Pattern Recognition

Pattern Recognition

Edited by

Peng-Yeng Yin

In-Tech
intechweb.org

Published by In-Teh

In-Teh
Olajnica 19/2, 32000 Vukovar, Croatia

www.intechweb.org
Additional copies can be obtained from:
publication@intechweb.org

First published October 2009
Printed in India

Technical Editor: Maja Jakobovic

Pattern Recognition,
Edited by Peng-Yeng Yin

p. cm.
ISBN 978-953-307-014-8

Preface

For more than 40 years, pattern recognition approaches are continuinglyimproving and have been used in an increasing number of areas with greatsuccess. This book discloses recent advances and new ideas in approachesand applications for pattern recognition.

Chapters 1 to 13 are devoted to new models and algorithms for genericpattern recognition problems ranging from improving primitivefunctionalities (calibration, segmentation, detection, registration, recognition,etc.) to the design of an integrated vision system. Since the use of biometricsin automatic control is given more attention, Chapters 14 to 17 present severalapplications in face recognition, posture estimation, and speaker recognition.

The intelligent processing of medical images is introduced in Chapters 18 to20. At last, a number of applications in various domains are summarized.Chapter 21 describes the recognition of color characters in scene images.Chapter 22 details the segmentation for license plate regions. Chapters 23 to25 propose approaches for recognition of plant branching, textile, and seabed,respectively. Chapter 26 introduces the use of pattern recognition in theprotection of power systems. Chapter 27 is devoted to the forecasting of airquality. The advent of Internet also promotes pattern recognition in newapplications such as spam recognition (Chapter 28), network security(Chapter 29), and content-based image retrieval (Chapter 30).

The 30 chapters selected in this book cover the major topics in patternrecognition. These chapters propose state-of-the-art approaches andcutting-edge research results. I could not thank enough to the contributions ofthe authors. This book would not have been possible without their support.

Peng-Yeng Yin
October 2009
Department of Information Management
National Chi Nan University
Nantou, Taiwan

Contents

1

Calibration of a Structured Light System Using Planar Objects

Koichiro Yamauchi, Hideo Saito and Yukio Sato
Keio University
Japan

1. Introduction

A triangulation-based structured light system consists of a camera and a projector. The system is similar to passive stereo vision system whose camera is replaced by the projector. Various light projection techniques, e.g. light section method and space encoding method, have been proposed (Shirai, 1972; Posdamer & Altschuler, 1982). These methods allow us to recover the 3D shape by the camera observing a light stripe projected from the projector. Then, the system using an electrically controlled liquid crystal device (Sato & Inokuchi, 1987) and the system using a semiconductor laser and a synchronized scanned mirror (Sato & Otsuki, 1993) have been proposed. These systems capture accurate 3D shape at high speed using highly intense light stripes.

Typically, the geometry of a structured light system is expressed by the pinhole model (Bolles et al., 1981). The camera geometry is represented by a 3 x 4 matrix having 11 degrees of freedom and the projector geometry is represented by a 2 x 4 matrix having 7 degrees of freedom. The two matrices allow 3D reconstruction of a target object (Li et al., 2003; Fukuda et al., 2006; Zhang et al., 2007). Although the pinhole model is suited for the camera geometry, it is not applicable to the projector geometry. For example, light stripes do not always pass through the optical center of the projector using a rotatable mirror, e.g. galvanometer mirror and polygon mirror.

Subsequently, the triangulation principle based on the baseline is also utilized for a structured light system. Given one side and two angles of a triangle determine the position of a target object. One side is the baseline which is defined as the distance between the camera and the projector. One of the angles indicates camera view and the other angle indicates projector view. The invariable baseline model (Matsuki & Ueda, 1989; Sansoni et al., 2000) fails to represent some projectors using a rotatable mirror, but the variable baseline model (Hattori & Sato, 1996; Reid, 1996) eases this problem. However, these models assume that a light stripe is vertical to the baseline. It is preferable to express the light stripe by a 3D plane disregarding the inner structure of the projector.

In this chapter, we present a new geometric model and calibration method for a structured light system to overcome the problems. The geometric model is defined such that the camera model is based on the pinhole model and the projector model is based on the equation of a plane model. If light stripes are projected in different directions, their projections are expressed accurately. In addition, the coefficients of the equation of a plane are estimated by observing a planar object from three viewpoints. It facilitates the procedure of user's tasks and provides a high degree of accuracy. Experimental results and comparisons demonstrate the effectiveness of our approach.

2. Geometric Model

A structured light system consists of a camera and a projector. The system captures a range data by the camera observing a target object illuminated from the projector. Fig. 1 is the geometric model of a structured light system. The camera model is based on the pinhole model and the projector model is based on the equation of a plane model. The geometric model is represented in the camera coordinate system and the reference plane is represented in the reference plane coordinate system.

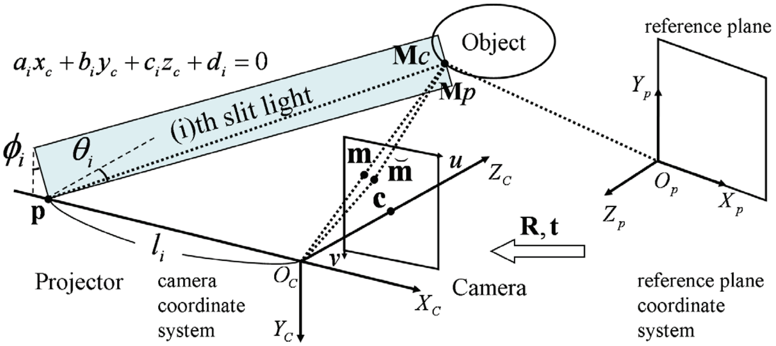


Fig. 1. Geometric model of a structured light system.

2.1 Camera model

Pinhole model is defined that light rays from an object pass through the optical center O_c for imaging. The principal point $\mathbf{c}$ at the intersection of the optical axis with the image plane is denoted by $[u_0,\quad v_0]$. X_c axis, Y_c axis, and Z_c axis are parallel to horizontal axis, vertical axis, and optical axis of the image plane. Here, a 2D point, i.e. image coordinates, $\mathbf{m}$ is denoted by $[u,\quad v]$ in the image plane, and a 3D point, camera coordinates, $\mathbf{M}_c$ is denoted by $[x_c,\quad y_c,\quad z_c]$ in the camera coordinate system $(O_c - X_c - Y_c - Z_c)$. In addition, X_p axis, Y_p axis, Z_p axis, and O_p are defined as horizontal axis, vertical axis, orthogonal axis, and the coordinate origin of the reference plane. Here, a 3D point, i.e. reference plane coordinates, $\mathbf{M}_p$ is denoted by $[x_p,\quad y_p,\quad z_p]$ in the reference plane coordinate system $(O_p - X_p - Y_p - Z_p)$. The perspective projection which maps the reference plane coordinates onto the image coordinates is given by

$$\tilde{\mathbf{m}} \cong \mathbf{A}[\mathbf{R} \quad \mathbf{t}]\tilde{\mathbf{M}}_p \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $\mathbf{A}$ is the camera intrinsic matrix with the scale factors, α, β, γ, and the principal point, u_0, v_0, i.e. the intrinsic parameters, and $[\mathbf{R} \quad \mathbf{t}]$ combines the rotation matrix and the translation vector, i.e. the extrinsic parameters. The tilde indicates the homogeneous coordinate by adding 1 for the additional element: $\tilde{\mathbf{m}} = [u, \quad v, \quad 1]$ and $\tilde{\mathbf{M}}_p = [x_p, \quad y_p, \quad z_p, \quad 1]$. The Euclidean transformation which transforms the reference plane coordinates to the camera coordinates is given by

$$\mathbf{M}_c \cong [\mathbf{R} \quad \mathbf{t}]\tilde{\mathbf{M}}_p \quad \text{with} \quad \mathbf{R} = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3] \tag{2}$$

where $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$ correspond to unit vectors to indicate the directions of X_p axis, Y_p axis, Z_p axis, respectively. $\mathbf{t}$ is the direction vector from O_p to O_c. Therefore, camera parameters provide the perspective projection and the Euclidian transformation. For more detail on camera geometry, refer to computer vision literatures (Faugeras & Luong, 2001; Hartley & Zisserman, 2004).

Let us consider camera lens distortion and its removal. The radial distortion causes the inward or outward displacement of the image coordinates from their ideal locations. This type of distortion is mainly caused by flawed radial curvature curve of the lens elements (Weng et al., 1992). Here, a distorted 2D point, i.e. real image coordinates, $\breve{\mathbf{m}}$ is denoted by $[\breve{u}, \quad \breve{v}]$. The discrepancy between the ideal image coordinates and the real image coordinates considering first two terms of radial distortion is given by

$$\breve{u} = u + (u - u_0)\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)\right]^2 \tag{3}$$

$$\breve{v} = v + (v - v_0)\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)\right]^2 \tag{4}$$

where k_1 and k_2 are the coefficients of the radial distortion, the center of which is the principal point. The normalized image coordinates $[x, \quad y]$ which suppose that the focal length is 1 (Wei & Ma, 1994) is give by

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{5}$$

Therefore, camera lens distortion can be corrected from captured images.

2.2 Projector model

The projector emits one to hundreds of light stripes for the measurement. We consider the case in which the light stripes are projected in different directions. It is difficult to assume that the projector model is based on the pinhole model, because they do not pass through the optical center. Therefore, we use the equation of a plane model to accurately represent the projector considering the projection of the light stripes which depend on the type of projector. In the camera coordinate system, the light stripe can be written as

$$a_i x_c + b_i y_c + c_i z_c + d_i = 0 \tag{6}$$

where i is the light stripe number, and a_i, b_i, c_i, d_i are the coefficients of the equation. There are an equal number of the equations of planes and the light stripes.

We define l_i is the baseline, i.e. the distance between the optical center of the camera and the light stripe of the projector, θ_i is the projection angle, i.e. the angle between Z_c axis and the light stripe, and ϕ_i is the tilt angle, i.e. the angle between Y_c axis and the light stripe. From the coefficients of the equation, these explicit parameters can be written as

$$l_i = d_i / a_i \tag{7}$$

$$\theta_i = \arctan(-c_i / a_i) \tag{8}$$

$$\phi_i = \arctan(-b_i / a_i) \tag{9}$$

Projector parameters are expressed by both implicit and explicit representations. The coefficients are used for computation of range data, but their values do not exhibit distinct features. In contrast, the baselines, projection angles, and tilt angles provide characteristic distributions.

2.3 Triangulation

To achieve range data, the projector emits light stripes to a target object, and then the camera observes the illuminated object. So, the camera coordinates is the intersection of the viewpoint of the camera and the equation of a plane of the projector. The linear equation $[x_c / z_c, \quad y_c / z_c, \quad 1]$ which is derived from (1), (2), and (6) is given by

$$\begin{bmatrix} \alpha & \gamma & 0 \\ 0 & \beta & 0 \\ a_i & b_i & d_i \end{bmatrix} \begin{bmatrix} x_c / z_c \\ y_c / z_c \\ 1 / z_c \end{bmatrix} = \begin{bmatrix} u - u_0 \\ v - v_0 \\ -c_i \end{bmatrix} \tag{10}$$

Consequently, we have

$$x_c = \frac{(u - u_0) - \gamma / \beta (v - v_0)}{\alpha} z_c \tag{11}$$

$$y_c = \frac{v - v_0}{\beta} z_c \tag{12}$$

$$z_c = \frac{d_i / a_i}{-c_i / a_i - \frac{(u-u_0) - \gamma / \beta (v - v_0)}{\alpha} - b_i / a_i \frac{(v-v_0)}{\beta}} \tag{13}$$

The coordinate z_c is computed by the relationship between the viewpoint of the camera and the equation of a plane of the projector. Then, the coordinate x_c and the coordinate y_c are computed by the similar triangle related to the camera. Therefore, the camera coordinates can be recovered by the camera and projector parameters.

The coordinate z_c which is expressed by the baseline, projection angle, and tilt angle instead of the coefficients can be written as

$$z_c = \frac{l_i}{\tan\theta_i - \frac{(u-u_0) - \gamma / \beta (v - v_0)}{\alpha} - \tan\phi_i \frac{(v-v_0)}{\beta}} \tag{14}$$

It indicates the triangulation principle based on one side and two angles of a triangle.

3. Calibration Method

In this section, we present a calibration method for a structured light system by observing a planar object from three viewpoints. Fig.2 is the calibration scene of a structure light system. The planar object, called reference plane, contains a checkered pattern, so that calibration points are detected as the intersection of line segments. To perform the calibration, the reference plane coordinates is assigned to the calibration points. Three sets of color images and slit light images, which include calibration points and light stripes on the reference planes respectively, are required. Our approach incorporates two separate stages: camera calibration and projector calibration.

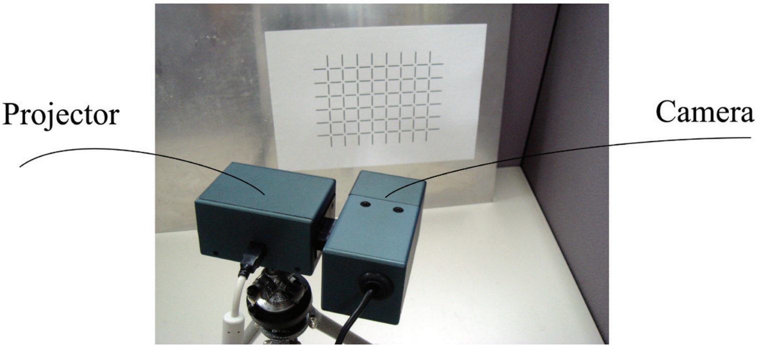


Fig. 2. Calibration scene.

3.1 Camera calibration

In the camera calibration stage, camera parameters are obtained by Zhang's method (Zhang, 2000). Fig. 3 shows the relationship between the reference plane and the image plane. The camera parameters are estimated by the correspondence between the reference plane coordinates and the image coordinates. Here, three color images must be captured from different positions changing orientations. If the reference plane undergoes pure translation, the camera parameters cannot be estimated (Zhang, 1998).

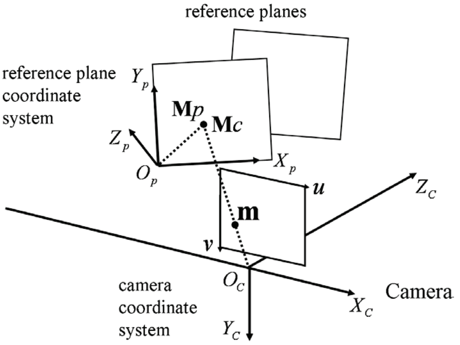


Fig. 3. Camera calibration.

3.2 Projector calibration

In the projector calibration stage, projector parameters are estimated by image-to-camera transformation matrix based on the perspective projection and the Euclidian transformation of the camera parameters which encapsulate the position and orientation of the reference planes. Fig. 4 shows the relationship among the reference plane, the image plane, and the light stripe. Here, the reference plane is on $z_p = 0$ and the coupled matrix $\mathbf{Q}$ is denoted by $[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]$. From (1), the perspective projection which maps the reference plane coordinates onto the image coordinates can be written as

$$\tilde{\mathbf{m}} \cong \mathbf{AQ}\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \tag{15}$$

From (2), the Euclidean transformation which transforms the reference plane coordinates to the camera coordinates can be written as

$$\mathbf{M}_c = \mathbf{Q}\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \tag{16}$$

Furthermore, the inverse of the coupled matix is given by

$$\mathbf{Q}^{-1} = \frac{1}{\mathbf{r}_3{}^{\mathrm{T}}\mathbf{t}} \begin{bmatrix} (\mathbf{r}_2 \times \mathbf{t})^{\mathrm{T}} \\ (\mathbf{t} \times \mathbf{r}_1)^{\mathrm{T}} \\ \mathbf{r}_3{}^{\mathrm{T}} \end{bmatrix} \tag{17}$$

where T indicates the transpose of a matrix. From (15), (16), and (17), the transformation matrix which maps the image coordinates into the camera coordinates is given by

$$\begin{aligned} \tilde{\mathbf{M}}_c = \begin{bmatrix} \mathbf{M}_c \\ 1 \end{bmatrix} &= \begin{bmatrix} \mathbf{Q}^{\mathrm{T}} \\ \mathbf{k} \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \\ &\cong \begin{bmatrix} \mathbf{Q}^{\mathrm{T}} \\ \mathbf{k} \end{bmatrix} \mathbf{Q}^{-1}\mathbf{A}^{-1}\tilde{\mathbf{m}} \\ &\cong \begin{bmatrix} \mathbf{I} \\ (\mathbf{r}_3{}^{\mathrm{T}}\mathbf{t})^{-1}\mathbf{r}_3{}^{\mathrm{T}} \end{bmatrix} \mathbf{A}^{-1}\tilde{\mathbf{m}} \end{aligned} \tag{18}$$

where the matrix $\mathbf{I}$ is denoted by $diag(1,1,1)$ and the vector $\mathbf{k}$ is denoted by $[0, \quad 0, \quad 1]$. The image-to-camera transformation matrix is directly estimated by camera parameters unlike other methods which necessitate recalculations. This matrix has eight degrees of freedom which is similar to the homography matrix in 2D.

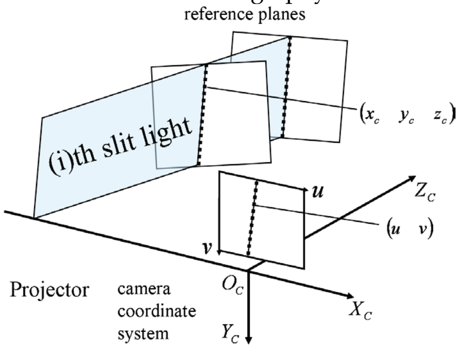


Fig. 4. Projector calibration.

For each light stripe, the image coordinates is transformed to the camera coordinates, so that the coefficients of the equation of a plane can be computed by the least square method at least three image coordinates. If the image coordinates of the light stripe are obtained from one reference plane, the equation of a plane cannot be computed. This is how all the light stripes are estimated.

3.3 Calibration procedure

The following is the recommended calibration procedure.

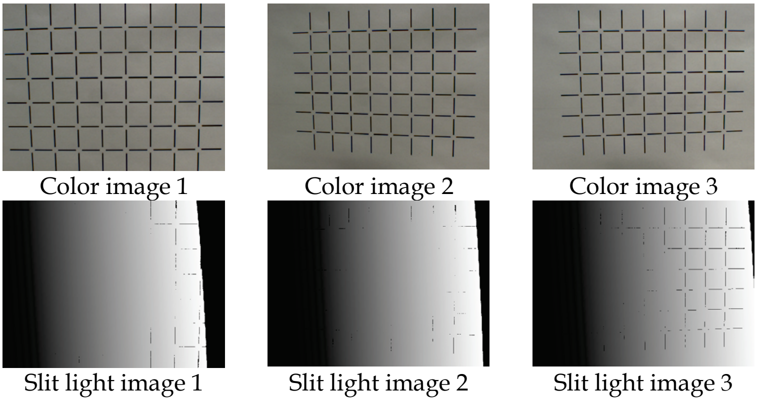
(a) Print a checkered pattern and attach it to a planar object
(b) Capture three sets of color images and slit light images from different position changing orientations by moving either the system or the plane
(c) Detect calibration points and correspond the image coordinates to the reference plane coordinates
(d) Estimate the camera parameters by Zhang's method
(e) Detect light stripes and correspond slit light numbers to the image coordinates
(f) Estimate the projector parameters by fitting a 3D plane as described in Sec. 3.2

4. Experimental Results

The data is captured by a structured light system, Cartesia 3D Handy Scanner of SPACEVISION. This system obtains range data in 0.5 seconds with 8 mm focal length, 640 x 480 pixels and 254 light stripes. The light stripes are scanned by a rotatable mirror (Hattori & Sato, 1996). The reference plane with the checkered pattern includes 48 calibration points with 20 mm horizontal and vertical intervals.

4.1 Calibration

Three sets of color images and slit light images are used for calibration as shown in Fig. 5. For the color images, one straight line is fitted to two horizontal line segments and the other straight line is fitted to two vertical segments. The calibration point is detected as the intersection of two straight lines. For slit light images, luminance values from 1 to 254 correspond to the light stripe number. The light stripes are projected to the reference plane vertically.



Color image 1 Color image 2 Color image 3

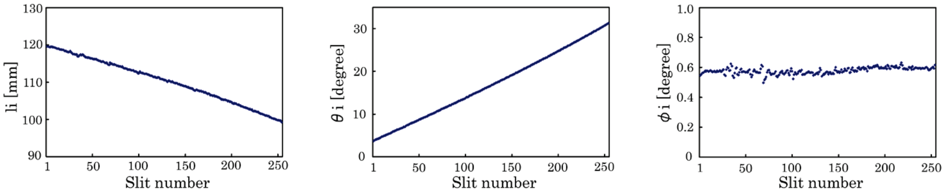
Slit light image 1 Slit light image 2 Slit light image 3

Fig. 5. Three sets of color images and slit light images.

Table 1 shows the camera intrinsic matrix and the coefficients of the radial distortion of the camera parameters. Fig. 6 shows the baselines, projection angles, and tilt angles of the projector parameters. When the light stripe number increases, the baselines gradually reduce, the projection angles increase, and the tilt angles remain almost constant. The camera and projector parameters enable the system to recover the camera coordinates of a target object.

A	$\begin{bmatrix} 1061.71 & -0.562002 & 350.08 \\ 0 & 1064.09 & 286.547 \\ 0 & 0 & 1 \end{bmatrix}$
k_1	-0.140279
k_2	-0.0916363

Table 1. Camera parameters.



Baseline Projection angle Tilt angle

Fig. 6. Projector parameters.

4.2 Evaluation

We evaluated the measurement accuracy using five spheres with 25 mm radius placed in front of the system. In our evaluation, the system captures range data, and then fit the ideal spheres to them. The measurement accuracy which is defined as the distance between the ideal radius $\hat{r}$ and the real radius r_i is given by

$$E = \frac{1}{N} \sum_{i=1}^{N} (r_i - \hat{r})^2 \tag{19}$$

where N is the number of measuring points. To show the effectiveness, we evaluated our approach by comparing with two conventional approaches.

(i) The pinhole model calibrated by slide stage

The camera is modeled by the 3 x 4 projection matrix, and the projector is modeled by the 2 x 4 projection matrix. The camera and projector parameters are estimated using the slide stage.

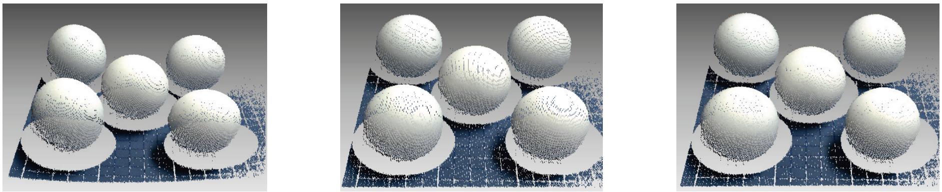
(ii) The equation of a plane model calibrated by slide stage

The camera model is based on the pinhole model, and the projector model is based on the equation of a plane model. The camera parameters are obtained by Tsai's method (Tsai, 1987), and the projector parameters are estimated using the reference plane.

(iii) The equation of a plane model calibrated by reference plane: Our approach

The camera model is based on the pinhole model, and the projector model is based on the equation of a plane model. The camera and projector parameters are estimated using the reference plane.

Fig. 7 is the range data of five spheres. The spheres are numbered from top left to bottom right. In the approach (i), left two spheres, i.e. No. 1 and No. 4, and the ground are distorted in contrast to the approach (ii) and (iii). Table 2 shows the measurement accuracy of five spheres. In the approach (i), the measurement accuracy is higher than the approach (ii) and (iii). The approach (ii) and (iii) achieve similar performance. Therefore, the equation of a plane model is applicable to the structured light system. In addition, the reference plane as a planer object provides a high degree of accuracy and has a high degree of availability compared to the slide stage as a cubic object. The experimental results demonstrate the effectiveness and efficiency of our approach.



Approach (i) Approach (ii) Approach (iii)

Fig. 7. Range data of five spheres.

Sphere number	No. 1	No. 2	No. 3	No. 4	No. 5
Measuring points	15,629	15,629	19,405	19,861	19,861
Approach (i)	0.41	0.38	0.26	0.26	0.31
Approach (ii)	0.22	0.31	0.19	0.13	0.20
Approach (iii)	0.23	0.32	0.21	0.15	0.21

Table 2. Measurement accuracy of five spheres.

5. Future Work

It has been challenging to capture range data of an entire body using multiple projector-camera pairs. In our previous works, we have developed the system consisting of four pole units with sixteen projector-camera pairs (Yamauchi & Sato, 2006). Then, we have developed the system consisting of three pole units with twelve projector-camera pairs (Yamauchi et al., 2007). Fig. 8 is the human body measurement system. The system acquires range data in 2-3 seconds with 3 mm depth resolution and 2 mm measurement accuracy. The range data of a mannequin and a man are shown in Fig. 9 and Fig. 10, respectively. The numbers of measurement points are approximately 1/2 to one million. The projector-camera pairs are calibrated by our approach, and then their local coordinate systems are integrated into a single coordinate system by an automatic alignment approach (Fujiwara et al., 2008). Although Fujiwara's method is performed in two stages, our approach allows fully automatic calibration for this type of system. In addition, it is possible to facilitate the calibration process and reduce the implementation time.

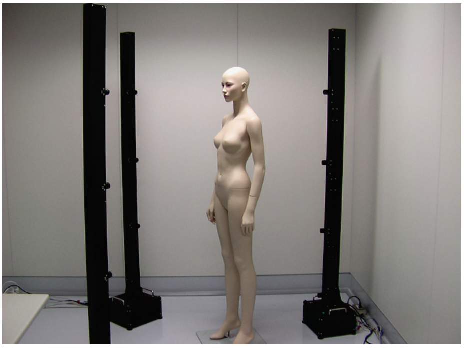


Fig. 8. Human body measurement system.

6. Conclusions

We presented a novel geometric model and calibration method for a structured light system using a planar object. The geometric model is defined such that the camera model is based on the pinhole model and the projector is based on the equation of a plane model. Although the light stripes do not exactly pass through the optical center, our model can approximate the system geometry. In addition, the camera and projector parameters are estimated by observing a planar object from three viewpoints. The camera parameters are obtained by Zhang's method and the projector parameters are estimated by using image-to-camera transformation matrix. Furthermore, we verify our approach provides a high degree of accuracy in the experiments. In the future we intend to apply for the human body measurement system using multiple projector-camera pairs.

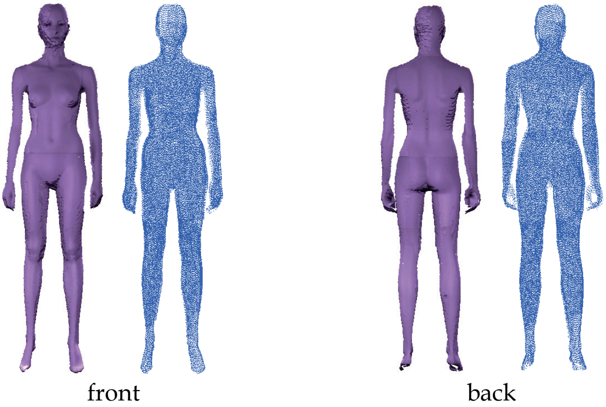


Fig. 9. Range data of a mannequin.

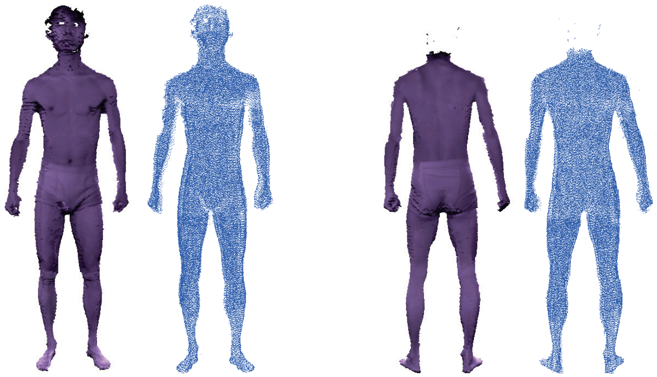


Fig. 10. Range data of a man.

7. References

Bolles, R. C.; Kremers, J. H. & Cain, R. A. (1981). A simple sensor to gather three-dimensional data, *Technical Report 249,* (July 1981) Stanford Research Institute.

Faugeras, O. & Luong, Q. T. (2001). *The geometry of multiple images,* MIT Press, ISBN: 978-0262062206, Cambridge, MA, United States.

Fujiwara, K.; Yamauchi, K. & Sato, Y. (2008). An automatic alighnment technique for multiple rangefinders, *Proceedings of the SPIE,* Vol. 6805, (January 2008).

Fukuda, M.; Miyasaka, T. & Araki, K. (2006). A prototype system for 3D measurement using flexible calibration method, *Proceedings of the SPIE,* Vol. 6056, (January 2006).

Hartley, R. & Zisserman, A. (2004). *Multiple View Geometry in Computer Vision,* Cambridge University Press, ISBN: 978-0521540513, Cambridge, United Kingdom.

Hattori, K. & Sato, Y. (1996). Accurate rangefinder with laser pattern shifting, *Proceedings of the 13th International Conference on Pattern Recognition,* Vol. 3, (August 1996) pp. 849-853.

Li, Y. F. & Chen, S. Y. (2003). Automatic recalibration of an active structured light vision system, *IEEE Transactions on Robotics and Automation,* Vol. 19, No. 2, (April 2003) pp. 259-268.

Matsuki, M. & Ueda, T. (1989). A real-time sectional image measuring system using time sequentially coded grating method, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 11, No. 11, (Novemeber 1989) pp. 1225-1228.

Posdamer, J. L. & Altschuler, M. D. (1982). Surface measurement by space-encoded projected beam systems, *Computer Graphics and Image Processing,* Vol. 18, (January 1982) pp. 1-17.

Reid, I. D. (1996). Projective calibration of a laser-stripe range finder, *Image and Vision Computing,* Vol. 14, No. 9, (October 1996) pp. 659-666.

Sansoni, G.; Carocci, M. & Rodella, R. (2000). Calibration and performance evaluation of a 3-D imaging sensor based on the projection of structured light, *IEEE Transactions on Instrumentation and Measurement,* Vol. 49, No. 3, (June 2000) pp. 628-636.

Sato, K. & Inokuchi, S. (1987). Range-imaging system utilizing nematic liquid crystal mask, *Proceedings of the 1st International Conference on Computer Vision,* (June 1987) pp. 657-661.

Sato, Y. & Otsuki, M. (1993). Three-dimensional shape reconstruction by active rangefinder, *Proceedings of the 9th IEEE Conference on Computer Vision and Pattern Recognition,* (June 2003) pp. 142-147.

Shirai, Y. (1972). Recognition of polyhedrons with a range-finder, *Pattern Recognition*, Vol. 4, No. 3, (October 1972) pp. 243-250.

Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lens, *IEEE Journal of Robotics and Automation,* Vol. 3, No. 4, (August 1987) pp. 323-344.

Wei, G. & Ma, S. (1994). Implicit and explicit camera calibration: theory and experiments, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 16, No. 5, (May 1994) pp. 469-480.

Weng, J.; Cohen, P. & Herniou, M. (1992). Camera calibration with distortion models and accuracy evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 14, No. 10, (October 1992) pp. 965-980.

Yamauchi, K. & Sato, Y. (2006). 3D human body measurement by multiple range images, *Proceedings of the 18th International Conference on Pattern Recognition,* Vol. 4, (August 2006) pp. 833-836.

Yamauchi, K.; Kameshima, H.; Saito, H. & Sato, Y. (2007). 3D reconstruction of a human body from multiple viewpoints, *Proceedings of the 2nd Pacific-Rim Symposium on Image and Video Technology,* LNCS 4872, (December 2007) pp. 439-448.

Zhang, B.; Li, Y. F. & Wu, Y. H. (2007). Self-recalibration of a structured light system via plane-based homography, *Pattern Recognition,* Vol. 40, No. 4, (April 2007) pp. 1368-1377.

Zhang, Z. (1998). A flexible new technique for camera calibration, *Technical Report MSR-TR-98-71,* (December 1998) Microsoft Research.

Zhang, Z. (2000). A flexible new technique for camera calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 22, No. 11, (November 2000) pp. 1330-1334.

Background modelling with associated confidence

J. Rosell-Ortega and G. Andreu-García
Computer Vision Group. DISCA. Technical University of Valencia
Spain

In this chapter we introduce an algorithm aimed to create background models which associates a confidence value to the obtained model. Our algorithm creates the model based on motion criteria of the secene. The goal of this value is to quantify the quality of the model after a number of frames have been used to build it. The algorithm is first designed in gray tones and for unimodal background models and through the chapter is extended for colour scenarios and with the possibility of using several models per pixel. Quantitative and qualitative experimental results are obtained with a well-known benchmark.

1. Introduction

Visual analysis of human motion (Wang .et. al., 2003) is currently one of the most active research topics in computer vision. This strong interest is driven by a wide spectrum of promising applications in many areas such as *virtual reality*, *smart surveillance* and *perceptual interface*, just to mention the most representative.

Visual analysis concerns the detection, tracking and recognition of objects in general, and particularly, people. Also the understanding of human behaviour in the case of image streams involving humans. Visual analysis of a scene starts from a segmentation of the scene in order to classify pixels as foreground or background; then, other steps may be taken depending on the application, such as motion analysis, object detection, object classification, tracking and activity understanding.

Background subtraction is usually mentioned in the literature concerning smart surveillance, as one of the most popular methods to detect regions of interest in frames. This technique consists in subtracting the acquired frame from a background model and classifying as foreground all those pixels whose difference with the background is over a threshold. Thus, the importance of producing an accurate background model and choosing a precise threshold is obvious.

1.1 Related work

A large number of different methods have been proposed in recent years and many different features have been used to maintain the background model and perform the background subtraction. Most of the methods rely on describing the background model with pixel's intensity or colour information (Sshoustarian & Bez, 2005), (Hu et. al., 2004), (Elgammal et. al. 2000). But some others rely on other kind of information, for instance, edge detection, optical flow or textures, (Mason and Duric, 2001), (Wixson, 2000), (Heikkilä & Pietikäinen, 2006).

One of the methods, which rely on the pixel's intensity, consists in modelling each pixel in a video frame with a Gaussian distribution. This is the underlying model for many background subtraction algorithms. A simple technique is to calculate an average image of the scene, to subtract each new video frame from it and to threshold the result. The adaptive version of this algorithm updates the model parameters recursively by using a simple adaptive filter. This single Gaussian model can be found in (Wren et. al., 2004).

This model however, does not work well when the background is not static. For instance, waves, clouds or any movement which does also belong to the background cannot be properly described using one Gaussian distribution. A solution is using more than one Gaussian to model the background, as proposed in (Stauffer & Grimson, 1999]. In (Zang & Klette, 2004) methods for shadow detection and per-pixel adaptation of the parameters of the Gaussians are developed.

Following with the methods based on mixture of Gaussians, in (Elgammal et. al. 2000), it is proposed to build a statistical representation of the background. This is done by estimating directly from data the probability density function, with no previous assumptions about the underlying distribution.

Other approaches which do not rely on Gaussian distributions to model the background can be found, for instance, in (Mason and Duric, 2001). In this paper, the algorithm proposed computes a histogram of edges in a block basis. These histograms are constructed using pixel-specific edge directions. A fusion of this approach with intensity information may be found in (Jabri et. al., 2000).

Motion may also be used to model the background. Authors of (Wixson, 2000) propose an algorithm that detects salient motion by integrating frame-to-frame optical flow over time. Salient motion is considered to be motion that tends to move in a consistent direction over time.

Radically different is the approach introduced in (Heikkilä & Pietikäinen, 2006). These authors propose using features bases on textures to model the scene and to detect moving objects. The features used are LBP (local binary pattern) and the algorithm models the background using these features by assigning each pixel with a set of LBP histograms. As authors state, their algorithm has a lot of parameters to tune.

Though the aforementioned approaches obtain good results in the tested scenarios, in general, all these approaches expect working in scenarios with low or null activity to build their first model. One of aspects which we miss in these approaches is that there is no measure of when a suitable background is achieved.

Besides the different approaches to background modelling, another issue related to this technique is the detection of corrupt models, that is, models which are not useful any more for surveillance purposes.

Few papers in the literature address this issue, as far as authors of this chapter are concerned. In the literature it is generally assumed that changes in the background will occur smoothly and abrupt changes are not considered. In (Toyama et.al., 1999), authors propose maintaining a database of models. In the case the background model is considered to be corrupt, by whichever the mean, a search in the database should be enough to find the most suitable model.

In our opinion, this is a very time consuming process, and does not solve completely the problem. We consider that recovering a corrupt background model is the same as creating a new one. In this chapter, we explore the possibility of giving a unique solution to both problems. Thus, only a method to detect corrupt models must be defined and the model recovery may just be considered as a restart of the system, building a new background model.

1.2 Goals

Our developments are constrained to concrete situations. We focus our attention specially on demanding scenarios, which are those in which there is always a significant activity level, making it difficult to obtain a clean model with traditional techniques, such as mean, mode... These scenarios may be found in public buildings or outdoor areas, for instance, airports, subway or railway stations, entrance of buildings and so on, in which there are always people walking or standing.

Scenarios such as airports or railway stations are on duty 24 hours a day with a constant activity. In this kind of scenario it is difficult obtaining images without people of the areas under surveillance, in the case it had to be done in a concrete moment. But it is also desirable to get as soon as possible a good model in the case of model corruption.

Hardware is another of our constraints. Algorithms discussed in the following sections are designed to be implemented in a DSP-based hardware with a limited memory. Thus, storing a big amount of background models is not possible.

Two questions arise when talking about corrupt background models. How can a model be considered to be corrupt? From the algorithmic point of view, a measure of the quality of a model is needed in order to be able to detect how the process of model recovery evolves and a mechanism to detect corrupt models is also needed.

And, how may the quality of a model be measured? These questions are not yet given an answer in the literature. We propose measuring the quality by taking into account for how long a model has properly described a pixel.

Our approach tries to obtain a background model which can provide the system with a suitable segmentation and a correct classification of objects in the scene.

The solution we propose tries to answer the two questions aforementioned and also, give a general technique for background reconstruction. We propose a mathematical definition of model corruption in terms of number of pixels classified as foreground with respect to total number of pixels contained in the scene. This definition may, of course, be tailored for any situation.

The aims of the algorithm are:

(1) Construct a background model by acquiring frames no matter how many objects appear in the scene.

(2) Define a measure of the quality of the background model obtained and a confidence of the pixels classified as foreground.

(3) Avoid storing background models, in case of failure, the model will be recomputed on the fly.

(4) Compute a confidence value associated to the model for each pixel $b \in B$, in order to evaluate the security with which this pixel is classified as belonging to background. The higher the confidence of the model, the better the background model is.

In the following sections we develop an algorithm which may quickly reconstruct a background model in the case it is corrupt. Our scope is being able to build it even if people are present in the scene, meeting the first four constraints.

The fourth requirement is achieved by means of a definition for background model quality. One of the problems of the methods mentioned in the introduction is that they cannot determine whether a suitable background model is built or not. For instance, averaging 50 images of a scene with no people present in any frames (or present in a little amount of them) will produce, more or less, the same result as taking the average of just a couple of images.

Using simple statistical methods gives no hint of the quality of the model constructed. If moving objects are present in the frames used to construct the background, blurred areas may appear as a mixture of the values of the objects and the values of the background will be done. The quality index we propose, is mainly used to give a quantitative measure of the background model's quality. However, its use is not only limited to this, but also may be helpful when defining a segmentation quality using this model.

This chapter is organized as follows, section 2 introduces BAC the background adaptive modelling algorithm (Rosell –Ortega et. al. 2008). This first version of the algorithm uses gray tones to describe the scene. In section 3, we explore the possibility of using the same segmentation schema of BAC with RGB coordinates. In section 4 we compare BAC with the Stauffer's approach. Finally, section 5 is devoted to conclusions and future works.

2. Background adaptive modelling algorithm (BAC)

Background models are traditionally generated using statistical measures. In this chapter, we propose not to use only statistical properties of pixels, but also their behaviour, to build the model. As stated in the introduction, the aim of the algorithm is reconstructing or creating a background model from the scratch, with no previous assumption about the scene activity. Similarity with the background and motion criteria are used to determine how the model must be updated.

We propose an algorithm that considers consecutive gray scale frames F(0), F(1), ... F(n), in which any pixel $p_{x,y} \in F(i)$ must belong either to foreground or to background and builds a background model B starting from a frame $F(i), i \geq 0$. In this first frame it is impossible to classify pixels as background or foreground, as no further information is given. To decide which pixels may be used to update the background model and which not, a new similarity and motion criteria is defined in next sections.

Section 2.2 describes the notion of similarity with the background and motion of a pixel. We use the previous knowledge of how a background pixel should behave to discriminate which values in each incoming frame belong to background and which do not. In section 2.3 we explain the algorithm. Section 2.4 explains the experiments we made with different real videos, comparing the result of using our method with mean, mode and median to construct a background model and shows the results we obtained.

2.1 Similarity criteria

Similarity between two pixels is usually tested by comparing the difference of their gray levels with a threshold. We propose to translate into a function the intuitive idea behind "very similar" or "similar" by using a continuous function defined as,

$$S(p,q)=e^{\frac{-|p-q|}{\kappa}} \qquad (1)$$

with $S:\Re \rightarrow [0,1]$, p and q are gray levels of two pixels, κ is a constant determined experimentally. This way, a difference degree and not an absolute value is calculated for pixels similarity. Figure 1 shows the evolution of this function.

2.2 Motion and similarity with the background

By using equation 1, it can be measured the similarity of each pixel with the background. Similarity between a pixel $q_{x,y} \in F(i)$ with a background pixel is then given by $S(q_{x,y}, b_{x,y})$, being $b_{x,y} \in B(i)$ the pixel in the background model.

Also, motion can be computed using equation 1. Motion of a pixel can be defined as its similarity with previous values of the pixel. Being $q_{x,y} \in F(t)$ a pixel in the current frame, $p_{x,y} \in F(i-1)$ and $r_{x,y} \in F(i-2)$; we define the motion of $q_{x,y}$ as,

$$M(q)=\frac{((1-S(p_{x,y},q_{x,y}))+(1-S(r_{x,y},q_{x,y}))}{2} \qquad (2)$$

This way, motion in the scene is detected by considering similarities of three consecutive frames.

2.3 Segmentation process

The background algorithm with confidence (BAC) starts by taking a frame F(i) to be the initial background model B(i) (the model in time i), and sets,

$$\forall b_{x,y} \in B(i) \ , c_{x,y}(i)=0 \wedge \sigma_{x,y}(i)=0 \qquad (3)$$

being $c_{x,y}(i)$ the confidence value of pixel $b_{(x,y)}$ and $\sigma_{x,y}(i)$ the filtered probability in time i.

Next two frames, F(i+1) and F(i+2), are ignored and used only to detect motion in frame F(i+3). For all the next incoming frames F(i), motion and similarities with B(i-1) are sought for. We define then the probability that any pixel $q \in F(i)$ belongs to foreground as,

$$P_{fore}(q)=\max(M(q),1-S(q,b)) \qquad (4)$$

because pixels will belong to foreground if either their motion value is high or their difference with the background is high. This way, we can include in the foreground set all pixels which, even being similar to the background but show significant motion and vice versa.

On the other side, the following expression,

$$P_{back}(q) = \max(1 - M(q), S(q,b)) \tag{5}$$

defines the probability that a pixel $q \in F(i)$ belongs to background if both its motion value is low and its similarity to current background is high (as stated in the constraints described before). It must be noted that the relationship,

$$P_{back} + P_{fore} = 1 \tag{6}$$

does necessary verify. According to definitions of both probabilities, it is easy to see that the chosen value for each probability is complementary of the other one.

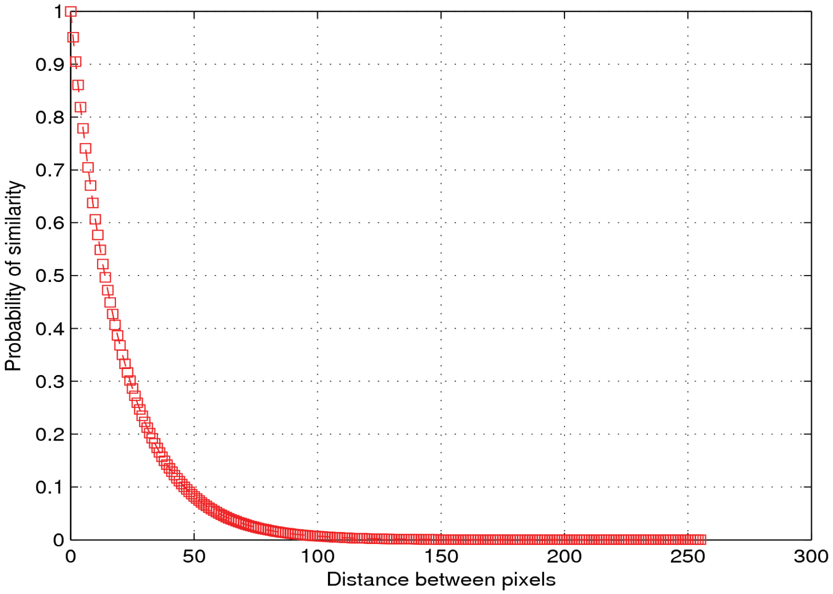


Fig. 1. Plot of function similarity for different distances.

Once F(i) is segmented, we must update the model B(i-1) to obtain B(i), using pixels in F(i). Not all pixels $b_{x,y} \in B(i-1)$ are updated in the same way, it depends on $P_{back}(b_{x,y})$, $c_{x,y}(i-1)$ and $\sigma_{x,y}(i)$. The segmentation separates pixels in two different sets; the foreground set (fSet) and the background set (bSet).

$$fSet = \{p_{x,y} \in F(\mathrm{i}) : P_{fore}(p_{x,y}) > 0.7\} \tag{7}$$

$$bSet = \{p_{x,y} \in F(\mathrm{i}) : p_{x,y} \notin fSet\} \tag{8}$$

We reduce the amount of pixels classified as foreground to only those whose foreground probability is high. This way, we get sure most of shadows will not be considered as foreground. On the other hand, anything which is not considered to be foreground, is classified as background.

The model probabilities are then updated,

$$\forall b_{x,y} \in B(i) : \sigma_{x,y}(i) = \frac{\sigma_{x,y}(i-1) \cdot c_{x,y}(i-1) + P_{back}(p_{x,y})}{\left(c_{x,y}(i-1)+1\right)} \tag{9}$$

being $\sigma_b(i)$ the filtered certainty of a pixel of belonging to background in time t. This probability is used, together with other measurements, to avoid that an object captured in the first model stays forever in the model. This filter accumulates the different background background probabilities obtained by a model pixel over time.

After filtering background probabilities, it may be seen that there are pixels whose probabilities diminish over time. For instance, this may be due to the fact that these pixels were still at the beginning of the process, and started to move later. But it may be also the case, that objects are moving over these pixels but they recover their values again after few frames.

In order to distinguish these two cases, two sets are added to the previous set definitions: pixels labelled as doubtful (dSet) and pixels in B(i) whose gray level will be replaced by the gray level of pixels in F(i) (cSet).

Doubtful pixels are those whose filtered background probability is under a threshold but whose confidence is still over a minimum. Recalling that the algorithm starts with zero knowledge about the scene, special care must be taken with such behaviour, in order to quickly change pixels which do not describe the background properly.

On the other side, pixels in the cSet represent pixels, whose confidence is under a threshold, and will be replaced by values from the current frame.

Equations 10 and 11 show how these sets are built. Doubtful pixels, dSet in equation 10, will be those which have a low filtered probability but still their confidence is high (at least, 80%). Equation 11 shows which are considered to be changed, those whose confidence is under 80% and also their filtered probability is low.

$$dSet = \left\{ p_{x,y} \in fSet : \sigma_{x,y}(i) < 0.8 \wedge \frac{c_{x,y}(i-1)}{c_{x,y}(i-1)+1} \geq 0.8 \right\} \tag{10}$$

$$cSet = \left\{ p_{x,y} \in fSet : \sigma_{x,y}(i) < 0.8 \wedge \frac{c_{x,y}(i-1)}{c_{x,y}(i-1)+1} < 0.8 \right\} \tag{11}$$

Values defining the previous sets were chosen to be very restrictive, this way, pixels which may yield low background similarity are quickly replaced. The regions of interest of frame F(i) are then defined by fSet. We define the following set for convenience,

$$C = bSet \cup dSet \tag{12}$$

We update pixels in a different way, depending on their observed behaviour. Those which have a high confidence and high filtered probability or their confidence is over a threshold, i.e., those which do not belong to cSet, are updated using the incoming values to cope with light changes.

Pixels which belong to the cSet are directly changed by values in the incoming frame. This way, regions which were labelled as foreground, become part of the background.

Being $q_{x,y} \in F(i)$,the model B(i) is updated as follows,

$$\forall b_{x,y} \in cSet : b_{x,y}(i) = q_{x,y}(i) \tag{13}$$

$$\forall b_{x,y} \in C : b_{x,y}(i) = q_{x,y}(i) + \alpha_{x,y}(i) \cdot \left(b_{x,y}(i-1) - q_{x,y}(i)\right) \tag{14}$$

Confidences of pixels are also updated distinguishing the set to which each pixel belongs to. In this case, pixels which do describe the background increase their confidence. Pixels whose $\sigma_{x,y}(i)$ reduces over time, do also reduce their confidence. On the other side, pixels which are copied from the image, take a confidence equal to zero.

As this operation is performed in a frame by frame basis, and pixels are reclassified after segmentation, any pixel whose confidence is reduced by a temporal occlusion by a foreground pixel will recover its previous confidence as soon as the occlusion finishes.

The confidence of pixels in B(i) is updated according to the following expressions,

$$\forall p_{x,y} \in bSet : c_{x,y}(i) = c_{x,y}(i-1) + 1 \tag{15}$$

$$\forall p_{x,y} \in dSet : c_{x,y}(i) = c_{x,y}(i-1) - 1 \tag{16}$$

$$\forall p_{x,y} \in cSet : c_{x,y}(i) = 0 \tag{17}$$

A difference with respect to other algorithms, is that we propose using a different adaption coefficient α for each pixel depending on the confidence they show. This way, we expect pixels which strongly described the scenario to update smoothly. On the other side, pixels whose confidence diminishes, recalling this means their background probability is descending, take a lower adaptation coefficient.

It is computed taking into account the confidence of the pixel in time i according to the following equations,

$$\forall p_{x,y} \in C : \alpha_{x,y}(i) = 0.98 \cdot \frac{c_{x,y}(i)}{\left(c_{x,y}(i) + 1\right)} \tag{18}$$

$$\forall p_{x,y} \in cSet : \alpha_{x,y}(i) = 0 \tag{19}$$

The adaptation coefficient takes values in the range [0,0.98). Being 0.98 the value which corresponds to pixels with a high confidence and 0 the value which corresponds to pixels which have been changed.

As said before, together with its gray level value, each pixel $b_{x,y} \in B(i)$ provides a confidence value which may be used to weight the quality of the segmentation. We define the segmentation confidence of the model B(i) as,

$$sc = \frac{1}{m \cdot n} \cdot \sum \frac{c_b(i)}{c_b(i)+1}, \forall b \in B(i) \tag{20}$$

being $m \cdot n$ the number of pixels of the model. The segmentation confidence (sc) is calculated for a target T_i with a size l in pixels of frame F(i), by particularizing this expression considering only the pixels segmented for this target.

Finally, in order to test when the background model is stable the mean square quadratic difference (msqd) between two consecutive models is calculated; the algorithm finishes if the following condition verifies,

$$msqd(B(i), B(i-1)) < 10^{-3} \wedge sc > 0.995 \tag{21}$$

2.4 Experiments

We made experiments to test two different issues. First, several random frames from test videos were chosen as the base to reconstruct the background model. We then compared the background model obtained with the BAC algorithm, with the one obtained by using median, mean and mode with the same frames used by BAC. Next experiments were aimed to control how accurate the segmentation was, by using BAC to segment frames while the algorithm was under construction.

Videos from different sources were used with the aim of reproducing different situations; videos recorded by ourselves, real videos from the airport and Wallflower benchmark. Videos had different lengths and were converted into grayscale when needed.

We compared the BAC's segmentation with a supervised segmentation in order to quantify the true positives (TP), which are pixels classified as foreground in the control image and by the algorithm, and the true negatives (TN), which are are pixels classified as background in the control image and by the algorithm. False positives (FP) and false negatives (FN) are defined as the complementary of the previous ones.

Good results were obtained with BAC, they may be found together with resulting models using mean aplied to test videos at www.vxc.upv.es/vision/proyectos/BAC.

A representative situation aim of our developments is analysed in this section. The video starts in F(0) with several people in a scene, simulating a surveillance system, in that moment B(0) is created with targets with sc=0, see figure 2 (a). In order to evaluate quantitatively the evolution of BAC, we segmented manually 22 frames randomly selected.

In table 1, segmentation results for frames F(90) and F(390) obtained with BAC and mean are compared; sc of pixels found in each target's segmentation with BAC is shown under column "confidence", for pixels not correctly segmented, sc was under 0.001. The original frames, together with segmentation result and the background model used may be found in figures 2,3 and 4.

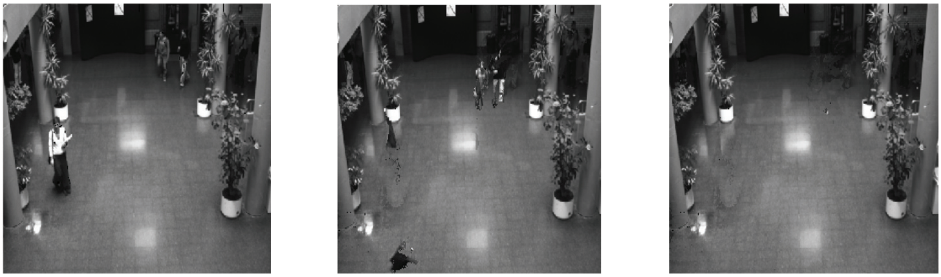
	Frame 90			Frame 390		
target	BAC	mean	Model background confidence	BAC	mean	confidence
1 st	0.69	0.74	0.946	0.96	0.88	0.997
2 nd	0.90	0.70	0.977	0.89	0.71	0.996
3 rd	0.37	0.49	0.986	0.51	0.69	0.997
4 th	0.47	0.49	0.933	-	-	-

Table. 1. Percentage of pixels found for each hand-segmented target in control frames 90 and 390. Targets are not the same in both frames.

In F(90), the four objects present in the scene are segmented with BAC and mean; only those dark objects which are far in the field of view of the camera are segmented more poorly (target 3); something similar happens with target 4, which is a group of two people moving still in the same area they occupied at the beginning of the movie. We consider that with at least 45% of the total size of pixels detected of a target is sufficient to continue with classification and tracking tasks, if they are grouped in an only blob.

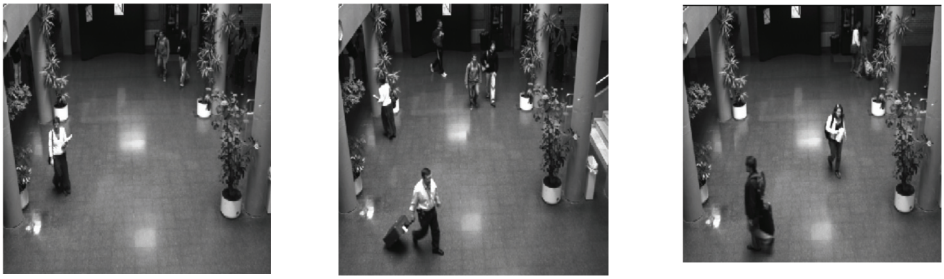
In figure 2 (a) and 2 (b) B(0) and B(89) are shown, it may be seen that in B(89), background model has achieved c = 0.982 and some targets have been removed. Improvement over time is evident as B(389) contains no target. This improvement manifests in F(390) with a better segmentation and a model with sc = 0.997.

Evolution of BAC's confidence, TP and TN, of BAC and mean are shown in figure 2. Objects standing still for long periods of time influence negatively the value of TP. The plot shows that BAC segments correctly more pixels than mean. In F(201) several objects leave the scene and others start coming in and in F(680) some objects stand still; this explains some foreground pixels not found. On the other side, TN, easily reach a high level as area of quiet targets is small compared to the image.



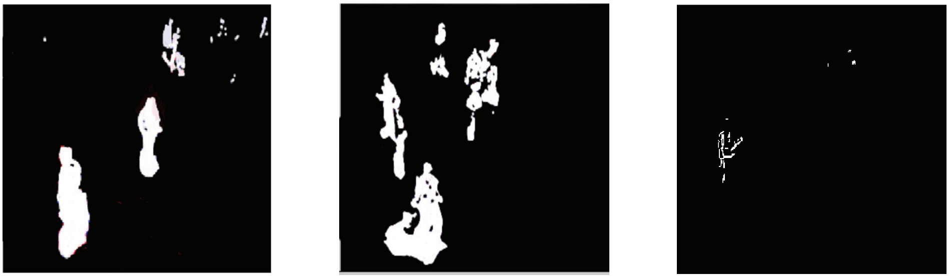
a) B(0)=0 b) B(90) = 0.982 c) B(390) = 0.997

Fig. 2. Background evolution, first figure correspond to background model in F (1). Figure (b) corresponds to the background model updated until F (90). Figure (c) corresponds to the background model in F (390).



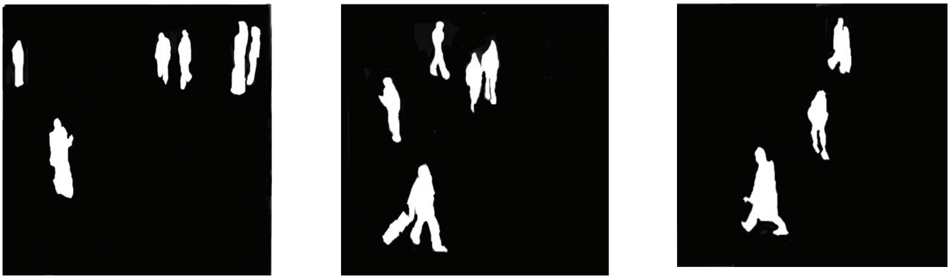
(a) (b) (c)

Fig. 3. Different frames showing the evolution of people in the scenario. From left to right, images correspond to frames 1, 90 and 390.



(a) (b) (c)

Fig. 4. From left to right, result of background subtraction of frames 1, 90 and 390 using the background models computed so far.



(a) (b) (c)

Fig. 5. From left to right, the expected result for background subtraction for frames 1, 90 and 390. These images are segmented manually, labeling with white expected foreground and in black the background.

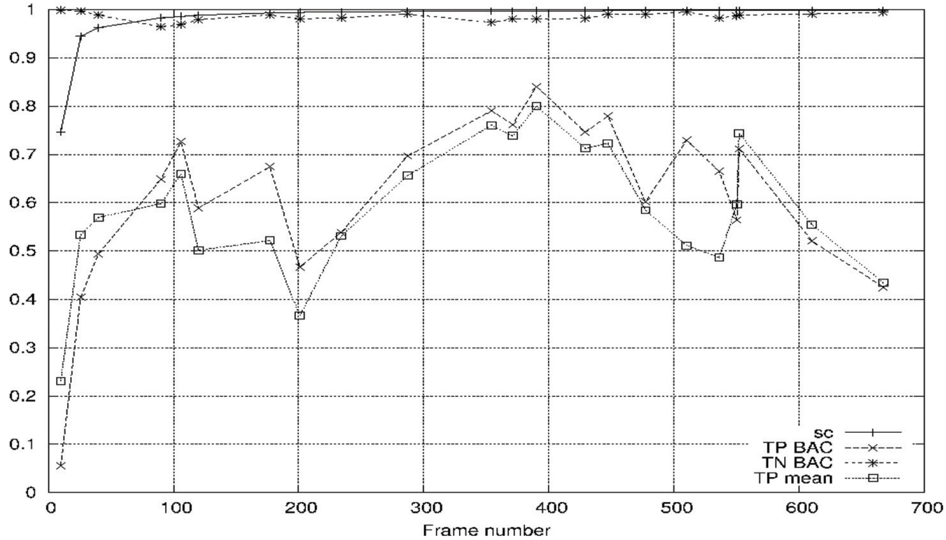


Fig. 6. Evolution of confidence, true positives and true negatives for the discussed video

Table 2 shows results for Wallflower benchmark. Values of true negatives are high, though videos were too short for BAC to converge. For sequence "wavingTree" sequence, fails due to the movement of the tree. In "lightSwitch", BAC was started in the moment lights were switched on.

Finally, in "bootstrap", brightness makes BAC fail to find correctly the targets, though it find most of the pixels associated to them.

Experiments show that BAC obtains background models equal to those obtained by using any statistical technique; with the added benefit of permitting segmentation from the very beginning of the process; as was the goal and together with a confidence measure of the obtained model. Also, the experiments performed with the Wallflower test set result promising. Our efforts should be address in near future to improve the response of the algorithm to shadows and brightness.

In the following sections, we extend the algorithm to improve segmentation results. We start by adding colour to the segmentation schema.

	bootstrap	camouflage	foregroun d	Light Switch	Moved Object	Time Of Day	Waving Trees
TP	0.48	0.73	0.51	0.44	------	0.36	0.73
TN	0.96	0.86	0.95	0.98	0.99	0.99	0.74

Table. 2. True positives and true negatives for the Wallflower benchmark.

3. BAC with colour

In order to improve results, one of the basic things that can be done is adding colour to the description of pixels. By doing this, better results for foreground and background are expected.

Choosing the colour coordinates depends on several factors. On one side, usually, cameras can produce images in RGB or YUV coordinates. Though mathematical methods exist that can convert coordinates of one system to another, this conversion may be very time consuming and have a severe impact on the system performance.

In the following sections, we develop BAC with colour by adding the use of RGB coordinates to the previous algorithm. Other systems exists, such as CIEL*a*b* or HSI which could be claimed to have better properties than RGB. We chose RGB system because it is a widely used system and it is easy to find cameras which reproduce images using this system.

3.1 Colour coordinates

There are different colour models that can be used in order to describe the colour of a pixel, RGB, HSI, CIEL*a*b*. CIEL*a*b*, for instance, has the advantage that is perceptually uniform. That means, that a change of the same amount in a colour value should produce a change of about the same visual importance. The distance between two colours represented in CIEL*a*b* coordinates is just the Euclidean difference of the two vectors representing them.

RGB on the other side, is not perceptually uniform because it was designed from the perspective of devices and not from a human perspective and CIEL*a*b*. Methods exist to convert the RGB coordinates into CIEL*a*b* coordinates and vice versa. In fact, the most common coordinates may be converted into each other through mathematical conversions.

In our case, we will use RGB coordinates. The similarity between two pixels, p and q is now given by a similar function as with gray tones. The only difference is that the distance between the pixels is extended to use the three RGB coordinates.

In this case, equation (1) is modified and the similarity function is given by,

$$S(p,q) = e^{-|dist|/\kappa} : \Re \rightarrow [0,1] \tag{22}$$

where the colour distance is computed as the Euclidean distance of two colours as in equation 23 and κ is a constant determined experimentally.

$$dist = \sqrt{(p_R - q_R)^2 + (p_G - q_G)^2 + (p_B - q_B)^2} \tag{23}$$

3.2 Experiments

Experiments were performed with the same benchmark as with BAC in order to compare results. It is obvious that adding colour to the image processing will improve results, as more information is being used in the segmentation.

Results for the Wallflower benchmark are shown in table 3. In this case, the improvement is evident, a bigger rate of foreground pixels is obtained in all sequences. The chosen segmentation seems to be very sensitive and a lower rate of background pixel is achieved in the sequences.

	Bootstrap	Camouflage	Foreground	Light Switch	Moved Object	Time Of Day	Waving Trees
TP	0.67	0.78	0.58	0.46	-	0.53	0.93
TN	0.85	0.70	0.87	0.97	0.99	0.98	0.59

Table. 3. True positives and true negatives for the Wallflower benchmark using colour in the pixels' description.

4. Comparison with other approaches

We compared the performance of the original BAC algorithm with another approach introduced in the paper by Stauffer and Grimson (Stauffer & Grimson, 1999). We implemented their algorithm and executed it with different parameters in order to seek for best results.

As in the other experiments, we used the Wallflower test for comparisons. We found some difficulties when trying to deal with shadows with this algorithm. Also, we used the parameters which seemed to be the best, keeping them the same for all sequences, as we made with BAC.

	Bootstrap	Camouflage	Foreground	Light Switch	Moved Object	Time Of Day	Waving Trees
TP	0.33	0.80	0.59	0.76	---	0.24	0.66
TN	0.97	0.62	0.55	0.08	1.00	0.99	0.85

Table. 4. True positives and true negatives for the Wallflower benchmark using the Stauffer & Grimson algorithm with a maximum number of models equal to 5, and T= 0.8.

Stauffer's algorithm uses several models per pixel to model the scene, and that is a big difference when motion in the background appears as in camouflage sequence or wavingTree sequence. Results for these two sequences outperform clearly BAC.

Despite results with Stauffer's algorithm could be improved with another set of parameters or initialization, the sequence that better illustrates the performance of BAC is lightSwitch.

In this sequence, a sudden change in light in the scenario is applied. The background rapidly changes from a dark room to an illuminated room.

This sudden corruption of the scene is caught by BAC, which quickly restarts the model. Stauffer's algorithm is slower when reducing the weights of the model to include the new model.

5. Conclusions and future work

A different approach to background modelling was introduced in this chapter. The aim of the algorithms developed is trying to give a quick response to two different problems with a common solution: building a background model and recovering a background model in demanding scenarios.

These scenarios are characterized by having always a significant activity level, making it difficult to obtain a clean model with traditional techniques. Results for the Wallflower benchmark and for the test videos result promising.

Several algorithms have been developed in order to meet the constraints we were facing. First algorithm, MBAC is the simplest of them. Its aim is building a background model and associates a confidence to it, in order to have a numerical description of how good the model is. This algorithm uses gray tone levels to describe the scene.

By adding colour to the BAC algorithm, more accuracy in the background description and the segmentation process is achieved. Results show that, as it was expected, the version of BAC with colour improves results. Other colour systems exists, such as CIEL*a*b* or HSI which could be claimed to have better properties than RGB. We chose RGB system because it is a widely used system and it is easy to find cameras which reproduce images using this system.

The reconstruction of background models on the fly proved to be useful for demanding scenarios, in which it may be difficult achieving good quality background models with traditional techniques. We tested the algorithm in several situations with test videos from different sources, we set a web-site where videos showing algorithm evolution is illustrated.

Further research should be done in improving the segmentation to include also shadows, which proved to be very difficult to classify with our method. Also, a review of the segmentation process should be done. Maybe the fact that RGB coordinates are not perceptual uniform affect the computation of distances and produces a high amount of missed background pixels

6. References

A. Elgammal, D. Harwood & L.S. Davis "Non-parametric model for background subtraction". *ECCV 2000*, pages 751 – 767. 2000.

I. Haritaoglu, D. Harwood & L. S. Davis "W4: Real-time Surveillance of people and their activities". . I*EEE Transactions on PAMI*. vol 22, num. 8, pages 809 – 830. 2000

M. M. Heikkilä & M. Pietikäinen. "A texture-based method for modeling the background and detecting moving objects" *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 28(4):657 - 662, April 2006.

W. Hu, T. Tan, L. Wang & S. Maybank.] "A Survey on Visual Surveillance of Object Motion and Behaviors". I*EEE Transactions on Systems, man, and Cybernetics*, vol 34, num.3. 2004

S. Jabri, Z. Duric, H. Wechsler & A. Rosenfeld. " Detection and location of people in video images unsing adaptive fusion of color and edge information" *Proc. International Conference Pattern Recognition,* pages 627 - 630, 2000.

M. Mason & Z. Duric " Using histograms to detect and track objects in color video". *Proc. Applied Imaginery Pattern Recognition Workshop,* pages 154 - 159, 2001.

J. Rosell-Ortega, G. García-Andreu, A. Rodas-Jordà, V. Atienza-Vanacloig. "Background modelling in demanding situations with confidence measure" *International Conference on Pattern Recognition*. ICPR08. 2008.

B. Shoushtarian & H.E. Bez. "A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking". *Pattern Recognition Letter*s 26, pages. 5 – 26. 2005

C. Stauffer & W. E. L. Grimson. " Adaptive background mixture models for real-time tracking" *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition,* pages 246 - 252, 1999.

K. Toyama, J. Krumm, B. Brumitt & B. Meyers. "Wallflower: Principles and Practice of Background Maintenance". *7th Intl. Conf. on Computer Vision*, Kerkyra, Greece. pages 255-261. 1999

L. Wixson. " Detecting salient motion by accumulating directionally-consistent flow". *IEEE Trans. Pattern Analysis and Machine Intelligence,* (8):774 - 780, 2000.

L. Wang, W. Hu & T. Tan. "Recent developments in human motion analysis". *Pattern Recognition*. pages 585-601,Vol .3 2003.

C. R. Wren, T. D. A. Azarbayenjani, & A. P. Pentland " Pfinder: rel-time tracking of the human body". *IEEE Trans. Pattern Analysis and Machine Intelligence*, (7):780 - 785, 1997.

Q. Zang & R. Klette. " Robust background subtraction and maintenance" *Proc. International Cong. Pattern Recognition*, pages 90 – 93, 2004.

New Trends in Motion Segmentation

Luca Zappella, Xavier Lladó and Joaquim Salvi
University of Girona, Girona, Spain

1. Introduction

Motion segmentation algorithms aim at decomposing a video in moving objects and background. In many computer vision tasks this decomposition is the first fundamental step. It is an essential building block for robotics, inspection, metrology, video surveillance, video indexing, traffic monitoring and many other applications. A great number of researchers has focused on the segmentation problem and this testifies the relevance of the topic. However, despite the vast literature, the performance of most of the algorithms still falls far behind human perception. In this chapter a review of the main motion segmentation approaches is presented, with the aim of pointing out their strengths and weaknesses and suggesting new research directions. The main features of motion segmentation algorithms are analysed and a classification of the recent and most important techniques is proposed. The conclusions summarise the review and present a vision on the future of motion segmentation algorithms.

2. State of the art

In this section a complete state of the art review on motion segmentation is presented. First the main problems and attributes of motion segmentation algorithms are analysed. Afterwards, a classification of the different techniques is proposed, describing the most significant works in this field. All the papers revised are summarised in table 1, which offers a compact at-a-glance overview with respect to the attributes presented in the next subsection.

2.1 Problems and attributes

In this subsection the common problems and the most important attributes of motion segmentation algorithms are analysed. Attributes describe in a compact way the assumptions made by an algorithms as well as its limitations and strengths.

One of the first choice that has to be taken when developing a motion segmentation algorithm is the *representation* of the motions: there are *feature-based* and *dense-based* approaches. In feature-based methods, the objects are represented by a limited number of salient points. Most of these methods rely on computing a homography corresponding to the motion of a planar object (Kumar et al, 2008). Features represent only part of an object hence the object can be tracked even in case of partial occlusions. In opposition to feature-based methods there are dense-based methods which do not use sparse points but compute a pixel-wise motion. The result is a more precise segmentation of the objects but the occlusion problem becomes harder to solve (Kumar et al, 2008).

Motion segmentation algorithms usually exploit temporal continuity. However, using only temporal clues a rather big part of the available information is thrown away and this lack of information can easily lead to problems. This is the reason why some techniques exploit also *spatial continuity*. In these cases each pixel is not considered as a single point but the information provided by its neighbourhood (in terms of spatial proximity) is taken into account. For example, one of the problems that are caused by the use of temporal information only, is the ability to deal with *temporary stopping*. In fact, many techniques Fail to segment when the objects stop moving even for a limited amount of time.

Another common problem of motion segmentation is the fact that objects that move are not always visible. The ability to deal with *missing data* is yet one of the most difficult problems. Missing data can be caused by many factors: presence of noise, occlusions, or feature points that are not in scene for the whole length of the sequence. The presence of noise is another cause of failure. Noise can affect the accuracy in the position of the tracked features, or the amount of outliers (erroneously tracked features). Hence, the *Robustness* of the algorithm against noise is an essential factor to take into account. For simplicity, in this chapter the term ˝robustness“ groups together the ability to deal with all the problems caused by noise, as well as the robustness against initialization (when an initial solution is required).

Another important attribute that has to be analysed is the ability to deal with different types of motion. There is a bit of confusion in the literature when it comes to "type of motion" as people tend to use different adjectives to describe the same property of the motion or the same adjective to describe different properties. Hence, it is important to clarify which is the exact meaning that is given to each adjective in this chapter. A motion can be described in terms of: *dependency* and *kind*.

The first classification is between independent and dependent motions. This is an attribute that describe the relationship between a pair of motions and is not a feature of one single motion. Motions are *independent* if the pairwise intersection of the generated subspaces is the zero vector. On the other hand, motions are *dependent* if the pairwise intersection of the subspaces is not empty. In this case the two motions can be seen as "similar", the dependency can be partial (which means that the subspaces intersect in some points) or complete (which means that one subspace is completely inside the other) (Rao et al, 2008).

The kind of motion is an attribute of the single motion. A motion is *rigid* when the trajectories generated by the points of a rigid object form a linear subspace of dimensions no more than 4 (Tomasi and Kanade, 1992). It is *non-rigid* if the trajectories generated by the points of a non-rigid object can be approximated by a combination of k weighted key basis shapes, and they form a linear subspace of dimension no more than 3k + 1 (Koterba et al, 2005; Llado et al, 2006). It has to be noted that the ability to deal with non-rigid motions is constrained to when the nonrigid structure has also a rigid motion component during its movement. And finally, a motion is *articulated* when is composed by two dependent motions M1 and M2 connected by a link. If the link is a joint, [R1|T1] and [R2|T2] must have T1 = T2 under the same coordinate system. Therefore, M1 and M2 lie in different linear subspaces but have 1-dimensional intersection. If the link is an axis, [R1|T1] and [R2|T2] must have T1 = T2 and exactly one column of R1 and R2 being the same under a proper coordinate system. So M1 and M2 lie in different linear subspaces but have 2-dimensional intersection (Yan and Pollefeys, 2006).

These are all the attributes, related to the description of motion, that will be taken into account in table 1. However, not always the authors clearly state under which conditions the algorithm would work, therefore the table is filled to the best of our knowledge given the information provided in the cited papers. There would be two more Attributes that, for the sake of completeness, are described here but they are not considered in the table as very few authors clearly explain these aspects. The first is called in this article "degeneracy". Many authors use it when they refer to dependent, non-rigid or articulated motions, but it is used here with a different meaning. Degeneracy is another aspect of a single motion. *Non Degenerate Motion* is a motion whose subspace dimension is the maximum (i.e. 4 for rigid motion, 3k + 1 for nonrigid motion, etc.). Whereas, *Degenerate Motion* is a motion whose subspace have a dimension which is lower than its theoretical maximum due to some degeneracies in the trajectories. The second attribute is the assumed camera model, which can be *affine, perspective, para-perspective or projective*

Furthermore, if the aim is to develop a generic algorithm able to deal in many unpredictable situations there are some algorithm features that may be considered as a drawback. For instance, one important aspect is the amount of *prior knowledge* required. In particular: number of moving objects and dimension of the generated subspaces. A second aspect is the fact that some algorithm require a *training* step. Training is not a negative point itself, however a trained algorithm tends to lose generality and it requires extra effort and a relevant amount of data that is not always available.

2.2 Strategies analysis

As motion segmentation has been a hot topic for many years its literature is particularly wide. In order to make the overview easier to read and to create a bit of order, the approaches will be divided into categories which represent the main principle underlying the algorithm. For each category some articles, among the most representative and the newest proposals, are provided. The division is not meant to be tight, in fact some of the algorithms could be placed in more than one category. The groups identified are: Image Difference, Statistical, Optical Flow, Wavelets, Layers, and Manifolds Clustering. As the amount of literature is notable only the main idea of each group of techniques is described while details about each paper are presented in the table 1.

2.2.1 Image difference

Image difference is one of the simplest and most used techniques for detecting changes. It consists in thresholding the intensity difference of two consecutive frames pixel by pixel. The result is a coarse map of the temporal changes. An example of an image sequence and the image difference result is shown in figure 1. Despite its simplicity, this technique cannot be used in its basic version because it is really sensitive to noise. Moreover, when the camera is moving the whole image changes and, if the frame rate is not high enough, the result would not provide any useful information. Works based on image difference usually focus on these two problems. For example, in (Cavallaro et al, 2005) the authors reinforce the motion difference using a probability-based test in order to change the threshold locally. In (Cheng and Chen, 2006) they exploit the wavelet decomposition in order to reduce the noise. Other proposals, like (Li et al, Aug. 2007), try to use temporal and spatial information simultaneously to be able to deal with noise and to solve other typical

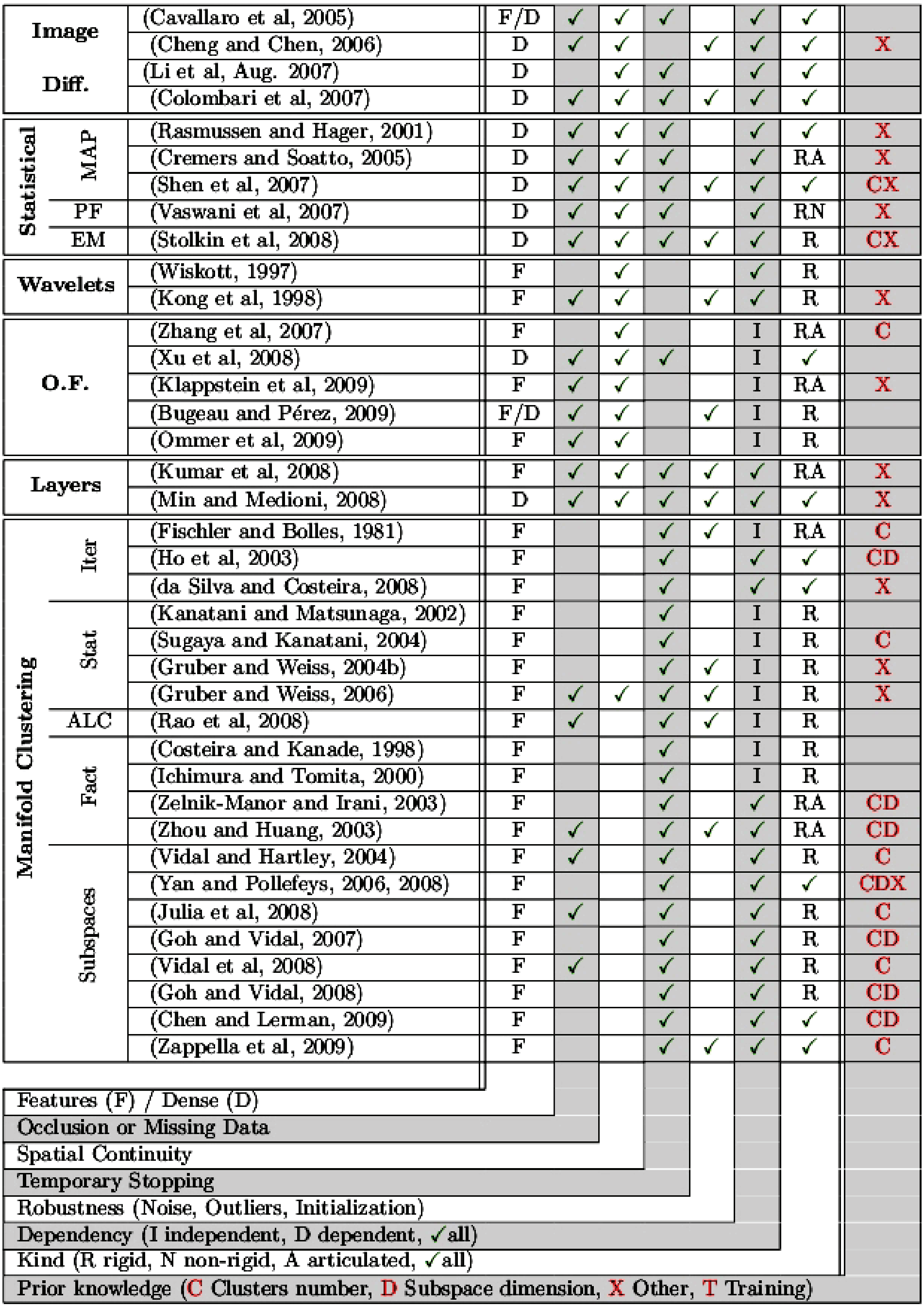


Image Diff.		(Cavallaro et al, 2005)	F/D	✓	✓	✓		✓	✓	
		(Cheng and Chen, 2006)	D	✓	✓		✓	✓	✓	X
		(Li et al, Aug. 2007)	D		✓	✓		✓	✓	
		(Colombari et al, 2007)	D	✓	✓	✓	✓	✓	✓	
Statistical	MAP	(Rasmussen and Hager, 2001)	D	✓	✓	✓		✓	✓	X
		(Cremers and Soatto, 2005)	D	✓	✓	✓		✓	RA	X
		(Shen et al, 2007)	D	✓	✓	✓	✓	✓	✓	CX
	PF	(Vaswani et al, 2007)	D	✓	✓	✓		✓	RN	X
	EM	(Stolkin et al, 2008)	D	✓	✓	✓	✓	✓	R	CX
Wavelets		(Wiskott, 1997)	F		✓			✓	R	
		(Kong et al, 1998)	F	✓	✓		✓	✓	R	X
O.F.		(Zhang et al, 2007)	F		✓			I	RA	C
		(Xu et al, 2008)	D	✓	✓	✓		I	✓	
		(Klappstein et al, 2009)	F	✓	✓			I	RA	X
		(Bugeau and Pérez, 2009)	F/D	✓	✓		✓	I	R	
		(Ommer et al, 2009)	F	✓	✓			I	R	
Layers		(Kumar et al, 2008)	F	✓	✓	✓	✓	✓	RA	X
		(Min and Medioni, 2008)	D	✓	✓	✓	✓	✓	✓	X
Manifold Clustering	Iter	(Fischler and Bolles, 1981)	F			✓	✓	I	RA	C
		(Ho et al, 2003)	F			✓		✓	✓	CD
		(da Silva and Costeira, 2008)	F			✓		✓	✓	X
	Stat	(Kanatani and Matsunaga, 2002)	F			✓		I	R	
		(Sugaya and Kanatani, 2004)	F			✓		I	R	C
		(Gruber and Weiss, 2004b)	F			✓	✓	I	R	X
		(Gruber and Weiss, 2006)	F	✓	✓	✓	✓	I	R	X
	ALC	(Rao et al, 2008)	F	✓		✓	✓	I	R	
	Fact	(Costeira and Kanade, 1998)	F			✓		I	R	
		(Ichimura and Tomita, 2000)	F			✓		I	R	
		(Zelnik-Manor and Irani, 2003)	F			✓		✓	RA	CD
		(Zhou and Huang, 2003)	F	✓		✓	✓	✓	RA	CD
	Subspaces	(Vidal and Hartley, 2004)	F	✓		✓		✓	R	C
		(Yan and Pollefeys, 2006, 2008)	F			✓		✓	✓	CDX
		(Julia et al, 2008)	F	✓		✓		✓	R	C
		(Goh and Vidal, 2007)	F			✓		✓	R	CD
		(Vidal et al, 2008)	F	✓		✓		✓	R	C
		(Goh and Vidal, 2008)	F			✓		✓	R	CD
		(Chen and Lerman, 2009)	F			✓		✓	✓	CD
		(Zappella et al, 2009)	F			✓	✓	✓	✓	C

Features (F) / Dense (D)
Occlusion or Missing Data
Spatial Continuity
Temporary Stopping
Robustness (Noise, Outliers, Initialization)
Dependency (I independent, D dependent, ✓all)
Kind (R rigid, N non-rigid, A articulated, ✓all)
Prior knowledge (C Clusters number, D Subspace dimension, X Other, T Training)

Table 1. Summary of the examined techniques with respect to the most important attributes.

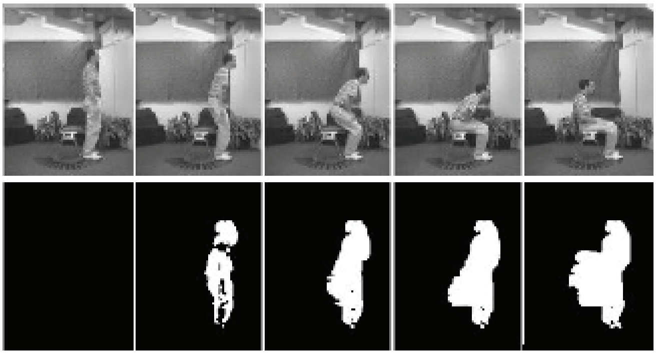


Fig. 1. Example of an image difference result (Bobick and Davis, 1996).

problems of image difference techniques such as dealing with temporary stopping. Another example is (Colombari et al, 2007), where in order to deal with noise and very small camera movements the authors propose a robust statistic to model the background.

As can be seen from the table 1, image difference is mainly based on dense representation of the objects. It combines simplicity and good overall results being able to deal with occlusions, multiple objects, independent motions, non-rigid and articulated objects. The main problem of these techniques is the difficulty to deal with temporary stopping and with moving cameras. In order to be successful in these situations a history model of the background needs to be built. Furthermore, image difference algorithms are still very sensitive to noise and to light changes, hence they cannot be considered an ideal choice in case of cluttered background.

2.2.2 Statistical framework

Statistical theory is widely used in the motion segmentation field. In fact, motion segmentation, in the simple case, can be seen as a classification problem where each pixel has to be classified as background or foreground. Statistical approaches can be further divided depending on the framework used. Common frameworks are Maximum A posteriori Probability (MAP), Particle Filter (PF) and Expectation Maximization (EM). Statistical approaches provide a general tool that can be used in very different ways depending on the specific technique.

In (Rasmussen and Hager, 2001), a MAP framework is used, namely they use the Kalman Filter and the Probabilistic Data Association Filter, to predict the most likely location of a known target in order to initialize the segmentation process. Another technique based on MAP is (Cremers and Soatto, 2005), where level sets (Sethian, 1998) are used to incorporate motion information. In (Shen et al, 2007), MAP formulation is proposed to iteratively update the motion fields and the segmentation fields along with the high-resolution image. The formulation is solved by a cyclic coordinate descent process that treats motion, segmentation, and high-resolution image as unknowns, and estimates them jointly using the available data. Another widely used statistical framework is PF. The main aim of PF is to track the evolution of a variable over time. The basis of the method is to construct a sample-based representation of the probability density function. Basically, a series of actions are taken, each of them modifying the state of the variable according to some model. Multiple copies of the variable state (particles) are kept, each one with a weight that signifies the quality of that specific particle. An estimation of the variable can be computed as a weighted sum of all the particles. The PF algorithm is iterative and each iteration is composed by prediction and update. After each action particles are modified according to the model (prediction), then each particle weight is re-evaluated according to the information extracted from an observation (update). At every iteration, particles with small weights are eliminated (Rekleitis, 2003). An example of PF applied to motion segmentation is (Vaswani et al, 2007), where some well known algorithms for object segmentation using spatial information, such as geometric active contours (Blake, 1999) and level sets (Sethian, 1998), are unified using a PF framework.

EM is also a frequently exploited framework in motion segmentation. The EM algorithm is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in presence of missing or hidden data. In ML the aim is to estimate the model parameter(s) for which the observed data is most likely to belongs to. Each iteration of the EM algorithm consists of an E-step and an M-step. In the E-step, using the conditional expectation the missing data are estimated. Whereas, in the M-step the likelihood function is maximized. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration (Borman, 2004). An example of EM applied to motion segmentation is (Stolkin et al, 2008), where the authors present an algorithm which uses EM and Extended-Markov Random Field (E-MRF). In order to track the camera trajectory (egomotion), the algorithm merges the observed data (the current image) with the prediction derived from prior knowledge of the object being viewed. The merging step is driven by the E-MRFs within a statistical framework.

Statistical approaches use mainly dense based representation. They work well with multiple objects and can deal with occlusions and temporary stopping. In general they are robust as long as the model reflects the actual situation but they degrade quickly as the model fails to represent the reality. Moreover, most of the statistic approaches require some kind of a priori knowledge.

2.2.3 Wavelets

Another group of motion segmentation algorithms is based on wavelets analysis. These methods exploit the ability of wavelets to perform analysis of the different frequency components of the images, and then study each component with a resolution matched to its scale. Usually wavelet multi-scale decomposition is used in order to reduce the noise and in conjunction with other approaches, such as optical flow, applied at different scales. However, there are a few proposals where wavelet is the main segmentation algorithm. In (Wiskott, 1997) the author combines Gabor and Mallat wavelet transform to overcome the aperture problem and the correspondence problem. The former transform is used to estimate the motion field and roughly cluster the image, while the latter is used to refine the clustering. The main limitation of this model is that it assumes that the objects only translate in front of the camera. A different approach is presented in (Kong et al, 1998) where the motion segmentation algorithm is based on Galilean wavelets. These wavelets behave as matched filters and perform minimum mean-squared error estimations of velocity, orientation, scale and spatio-temporal positions. This information is finally used for tracking and segmenting the objects.

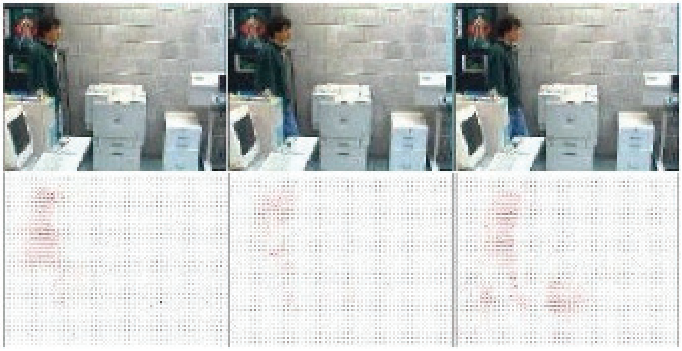


Fig. 2. Example of OF, in red the vectors of the flow of the moving person

Wavelets solutions seem to provide overall good results but limited to simple cases (such as translation in front of the camera). Wavelets were in fashion during the 90s, nowadays the research interest seems to be less active, at least in relation to motion segmentation.

2.2.4 Optical flow

Optical flow (OF) can be defined as the apparent motion of image brightness patterns in an image sequence. An example of OF can be seen in figure 2. Like image difference, OF is an old concept greatly exploited in computer vision. It was first formalized and computed for image sequences by Horn and Schunck in the 1980 (Horn and Schunck, 1980). However, the idea of using discontinuities in the optical flow in order to segment moving objects is even older, in (Horn and Schunck, 1980) there is a list of older methods based on this idea but they all assume the optical flow is already known. Since the work of Horn and Schunck, many other approaches have been proposed. In the past, the main limitation of such methods was the high sensitivity to noise and the high computational cost. Until recently, OF was more often used in hardware implementations in order to overcome the computational cost, as in (Jos et al, 2005). Nowadays, thanks to the high computational speed and to improvements made by research, OF is widely used also in software implementation. In (Xu et al, 2008) is presented a variational formulation of OF combined with color segmentation obtained by the Mean-shift algorithm. The authors of (Klappstein et al, 2009) exploit OF in order to built a robust obstacle detection for driver assistance purposes. The work is done both with monocular (exploiting some motion constraints) and stereo (using Extended Kalman Filter) vision. In (Bugeau and Pérez, 2009) the segmentation problem is addressed by combining motion information, spatial continuity and photometric information. In (Ommer et al, 2009) an algorithm for segmentation, tracking and object recognition is presented. The segmentation and tracking parts are done by OF (using salient features and KLT tracking). The algorithm is based on grouping together salient features following a proximity criteria. The features are tracked by KLT and the mean flow is computed. The position of the group of features is predicted using the previous mean flow in order to constrain the tracking area. At every iteration the mean flow is updated taking into account the old flows with an exponential decay over time.

OF is, theoretically, a good clue in order to segment motion. However, alone it is not enough because it cannot deal with occlusions and temporal stopping. Statistical techniques or

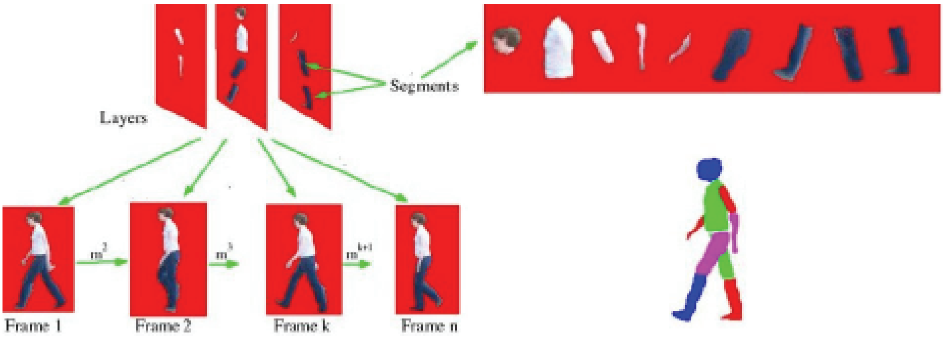


Fig. 3. Example of layers segmentation (Kumar at al, 2008)

spatial analysis (like colour or texture) could help to increase the robustness as OF is still very sensitive to noise and light changes.

2.2.5 Layers

The first layers technique was proposed by J. Wang and E. Adelson in 1993 (Wang and Adelson, 1993). The key idea of layers based techniques is to divide the image into layers with uniform motion. Furthermore, each layer is associated with a depth level and a "transparency" level that determines the behaviour of the layers in case of overlapping. This approach is often used in stereo vision as the depth distance can be recovered easily. However, even without computing the depth it is possible to estimate which objects move on similar planes. This is extremely useful as it helps to solve the occlusion problem. Recently, new interest raised around this idea (Kumar et al, 2008; Min and Medioni, 2008). The authors of (Kumar et al, 2008) propose a method for learning a layered representation of the scene. They initialize the algorithm by first finding coarse moving components between every pair of frames. They divide the image in patches and find the rigid transformation that moved the patch from frame j to frame j + 1. The initial estimate is then refined using αβ-swap and α-expansion algorithms (Boykov et al, 1999). More recently, in (Min and Medioni, 2008), a new layer based technique was presented. This technique exploits a 5 dimensional representation of each feature, the 5D token is composed by: position (x, y), time (t), and velocity (v_x, v_y). The layers are seen as 3D variety which can be extracted from the 5D tensor (using neighbours tokens) by tensor voting framework. In order to produce accurate results pre-segmented areas based on color segmentation (performed by Mean-shift) are required. An example of a layer segmentation is shown in figure 3.

Layers solutions are very interesting. It is probably the most natural solution for the occlusion problem: human beings also use depth perception to solve this issue. The main drawback is the level of complexity of these algorithms and the number of parameters that have to be tuned manually. Furthermore, a deeper evaluation should be carried out as none of the presented algorithms has exhaustive tests with more than two motions.

2.2.6 Manifold clustering

Manifold clustering techniques consist in projecting the original data into a smaller space (if necessary, otherwise the ambient space could be directly used) and trying to cluster together data that has common properties by, for example, fitting a set of hyperplanes to the data. Nowadays, manifold clustering is a "hot" topic and it is applied in many fields. Segmentation seems one of the most natural applications, particularly motion segmentation. This class of solutions is usually based on feature points. They provide not only the segmentation but they can be naturally extended to Structure from Motion (SfM) in order to recover the 3D structure of the objects and the motion of the camera. Furthermore, they do not have any problem with temporary stopping because features can be tracked even if the object is not moving (provided that this is a temporary situation). Most of these techniques assume an affine camera model, however, it is possible to extend them to the projective case by an iterative process as shown in (Li et al, 2007). A common drawback to all these techniques is that they can deal very well when the assumptions of rigid, independent and non degenerate motions, are respected, but if one of these assumptions fails, then problems start to arise as the properties of motions have to be taken into account explicitly. This group of techniques is rather big, hence, a further classification helps to give some order. Manifold clustering can be divided into, Iterative solutions, Statistical solutions (solutions that fall inside this category could be placed in the previous Statistical group, but in this case we refer to statistical frameworks specifically applied to manifold clustering), Agglomerate Lossy Compression (ALC), Factorization solutions and Subspace Estimation solutions.

An iterative solution is presented in (Fischler and Bolles, 1981) where the RANdom SAmple Consensus (RANSAC) algorithm is used. RANSAC tries to fit a model to the data randomly sampling n points, then it computes the residual of each point to the model and those points whose residual is below a threshold are considered inliers. The procedure is repeated until the number of inliers is above a threshold, or enough samples have been drawn. Another iterative algorithm called "K-Subpsace Clustering" is presented in (Ho et al, 2003) for face clustering, however, the same idea could be adopted to solve the motion segmentation problem. K-Subpsace can be seen as a variant of K-means. K-Subspace iteratively assigns points to the nearest subspace, than that subspace is updated computing the new bases that minimize the sum of the square distances to all the points of that cluster. The algorithm ends after a predefined number of iterations. The authors of (da Silva and Costeira, 2008) propose a subspace segmentation algorithm based on a Grassmannian minimization approach. The technique consists in estimating the subspace with the maximum consensus (MCS): maximum number of data that are inside the subspace. Then, the algorithm is recursively applied to the data inside the subspace in order to look for smaller subspaces included in it.

Iterative approaches are in general robust to noise and outliers, and they provide good solutions if the number of clusters and the dimension of the subspaces are known. This prior knowledge can be clearly seen as their limitation as this information is not always available. Moreover, they require an initial estimation and they are not robust against bad initializations, so when the initialization is not close enough to the correct solution the algorithms are not guaranteed to converge.

Another manifold clustering group is composed by statistical solutions. In (Kanatani and Matsunaga, 2002) the authors use a statistical framework for detecting degeneracies of a geometric model. They use the geometric information criterion (AIC) defined in (ichi Kanatani, 1997) in order to evaluate whether two clouds of points should be merged or not.

Another statistical based technique is (Sugaya and Kanatani, 2004). This paper analyses the geometric structure of the degeneracy of the motion model, and suggests a multi-stage unsupervised learning scheme first using the degenerate motion model and then using the general 3-D motion model. The authors of (Gruber and Weiss, 2004b) extend the EM algorithm already proposed in (Gruber and Weiss, 2004a) for the single object case in order to deal with multiple objects and missing data. In (Gruber and Weiss, 2006) the same authors further extend the method incorporating non-motion cues (such as spatial coherence) into the M step of the algorithm.

Statistical solutions have more or less the same strengths and weaknesses of iterative techniques. They can be robust against noise whenever the statistical model is built taking the noise explicitly into account. However, when noise is not considered or is not modeled properly their performances degenerate rapidly. As previously said for general statistical approaches: they are robust as long as the model reflects the actual situation.

A completely different idea is the basis of (Rao et al, 2008) which uses the Agglomerative Lossy Compression (ALC) algorithm (Ma et al, 2007). This technique consists in minimizing a cost function by grouping together trajectories. Roughly speaking, the cost function is given by the amount of information required to represent each manifold given a particular segmentation.

ALC provides a connection between coding theory and space representation. It performs extremely well with a variety of motions. However, it has some problems to deal with a lot of data (curse of dimensionality). Furthermore, the algorithm depends on a parameter that has to be tuned per each sequence depending on the number of motions and the amount of noise. Although the tuning can be automated trying many different values and choosing at the end the solution with the lowest cost, this process is highly time-consuming.

Factorization techniques are based on the approach introduced by Tomasi and Kanade in 1992 (Tomasi and Kanade, 1992) to recover structure and motion using features tracked through a sequence of images. In (Costeira and Kanade, 1998) the framework of Tomasi and Kanade is first used in order to factorize the trajectory matrix W by Singular Value Decomposition into the matrices U, D, and V. Then a matrix called "shape interaction matrix" $Q = VV^T$ is built. The shape interaction matrix has, among other properties, zero entries if the two indexes represent features belonging to different objects, non-zero otherwise. Hence, the algorithm focuses on finding the permutation of the interaction matrix that gives a block diagonal matrix structure as shown in figure 4. In (Ichimura and Tomita, 2000), once the rank r of the trajectory matrix is estimated they perform the QR decomposition of the shape interaction matrix and select the r bases of the shape space which gives the segmentation among those features. Finally, the remaining features are segmented by using the orthogonal projection matrix. The two previous factorization techniques assume that the objects have independent motions. In (Zelnik-Manor and Irani, 2003) the authors study the degeneracy in case of dependent motion. They propose a factorization method that consists in building an affinity matrix by using only the dominant eigenvector and estimating the rank of the trajectory matrix by studying the ratio between the eigenvalues. In (Zhou and Huang, 2003) a hierarchical factorization method for recovering articulated hand motion under weak perspective projection is presented. They consider each part of the articulated object as independent and they use any of the techniques able to deal with missing data to fill the gaps. In the second step, they guarantee that the end of the consecutive objects are linked in the recovered motion.

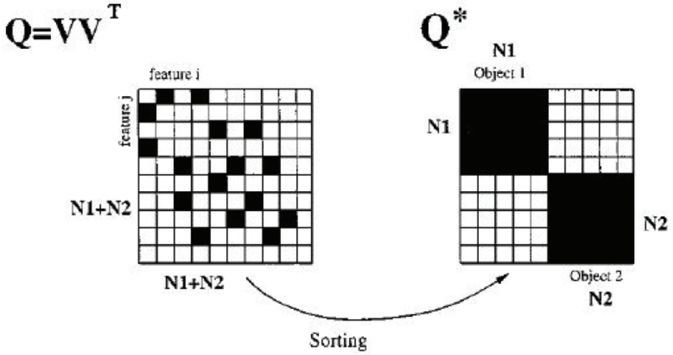


Fig. 4. (Costeira and Kanade, 1998) computes the interaction matrix Q and finds the permutation of rows and columns that gives a block diagonal matrix.

Factorization techniques are based on a very simple and elegant framework. However, factorization methods are particularly sensitive to noise and cannot deal very well with outliers. Moreover, most of these techniques assume rigid and independent motions.

The last category of manifold clustering is the subspace estimation techniques.

The work presented in (Vidal and Hartley, 2004) belongs to this group. First, exploiting the fact that trajectories of rigid and independent motion generate subspaces at most of dimension four, they project the trajectories onto a five dimensional space using PowerFactorization. Then, the Generalized Principal Component Analysis (GPCA) is used to fit a polynomial of degree n, where n is the number of subspaces (i.e. the number of motions), through the data and estimate the bases of the subspaces using the derivatives of the polynomial. More recently, the same authors in (Vidal et al, 2008) extended the previous explained framework using RANSAC to perform the space projection in order to be able to deal with outliers. Another well known technique is the Local Subspace Affinty (LSA) (Yan and Pollefeys, 2006, 2008). LSA is able to deal with different types of motion: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. The key idea is that different motion trajectories lie in subspaces of different dimension. Thus, the subspaces are estimated and an affinity matrix is built using principal angles. The final segmentation is obtained by clustering the affinity matrix. The main limitations of LSA are the difficulty of estimating the size of the global and local subspaces without manual tuning, and the fact that a full trajectory matrix without missing data is assumed. In (Julia et al, 2008) a technique similar to LSA is presented in order to deal with missing data. The idea is to fill the missing data using a frequency spectra representation for the matrix estimation. When a full trajectory matrix is obtained an affinity matrix is built and a cluster algorithm based on normalized cuts is applied in order to provide the segmentation. In (Chen and Lerman, 2009) the authors propose a generalization

of LSA called Spectral Curvature Clustering (SCC). SCC differs from LSA for two main reasons. The first reason is related to the affinity measure, SCC uses polar curvature while LSA uses principal angles. In SCC the affinity between a point i and the other points is given by the polar curvature of the space generated by i and some combination of other d + 1 points (where d is the size of the generated subspace). The second reason is how they select which points have to be combined with I: SCC uses an iterative solution while LSA uses a nearest neighbour solution. Theoretically, in SCC all the possible combination of points should be tried but this may not be computationally feasible, instead, only one combination of d + 1 points is randomly selected among the points that belong to the same cluster of i. Of course, the first time this selection is done, there is no information about which point belong to which cluster, hence at the first iteration the points are randomly selected among all of them. At the second iteration, the clustering result of the first iteration is used to constrain the selection among the points that were clustered with i. A completely different strategy is presented in (Goh and Vidal, 2007) where, starting from the Locally Linear Embedding algorithm (Saul and Roweis, 2003), they propose the Locally Linear Manifold Clustering Algorithm (LLMC). With LLMC the authors try to deal with linear and non-linear manifolds. The same authors extended this idea to Riemannian manifolds (Goh and Vidal, 2008). They project the data from the Euclidean space to a Riemannian space and reduce the clustering to a central clustering problem. Finally, in (Zappella et al, 2009) the authors enforce the LSA algorithm proposing a new Enhanced Model Selection (EMS) technique. EMS is a generic rank estimation tool, in this case it is used in order to estimate the size of the global and local subspaces in an automatic fashion, auto-tuning the parameters in order to deal with different noise conditions and different number of motions.

Subspace estimation techniques can deal with intersection of the subspaces and generally they do not need any initialization. However, all these techniques suffer from common problems: curse of dimensionality, weak estimations of number of motions and subspaces dimension. The curse of dimensionality is mainly solved in two ways: projection into smaller subspaces or random sampling. Whereas the number of motions and the subspace dimension estimations are commonly two open issues.

3. Discussions and conclusions

Table 2 summarises and generalises the advantages and disadvantages of each group of techniques. This review should have given an idea of how vast the motion segmentation literature is, and the fact that research in this field is still active (most of the papers presented here were published after 2005) is a sign of the importance of this problem. On the other hand, effervescent research activity signifies also that many problems have still to be solved and there is not an outstanding solution yet. From the analysis it is possible to state that manifold clustering algorithms seems one of the most natural solutions for motion segmentation. Recently manifold clustering has been studied and exploited deeply in order to solve the motion segmentation problem. This class of techniques have already good performances, nevertheless there is space for further improvements. A quick glance at table 1 may catch the attention on the fact that for manifold clustering techniques, the price to pay in order to be able to deal with different kind of motions and with dependent motions is a higher amount of prior knowledge (in particular about the dimension of the generated subspaces). The amount of prior knowledge is another limitation that in future should be overcome. In order to obtain more robust results it would be interesting to study different ways of merging spatial information, and to exploit the ability of statistical frameworks to find hidden information and outliers.

Techniques		Pros		Cons	
Image Diff.		- Simple	- Dependency	- Sensitive to noise	- Moving camera
		- Occlusions	- Kind of motion	- Temporary stopping	
Statistical		- Occlusions		- Sensitive to model	- Dependency
		- Temporary stopping		- Prior knowledge	
Wavelets		- Depth estimation	- Dependency	- Multiple motions	- Kind of motion
O.F.		- Simple		- Dependency	- Sensitive to noise
				- Non-rigid motions	
Layers		- Occlusions		- Complexity	- Many Parameters
Manifold Clustering	Iter	- Extension to SfM		- Prior knowledge	- Occlusions
		- Temporary stopping		- Sensitive to initialization	- Dependency
				- Kind of motion	
	Stat	- Extension to SfM		- Prior knowledge	- Occlusions
		- Temporary stopping		- Dependency	- Sensitive to model
				- Kind of motion	
	ALC	- Extension to SfM	- Occlusions	- Time consuming	- No justification
		- Temporary stopping	- Misclassification	- Dependency	- Curse of dimensionality
	Fact	- Extension to SfM	- Elegant	- Prior knowledge	- Occlusions
		- Temporary stopping		- Dependency	- Sensitive to noise
				- Kind of motion	
	Sub	- Extension to SfM	- Misclassification	- Prior knowledge	- Occlusions
		- Temporary stopping		- Curse of dimensionality	

Table 2. Summary and generalisation of pros and cons of each group of techniques.

Nowadays the misclassification rates knowing the number of motions are already quite good. Despite the fact that the misclassification rates could be further improved, it is the opinion of the authors that future works should focus on the ability to estimate the number of clusters in a more efficient way. In general feature based techniques are preferred over dense based approaches as the amount of computation required by dense approaches is very large. However, feature based techniques have to rely on the ability of the tracker to find salient points and track them successfully through the video sequence. Today, such an assumption is not too constraining but it is important to develop algorithms able to deal only with few points (from four to six) per motion instead of requiring lots of them. Moreover, in order to have a useful system for real time applications, future motion segmentation algorithms should be able to work incrementally. An ideal incremental algorithm should be able to refine the segmentation at every new frame (or every group of few frames) without recomputing the whole solution from the beginning.

4. Acknowledgements

This work has been supported by the Spanish Ministry of Science projects DPI2007-66796-C03-02 and DPI2008-06548-C03-03/DPI. L. Zappella is supported by the Catalan government scholarship 2007FI_A 00765.

5. References

Blake A (1999) Active contours. Robotica 17(4):459–462

Bobick A, Davis J (1996) An appearance-based representation of action. IEEE InternationalConference on Pattern Recognition pp 307–312

Borman S (2004) The expectation maximization algorithm – a short tutorial

Boykov Y, Veksler O, Zabih R (1999) Fast approximate energy minimization via graph cuts. In: International Conference on Computer Vision, pp 377–384

Bugeau A, Perez P (2009) Detection and segmentation of moving objects in complex scenes. Computer Vision and Image Understanding 113:459–476

Cavallaro A, Steiger O, Ebrahimi T (2005) Tracking Video Objects in Cluttered Background. IEEE Transactions on Circuits and Systems for Video Technology 15(4):575–584

Chen G, Lerman G (2009) Spectral curvature clustering (scc). International Journal of Computer Vision 81:317–330

Cheng FH, Chen YL (2006) Real time multiple objects tracking and identification based on discrete wavelet transform. Pattern Recognition 39(6):1126–1139

Colombari A, Fusiello A, Murino V (2007) Segmentation and tracking of multiple video objects. Pattern Recognition 40(4):1307–1317

Costeira JP, Kanade T (1998) A multibody factorization method for independently moving objects. International Journal of Computer Vision 29(3):159–179

Cremers D, Soatto S (2005) Motion competition: A variational approach to piecewise parametric motion segmentation. International Journal of Computer Vision 62(3):249–265

Fischler MA, Bolles RC (1981) Ransac random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24:381–395

Goh A, Vidal R (2007) Segmenting motions of different types by unsupervised manifoldclustering. IEEE Conference on Computer Vision and Pattern Recognition pp 1–6

Goh A, Vidal R (2008) Clustering and dimensionality reduction on riemannian manifolds. In: IEEE Conference on Computer Vision and Pattern Recognition

Gruber A, Weiss Y (2004a) Factorization with uncertainty and missing data: exploiting temporal coherence. Advances in Neural Information Processing Systems

Gruber A, Weiss Y (2004b) Multibody factorization with uncertainty and missingdata using the em algorithm. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1:707–714

Gruber A, Weiss Y (2006) Incorporating non-motion cues into 3d motion segmentation. In: European Conference on Computer Vision, pp 84–97

Ho J, Yang MH, Lim J, Lee KC, Kriegman D (2003) Clustering appearances of objects under varying illumination conditions. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 1, pp 11–18

Horn BK, Schunck BG (1980) Determining optical flow. Tech. rep., Cambridge, MA, USA

Ichimura N, Tomita F (2000) Motion segmentation based on feature selection from shape matrix. Systems and Computers in Japan 31(4):32–42

Jos LM, Zuloaga A, Cuadrado C, Lzaro J, Bidarte U (2005) Hardware implementation of optical flow constraint equation using fpgas. Computer Vision and Image Understanding 98(3):462–490

Julia C, Sappa A, Lumbreras F, Serrat J, Lopez A (2008) Rank estimation in 3d multibody motion segmentation. Electronics Letters 44(4):279–280

Ichi Kanatani K (1997) Statistical optimization and geometric visual inference. In: AFPAC '97: Proceedings of the International Workshop on Algebraic Frames for the Perception-Action Cycle, Springer-Verlag, pp 306–322

Kanatani K, Matsunaga C (2002) Estimating the number of independent motions for multibody motion segmentation. In: Proceedings of the Fifth Asian Conference on Computer Vision, vol 1, pp 7–12

Klappstein J, Vaudrey T, Rabe1 C, Wedel A, Klette R (2009) Moving object segmentation using optical flow and depth information. In: Pacific-Rim Symposium on Image and Video Technology, p 611623

Kong M, Leduc JP, Ghosh B, Wickerhauser V (1998) Spatio-temporal continuous wavelet transforms for motion-based segmentation in real image sequences. Proceedings of the International Conference on Image Processing 2:662–666

Koterba S, Baker S, Matthews I, Hu C, Xiao J, Cohn JF, Kanade T (2005) Multi-view aam fitting and camera calibration. In: ICCV, pp 511–518

Kumar MP, Torr PH, Zisserman A (2008) Learning layered motion segmentations of video. International Journal of Computer Vision 76(3):301–319

Li R, Yu S, Yang X (Aug. 2007) Efficient spatio-temporal segmentation for extracting moving objects in video sequences. IEEE Transactions on Consumer Electronics 53(3):1161–1167

Li T, Kallem V, Singaraju D, Vidal R (2007) Projective factorization of multiple rigid-body motions. pp 1–6

Llado X, Bue AD, Agapito L (2006) Euclidean reconstruction of deformable structure using a perspective camera with varying intrinsic parameters. 18th International Conference on Pattern Recognition 1:139–142

Ma Y, Derksen H, Hong W, Wright J (2007) Segmentation of multivariate mixed data via lossy data coding and compression. IEEE transactions on pattern analysis and machine intelligence 29(9):1546 –1562

Min C, Medioni G (2008) Inferring segmented dense motion layers using 5d tensor voting. IEEE transactions on pattern analysis and machine intelligence 30(9):1589–1602

Ommer B, Mader T, Buhmann JM (2009) Seeing the objects behind the dots: Recognition in videos from a moving camera. International Journal of Computer Vision 83:57–71

Rao SR, Tron R, Vidal R, Ma Y (2008) Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: IEEE Conference on Computer Vision and Pattern Recognition

Rasmussen C, Hager GD (2001) Probabilistic data association methods for tracking complex visual objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6):560–576

Rekleitis I (2003) Cooperative localization and multi-robot exploration. PhD in computer science, School of Computer Science, McGill University, Montreal, Quebec, Canada

Saul LK, Roweis ST (2003) Think globally, fit locally: unsupervised learning of low dimensional manifolds. J Mach Learn Res 4:119–155

Sethian J (1998) Level set methods and fast marching methods: Evolving interfaces in computational geometry

Shen H, Zhang L, Huang B, Li P (2007) A map approach for joint motion estimation, segmentation, and super resolution. IEEE Transactions on Image Processing 16(2):479–490

da Silva NP, Costeira JP (2008) Subspace segmentation with outliers: a grassmannian approach to the maximum consensus subspace. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 1–6

Stolkin R, Greig A, Hodgetts M, Gilby J (2008) An em/e-mrf algorithm for adaptive model based tracking in extremely poor visibility. Image and Vision Computing 26(4):480–495

Sugaya Y, Kanatani K (2004) Geometric structure of degeneracy for multi-body motion segmentation. In: Statistical Methods in Video Processing, pp 13–25

Tomasi C, Kanade T (1992) Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision 9(2):137–154

Vaswani N, Tannenbaum A, Yezzi A (2007) Tracking deforming objects using particle filtering for geometric active contours. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(8):1470–1475

Vidal R, Hartley R (2004) Motion segmentation with missing data using powerfactorizationand gpca. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2:310–316

Vidal R, Tron R, Hartley R (2008) Multiframe motion segmentation with missing data using powerfactorization and gpca. International Journal of Computer Vision 79:85–105

Wang J, Adelson E (1993) Layered representation for motion analysis. pp 361–366

Wiskott L (1997) Segmentation from motion: Combining Gabor- and Mallat-wavelets to overcome aperture and correspondence problem. In: Sommer G, Daniilidis K, Pauli J (eds) Proceedings of the 7th International Conference on Computer Analysis of Images and Patterns, Springer-Verlag, Heidelberg, vol 1296, pp 329–336

Xu L, Chen J, Jia J (2008) A segmentation based variational model for accurate optical flow estimation. In: European Conference on Computer Vision, pp 671–684

Yan J, Pollefeys M (2006) A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: Computer Vision ECCV 2006, vol 3954, pp 94–106

Yan J, Pollefeys M (2008) A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5):865–877

Zappella L, Llado X, Salvi J (2009) Rank estimation of trajectory matrix in motion segmentation. Electronics Letters 45(11):540–541

Zelnik-Manor L, Irani M (2003) Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2:287–93

Zhang J, Shi F, Wang J, Liu Y (2007) 3d motion segmentation from straight-line optical flow. In: Multimedia Content Analysis and Mining, pp 85–94

Zhou H, Huang TS (2003) Recovering articulated motion with a hierarchical factorization method. In: Gesture Workshop, pp 140–151

4

Volume Decomposition and Hierarchical Skeletonization for Shape Analysis

Xiaopeng Zhang[1], Bo Xiang[1], Wujun Che[1] and Marc Jaeger[2]
[1]*LIAMA-NLPR, Institute of Automation, CAS*
China
[2]*CIRAD-AMAP / INRIA-Saclay*
France

1. Introduction

Volume is one of the important shape types in our world, from daily tools to complex and precise equipments, and even to life phenomena. Shape analysis techniques help understanding and using shape information for various applications. With the development of shape data acquisition and digitalisation techniques, more and more high-resolution shape data sets are available. Increasing demand for their compact shape description in applications inspires the need to reduce the data to a description of more concise remnant.

Skeletonization and volume decomposition are fundamental tools for shape information processing and understanding, being widely used in many applications, such as character animation, measurement and navigation planning in virtual colonoscopy.

A skeleton is an ideal shape representation with significant data compression while highlighting topological structures. The skeleton of a solid object, accompanied with radius information, exhibits its shape variation and spatial expansion. Skeletonization is a process of data abstraction to extract skeletons of an object, being promising and efficient due to linearity and simplicity of skeletons. There is an extensive body of scientific literature on 2D Skeletonization. 3D skeletonization is widely cared about in recent years. It has been linked to different shape-related techniques, like shape manipulation (Katz & Tal, 2003), shape matching (Funkhouser et al., 2004), shape retrieval (Tung & Schmitt, 2004) and collision detection (Li et al., 2001).

Shape decomposition is a practical process of dividing complex structures of an object into simple components. It has been indicated that good shape decomposition can result in skeleton extraction of high quality (Katz & Tal, 2003), and that a high quality skeletonization may lead to a meaningful decomposition (Li et al., 2001). But we will have much work to do yet in specifying the relationship between shape decomposition and skeletonization of 3D shapes (Lien & Amato, 2007). The definition of shape decomposition of a volume is still challenging since it is hard to specify the cut-surface.

2. Related Work

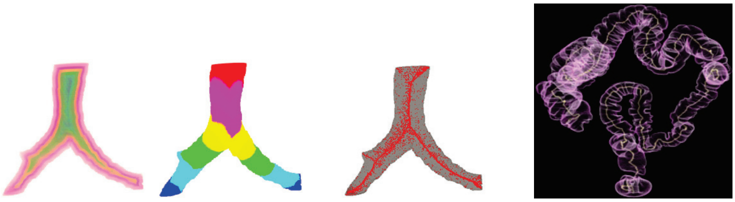
Volume elements, usually known as *voxels* in literature, are evenly distributed in three axis directions, constituting the space occupation of the volume. Skeletons and shape components are typical shape features. The skeleton of a volume is 1D thinning structure to represent its shape and topology (Brostow et al., 2004), and it is a concise representation of the volume shape. Shape decomposition is a basic technique to describe its complex structure. Related work on 3D skeletonization, shape decomposition and their applications are briefly discussed in this section.

2.1 Distance based volume skeletons

Distance transformation of a volume converts voxels into layers in terms of a feature point or the boundary surface. A skilful concept was proposed in (Zhou & Toga, 1999) to find a shortest path skeleton through the combination of DFS-distance (Distance From a Starting point) transformation with DFB-distance (Distance From Boundary) transformation. Fig. 1(a) shows a DFS-distance map and Fig. 1(b) shows a DFB-distance map. A global minimum cost path-searching algorithm was provided in (Bitter et al., 2000) with penalty distance through a heuristic combination of DFS-distance and DFB-distance. It was extended in (Sato et al., 2000) to branched skeletons with an adaptive sphere. The penalty distance algorithm was further improved in (Bitter et al., 2001) to correct mistakes cased in (Bitter et al., 2000) and (Sato et al., 2000).

A thinness parameter was adopted in (Gagvani & Silver, 1999) to control the candidacy of a voxel on a skeleton using DFB-distance transformation. A set of underlying skeleton points is defined by maximum central point map (MCP) (Fig. 1(c)).

The corner-cutting problem of volume skeletonization was solved in (Wan et al., 2002) by delivering a centred path rather than a shortest one with exact Euclidean distance and using minimum-spanning tree (Fig. 1(d)). But the influences of side-branches on the main ones are not considered.



(a) (b) (c) (d)

Fig. 1. Distance based volume skeletons. (a) DFS-distance; (b) DFB-distance; (c) MCP; (d) Centreline of a colon

2.2 Thinning based volume skeletons

Parallel volume thinning is an efficient way to find the centreline by deleting outer voxels iteratively if the deletion does not destroy the original topology of the object.

Ma & Sonka (1996) proposed a fully parallel and connectivity-preserving thinning algorithm to reduce computational cost. 3D thinning erodes a 3D binary image layer by layer through templates to extract the skeletons. Expensive testing of feature points is avoided by matching the 26-neighborhood of the points with predefined templates (Fig. 2(a)). However, the final centreline may be disconnected and many spurious branches may be generated in the resultant skeleton too. The connectivity of the skeleton is kept in (Liu et al., 2005) by adjusting a point deletion with its neighbour points in different iterations and a length parameter is also applied to removing the creation of spurious branches. However, these methods are ineffective if the length of a spurious branch is longer than that of a real one. Ma's Templates was modified to preserve connectivity in (Wang & Basu, 2007), as shown in Fig. 2(b), where the left is the original object, the middle is the skeleton with the approach of (Ma & Sonka, 1996) and the right is with that of (Wang & Basu, 2007).

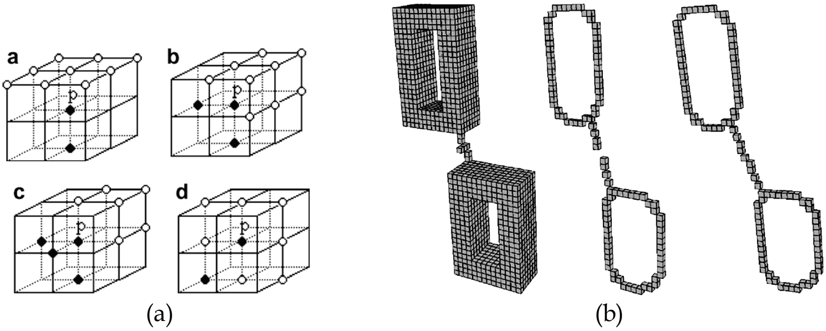
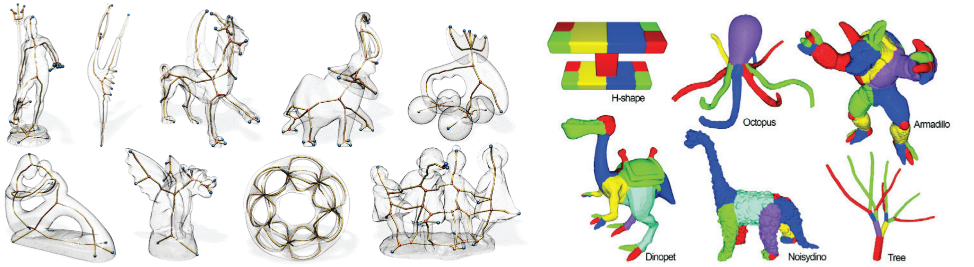


Fig. 2. Thinning based skeletons. (a) Ma's templates. (b) Modified Ma's Templates

2.3 Shape decomposition and skeleton hierarchy

Shape decomposition and skeleton hierarchy are significant topics of many shape information processing, such as shape analysis and understanding of 2D models (Rom & Medioni, 1993; Simmons & Sequin, 1998), 2D images (Siddiqi & Kimia, 1995; Telea et al., 2004), boundary represented models (Au et al., 2008) and volumes data (Cornea et al., 2005). Lien & Amato (2004) used approximate convex decomposition (ACD) to partition the mesh into nearly convex components and skeletons are then extracted from their convex hulls, respectively. Au et al. (2008) presented a simple and robust skeleton extraction method based on mesh contraction. As shown in Fig. 3(a), the 1D skeleton shape is achieved by performing geometric contraction using constrained Laplacian smoothing.

A framework was presented in (Reniers & Telea, 2007) to segment a 3D shape into meaningful components using curve skeletons. Critical points, or ramification point, are used to construct a partition of the object surface with geodesics, and the segments have minimally-twisting smooth borders. The resultant segmentation of the shape reflects the hierarchical structure of the curve skeleton, as illustrated with Fig. 3(b). But the application of volume decomposition to hierarchical skeletons is not considered. The work of (Reniers & Telea, 2007) is based on an assumption that all skeletons compete equally around each ramification while our work proposed here un-equally.



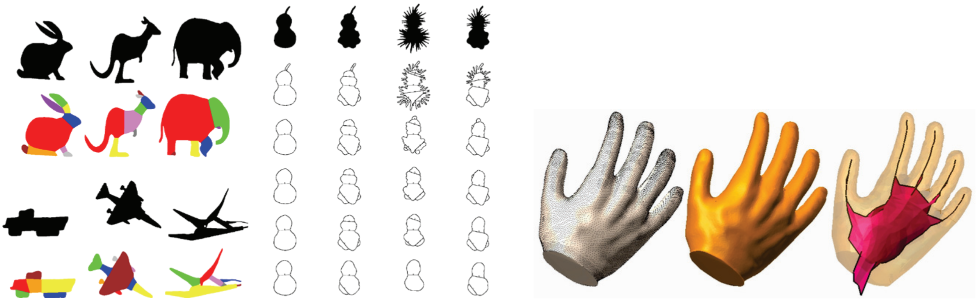
(a) (b)

Fig. 3. Shape decomposition and skeleton hierarchy; (b) Skeleton by Mesh Contraction; (a) Segmentations through curve skeletons

The concept of hierarchical curve-skeleton was proposed in (Cornea et al., 2005) and (Cornea & Min, 2007). A family of hierarchical curve-skeletons is extracted robustly for varied 3D objects, e.g. volumetric shapes, polygonal models or scattered point sets. These algorithms are based upon computing a repulsive force field over a discretization of the 3D object. Topological characteristics of the resulting vector field, such as critical points and critical curves, are used to extract the curve-skeletons.

2.4 Applications of shape features

2D shape decomposition was concerned for shape recognition before 3D case. In the work of (Siddiqi & Kimia, 1995), a partition scheme was proposed and used to segment a variety of 2D shapes. Different scales of partitions are intuitive by relating segmented parts to semantic portions of the original object, which are rather useful to shape recognition (Fig. 4(a)).



(a) (b)

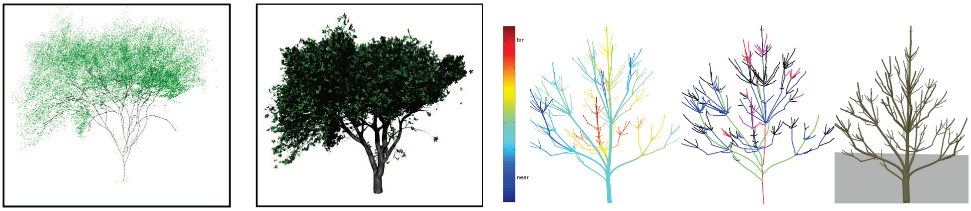
Fig. 4. Applications of shape features to shape retrieval and analysis; (a) Recognition of 2D shape through decomposition; (b) Mesh reconstruction through medial axis

Shape decomposition and skeleton hierarchy provide shape features of a volume for various applications. Ju et al. (2007) proposed a new method by alternating thinning and a novel skeleton pruning routine to extract skeletons of volumetric models for shape description. This technique is simple and meaningful, but cannot be used to segmentation, matching and recognition.

Hierarchical skeletons can be used for solid reconstruction. Local differential geometry of different kinds of points forming 3D symmetry sets was analysed in (Giblin & Kimia, 2004). With this approach, the extracted skeletons can be used to reconstruct the original surface with full geometrical information. Linked with a real-time simulation of stroke therapy, a mechanism was proposed in (Luboz et al., 2005) to segment and to reconstruct 3D human vasculature models with a balance of smoothness, number of triangles and distance error.

The idea of power crust presented in (Amenta et al., 2001) was used to approximate the medial axis transform (MAT) of an object. An inverse transform was then applied to producing mesh representation of the surface from MAT. Examples in Fig. 4(b) suggest the capability of surface reconstruction from laser range data.

Skeletonization of a range image from a real tree is a very new development. Xu et al. (2007) and Cheng et al. (2007) used range images to reconstruct the geometric model of a tree. The former is for producing full a polygonal model of a range scanned tree through skeletonization of the trunk and main branches of the tree (Fig. 5(a)), while the latter is through 2D skeletonization, cylinder fitting and generalized surface (Fig. 5(b)).



(a) (b)

Fig. 5. Applications to tree reconstruction; (a) Knowledge and heuristic-based modelling of laser-scanned trees; (b) Tree Branch Reconstruction through cylinder fitting

2.5 Contributions of our work

A new approach of volume decomposition and efficient extraction of hierarchical skeletons are systematically described in this chapter based on our work as in (Zhang et al., 2008, 2009).

Five aspects are discussed: computing multiple distance transformations to find the hierarchical structure of the object volume; decomposing the volume into simple components; extracting compact and smooth skeletal segments corresponding to each independent components; efficient skeleton extraction from noise data of laser scan; and an application of this technique to plant reconstruction. Technical contributions are fourfolds: (1). The influence of side branches to the main one is avoided via volume decomposition; (2). The extracted hierarchical skeletons keep well the topology of the original object; (3). Skeleton nodes sampling is adjusted to smooth the skeletons; (4). The new approach is applied to handling the shape of more complex topology, e.g. with a loop.

The deviation problem in skeletonization is solved with volume decomposition, compact skeletonization and hierarchical skeletons. The relationship between shape decomposition and skeletonization is emphasized and specified with technical details. The construction of volume decomposition surfaces, or cut-surfaces, and skeleton point sampling are extended so that curve-skeletons become smoother and keeps well the shape of the branched volume than our early work. The extraction is fully automatic and much faster, even to a shape with complex topology like a ring volume.

3. System Overview

In this chapter we process tree-like object volumes possibly with some loop structures. The main technical line is to decompose the volume into components, based on which hierarchical skeletons are extracted.

Volume decomposition is to separate a complex branched volume into a series of simple components around its ramification points, each of which is topologically equivalent to a single column. The term *hierarchical* in this chapter does not mean a hierarchical process for skeleton extraction but a hierarchical decomposition of a volume into components. Skeletons are extracted from all components and then are connected to form a hierarchical structure. The object data should have an evident root point, i.e. the centre at the bottom of the trunk.

Three main concepts are concerned as classification-skeleton for volume decomposition, decomposition for hierarchical connection of skeletons and path growing for efficient skeletonization. *Classification-skeleton* shows the hierarchical structure of the object volume, so it is chosen as the key criterion for volume decomposition. *Hierarchical connection* means a connection of skeletons into a tree-like structure in the same topology with the original volume after decomposition. Classification-skeleton is the standard for connection. *Path growing* means voxel propagation from a seed point in one direction until some conditions are satisfied.

The classification-skeleton is extracted through distance transformation from the boundary or a seed point and through central cluster graphing. With the help of classification-skeleton, volume decomposition is performed by cross-section surfaces, called cut-surfaces; each decomposition component should be close to the ramification points so that the deviating affect of side branches is greatly reduced.

Finally, the hierarchical branched skeleton is organized by connecting all compact skeleton segments according to the classification-skeleton so that the final hierarchical skeleton reflects the shape of the original volume.

In order to extract skeletons efficiently, distance transformations from multiple source points are made from all tip points so that all components of the object can compete to reach the root through ramifications. They are used for both decomposition and skeletonization. A cluster interval is adopted to generate equivalent skeleton point range for each component so that the skeleton becomes smoother and more concise.

The system includes four main aspects: (1) Voxel classification into a hierarchical structure through integer seed point distance transformation and boundary distance transformation, then classification-skeletons for branched structure; (2) Volume decomposition at ramification points through real-valued distance transformation based on classification-skeletons; (3) Compact skeletonization of each decomposed component through real-valued distance transformation and connection of hierarchical skeleton on reference of classification-skeletons; (4) Path competition for decomposition and efficient extraction of smooth skeletons. Fig. 6 details the contents and the structure of the system.

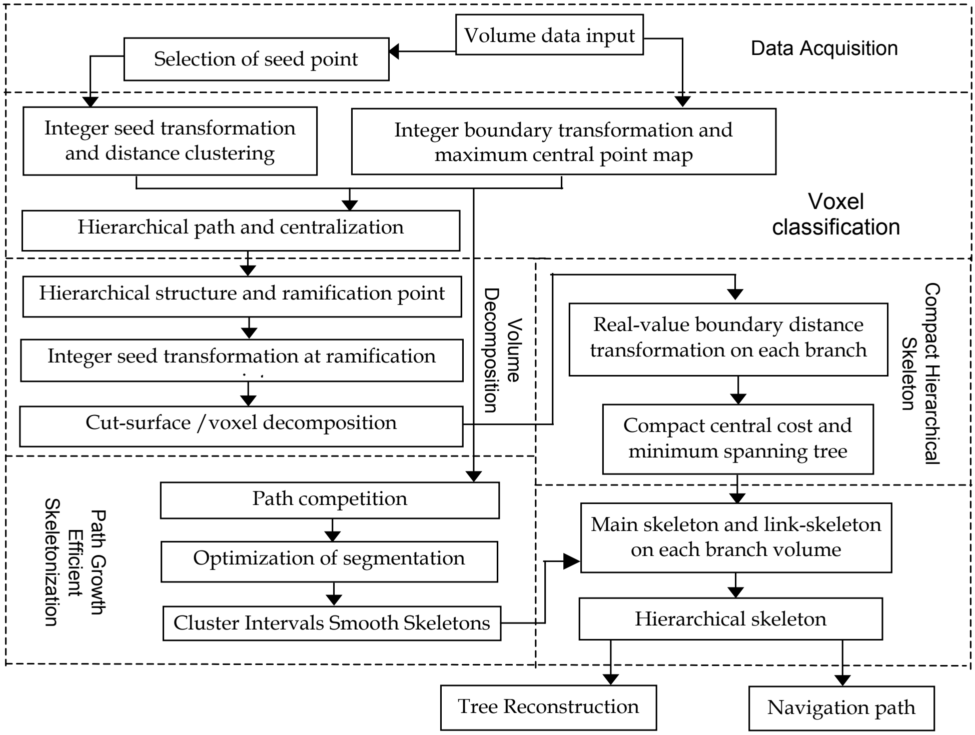

Volume data input
Selection of seed point
Data Acquisition
Integer seed transformation and distance clustering
Integer boundary transformation and maximum central point map
Voxel classification
Hierarchical path and centralization
Hierarchical structure and ramification point
Integer seed transformation at ramification
Cut-surface /voxel decomposition
Volume Decomposition
Real-value boundary distance transformation on each branch
Compact central cost and minimum spanning tree
Compact Hierarchical Skeleton
Path competition
Optimization of segmentation
Cluster Intervals Smooth Skeletons
Path Growth Efficient Skeletonization
Main skeleton and link-skeleton on each branch volume
Hierarchical skeleton
Tree Reconstruction
Navigation path

Fig. 6. Overview of the system

4. Voxel Classification

Classification of voxels according volume topological structure consists of five steps.

4.1 Distance transformation

Distance transformation is a tool for voxel classification. It is a propagation process of voxel values from a voxel sub-set S to the whole set V. At the beginning, the value of each voxel in V is initialized, usually as 0 or 1. We take W as the work set, and S as the initial values of W. In each step of propagation, all voxels in the n-neighbourhood of a voxel in W are set as m_n+m if its initial value is smaller than it, where m is the distance of its n-neighbourhood voxel in W and m_n is the scheme for distance transmission between two n-neighbouring voxels. Often the evaluation rule is represented as m_6-m_{18}-m_{26}, where m_6, m_{18} and m_{26} are the evaluation transmissions to 6-neighborhoods, 18-neighborhoods and 26-neighborhoods respectively. They are selected according to desired transformation precision, such as 1-2-3, 4-5-6 and 1-$\sqrt{2}$ - $\sqrt{3}$.

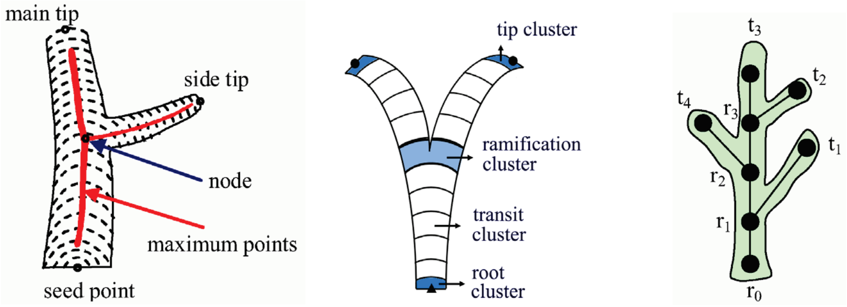
m_n can be an integer, or more precisely, a real. Integer number is good for concise classification and real number is good for precise shape acquisition. In this section integer is adapted before volume decomposition,, such as 1-2-3 or 3-4-5, so that distance can be used as a criterion to distinguish voxels according to branch structures. After volume decomposition, real is adapted instead for a precise skeleton.

Voxel points can be organized after integer distance transformation. A *cluster* is defined as a set of consecutive points with the same integer distance. *Classification skeleton* is a structure of skeletons linking the volume root to each branch tip, which shows the hierarchical structure of the object volume. *Volume decomposition* is the separation of a complex branched volume around its ramification points into a series of simple components, each of which is topologically equivalent to a single column. *Hierarchical connection* means a connection of skeletons into a branched structure of the same topology with the original volume.

Classification skeleton is a direct application of integer distance transformation. It is used to find out tip points, ramification points, hierarchical structure and a rough central skeleton. The skeleton obtained this way is not well central and not 26-neighboring either, but it can be used as a criterion for voxel classification due to its integer value.

4.2 Central clustering

Integer seed point transformation is performed with the evaluation scheme 1-2-3 and 6-neighborhood propagation, where the seed point is the volume root specified by the user or by the system. The main purpose of the transformation is to search all tip points of branches and all ramification points so that decomposition could be properly performed at each ramification point.



(a) S-cluster (b) Structure of clusters (c) Branch graph

Fig. 7. Central clustering on root distance

An *S-cluster* is the set of all the points having the same S-distance value, which is similar in part to sphere waves from the seed point as its centre (Fig. 7(a)). S-cluster is often abbreviated as a *cluster* here. All clusters constitute a graph according to their neighbouring relation. The *positive direction* of this graph is defined as from the root to one tip.

A *tip cluster* is composed of points with local maximum S-distance, which corresponds to a volume tip. A *ramification cluster* is one whose oneness breaks up at the next neighbour clusters in the positive direction of the cluster graph. Seeing locally from a ramification cluster in the positive direction, the S-distance of the next neighbouring clusters will increase by one. A ramification cluster corresponds to a volume ramification. The tip cluster and ramification cluster represent the topological structure of the cluster graph.

A *transition cluster* is one that is not a seed cluster, not a tip cluster or not a ramification cluster. A transition cluster has only one preceding cluster and only one succeeding cluster.

Fig. 7(b) shows the structure of all clusters, where circles represent tip points, and triangle represents the root point. After omitting all transition clusters, the cluster graph becomes a directed graph with each cluster as a node of the graph (Fig. 7(c)). This graph is referred to as a *cluster graph.*

4.3 Construction of the branch-link path

A *branch-link path* is a sequence of points with two ends, i.e. the root point and one tip point. All branches are correctly connected with branch-link paths. A branch-link path is unnecessarily the centreline of the volume, but it is the basic for finding a central path of the volume. It is calculated as follows.

All tip clusters are found first. For each tip cluster, the maximum B-distance point, or the barycentric point of all the maximum B-distance points if they are more than one, is chosen as the initial current point. For each current point, we search for the point in its 6-neighborhood with least S-distance as the next point, which is either in the same cluster or in its neighbour cluster. This process is repeated until the root seed point is reached; then a sequence of points is obtained, starting from a tip point to the root point with decreasing S-distances in turn.

Because the distance between two neighbour clusters is no more than 1, a branch-link path is guaranteed to converge to the root point in the decreasing order.

Since 6-neighborhood is used, on the other hand, there is always a point in the path with the distance of any integer number between 1 and the distance number from the tip point to the seed one. Therefore, this sequence contains all the clusters between the tip and the seed.

Only one representative of a cluster is reserved and the connection of all representatives from the root to the tips will become a bundle graph or a divergent graph. This graph is called the *branch-link path*. It is not a medial axis of the volume yet.

4.4 Centralization of the branch-link path

Centralization of a branch-link path means moving this path to the centreline or replacing each point of the path with another one in the same cluster closer to the cluster centre, so that the entire path approaches the medial axis of the object volume. Two concepts are useful for centralization. One is *Maximum central point set C* of all points in the maximum central point map. *Boundary-distance maximum point set B* is the point set with maximum boundary distance.

The centralization begins from the tip point of each branch in the direction of decreasing S-distance. If *C* in this cluster is not empty, its barycentre is accepted as a representative of the cluster; or else, the barycentre of *B* is selected; and the boundary distance is recorded as the corresponding radius. If both *C* and *B* are empty, the one with smallest distance to the immediately preceding point is chosen, and the boundary distance is the corresponding radius.

In the method of (Shahrokni et al., 2001), either the point with the maximum B-distance is chosen, or, if they are multiple, their barycentric point is chosen. However, since the connection relationship of neighbouring clusters is not well considered, the resultant central point may destroy the original topology of the volume. The green circle in Fig. 8(a) shows this wrong node location and wrong topology. We exploit both *C* and *B* to centralize the branch-link path so that the final path is correct in centrality and topology (Fig. 8 (b)).

4.5 Main branch and branched structure

Fixing the main branch is to find a primary and secondary order of all branches around each node, then to select the principal path and to unite all of them around the node according to their mutual distances. The determination of filiation and brotherhood depends on the radius of each branch at the node. A simple way is to judge by the length, but it sometimes fails since the main branch is not always the longest, but more often the thickest.

After the determination of all paths and all ramification points, all filiations are recorded and all these relationships constitute the branched structure. Considering the computation errors due to integer distance transformation and the influence of side branches on the parent branch, this hierarchical skeleton is not ideal, but can be used to find the ramification location and volume decomposition at the ramification. It contains the information of hierarchical classification, centrality and connectivity, so it is called the *classification skeleton*.

5. Volume Decomposition and Compact Skeletons

5.1 Volume Decomposition

Volume decomposition means the separation of a branched volume around ramification points into a hierarchy of components, each of which is topologically equivalent to a single column. The crucial technique for decomposition is how to position cut-surfaces to divide the volume into parts.

The object volume is regarded as of clear hierarchical structure of parent-child relation at each ramification point. After separation, the main branch is called a *parent branch* or a *principal branch*, and the others are called a *subordinate branch*. All cut-surfaces are cross sections almost orthogonal to the skeleton direction. The basic consideration in constructing a cut-surface is to find a surface on each branch and to avoid any two cut-surfaces intersecting with each other. Additionally, a cut-surface is as close to a ramification as possible so that as many voxels as possible will be assigned to the branch from its parent. The search for cut-surfaces starts from tips to the root.

The general topology of the volume and hierarchical connection relationship of components can be deduced through the classification-skeleton obtained above.

For clarity, we only describe the construction algorithm for a bifurcating point. We take any skeleton node location as a seed point and perform an integer seed point transformation in the entire volume. This is the first stage of transformation for volume decomposition. For each integer distance value, there must be a cut-surface or some disconnected cross-section surfaces associated with this distance. The branched structure of the volume around a ramification is responsible for these surfaces, called *wave-pairs*. Each surface is similar to a part of a sphere, and it is almost orthogonal to the classification-skeleton. Each wave pair is chosen to be as far as possible from the ramification position. Then a pair of surface transformations is performed backward from the wave-surface-patch pair surfaces to the ramification location until these pair surfaces intersect with each other. This is the second stage of volume decomposition. The reason for the progress from farthest to nearest is to make the cut-surface as orthogonal to the skeleton as possible. Also, cut-surfaces too far from the ramification location will leave more voxels in the parent branch.

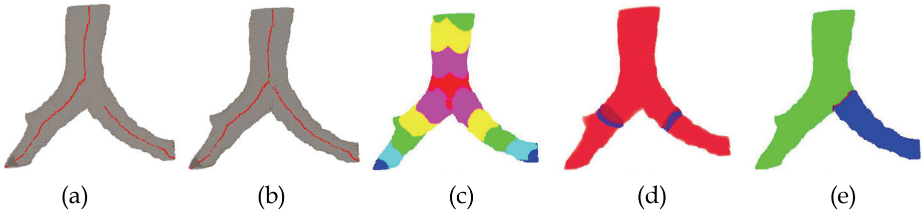


Fig. 8. Cluster graph and segmentation; (a) Improper cluster graph; (b) Proper cluster graph; (c) the seed point transformation with the ramification as the seed point; (d) a pair of wave clusters are selected from the seed point transformation; (e) the cut-surface

We set the cut-surface into the child branch, and it is called the root-surface of the child branch. It centre is called the *main source point* of the child branch. The corresponding surface consists of voxels in the parent branch but 26-neighboring to some voxels of the root-surface of the child branch. It is called the side-surface of the parent branch.

Fig. 8 (c)-(e) depicts an example of volume decomposition at a ramification. Fig. 8 (c) is the seed point transformation of a trachea at the ramification. Fig. 8 (d) displays two separate cut-surfaces with the same distance from the seed to the ramification in two different branches. Fig. 8(e) illustrates the final result of the volume decomposition, green corresponding to the parent branch, blue corresponding to the filial branch, and red representing the cut-surface.

After decomposition, the shape of each component has several mouths, as shown in Fig. 9(a). The barycentric voxel of the main tip mouth of this branch component is called the *main tip*. The barycentric voxel of the mouth corresponding to its child volume is called a *side tip*. The barycentric voxel of the mouth cut from its parent branch or the root mouth is called a *main source*, which is the seed point for this component.

5.2 Compact Skeletons

The results of volume decomposition are used to extract skeleton segments of component *V*, to record the connection relation and to connect them with more skeleton segments into the hierarchical topology equivalent to that of the volume.

Our method of hierarchical skeleton extraction takes three steps: (a) generation of a 3-D directed weighted graph from each component data; (b) using a minimum spanning tree (MST-tree) to get a skeleton segment for each component; (c) connection of all skeleton segments. For each component, the main source point is chosen as the seed for point distance transformation. The evaluation scheme is real value 1- $\sqrt{2}$ - $\sqrt{3}$ with 26-neighborhood propagation for better precision. Then, *C-Cost* function $C(p)$, i.e. a *Compact Centrality Cost* Function, is constructed. Finally, a branched structure is constructed of minimum central cost connecting the main source point and the main tip point, and this topological structure is equivalent to that of the volume before volume decomposition.

5.2.1 Compact centrality cost

The central line is the feature inverse to the boundary, so the distance to the boundary is chosen as the basic opposing element of Centrality. In order to have centrality and compactness connection of the skeleton both be considered, $C(p)$ is defined as an addition of two parts in (1): boundary cost, *B-cost* $B(p)$,and compactness cost, *Q-cost* $Q(p)$.

$$\mathrm{C(p)}=B(\mathrm{p})+Q(\mathrm{p}) \tag{1}$$

where $B(p)$ is defined as $1/\mathrm{b}(p)$, and $Q(p)$ as a scaled accumulated seed distance cost $T(p)$ in (2).

$$Q(p)=k(V)\ \xi(V)\ T(p) \tag{2}$$

where $k(V)$, $\xi(V)$, $T(p)$ are calculated by (3), (4) and (5) respectively.

$T(p)$ is defined in a recursive manner starting from the main source point in Fig. 9(a), where the main source point is chosen as the seed point for transformation, and its $T(p)$ is set as 0 at this point. Then the $T(p)$ is evaluated recursively in (3), like a seed distance cost expansion from the seed point.

$$T(p) = \min\{S(q) + (1 - M * b(p))E(p,q) \mid q \in N_n^*(p)\} \tag{3}$$

where $M = \min\{B(p) \mid p \in V\}$ is the minimum cost in current component V. The effect of $b(p)$ is to make p closer to the centre and $1 - M * b(p) \in [0,1]$. $E(p,q)$ is defined as $d(p,q)$-0.75, ($q \in N_n^*(p)$), where d(p,q) is the Euclidean distance. The effect of -0.75 is to make the influence of the distance between neighbouring voxels lesser. For any two neighbour voxels p and q, $d(p,q)$ is in [1, $\sqrt{3}$], so $E(p,q)$ is in [0.25,0.919].

$\xi(V)$ in (2) is defined as the minimum absolute difference of any two B-distances of two voxel points in volume V in (4).

$$\xi(V) = \min\{|b(p) - b(q)| : |b(p) - b(q)| \geq \nu; p,q \in V\} \tag{4}$$

Thus the Q-cost at any point must be smaller than difference between any two points in the volume. Considering that real-valued B-distance could make $\xi(V)$ very small, a double precision real number is used to represent this cost. On the other hand, we perform a simple transformation on all of the B-costs and they will be treated as equal when the difference of two B-costs is smaller than a small value ν,. Therefore, the effect of $\xi(V)$ is that Q-cost will differentiate two points of the same B-cost; a voxel closer to the centre has a bigger Q-cost. For voxels of different B-cost, Q-cost will have no effect since they have no impact on the two different B-costs. Therefore, the selected point will be closer to the centre and closer to the seed point simultaneously so that centrality and compactness are both satisfied.

The coefficient $k(V)$ in (2) is a constant for component V and it is defined as (5).

$$k = 1/T\ ,\ T = \min\{T(p) \mid p \in V\} \tag{5}$$

The centrality cost is the global cost of non-centrality and non-compactness.

5.2.2 Construction of a minimum spanning tree

After the determination of the compact centrality cost for each voxel, the volumetric dataset is converted into a 3D directed weighted graph. This graph includes all the links of any two 26-neighboring voxels. The connection directions of any two voxels are bi-directional, with different values for different directions. The weight is determined by the C-cost

Similar to (Wan et al., 2002), all component data are transformed to 3D directed weighted graphs at first, and then a minimum-spanning tree is constructed for each component. This is expressed in the following procedure:

Step 1) Mark source point S as the seed point for C-cost. Mark S, set its parent pointer *parent-link* to NULL.

Step 2) Select the seed point as the beginning current point T.

Step 3) For each current point T, push each of its unmarked 26-neighborhood B_i into a sorted heap in increasing order of C-cost, so that the one with the smallest C-cost is always at the top. If the C-cost of the *parent-link* of the node B_i is larger than that at T, set the *parent-link* of B_i to T.

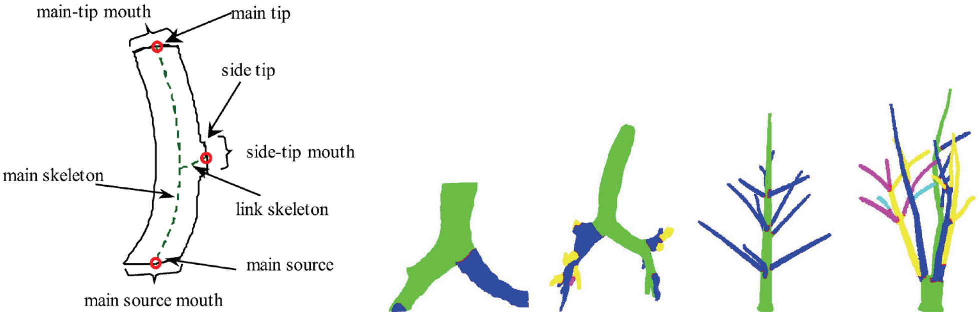
Step 4) Pop the top node of the heap, mark it, and set it as the current node.

Step 5) Repeat Step 3) and Step 4) until the heap is empty.

This is a little similar to the algorithm described in (Wan et al., 2002), but they did not consider ramifications. Our method addresses this issue, more concisely, compactly and robustly due to formula (3). The resultant skeleton always stays away from the boundary and the tendency to go roundabout is avoided.

5.2.3 Skeleton extraction and hierarchical connection

Skeleton extraction is in two steps. The main tip and side tips are calculated at first for the minimum-spanning tree of the current component (Fig. 9(a)). The skeleton segments with two ends of these two tips are called the *main-skeleton* and *side-skeleton*, respectively. The main-skeleton connects the main source and the main tip, whereas the side skeleton connects one side tip to the main-skeleton, so that the centrality of the skeleton, hierarchy, and topological connection are maintained.



(a) Volume structure (b); (c) ; (d); (e)

Fig. 9. Hierarchical volume decomposition; (a) Volume structure after decomposition; (b) Airway 1, (c) Airway 2, (d) Larch tree, (e) Willow tree

Then the main-skeleton is extracted by tracing along the MST backward from the main tip point until the main source is met. The radius values of the skeleton segment are the distances of the path points to corresponding boundary points. Skeleton extraction is finished after recording this path along with the associated radii. For the link-skeleton, the process is similar, except that the beginning point is chosen as the side tip point until one voxel in the main-skeleton is reached. From the definition of C-cost, the link-skeleton will converge to the main-skeleton since the latter is the skeleton of this component.

After the extraction of skeleton segments, i.e. main skeletons and link skeletons, they will be connected together according to their original topological relations, including all paths from each tip to root passing all nodes. This is where the hierarchical skeleton can help, with its excellent centrality and classification.

Main skeleton can be used for pattern recognition of 3D objects.

5.2.4 Experiments on volume decomposition and hierarchical skeletons

Table 1 lists the details about the data with compact skeletonization, including the number of voxels in the volume, the number of voxels in the skeleton, and the number of branches. Fig. 9 (b)-(e) shows the results of volume decomposition, where green, blue, yellow, pink and cyan represent the levels of branches in increasing order. Objects here are two human airways, above year-old larch tree and a four year-old willow tree. It can be seen that each component has a single-column shape and their borders are kept well. Fig. 9 (b)-(e) also demonstrates that our volume decomposition technique is valid for multiple furcations.

Name	Volume Voxel No	Skeleton Voxel Number	Branch Number
Airway 1	552595	408	3
Airway 2	581539	1229	14
Larch	29657	910	12
Willow	49883	1622	18

Table 1: Volume and skeleton specifications with compact skeletonization

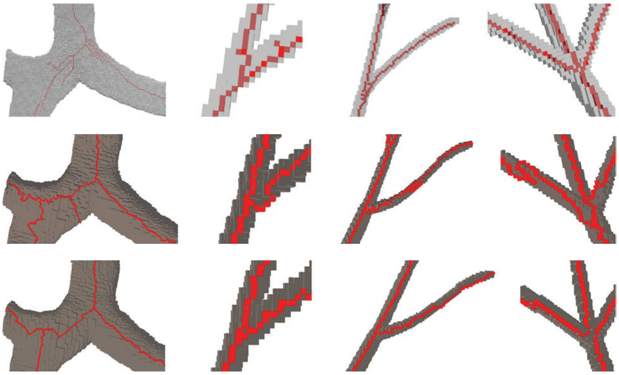


Fig. 10. Effect comparison among three approaches on three samples

The results of skeleton extraction by Ma's algorithm (Ma & Sonka, 1996), Wan's approach (Wan et al., 2002) and our algorithm are compared in Fig. 10. The skeletons of Ma's are shown in the first row, those of Wan's in the second row and those of our algorithm in the third. The first column displays the results of the three algorithms at the ramification of an airway. It can be seen that spurious branches are pruned in our algorithm while they still exist in the other two. The second column illustrates their results with a simple ramification of a willow tree. Here we find that our algorithm can be used to eliminate the influence of deviation, but the other two fail. The third column is the skeleton of a willow branch. Ma's method generates some spurious branches in the ramification; Wan's generates a "roundabout" shape in the skeleton, but all these mistakes are removed in our algorithm. The fourth column illustrates the skeletons in a multi-ramification of the willow tree; our algorithm keeps the skeleton in the centre while the skeletons generated by the others do not follow the medial axis.

Overlaying of the final skeletons over the objects is illustrated in Fig. 11 from all four models in Fig. 9, where they are viewed from three angles. Non-boundary voxels are displayed with a non-transparent single colour model, and boundary voxels are with a semi-transparent and lighting model. The frontward boundary voxels are set semi-transparent, and rearward voxels are set to be non-transparent. The results demonstrate that skeletons extracted with our algorithm are more centred and more compact, and the deviation is also eliminated in the ramification part through compact centrality cost when there exist many candidate points with the same distance from the boundary.

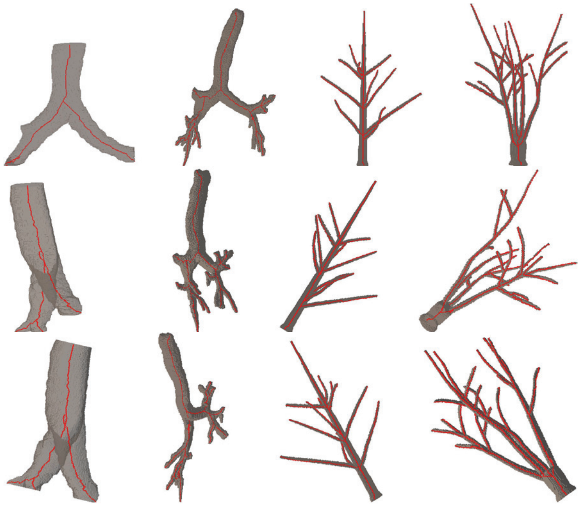


Fig. 11. Overlaying skeletons on 3D models

6. Efficient Extraction of Skeletons

The work here on efficient extraction of skeletons is based on all the techniques described above. The extended work in (Xiang et al., 2008; Ma et al., 2008; Ma et al., 2009) makes skeleton extraction more efficient and more robust. The skeleton extraction method supports shapes with loop structures because each voxel in the volume, including those in the loop, can be reached in the path growing and labelled to a component of the volume, while it is not considered in (Zhang et al., 2008) with the minimum spanning tree.

DFS-distance transformation is applied to the volume data once. No skeleton refinement and connection are needed after the skeleton extraction so that make the method efficient.

6.1 Automatic detection of feature points

All feature features of the object volume are thought of as tip points,. The most important is the root point. Instead of specifying it by user as an input in (Zhou & Toga, 1999) and (Xiang et al., 2008), feature points are detected automatically in our work here.

A voxel v is chosen randomly in the volume as the seed to perform the DFS-distance transformation. A new feature point s is chosen as the centre of those voxels with the largest distance away from v. Point s is called the opposite feature point to v. s can be thought of as a potential root point related to v, but it is not if v is too close to the actual root position. The DFS-distance transformation is applied then again with s as the seed, and its opposite feature point p is obtained. p is set as the root point if its radius is bigger; or else s is.

Concurrently with the specification of the root point, all tip features are detected also based on the DFS-distance transformation to the root point. The tip feature of each branch of the volume is detected from the local maximum cluster on the root point distance map. The number of all tip points is noted as N in this subsection.

6.2 Distance transformation from multiple source points

The DFMS transformation (distance from multiple source points) proposed in (Xiang et al., 2008) takes multiple DFS transformations regarding each seed point as a tip; the distance evaluation scheme is 1-2-3.

Ramification points are calculated with the method in Subsection 4.2. In the process of DFMS transformation, a DFS-distance transformation stops once a ramification point r is met. Then, all the DFS-distance transformations at the ramification node compete against each other to propagate toward the root point. The losers stop while the winner continues until another ramification or the root point is met.

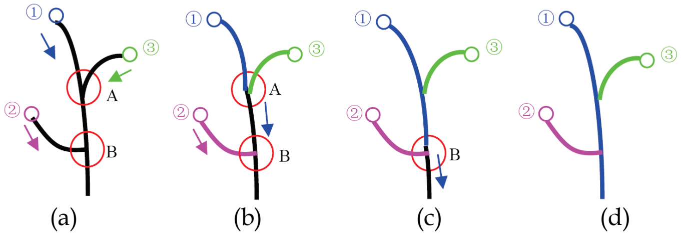


Fig. 12. Decomposition based on path competition; (a) three paths compete for growing; (b) *Path* 1 and *Path* 3 meet at the ramification *A*; (c) *Path* 1 wins and continues to compete with *Path* 2; (d) *Path* 1 wins all and become the trunk

The number of DFS-distance transformations corresponds to that of components passing the ramification point r. This number, denoted as $m(r)$, is saved as an index of r. All these DFS-distance transformations will be reused for efficient skeletonization.

Fig. 12 demonstrates the procedure of DFMS transformations for decomposition on a volume model with three branches. Fig. 12 (a) shows that three paths start from three tips, respectively. Fig. 12 (b) shows that a path stops growing when it reaches a ramification. *Path*

1 and *Path* 3 meet at the ramification *A* and only one branch is left, along which both paths compete for growing. *Path* 2 grows until meeting the ramification *B*, waiting other path to meet with. In Fig. 12 (c), *Path* 1 wins over *Path* 2 and it continues growing until reaching *B*. In Fig. 12 (d), *Path* 1 wins over *Path* 2 and it continues growing until all voxels in the volume are traversed. In fact, volume decomposition finishes when *Path* 1 meets the root.

6.3 Optimisation of segmentation surface

Segmentation at ramifications plays an important role in volumetric decomposition to obtain smooth and appropriate cutting boundaries for different components. Our segmentation method is based on the DFMS transformation above, and it is an extension of the approach in Subsection 5.1.

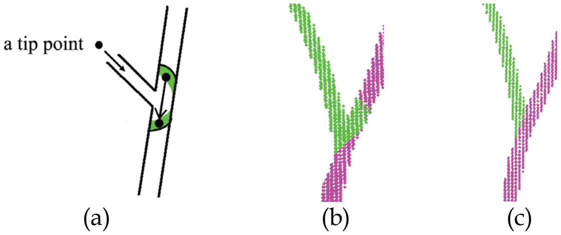


Fig. 13. Segmentation at a ramification; (a) direction vector; (b) Segmentation without optimisation; (c) Segmentation with optimisation

For simplicity, we only describe the cut-surface for a sub-branch corresponding to a tip point *s*. The ramification cluster found by a DFS-distance transformation with tip *s* as the seed can be divided into some self-connected groups, shown as two sets in green in Fig. 13(a). In Fig. 13 (a), the green regions stand for the neighbour clusters at the ramification, whose DFS-distance value is larger by 1 than that of the ramification cluster. We connect their centres and obtain a direction vector as the black arrow between them. Along this direction, we search for all the voxels on the shortest path linking the two sets. After cutting off these voxels, we mark the left in the clusters as parts belonging to the sub-branch indicated by the starting tip feature. This is the process of the optimisation of a cut-surface. The segmentation result around a ramification before using optimisation is demonstrated in Fig. 13(b) and the one after optimisation is in Fig. 13(c). The optimisation gives a better segmentation in Fig. 13 (c) than without the optimisation in Fig. 13 (b). The above algorithm is based on an assumption that components are approximately cylindrical.

6.4 Path growing for shape segmentation

From *N* tip features, we have *N* paths corresponding to *N* components, respectively. When a voxel is reached by a certain path, we mark the voxel with the feature component corresponding to this path. Those paths growing from tips to the root point will stop when a ramification appears. Growing here does not mean the development of plant architecture, but the voxel propagation from a tip point toward the root point until the specified conditions are satisfied. Rules for path growing are as follows:

Step 1) A path grows from its tip point of each component. Growing stops when the path meets a ramification. The state of a growing is either continuing, waiting or stopping; all voxels on the path are labelled with the index of the tip point;

Step 2) For each ramification point *r*, all paths reaching it are checked. Let *m*=*m*(*r*). If at least two paths are not indexed, they are waiting here. If only one path is not, all the other *m-1* paths will compete. The winner will continue to grow, but the others fail. The winning criterion is that the distance to a tip point is the biggest than those to other tips.

Step 3) All pathes continue to grow until a new ramification is met, and return to Step 2;

Step 4) If only one path can grow currently, the path will expand in the direction of increasing distance. If all voxels are indexed, decomposition finishes. If there are un-indexed voxels left and no paths can grow, a loop appears in the shape. They are classified as new components according to connectivity;

Step 5) All cutting boundaries are optimised.

The result of volume segmentation based on path growing is a hierarchical structure too. It represents the topological structure of the shape and it will be used to construct the skeleton hierarchy.

6.5 Efficient extraction of smooth skeletons

So far, the branched volume has been decomposed into *N* components, each of which corresponds to a tip feature. Here, we proceed to extract skeletons for each sub-branch component.

In voxel space of each component, DFS transformation with respective to its tip point is useful. Since DFS transformations have been calculated in decomposition, however, they can be reused here for efficient computation.

The distance value of each cluster will increase when path growing. The barycentre of each cluster is calculated and considered as a skeleton node since the topology of each component is simple. Each node has the distance value one larger or smaller than its neighbour nodes; hence all skeleton nodes are connected in distance-increasing order. Each skeleton is from a tip point to a ramification or the root point.

Because the connection of neighbour nodes may produce a large number of skeleton points. An appropriate cluster interval, as the distance sampling in (Xu et al., 2007), is adopted to generate equivalent skeleton point range for each sub-branch, in order for both less skeleton points and higher data compression rate. A cluster interval is a group of consecutive clusters in a component. A skeleton node is the barycentre of all the points in a cluster interval. The application of interval is a filtering of skeleton shape.

In order to deal with noisy data robustly, we compute a skeleton point $\vec{X}_t$ of a cluster interval as

$$\vec{X}_t = w_1\left[\frac{(\vec{X}_{t-1}-\vec{X}_{t-2})\left\|\vec{C}_t-\vec{X}_{t-1}\right\|}{\left\|\vec{X}_{t-1}-\vec{X}_{t-2}\right\|}+\vec{X}_{t-1}\right]+w_2\vec{C}_t \qquad (6)$$

where $\vec{C}_t$ is the barycentre of the cluster interval, $\vec{X}_{t-1}$ and $\vec{X}_{t-2}$ denote the skeleton nodes in the last two cluster intervals respectively, and w_1 and w_2 are two weighting coefficients.

The first term in (6) is a smoothness constraint. For each skeleton node, its radius is evaluated based upon the projections of voxels of the cluster on the plane, through the cluster centre and perpendicular to the skeleton.

For robustness to noises, we re-compute the radii along the skeleton varying too seriously. In our following experiments, w_1 and w_2 are set as 0.4 and 0.6 respectively.

Finally, skeletons of components are connected according to the hierarchical structure obtained in volume decomposition in Subsection 6.4.

6.6 Application to Tree Branch Reconstruction

To reconstruct real plants based on skeletonization, the point cloud data of two real bonsai trees are acquired by laser scanning from six angles (Fig. 19 and Fig. 20). They are then registered in a single coordinate system. The volume data of a tree is obtained from the resultant point cloud using an octree space subdivision. All skeletons of the volume data are extracted, where each skeleton node is accompanied with a radius representing its distance to the boundary. A generalized cylinder is then constructed based on each skeleton. Thus a mesh model is available when each cylinder is converted to polygons.

6.7 Experiments on efficient skeletonization

We make experiments upon the proposed algorithms of volume decomposition, efficient skeletonization and surface reconstruction using virtual volume models and those converted from laser scan data. Virtual plant models without leaves are generated by the software AMAP Gensis^{TM}. Other virtual models are generated from mathematical formula.

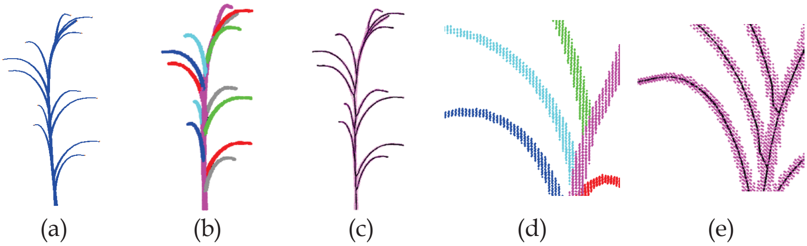


Fig. 14. Synthetic Sunflower branched volume; (a) the volume data; (b) volume decomposition; (c) extracted skeletons; (d) a zoom-in to (b); (e) a zoom-in to (c)

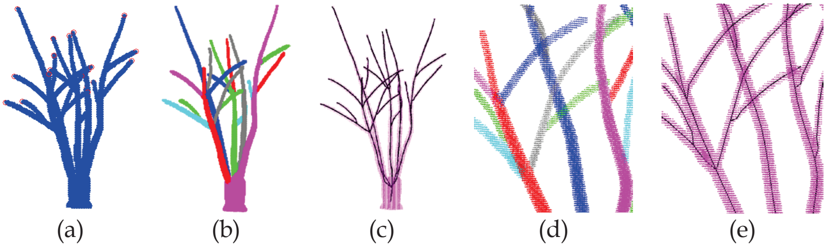


Fig. 15. Synthetic sweep willow branched volume; (a) the volume data; (b) volume decomposition; (c) extracted skeletons; (d) a zoom-in to (b); (e) a zoom-in to (c)

Fig. 14 and Fig. 15 present the results about two synthetic data sets: a virtual sunflower branched volume model and a virtual sweep willow branched volume model. Different colours are adopted to distinguish distinct components. It can be seen that the algorithms decompose the branches into components with appropriate boundaries and the extracted skeletons preserve the volume shape.

Fig. 16 shows the result of skeletons of a virtual colon shell volume data generated by a generalized cylinder spanned from an interpolation curve. The black curve is the extracted skeleton while the blue one is interpolation curve defining the volume surface. It can be seen that the extracted skeleton is very similar to the mathematical skeleton of the cylinder, and its centricity is well kept if the shape curves sharply.

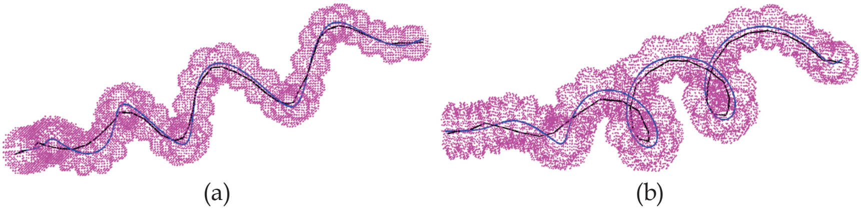


Fig. 16. Synthetic volume data in two views: a virtual colon; (a) the volume and its skeleton curves; (b) a new view of (a)

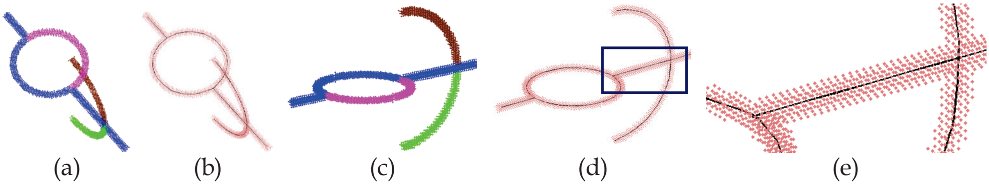


Fig. 17 Synthetic volume data: a mechanical part with a loop; (a) the volume; (b) the extracted skeleton with voxel points as the background; (c) a new view of (a); (d) a new view of (b); (e) a close view of (d).

Fig. 17 shows the results of volume decomposition and skeletons of a synthetic data generated by straight lines and circles. It can be seen that the new approach works robustly on such a shape with a loop, and the volume decomposition is consistent to human perception. The extracted skeletons reflect the topology of the volume and they are smooth.

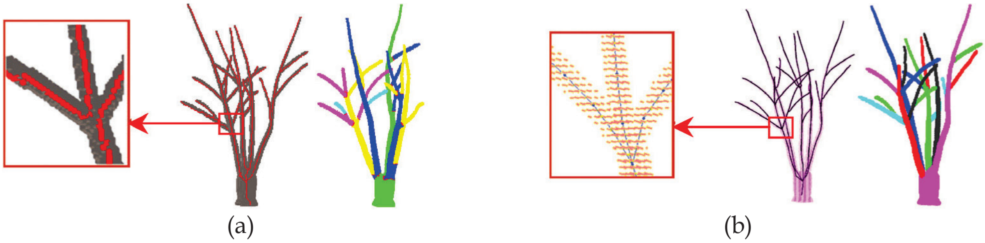


Fig. 18. Comparison on the results of skeletonization; (a) The result of (Zhang et al., 2008); (b) The result of (Xiang et al., 2008)

A comparison is made in Fig. 18 on the qualities between by volume decomposition and skeleton extraction of (Zhang et al., 2008) and by those of (Xiang et al., 2008). It displays that the skeletons in Fig. 18(a) are smoother and more centred than those in Fig. 18(b). In their zoom-in figures, the skeletons extracted by (Xiang et al., 2008) is connected more properly and better centred in the volume than those by (Zhang et al., 2008).
Fig. 19 and Fig. 20 demonstrate the experimental results of the reconstruction of real bonsai trees with high fidelity. Fig. 19 (a) and Fig. 20 (a) are the raw volume data from laser scan and those of (b) are decomposition volume data. Their (c) and (d) are extracted skeletons and final reconstructed mesh models of the branches, respectively. It can be seen that the reconstruction keeps well the tree shape noise-robustly.

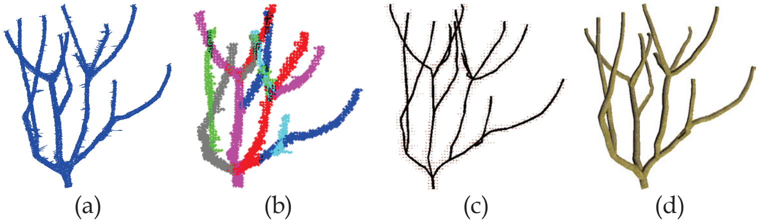


Fig. 19. Reconstruction of a bonsai tree *murraya*; (a) raw data after the detection of tip features; (b) decomposition results; (c) extracted skeletons; (d) final reconstructed models;

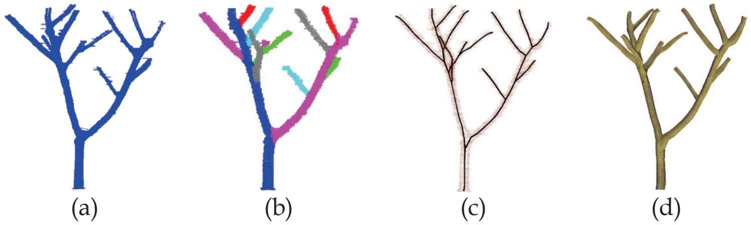


Fig. 20. Reconstruction of a bonsai tree *asclepiadaceae*; (a) raw data after the detection of tip features; (b) decomposition results; (c) extracted skeletons; (d) final reconstructed models;

Table 2 lists the details about the data with our efficient skeletonization algorithm of the six examples, which are the number of voxels in the volume, the number of branches of the data, the skeleton sample steps, and the number of skeleton nodes.

Name of volume model	**Number of Voxels**	**Number of Branches**	**Number of Clusters in an Interval**	**Number of Skeleton Nodes**
Sunflower (Fig. 14)	12,367	13	10	203
Willow (Fig. 15)	49,883	18	10	297
Virtual Colon (Fig. 16)	15,392	1	10	40
Part with a loop (Fig. 17)	3,361	5	5	71
Tree *murraya* (Fig. 18)	3,788	12	5	181
Tree *asclepiadaceae* (Fig. 19)	2,746	13	10	99

Table 2: Volume and skeleton specifications with efficient skeletonization algorithm

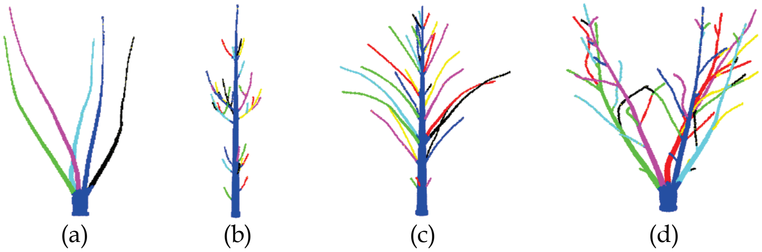


Fig. 21 Volume decomposition of four synthetic tree volume when segmentation surfaces are optimized; (a) a prone willow tree; (b) a holly tree; (c) a poplar tree; (d) a will tree

Fig. 21 and Fig. 22 show the results of volume decomposition and skeleton nodes of the volume data of four virtual trees with more complex topology. We can see that our methods still work well.

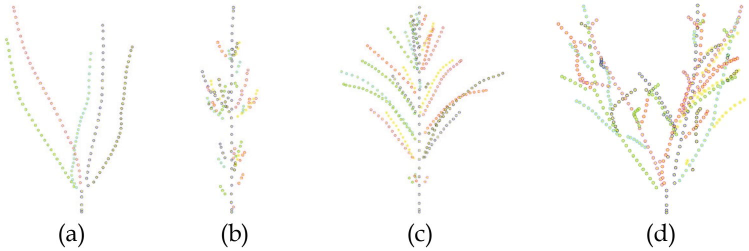


Fig. 22 skeleton nodes of the four synthetic tree volumes in Fig. 17; (a) a prone willow tree; (b) a holly tree; (c) a poplar tree; (d) a will tree

7. Conclusion and Future Work

We proposed algorithms on volume decomposition and hierarchical skeletonization for branched objects in (Zhang et al., 2008). The extracted skeletons are highly centred. One important advantage of our work is the help of volume decomposition to hierarchical skeletons, so that the final extracted skeletons keep the structure of the volume data well.
This work of (Zhang et al., 2008) is extended in (Xiang et al., 2008; Ma et al., 2008; Ma et al., 2009) with the following contributions: segmentations are optimised at ramifications; extracted skeletons are smoother and more robust; the new method is applicable to shapes with a loop and shell volume models; it is more efficient since skeleton reconnection is unnecessary. The experiments are of high fidelity on reconstruction of real *bonsai* trees
We plan to continue our work in three aspects: internal navigation, segmentation of mesh models based on skeletons and the reconstruction of big trees. We would like to guide internal navigation hierarchically inside human organs with a complex topological structure (Bartz, 2005). The work of reconstructing real big trees from laser scan data in (Xu et al., 2007) and (Cheng et al., 2007) could be improved with shape decomposition. To avoid occlusion of laser scan data for the reconstruction of big real trees, the technique of particle flow can be applied. Like the work of (Reniers & Telea, 2007), this approach can also be applied to shape animation in our future work.

8. Acknowledgement

This work is supported in part by National Natural Science Foundation of China with projects No. 60672148 and No. 60872120; in part by the National High Technology Development 863 Program of China under Grant No. 2008AA01Z301, No. 2006AA01Z301, and 2008AA10Z218; in part by Science and Technology Commission of Shanghai Municipality under Grant No. 08511501002; in part by the project; and in part by the Project Arcus 2006 Languedoc-Roussillon/Chine.

9. References

Au, O.; Tai, C.; Chu, H.; Cohen-Or, D. & Lee, T. (2008). Skeleton extraction by mesh contraction. *ACM Transactions on Graphics,* Vol.27, No 3, pp. 1-10, ISSN: 0730-0301.

Amenta, N.; Choi, S.; Kolluri R. (2001) The power crust. in *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pp. 249 - 266; ISBN:1-58113-366-9; Ann Arbor, Michigan, United States; June , 2001; ACM New York, NY, USA

Bartz, D. (2005). Virtual Endoscopy in Research and Clinical Practice, *Computer Graphics Forum ,* Vol.24, No.1, pp. 111-126, ISSN: 0167-7055.

Bitter, I.; Sato, M.; Bender, M.; Mcdonnell, K.; Kaufman, A. & Wan, M. (2000). Ceasar, A smooth, accurate and robust centerline extraction algorithm. In *Proc. Visualization 2000*, pp. 45-52, ISBN: 0-7803-6478-3; Salt Lake City, Utah, United States; October 2000; IEEE Computer Society Press Los Alamitos, CA, USA.

Bitter, I.; Kaufman, A. & Sato, M. (2001). Penalized-distance volumetric skeleton algorithm. *IEEE Transactions on Visualization and Computer Graphics* , Vol.7, No.3, pp. 195-206, ISSN: 1077-2626.

Brostow, G. J. ; Essa, I. ; Steedly, D. & Kwatra, V. (2004). Novel skeletal representation for articulated creatures. In *Proceedings of the European Conference on Computer Vision (ECCV04)*, vol. III, pp.66-78, ISBN: 3-540-21984-6, Prague, Czech Republic; May, 2004; Springer-Verlag, Berlin Heidelberg.

Cheng, C.; Zhang, X. & Chen, B. (2007). Simple reconstruction of tree branches from a single range image. *Journal of Computer Science and Technology* , Vol.22, No.6, pp. 846-858, ISSN: 1000-9000 (print version) ISSN: 1860-4749 (electronic version).

Cornea, N.; Silver, D., Yuan, X. & Balasubrama-Nian, R. (2005). Computing hierarchical curve-skeletons of 3d objects. *The Visual Computer* , Vol.21, no.11, pp. 945-955, ISSN: 0178-2789 (print version), ISSN: 1432-2315 (electronic version).

Cornea, D. & Min, P. (2007). Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics* , Vol.13, No.3, pp. 530-548, ISSN: 1077-2626.

Funkhouser, T.; Kazhdan, M.; Shilane, P.; Min, P., Kiefer, W.; Tal, A.; Rusinkiewicz, S. & Dobkin, D. (2004). Modeling by example, *ACM Transactions on Graphics,* Vol. 23, No. 3, 652-663, ISSN: 0730-0301.

Gagvani, N. & Silver, D. (1999). Parameter controlled volume thinning. *Graphical Models and Image Processing* , Vol.61, no.3, pp.149-164, ISSN: 1049-9652.

Giblin, P. & Kimia, B. (2004). A formal classification of 3d medial axis points and their local geometry *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 26, No.2, pp. 238-251, ISSN: 0162-8828.

Ju, T.; M. Baker, L. & Chiu, W. (2007). Computing a family of skeletons of volumetric models for shape description. *Comput. Aided Des,* Vol.39, No.5, pp. 352-360, ISSN: 0010-4485.

Katz S. & Tal A. (2003). Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Computer Graphics (Proc. of SIGGRAPH 2003)* , Vol. 22, No. 3, 954–961, ISSN: 0730-0301.

Li, X.; T. Woon, W.; Tan, T. S. & Huang, Z. (2001), Decomposing polygon meshes for interactive applications. In I3D '01: *Proceedings of the 2001 symposium on Interactive 3D graphics*, pp.35-42, ISBN:1-58113-292-1, Research Triangle Park, North Carolina, USA, March 2001, ACM Press, New York, NY, USA.

Lien, J. -M. & Amato, N. M. (2004). Approximate convex decomposition of polyhedra. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Posters*, pp. 2, ISBN: 1-58113-896-2, Los Angeles, California, August 2004, ACM Press, New York, NY, USA.

Lien, J. & Amato, N. (2005). Simultaneous shape decomposition and skeletonization using approximate convex decomposition. In *Technical Report, TR05-004,* pp. 44-47, Parasol Laboratory, Department of Computer Science, Texas A M University; College Station, Texas, USA; Dec 2005.

Lien, J. -M. & Amato, N. M. (2007). Approximate convex de-composition of polyhedra. In *Symposium on Solid and Physical Modeling*, pp.121-131, ISBN:1-59593-358-1, Cardiff, Wales, UK; Jun 2006; ACM Press, New York, NY, USA.

Liu, J.; Zhang, X. & Blaise, F. (2004). Distance contained centerline for virtual endoscopy. In *IEEE International Symp. Biomedical Imaging Macro to Nano (ISBI)*, pp. 261-264, ISBN: 0-7803-8388-5, Arlington, VA, USA, April 2004, IEEE Press, New Jersey, USA.

Luboz, V.; Wu, X., Krissian, K.; Westin, C. -F.; Kikinis, R.; Cotin, S. & Dawson, S. (2005). A segmentation and reconstruction technique for 3d vascular structures, In *Proceedings Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005*, pp. 43-50, ISSN: 0938-7994 (Print) 1432-1084 (Online); Palm Springs, CA, USA; October 26-29, 2005; Springer-Verlag, Berlin, Heidelberg.

Ma, C. M. & Sonka, M. (1996). A fully parallel 3D thinning algorithm and its applications. *Computer Vision and Image Understanding,* Vol.64, No.3, pp. 420-433, ISSN: 1077-3142.

Ma, W.; Xiang, B.; Zhang, X. & Zha H. (2008), Decomposition of Branching Volume Data by Tip Detection. In ICIP08: *IEEE International Conference on Image Processing(ICIP)*. pp. 1948-1951. ISBN: 978-1-4244-1765-0; San Diego, California, U.S.A; October, 2008; IEEE Press, New Jersey, USA.

Ma, W.; Xiang, B.; Zha, H.; Liu, J. & Zhang, X. (2009). Modeling Plants with Sensor Data, *Science in China Series F: Information Sciences*, Vol. 52, No. 3, pp. 500-510, ISSN: 1009-2757 (Print) 1862-2836 (Online).

Reniers, D. & Telea, A. (2007), Skeleton-based Hierarchical Shape Segmentation, in: *Proceedings of IEEE International Conference on Shape Modeling and Applications, 2007. SMI '07.* pp. 179-188; ISBN: 0-7695-2815-5; Lyon, France; 13-15 June 2007, IEEE Computer Society Washington, DC, USA

Rom, H. & Medioni, G. (1993). Hierarchical decomposition and axial shape description. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , Vol.15, No.10, pp. 973-98, ISSN: 0162-8828.

Sato, M.; Bitter, I.; Bender, M. & Kaufman, A. (2000). Teasar. Tree-structure extraction algorithm for accurate and robust skeletons. In *Proc. 8th Pacific conf. Computer Graphics and Applications*, pp. 281-289, ISBN: 0-7695-0868-5; Hong Kong, China, October 2000; IEEE Computer Society Washington, DC, USA.

Shahrokni, A.; Zoroofi, R. & Soltanian-Zadeh, H. (2001). Fast skeletonization algorithm for 3d elongated objects, In *Proc. SPIE 4322, Vol. 4322, Medical Imaging 2001*, pp. 323-330, ISBN: 0-8194-4008-6, February 2001, San Diego, CA, USA, SPIE Press, Bellingham, USA.

Siddiqi, K. and Kimia, B.(1995), Parts of visual form: Computational aspects, , *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 17, No. 3, pp. 239-251,. ISSN: 0162-8828

Simmons, M. & Sequin, C. H. (1998). 2D shape decomposition and the automatic generation of hierarchical representations. *International Journal of Shape Modeling,* Vol., No.4, pp. 63-78. ISSN: 0218-6543.

Telea A.; Sminchisescu C. & Dickinson S. (2004). Optimal inference for hierarchical skeleton abstraction. In *ICPR 2004*, pp. 19-22; ISBN:1051-4651; Cambridge, UK; August, 2004; IEEE Computer Society Washington, DC, USA.

Tung, T. & Schmitt, F. (2004). Augmented reeb graphs for content-based retrieval of 3d mesh models. In: *Proceedings of Shape Modeling Applications 2004.*, pp.157-166, ISBN: 0-7695-2075-8; Genoa, Italy; June 2004; IEEE Computer Society, Los Alamitos, California, USA.

Wan, M.; Liang, Z.; Ke, Q.; Hong, L.; Bitter, I. & Kaufman, A. (2002). Automatic centerline extraction for virtual colonoscopy. *IEEE Transactions on Medical Imaging* , Vol.21, No 12, pp. 1450-1460. ISSN: 0278-0062.

Wang, T. & Basu, A. (2007). A note on 'A fully parallel 3D thinning algorithm and its applications'. *Pattern Recognition Letters*, Vol. 28, No. 4. pp. 501-506. ISSN:0167-8655.

Xiang, B.; Zhang, X.; Ma, W. and Zha, H (2008). Skeletonization of Branched Volume by Shape Decomposition. In CCPR08: *Chinese Conference on Pattern Recognition*. pp. 116-121. ISBN: 978-1-4244-2316-3; Beijing, China; October 2008; IEEE Press, New Jersey, USA.

Xu, H.; Gossett, N. & Chen, B. (2007). Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics* , Vol. 26, No.4, pp. 19. ISSN: 0730-0301.

Zhang, X.; Liu, J.; Li, Z. & Jaeger, M (2008). Volume decomposition and hierarchical skeletonization. In VRCAI '08: *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*. ISBN:978-1-60558-335-8; Singapore; December 2008; ACM Press, New York, NY, USA.

Zhang, X.; Liu, J.; Jaeger, M. & Li, Z. (2009). Volume Decomposition for Hierarchical Skeletonization, *The International Journal of Virtual Reality*; to appear

Zhou, Y. & Toga, A. W. (1999). Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics,* Vol. 5, No.3, pp. 196-209. ISSN: 1077-2626.

Structure and Motion from Image Sequences based on Multi-Scale Bayesian Network

Norio Tagawa and Shoichi Naganuma
Tokyo Metropolitan University
Japan

1. Introduction

A lot of studies have been reported on the problem of structure from motion (SFM) as a central theme of computer vision (CV). In the beginning of the research in this field, the principles of 3-D depth recovery and 3-D motion estimation from the viewpoint of mathematics have the attention of a lot of researchers (Adiv, 1985; Huang & Faugeras, 1989; Kanatani, 1993; Longuet-Higgins, 1981; Maybank, 1990; Tagawa et al., 1993; Tsai & Huang, 1984; Zhuang et al., 1988). Subsequently accurate recovery methods have been examined and simultaneously the idea and the role of stochastic inference in CV have been discussed and analyzed (Daniilidis & Nagel, 1990; Kanatani, 1996; Tagawa et al., 1994; Tagawa et al. 1996). Recently, stable and efficient methods represented by the factorization technique (Han & Kanade, 2002; Ke & Kanade, 2005) and methods with no use of camera calibration (Han & Kanade, 2002) have been proposed. Using these methods the development of the virtual reality technique has been intensely advanced. However, the intuitive difficulties of SFM, such as accuracy, high-resolution, computational cost and so on, have not been solved completely. Namely, a practical method for accurately detecting dense motion fields in images and/or relative depth maps between a camera and a target object, keeping spatial discontinuity with low computational cost, has not been established, and many studies have progressed on this problem (Brox et al., 2004; Farnebäck, 2001). Although the method based on the Markov random field (MRF) including a line process (Geman & Geman, 1984) is systematic and the resultant accuracy is significant, its realization in human vision system is not easy because of its computational complexity. On the other hand, increasing the observation information by unifying multiple frames is an important strategy, and the recently regarded methods described above use multi-frame information suitably (Bruhn & Weickert, 2005). However, these techniques assume that tracking of sparse feature points has been performed in advance, and hence, reliable detection caused by integrating temporal information is derived only at the sparse pixels. In this study we introduce a method which can recover dense and accurate depth maps based on two successive frames with no use of the complex MRF including a line process. In this framework, we are going to consider temporal unification in our future research.

Since the detection of 2-dimensional motion field, called optical flow, based on the gradient equations is an ill-posed problem, another condition is required to determine the value of optical flow at each pixel. This issue is called “aperture problem,” and it is a fundamental difficulty causing heteroptics in human vision. In the direct method (Horn & Welden Jr, 1988; Stein & Shashua, 1997) which recovers depth without explicit detection of optical flow, the aperture problem also arises. The aperture problem has been conventionally avoided by either the local optimization (Lucas & Kanade, 1981; Kearney et al., 1987) or the global optimization (Horn & Schunk, 1981). The former assumes that optical flow or depth is locally constant. The latter assumes that the optical flow or depth changes smoothly in the spatial and/or the temporal domains. However, these assumptions impose constraints on the shape of the target directly or indirectly and cause resolution deterioration of the recovered structure. Especially, at the pixels where depth varies discontinuously, for example the contour of the object, the recovered depth might be inaccurate. Lately, in consideration of the case in which intensity invariableness before and after relative camera motion does not hold, methods using intensity constraint as well as geometric constraint are examined (Maki et al. 2002). Although it is possible that these schemes can solve the aperture problem, the research on this issue is still in progress.

As another difficulty with respect to the optical flow detection, the alias problem should be solved. When the intensity pattern with short wavelength in comparison with the size of optical flow is used to detect optical flow, large detection error is observed. This detection error coincides with a usual aliasing phenomenon, and hence a frame rate is not enough to get complete information of the intensity variation. In an active vision scheme, the alias problem can be avoided by making the size of optical flow under a certain value constantly with a suitable time-sampling interval. However, for usual applications, passively taken image sequences are often used to recover depth maps, and then, a method based on a signal processing scheme is desirable. Most of the conventional methods extract spatially smooth intensity patterns by low-pass filtering. Therefore using these techniques, only a low spatial-resolution structure is recovered.

In our study, original images are decomposed into multi-scale images, and the depth information detected using low resolution images is propagated to high resolution images in order to avoid the alias problem and to realize stable and high-resolution recovery using the dynamic Bayesian network spreading to a resolution direction. In this statistical inference processing, by applying the local optimization method and making the local region size smaller as the resolution of the treated images increases, depth discontinuity can be exactly recovered with low computational cost in comparison with the MRF. We introduce an algorithm by which depth map is recovered directly from the spatial and temporal variations of the image intensity without detecting optical flow. Although most of the related research firstly detects optical flow (Tagawa et al., 1995), and recovers depth map using the detected optical flow as an intervening measurement (Tagawa et al., 1996), it is natural for human vision system to extract and recognize optical flow, which is not necessarily caused by rigid motion, and depth map in parallel from the temporal variation of the light intensity received on a retina. From a computational theory, firstly detected optical flow without rigid motion constraints can be regarded as an intermediate solution derived by expanding the solution space, and hence it is not sure that the finally obtained depth map satisfies the exact constraints.

In the dynamic Bayesian network used in this study, each unknown parameter is represented as a node as well as the depth corresponding to each pixel to be recovered and the observed image information. We call this graphical model a multi-scale Bayesian network. If the parameters, including relative 3-D motion between the camera and the object, are determined in advance, the inference of depth map is realized by the Kalman filter (Huang & Ho, 1999; Matthies & Kanade, 1989). For optical flow detection, Simoncelli proposed a method based on the same network in which optical flow is considered as a node and parameters are assumed to be known (Simoncelli, 1999).

The scheme we propose in this chapter is to estimate depth map and parameters simultaneously from image observations using the above described Bayesian network. The parameters to be estimated are common to all multi-scale images, and hence, complete information propagation of both depth and parameters is complicated. Therefore, a suitable approximation has to be adopted. For that purpose we proposed a method based on the MAP-EM algorithm, in which the Laplace approximation is applied to the posterior probabilities of the parameters, and the saddle point approximation is introduced to the posterior of the depth (Tagawa et al., 2008). However, in this method, the variances of the parameters are computed by naïve numerical evaluation of the second order differentials of the log likelihood, which estimation is unstable and inaccurate. In this chapter, we present a new algorithm, in which the Supplemented MAP-EM algorithm (Meng & Rubin, 1991; van Dyk et al., 1995) is adopted to achieve a stable and accurate estimator of the above variances.

2. Principle of Direct Structure and Motion Recovery

2.1 Optical flow caused by rigid motion

We use perspective projection as our camera-imaging model shown in Fig. 1. The camera is fixed with an (X, Y, Z) coordinate system, where the viewpoint, i.e., lens center, is at origin O and the optical axis is along the Z-axis. The projection plane, i.e. image plane, $Z = 1$ can be used without any loss of generality, which means that the focal length equals 1. A space point (X, Y, Z) on the object is projected to the image point (x, y). At each (x, y), the optical flow $[v_x, v_y]^T$ is formulated with an inverse depth $d(x, y) = 1/Z(x, y)$ and the camera's translational and rotational vectors $\mathbf{u} = [u_x, u_y, u_z]^T$, and $\mathbf{r} = [r_x, r_y, r_z]^T$, respectively, are given as follows:

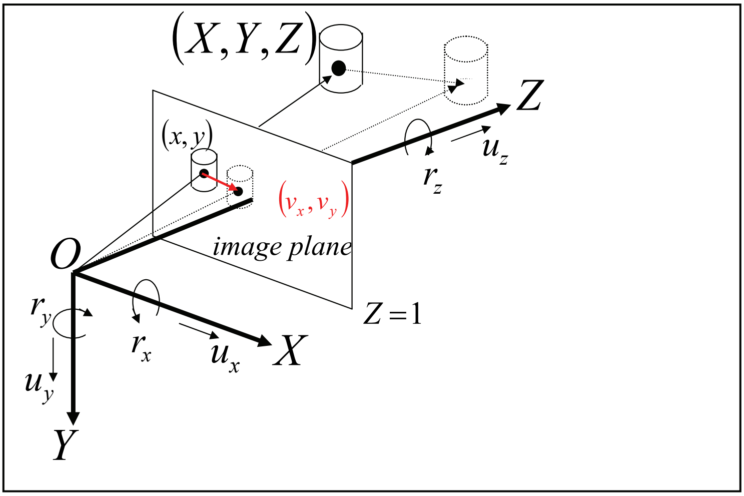


Fig. 1. Camera-imaging model used in this chapter

$$v_x = xyr_x - \left(1 + x^2\right)r_y + yr_z - \left(u_x - xu_z\right)d, \tag{1}$$

$$v_y = \left(1 + y^2\right)r_x - xyr_y - xr_z - \left(u_y - yu_z\right)d. \tag{2}$$

In the above equations, d is an unknown variable at each pixel position, and **u** and **r** are unknown parameters common for the whole image. In the following, Eqs. 1 and 2 are rewritten as

$$v_x = v_x^r(\mathbf{r}) + v_x^u(\mathbf{u})d, \tag{3}$$

$$v_y = v_y^r(\mathbf{r}) + v_y^u(\mathbf{u})d. \tag{4}$$

2.2 Gradient equation for rigid motion

The optical flow constraint equation, which is called the "gradient equation," is the first approximation of the assumption that image intensity is invariable before and after the relative 3-D motion between a camera and an object. At each pixel (x, y) in the image, the gradient equation is formulated with the partial differentials f_x, f_y and f_t, where t denotes time, of the image intensity $f(x, y, t)$ and the optical flow

$$f_t = -f_x v_x - f_y v_y. \tag{5}$$

By substituting Eqs. 3 and 4 into Eq. 5, the gradient equation representing a rigid motion constraint can be derived explicitly as

$$\begin{aligned} f_t &= -\left(f_x v_x^r + f_y v_y^r\right) - \left(f_x v_x^u + f_y v_y^u\right)d \\ &\equiv -f^r - f^u d. \end{aligned} \tag{6}$$

This equation can be used for the direct recovering the structure and the motion (Horn & Weldon Jr, 1988; Stein & Shashua, 1997). In this study, we use a convolution kernel defined as a derivative of the suitable interpolator kernel (Farid & Simoncelli, 1997), which is described in Sec.6.2 in detail, to calculate f_x and f_y accurately, and f_t is detected as the finite difference using two successive frames. Hence, we suppose that only f_t contains observation error, and we use Eq. 6 as the observation equation.

It is obvious that from Eq. 6 the norm of **u** and d cannot be uniquely determined. Therefore, we suppose that the norm of **u** equals 1, and consider **u** to be **u** = $[u_x, u_y, (1-u_x^2-u_y^2)^{1/2}]^T$.

3. Outline of Depth Recovery with Multi-Scale Processing

3.1 Problems definition

In order to recover accurate and high resolution depth map, the alias problem and the aperture problem have to be avoided. Firstly, we briefly explain the aperture problem. By representing the optical flow at each pixel as $[v_x, v_y]^T$, from the relation $k_t = v_x k_x + v_y k_y$ between the spatial wavenumber (k_x, k_y) and the temporal wavenumber k_t, it can be known that the Nyquist frequency of time-sampling takes different value for each spatial frequency.

However, usual time-sampling rate for each pixel, i.e. frame rate, is constant independently of the spatial frequencies. Hence, the possibility that the time-sampling rate for high spatial frequencies can not satisfy the sampling theory is higher than such possibility for low spatial frequencies. Therefore, a lot of conventinal methods analyze only the low resolution images extracted by low-pass filtering. In these techniques, high resolution information appears to be lost, which causes deterioration of the recovered structure.

Subsequently, the aperture problem means that if some pairs of (f_x, f_y) take the same values in the local region in the image, where the optical flow can be assumed to be constant, we can not determine the optical flow uniquely by solving simultaneously the multiple equations corresponding to Eq. 5 obtained at the pixels in such local region. The essential cause for the aperture problem is the fact that Eq. 5 is ill-posed. The conventional methods which can cope with the problem are roughly devided into two types: the local optimization method and the global optimization method. The local optimization method assumes that the optical flow or the corresponding depth in each spatial local region and/or in each temporal local region is constant. Such the local regions have to be determined so that (f_x, f_y)s take various values. Then, each optical flow and each corresponding depth are obtained using the multiple equations in Eq. 5. By this approach, local constraints are introduced into the unknown structure, and only linear computations are required. On the other hand, the global optimization method uses the global constraints represented by the spatial smoothness of the optical flow or the depth, and concretely solves differential equations derived through the variational principle.

As mentioned above, introducing limitations to the image information used to recover depth, i.e. low frequency components are extracted and used, for the alias problem, and introducing limitations to the unknown structure, i.e. resolution and hence degree of freedom of structure are lowered, for the aperture problem, are important approaches. Much conventional methods solve the aperture problem and the alias problem by applying such kinds of limitations. However, they cause resolution lowering of the obtained structure as compared with the raw image resolution, and then, accuracy of recovering is decreased especially at the discontinuous parts of the structure.

3.2 Fundamental concept of proposed solution

In the method described in this chapter, we aim to solve the above described two problems without lowering the resolution of recovered structure. Fundamental ideas are (i) decomposing images into multi-scale images each of which has proper spatial frequencies, (ii) adapting the assumed resolution of the structure to the resolution of the used image at each scale processing, and (iii) advancing the processing from low resolutions to high resolutions sequentially.

The combination of (i) and (iii) deals with the alias problem. Although it is desirable that low resolution images are used in order to avoid aliasing, it is important that high resolution images are suitably analyzed in order not to lower the resolution of the recovered structure. Therefore, the employment of high resolution information with avoiding the aliasing can be performed by sequentially propagating the stable depth map with no aliasing obtained from low resolution step to high resolution step. This strategy is the same as the one proposed in the previous study (Simoncelli, 1999) for optical flow detection. By adopting the Bayesian inference as an information propagation scheme, depth error in low resolution can be compensated by high resolution processing.

The combination of (i) and (ii) deals with the aperture problem. If we adopt the scheme (i) to avoid the aliasing, the obtained multi-scale depth information has naturally a hierarchical structure. Hence, it is appropriate that spatial resolution of the depth variable is lowered for low resolution step and inversely it is improved for high resolution step. Such control of the resolution depending on the image resolution can be done by both the local optimization method and the global optimization method, and in this study, we use the local optimization scheme, since it can be performed by local computations. Basically, for the low resolution step the size of the region where depth is assumed to be constant is expanded, and as resolution becomes high, this size is contracted.

In addition to the methodological feature of our solution described above, the following technical originality is asserted in this study. If the parameters contained in our statistical model are determined in advance, the Kalman filter can be applied to the inference of depth map (Huang & Ho, 1999; Matthies & Kanade, 1989). However, such condition can not be generally supposed, and hence, our solution can estimate both depth map and parameters by information propagation on the dynamic Bayesian network spreading to the image resolution direction. In this scheme, a strategy with low computational cost is quite important, since simultaneous complete belief propagation (BP) is a complicated problem. Therefore, we adopt suitable approximations, i.e. the Laplace approximation is applied to the posterior probabilities of the parameters and the saddle point approximation is introduced to the posterior of the depth, and additionally, the Supplemented MAP-EM algorithm (Meng & Rubin, 1991) is applied as a stable and effective processing. The remarkable feature of the Supplemented MAP-EM algorithm compared with the simple MAP-EM algorithm is that by the Supplemented MAP-EM algorithm the asymptotic variance-covariance of the parameters can be estimated using only the code for the MAP-EM algorithm itself and the code for standard matrix operation instead of the numerical differentials. The asymptotic variance-covariance matrix is necessary for BP, and then, the application of the Supplemented MAP-EM algorithm is a strong assertion in this study. This framework can be used for a lot of fields in which the usual Kalman filter can be applied effectively.

4. Computation Principle based on Multi-Scale Processing

4.1 Image decomposition and probabilistic models

Firstly, we explain decomposition of observed images into multi-scale images. In order to simply decompose those into multiple images having different resolutions, it is reasonable to vary the cut-off frequency of the spatial low-pass filter. However, since we assume for the following processing that the observation errors contained in the temporal differentials of the image intensity have statistically independency among different resolutions, we adopt the decomposition using a spatial band-pass filter. In the following, index $l = 1, 2, \dots, L$ represents the difference of the frequency band of the image, and $l = 1$ indicates the lowest resolution image, called the lowest image hereafter. The resolution of the depth map for each resolution image is defined according to its frequency band which corresponds to the spatial wavelength. Namely, for the image indexed by l, depth is assumed to be constant in a local region which size is N_l, and size N_l is determined along with a rule that $N_{l_1} < N_{l_2}$ if

$l_1 > l_2$. Although a pyramid structure of multi-scale images, for example the Wavelet transform, is suitable for effective treatment of the information, all information observed at all pixels is used without performing down sampling.

Next, we define the probabilistic models. As mentioned in Sec. 2.2, translational vector **u** is described as $\mathbf{u} = [u_x, u_y, (1-u_x^2-u_y^2)^{1/2}]^T$, hence the scale of d should be fixed by this normalization of **u**. As mentioned above, d is a constant in the local region for each resolution image, and the constants for all the local regions are independent. Among the different resolutions, d is supposed to be a conditionally independent unknown variable. This is based on the definition

$$d^{(l+1)} = \mathrm{I}^{(l)}\left[d^{(l)}\right] + n_0^{(l)}, \tag{7}$$

where $\mathrm{I}^{(l)}$ indicates a linear interpolation operator from l to l+1. In this equation, $n_0^{(l)}$ is a Gaussian random variable with zero mean and variance σ_0^2 common to all local regions and all resolutions, which means a perturbation from the interpolation value and it is statistically independent of other random variables. The probability density function of $d^{(l+1)}$ is given as

$$p\left(d^{(l+1)} \middle| \sigma_0^2\right) = \frac{1}{\sqrt{2\pi\sigma_d^{2(l+1)}}} \exp\left[-\frac{\left(d^{(l+1)} - m_d^{(l+1)}\right)^2}{2\sigma_d^{2(l+1)}}\right], \tag{8}$$

where the mean $m_d^{(l+1)}$ and the variance $\sigma_d^{2(l+1)}$ can be formulated recuirsively as follows:

$$m_d^{(l+1)} = \mathrm{I}^{(l)}\left[m_d^{(l)}\right], \tag{9}$$

$$\sigma_d^{2(l+1)} = \mathrm{I}^{(l)2}\left[\sigma_d^{2(l)}\right] + \sigma_0^2. \tag{10}$$

Equation 10 indicates an approximated representation in which the covariance terms of $d^{(l)}$ are neglected and hence only the variance terms are considered. Moreover, $\mathrm{I}^{(l)^2}$ is also the linear interpolation operator, the weight coefficients of which correspond to the power of each of the corresponding coefficient of $\mathrm{I}^{(l)}$.

Subsequently, we define a probabilistic model for an observation f_t. An observation equation containing an observation error is given based on Eq. 6 as follows:

$$f_t^{(l)} = -f^{r(l)} - f^{u(l)} d^{(l)} + n_1^{(l)}. \tag{11}$$

There have been several discussions with respect to a noise model of Eq. 11 (Nestares et al., 2000; Weiss & Fleet, 2001). For simplicity, the observation error $n_1^{(l)}$ is a Gaussian random variable with zero mean and variance σ_1^2 common to all local regions and all resolutions and it is statistically independent of other random variables. From Eq. 11, $f_t^{(l)}$ is also a Gaussian random variable and its conditional probability density can be formulated as

$$p\left(f_t^{(l)}\middle|d^{(l)},\sigma_1^2\right)=\frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\left[-\frac{\left(f_t^{(l)}+f^{r(l)}+f^{u(l)}d^{(l)}\right)^2}{2\sigma_1^2}\right]. \qquad (12)$$

The probabilistic model defined above can be represented as a graphical model, i.e. the Bayesian network shown in Fig. 2. In this network, the parameters $\Theta \equiv \left\{u_x,u_y,\mathbf{r},\sigma_0^2,\sigma_1^2\right\}$, which are also shown in Fig. 2, are regarded as probabilistic variables estimated through BP, which is described in the following section. In this figure, the parameters are shown not to be common to the layers formally, but they are assumed to be constants with respect to the layers in this study.

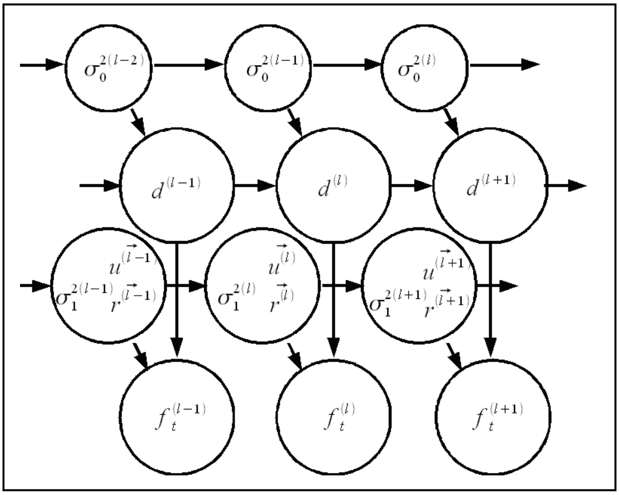


Fig. 2. Graphical model used in this chapter

4.2 Hierarchical estimation of depth

Based on the probabilistic models defined in Sec. 4.1, we are going to find a minimum variance estimator $\hat{d}^{(l+1)}$, when $\left\{f_t^{(l+1)}\right\},\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\}$ are observed, where the symbol $\{a\}$ represents a set of arbitrary values a defined at all the pixels. Such $\hat{d}^{(l+1)}$ corresponds to the mean of a posterior probability of $d^{(l+1)}$ after obtaining all observations. This posterior probability is introduced as follows.

Let $\left\{f_t^{(l)}\right\}_{\mathrm{N}}$ be a set of $f_t^{(l)}$ in a local region N in an image where $d^{(l)}$ is assumed to be constant, and the next holds

$$p\left(d^{(l+1)}\middle|\left\{f_t^{(l+1)}\right\},\left\{f_t^{(l)}\right\},\cdots\right)=p\left(d^{(l+1)}\middle|\left\{f_t^{(l+1)}\right\}_{\mathrm{N}},\left\{f_t^{(l)}\right\},\cdots\right) \tag{13}$$

This can be rewritten using the Bayes formula as follows:

$$p\left(d^{(l+1)}\middle|\left\{f_t^{(l+1)}\right\}_{\mathrm{N}},\left\{f_t^{(l)}\right\},\cdots\right)=\frac{p\left(\left\{f_t^{(l+1)}\right\}_{\mathrm{N}}\middle|d^{(l+1)}\right)p\left(d^{(l+1)}\middle|\left\{f_t^{(l)}\right\},\cdots\right)}{p\left(\left\{f_t^{(l+1)}\right\}_{\mathrm{N}}\middle|\left\{f_t^{(l)}\right\},\cdots\right)}. \tag{14}$$

The numerator of the right hand side of Eq. 14 can be concretely shown. Firstly, the second term of this numerator can be written using $d^{(l)}$ which resolution is one-step lower than $d^{(l+1)}$

$$p\left(d^{(l+1)}\middle|\left\{f_t^{(l)}\right\},\cdots\right)=\int p\left(d^{(l+1)}\middle|\left\{d^{(l)}\right\}_{\mathrm{I}}\right)p\left(\left\{d^{(l)}\right\}_{\mathrm{I}}\middle|\left\{f_t^{(l)}\right\},\cdots\right)d\left\{d^{(l)}\right\}_{\mathrm{I}}, \tag{15}$$

$$p\left(\left\{d^{(l)}\right\}_{\mathrm{I}}\middle|\left\{f_t^{(l)}\right\},\cdots\right)=\prod_{\mathrm{I}} p\left(d^{(l)}\middle|\left\{f_t^{(l)}\right\}_{\mathrm{N}},\cdots\right), \tag{16}$$

where $\left\{\cdot\right\}_{\mathrm{I}}$ indicates a set of the elements used for the linear interpolation. From Eqs. 7 and 11, Eq. 15 also represents the Gaussian distribution, and using the mean $\tilde{d}^{(l)}$ and the variance $\tilde{\sigma}_d^{2(l)}$ of the posterior distribution $p(d^{(l)}|\left\{f_t^{(l)}\right\},\cdots)$ for resolution l, Eq. 15 can be represented as Eq. 8, i.e.,

$$p\left(d^{(l+1)}\middle|\left\{f_t^{(l)}\right\},\cdots\right)=\frac{1}{\sqrt{2\pi\left(\mathrm{I}^{(l)^2}\left[\tilde{\sigma}_d^{2(l)}\right]+\sigma_0^2\right)}}\exp\left[-\frac{\left(d^{(l+1)}-\mathrm{I}^{(l)}\left[\tilde{d}^{(l)}\right]\right)^2}{2\left(\mathrm{I}^{(l)^2}\left[\tilde{\sigma}_d^{2(l)}\right]+\sigma_0^2\right)}\right]. \tag{17}$$

Therefore, the numerator of Eq. 14 can be formulated as follows:

$$p\left(\left\{f_t^{(l+1)}\right\}_{\mathrm{N}}\middle|d^{(l+1)}\right)p\left(d^{(l+1)}\middle|\left\{f_t^{(l)}\right\},\cdots\right)=\left[\prod_{\mathrm{N}} p\left(f_t^{(l+1)}\middle|d^{(l+1)}\right)\right]p\left(d^{(l+1)}\middle|\left\{f_t^{(l)}\right\},\cdots\right)=\frac{\exp(-H)}{Z}, \tag{18}$$

$$Z=\left(\sqrt{2\pi\sigma_1^2}\right)^N\sqrt{2\pi\left(\mathrm{I}^{(l)2}\left[\tilde{\sigma}_d^{2(l)}\right]+\sigma_0^2\right)}, \tag{19}$$

$$H=\frac{\sum_{\mathrm{N}}\left(f_t^{(l+1)}+f^{r(l+1)}+f^{u(l+1)}d^{(l+1)}\right)^2}{2\sigma_1^2}+\frac{\left(d^{(l+1)}-\mathrm{I}^{(l)}\left[\tilde{d}^{(l)}\right]\right)^2}{2\left(\mathrm{I}^{(l)2}\left[\tilde{\sigma}_d^{2(l)}\right]+\sigma_0^2\right)}=\frac{\left(d^{(l+1)}-\tilde{d}^{(l+1)}\right)^2}{2\tilde{\sigma}_d^{2(l+1)}}+\mathrm{const.}, \tag{20}$$

$$\tilde{d}^{(l+1)} = \tilde{\sigma}_d^{2(l+1)} \left(\frac{\mathrm{I}^{(l)}\left[\tilde{d}^{(l)}\right]}{\mathrm{I}^{(l)2}\left[\tilde{\sigma}_d^{2(l)}\right] + \sigma_0^2} - \frac{\sum_{\mathrm{N}} \left(f_t^{(l+1)} + f^{r(l+1)} \right) f^{u(l+1)}}{\sigma_1^2} \right), \tag{21}$$

$$\tilde{\sigma}_d^{2(l+1)} = \left(\frac{1}{\mathrm{I}^{(l)2}\left[\tilde{\sigma}_d^{2(l)}\right] + \sigma_0^2} + \frac{\sum_{\mathrm{N}} f^{u(l+1)^2}}{\sigma_1^2} \right)^{-1}, \tag{22}$$

where N is the number of pixels in the region N. The estimator of $d^{(l+1)}$ minimizing H formulated by Eq. 20 is the MAP (Maximum A Posteriori) estimator $d_{MAP}^{(l+1)}$ and it coincides with the mean of the posterior probability in Eq. 14. Hence, $d_{MAP}^{(l+1)} = \tilde{d}^{(l+1)}$ holds. For the Gaussian distribution, the MAP estimator is also the minimum variance estimator. It can be known that $\tilde{d}^{(l+1)}$ is computed recursively using Eq. 21, and this procedure is an essential operation of the Kalman filter. Equations 7 and 11 correspond to the state equation and the observation equation, respectively.

The Kalman filter is usually used to estimate effectively a marginal posterior probability of inner state at the present time using observations already measured. Since we aim to estimate $\left\{d^{(L)}\right\}$, and $\left\{d^{(l)}\right\}(l < L)$ is not explicitely needed, in the proposed scheme, we can determine $\left\{d^{(L)}\right\}$ in a way similar to the Kalman filter strategy, i.e. we use the marginal posterior probability $p(\left\{d^{(L)}\right\} | \left\{f_t^{(L)}\right\}, \left\{f_t^{(L-1)}\right\}, \cdots)$ instead of the simultaneous posterior probability $p(\left\{d^{(L)}\right\}, \left\{d^{(L-1)}\right\}, \cdots | \left\{f_t^{(L)}\right\}, \left\{f_t^{(L-1)}\right\}, \cdots)$ to determine $\left\{d^{(L)}\right\}$. This estimator is often called the MPM (Marginal Posterior Mode) estimator. Another essential reason for using the Kalman filter strategy is the fact that the alias problem should be avoided. When $f_t^{(l)}$ is measured as a difference using two succsessive frames, if the amplitude of the 2-D motion is larger than the spatial wavelength corresponding to the spatial frequency band of $f^{(l)}$, the undesirable aliasing occurs. Therefore, we define $f_t^{(l)}$ in which alising can not be seen using the estimated optical flow $\hat{\mathbf{v}}^{(l-1)} = \left[\hat{v}_x^{(l-1)}, \hat{v}_y^{(l-1)}\right]^{\mathrm{T}}$ calculated indirectly from the depth $\tilde{d}^{(l-1)}$ obtained for the one-step low resolution images $f^{(l-1)}$ as follows:

$$f_t^{(l)} = -f_x^{(l)} \mathrm{I}^{(l-1)}\left[\hat{v}_x^{(l-1)}\right] - f_y^{(l)} \mathrm{I}^{(l-1)}\left[\hat{v}_y^{(l-1)}\right] + \frac{\partial}{\partial t} \mathrm{W}\left[f^{(l)}, \mathrm{I}^{(l-1)}\left[\hat{\mathbf{v}}^{(l-1)}\right]\right], \tag{23}$$

where a warp operation is defined as

$$\mathrm{W}[f, \mathbf{v}](\mathbf{x}, t + \delta t) \equiv f(\mathbf{x} - \mathbf{v}\delta t, t + \delta t), \tag{24}$$

and partial differential $\partial / \partial t$ is done as a finite difference. From the above definitions, the optical flow estimations at the lower resolution step, which estimations have little risk of aliasing, should be used successively, so as to avoid aliasing and detect stable $f_t^{(l)}$. For such purpose, the sequential procedure from low resolution step to high resolution step is required.

4.3 Computation flow including parameter determination

For the successive estimation described in the above section, there are some parameters which should be known in advance. It is usual that parameters are treated as definite variables and are determined as a maximum likelihood (ML) estimator. On the other hand, it is known that the MAP estimator obtained by considering a parameter as a probabilistic variable formally having a uniform distribution coincides with a ML estimator. Hence, in this study, we suppose that the parameters are probabilistic variables as well as depth.

Since we assume that σ_0^2 and σ_1^2 are common to all resolution steps, $\Theta = \{u_x, u_y, \mathbf{r}, \sigma_0^2, \sigma_1^2\}$ has to be determined with no dependence of each resolution processing. The information of $\{d^{(l)}\}$ is propagated from low resolution to high resolution, therefore we estimate Θ by the same scheme and adopt the Bayesian inference formally supposing a prior of Θ . The posterior probability density of Θ can be decomposed in the following way

$$p\left(\Theta \middle| \{f_t^{(L)}\},\cdots,\{f_t^{(1)}\}\right) \tag{25}$$
$$= \frac{p\left(\{f_t^{(L)}\}\middle|\{f_t^{(L-1)}\},\cdots,\{f_t^{(1)}\},\Theta\right)}{p\left(\{f_t^{(L)}\}\middle|\{f_t^{(L-1)}\},\cdots,\{f_t^{(1)}\}\right)} \frac{p\left(\{f_t^{(L-1)}\}\middle|\{f_t^{(L-2)}\},\cdots,\{f_t^{(1)}\},\Theta\right)}{p\left(\{f_t^{(L-1)}\}\middle|\{f_t^{(L-2)}\},\cdots,\{f_t^{(1)}\}\right)} \cdots \frac{p\left(\{f_t^{(1)}\}\middle|\Theta\right)p(\Theta)}{p\left(\{f_t^{(1)}\}\right)}.$$

We assume that the last term at the right hand side of Eq. 25, i.e. $p\left(\Theta\middle|\{f_t^{(1)}\}\right)$, is explicitly known. After observing $\{f_t^{(2)}\}$ from the next higher resolution images, we can compute

$$\frac{p\left(\{f_t^{(2)}\}\middle|\{f_t^{(1)}\},\Theta\right)}{p\left(\{f_t^{(2)}\}\middle|\{f_t^{(1)}\}\right)} p\left(\Theta\middle|\{f_t^{(1)}\}\right) = \frac{p\left(\{f_t^{(2)}\},\Theta\middle|\{f_t^{(1)}\}\right)}{p\left(\{f_t^{(2)}\}\middle|\{f_t^{(1)}\}\right)} = p\left(\Theta\middle|\{f_t^{(2)}\},\{f_t^{(1)}\}\right) \tag{26}$$

In this equation, the term $p\left(\{f_t^{(2)}\}\middle|\{f_t^{(1)}\},\Theta\right)$ can be derived using Eqs. 12 and 17 as follows:

$$p\left(\{f_t^{(2)}\}\middle|\{f_t^{(1)}\},\Theta\right) = \int p\left(\{f_t^{(2)}\}\middle|\{d^{(2)}\},\Theta\right) p\left(\{d^{(2)}\}\middle|\{f_t^{(1)}\},\Theta\right) d\{d^{(2)}\}. \tag{27}$$

It should be noticed that Θ is omitted in Eqs. 12 and 17 and in these equations Θ means a true value, but in Eq. 27 Θ is considered as a variable. By propagating the computation of Eq. 26 successively from low resolution images to high resolution images, we can finally obtain the left hand side of Eq. 25. This procedure also coincides with Kalman filtering for a state variable having no dynamic transition. By assuming a prior having a large entropy for $p(\Theta)$, for example a uniform distribution, the Bayesian estimator Θ_{MAP} approximately equals to the ML estimator.

We again refer the successive estimation of $\left\{d^{(l)}\right\}$, and also in Eqs. 14, 21 and 22 the true value of Θ is necessary to be known. Correctly, Eq. 14 should be written as $p\left(d^{(l+1)}\middle|\left\{f_t^{(l+1)}\right\}_{\mathrm{N}},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)$. Hence, in order to solve a problem, in which Θ is unknown as well as $\left\{d^{(L)}\right\}$, the marginalization with respect to Θ is required to propagate the information of depth as follows:

$$p\left(d^{(l+1)}\middle|\left\{f_t^{(l+1)}\right\}_{\mathrm{N}},\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)=\int p\left(d^{(l+1)}\middle|\left\{f_t^{(l+1)}\right\}_{\mathrm{N}},\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)p\left(\Theta\middle|\left\{f_t^{(l+1)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)d\Theta. \tag{28}$$

From the above discussion, the flow of procedures at each resolution step is summarized in the following five steps. We assume that at the lth layer the posteriors of Θ and $\left\{d^{(l)}\right\}$ are already derived as $p\left(\Theta\middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)$ and $p\left(d^{(l)}\middle|\left\{f_t^{(l)}\right\}_{\mathrm{N}},\cdots,\left\{f_t^{(1)}\right\}\right)$, respectively.

(i) Using Eq. 17, compute the predictive posterior probability $p\left(d^{(l+1)}\middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)$ from $p\left(d^{(l)}\middle|\left\{f_t^{(l)}\right\}_{\mathrm{N}},\cdots,\left\{f_t^{(1)}\right\}\right)$.

(ii) By generalizing Eqs. 26 and 27 for (l+1)th layer, compute the posteriors of the parameters $p\left(\Theta\middle|\left\{f_t^{(l+1)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)$ from $p\left(\Theta\middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)$ at the previous layer, $p\left(d^{(l+1)}\middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)$ obtained at (i) and $p\left(f_t^{(l+1)}\middle|d^{(l+1)},\Theta\right)$ definded in Eq. 12.

(iii) Using $p\left(d^{(l+1)}\middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)$ and $p\left(f_t^{(l+1)}\middle|d^{(l+1)},\Theta\right)$ alike in (ii), compute the posterior of depth conditioned on the parameters $p\left(d^{(l+1)}\middle|\left\{f_t^{(l+1)}\right\}_{\mathrm{N}},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)$.

(iv) Using Eq. 28, compute the marginal posterior of depth $p\left(d^{(l+1)}\middle|\left\{f_t^{(l+1)}\right\}_{\mathrm{N}},\cdots,\left\{f_t^{(1)}\right\}\right)$ from $p\left(d^{(l+1)}\middle|\left\{f_t^{(l+1)}\right\}_{\mathrm{N}},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)$ obtained in (iii) and $p\left(\Theta\middle|\left\{f_t^{(l+1)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)$ obtained in (ii).

(v) The above procedures should be repeated for the next resolution layer.

There is another scheme to determine the parameters besides the above one. By considering both $\left\{d^{(l)}\right\}$ and Θ as state variables, their successive updating can be performed. Based on the extended Kalman filter theory, this can be formulated by linearizing locally the observation equation and the state equation with respect to Θ. However, the obtained estimator is the Bayesian estimator based on the simultaneous posterior $p\left(\left\{d^{(l)}\right\},\Theta\middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)$. On the other hand, our scheme can obtain the estimators based on both marginal posteriors $p\left(\left\{d^{(l)}\right\}\middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)$ and $p\left(\Theta\middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)$. Since $\left\{d^{(l)}\right\}$ and Θ have no special meaning as a pair, the estimation scheme in our study is adequate.

5. Computing Algorithm

5.1 Application of EM algorithm for approximation

In order to approximately execute the BP procedure mentioned in the above section, we can use a framework with the EM algorithm (Dempster et al., 1977). The EM algorithm is an effective scheme if applied to the problem which is easy to be solved if some unknown variables, often called the ˝hidden variables˝, are observed in addition to actual observations. Such hidden variables and observations considered together are called the ˝complete data,˝ and the observations themselves are called the ˝incomplete data.˝ By the EM algorithm, the posterior probabilities of the hidden variables and the ML estimators of the parameters can be obtained through iterative procedures. Additionally, by the MAP-EM algorithm extended version of EM algorithm, the MAP estimation of the parameters can be performed based on the posterior probabilities of the parameters marginalized with respect to the hidden variables. Although the details are omitted, the following two steps are executed at the each iteration:

[E-step]: The posterior probabilities of the hidden variables are derived using the parameters values estimated at the previous iteration. Using these probabilities the evaluation function, called Q function, needed for the parameters updating is introduced.

[M-step]: The parameters values are updated by maximizing the Q function introduced in the E-step.

In this study, the MAP-EM algorithm is applied for each resolution l, and hence, the MAP estimator Θ_{MAP} using the prior $p\left(\Theta \middle| \left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)$ and the MAP estimator $\left\{d_{MAP}^{(l)}\right\}$ based on its posterior conditioned by Θ_{MAP} are determined. These estimators are not completely the ones given in Eqs. 25 and 28. However, the application of the MAP-EM algorithm with supplemented function described in the following subsections in details enables an efficient computation algorithm for our problem, and the estimators derived by this algorithm can be assumed to be appropriate approximations.

5.2 MAP-EM algorithm for determining depth and motion

In this section, we introduce the explicit formulations for the resolution l, i.e. the layer l, to obtain Θ_{MAP} and $\left\{d_{MAP}^{(l)}\right\}$ using the images which resolutions are not exceeding l. At the E-step, the following Q function with respect to Θ is constructed

$$Q\left(\Theta;\hat{\Theta}\right)=\mathrm{E}\left[\ln p\left(\left\{f_t^{(l)}\right\},\left\{d^{(l)}\right\}\middle|\left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)+\ln p\left(\Theta\middle|\left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)\ \middle|\ \left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\},\hat{\Theta}\right], \tag{29}$$

where $\mathrm{E}\left[\ \cdot\ \middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\},\hat{\Theta}\right]$ denotes the conditional expectation using $p\left(\left\{d^{(l)}\right\}\middle|\left\{f_t^{(l)}\right\},\cdots,\left\{f_t^{(1)}\right\},\hat{\Theta}\right)$, and in the following formulations including Eq. 29, the symbol $\hat{}$ indicates the estimate or variable depending on the estimate derived in this iteration. In Eq. 29, the simultaneous probability of $\left\{f_t^{(l)}\right\}$and $\left\{d^{(l)}\right\}$ is

$$p\left(\left\{f_t^{(l)}\right\},\left\{d^{(l)}\right\}\middle|\left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)=p\left(\left\{f_t^{(l)}\right\}\middle|\left\{d^{(l)}\right\},\Theta\right)p\left(\left\{d^{(l)}\right\}\middle|\left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\},\Theta\right) \tag{30}$$
$$=\prod_{i=1}^{M^{(l)}}\left[\prod_{\mathrm{N}_i^{(l)}}p\left(f_t^{(l)}\middle|d_i^{(l)},\Theta\right)\right]\prod_{i=1}^{M^{(l)}}p\left(d_i^{(l)}\middle|\left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\},\Theta\right)$$

In Eq. 30, we suppose that the number of pixels in the local region $\mathrm{N}_i^{(l)}$ is constant with respect to the local region index i and it takes value $N^{(l)}$. Additionally, the number of the local regions in an image is given by $M^{(l)}$. The function which expectation is computed in Eq. 29 is concretely written using Eqs. 12 and 17 as follows:

$$-\frac{N^{(l)}M^{(l)}}{2}\ln\sigma_1^2-\frac{1}{2}\sum_{i=1}^{M^{(l)}}\ln\left(\mathrm{I}^{(l-1)^2}\left[\tilde{\sigma}_d^{2(l-1)}\right]_i+\sigma_0^2\right)-\frac{1}{2\sigma_1^2}\sum_{i=1}^{M^{(l)}}\sum_{\mathrm{N}^{(l)}}\left(f_t^{(l)}+f^{r(l)}+f^{u(l)}d_i^{(l)}\right)^2 \tag{31}$$
$$-\frac{1}{2}\sum_{i=1}^{M^{(l)}}\frac{\left(d_i^{(l)}-\mathrm{I}^{(l-1)}\left[\tilde{d}^{(l-1)}\right]_i\right)^2}{\left(\mathrm{I}^{(l-1)^2}\left[\tilde{\sigma}_d^{2(l-1)}\right]_i+\sigma_0^2\right)}-\frac{1}{2}\left(\Theta-\tilde{\Theta}\right)^{\mathrm{T}}\tilde{\mathbf{V}}^{-1}\left(\Theta-\tilde{\Theta}\right)+\mathrm{Const.},$$

where the Laplace approximation is applied to $p\left(\Theta\middle|\left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)$, i.e. it is approximated by a Gaussian distribution, since it has a complex form. To simplify the computations, we assume that there is no correlation between each two of $\mathbf{m}\equiv\left\{u_x,u_y,\mathbf{r}\right\}$, σ_0^2 and σ_1^2. As parameters of the Laplace approximation, the mean $\tilde{\Theta}$ and the covariance $\tilde{\mathbf{V}}$ of Θ at layer l-1 are used. Their estimation method is described in the next section.

On the other hand, $p\left(d_i^{(l)}\middle|\left\{f_t^{(l)}\right\}_{\mathrm{N}_i},\cdots,\left\{f_t^{(1)}\right\},\hat{\Theta}\right)$ used for the expectation in Eq. 29 has a Gaussian distribution, and its mean and variance are represented using $\hat{\Theta}$ and Eqs. 21 and 22

$$\hat{\tilde{d}}_i^{(l)}=\hat{\tilde{\sigma}}_{d_i}^{2\,(l)}\left(\frac{\mathrm{I}^{(l-1)}\left[\tilde{d}^{(l-1)}\right]_i}{\mathrm{I}^{(l-1)2}\left[\tilde{\sigma}_d^{2(l-1)}\right]_i+\hat{\sigma}_0^2}-\frac{\sum_{\mathrm{N}_i^{(l)}}\left(f_t^{(l)}+\hat{f}^{r(l)}\right)\hat{f}^{u(l)}}{\hat{\sigma}_1^2}\right), \tag{32}$$

$$\hat{\tilde{\sigma}}_{d_i}^{2\,(l)}=\left(\frac{1}{\mathrm{I}^{(l-1)2}\left[\tilde{\sigma}_d^{2(l-1)}\right]_i+\hat{\sigma}_0^2}+\frac{\sum_{\mathrm{N}_i^{(l)}}\hat{f}^{u(l)2}}{\hat{\sigma}_1^2}\right)^{-1}. \tag{33}$$

By taking the expectation of Eq. 31 with respect to $p\left(d_i^{(l)}\middle|\left\{f_t^{(l)}\right\}_{\mathrm{N}_i},\cdots,\left\{f_t^{(1)}\right\},\hat{\Theta}\right)$ using Eqs. 32 and 33, $Q\left(\Theta;\hat{\Theta}\right)$ in Eq. 29 can be concretely derived. At the M-step, $Q\left(\Theta;\hat{\Theta}\right)$ should be maximized with respect to Θ in order to update Θ. Therefore, to update Θ we can minimize the following function, which is obtained by multiplying $Q\left(\Theta;\hat{\Theta}\right)$ by -2 and neglecting the constant value

$$J(\Theta) = N^{(l)} M^{(l)} \ln \sigma_1^2 + \sum_{i=1}^{M^{(l)}} \ln\left(\mathrm{I}^{(l-1)^2} \left[\tilde{\sigma}_d^{2(l-1)} \right]_i + \sigma_0^2 \right) + J_f\left(\mathbf{m}, \sigma_1^2\right) + J_d\left(\sigma_0^2\right) + J_p(\Theta), \tag{34}$$

$$\begin{aligned} & J_f\left(\mathbf{m}, \sigma_1^2\right) \\ & = \frac{1}{\sigma_1^2} \left\{ \sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i^{(l)}} \left(f_t^{(l)} + f^{r(l)} \right)^2 + 2 \sum_{i=1}^{M^{(l)}} \hat{\tilde{d}}_i^{(l)} \sum_{\mathrm{N}_i^{(l)}} f^{u(l)} \left(f_t^{(l)} + f^{r(l)} \right) + \sum_{i=1}^{M^{(l)}} \left(\hat{\tilde{d}}_i^{(l)2} + \hat{\tilde{\sigma}}_{d_i}^{2\,(l)} \right) \sum_{\mathrm{N}_i^{(l)}} f^{u(l)^2} \right\}, \end{aligned} \tag{35}$$

$$J_d\left(\sigma_0^2\right) = \sum_{i=1}^{M^{(l)}} \frac{ \hat{\tilde{d}}_i^{(l)2} + \hat{\tilde{\sigma}}_{d_i}^{2\,(l)} - 2 \hat{\tilde{d}}_i^{(l)} \mathrm{I}^{(l-1)} \left[\tilde{d}^{(l-1)} \right]_i + \left(\mathrm{I}^{(l-1)} \left[\tilde{d}^{(l-1)} \right]_i \right)^2 }{ \mathrm{I}^{(l-1)^2} \left[\tilde{\sigma}_d^{2(l-1)} \right]_i + \sigma_0^2 }, \tag{36}$$

$$J_p(\Theta) = \left(\mathbf{m} - \tilde{\mathbf{m}} \right)^{\mathrm{T}} \tilde{\mathbf{V}}_m^{-1} \left(\mathbf{m} - \tilde{\mathbf{m}} \right) + \frac{\left(\sigma_0^2 - \tilde{\sigma}_0^2 \right)^2}{\tilde{\sigma}_{\sigma 0}^2} + \frac{\left(\sigma_1^2 - \tilde{\sigma}_1^2 \right)^2}{\tilde{\sigma}_{\sigma 1}^2}. \tag{37}$$

Minimization of $J(\Theta)$ in Eq. 34 cannot be done analytically; therefore, numerical search has to be performed. In general, the value completely minimizing an objective function is difficult to be found. In such case, we can use the generalized MAP-EM algorithm in which, at the M-step, parameter updating is done so as to enlarge the value of the Q function more than that for the parameters values obtained at the previous iteration. By the generalized MAP-EM algorithm, the computational cost for each M-step often decreases, although the number of iterations may increase.

After convergence of the above two steps for each layer l, if the initial values of Θ for the iteration are suitable, we can obtain Θ_{MAP}, which maximizes $p\left(\Theta \middle| \left\{f_t^{(l)}\right\}, \cdots, \left\{f_t^{(1)}\right\}\right)$ and coincides with the mean $\tilde{\Theta}$ of this probability because of the Laplace approximation, and $d_{MAP}^{(l)}$, which maximizes $p\left(d^{(l)} \middle| \left\{f_t^{(l)}\right\}_{\mathrm{N}}, \cdots, \left\{f_t^{(1)}\right\}, \tilde{\Theta}\right)$ and also coincides with the mean $\tilde{d}^{(l)}$. The probability which should be actually evaluated corresponds to Eq. 28, and the integration for this marginalization requires a numerical computation or a random sampling technique. Hence, in this study, we justify the solution of the above MAP-EM algorithm by applying the saddle point approximation to $p\left(\Theta \middle| \left\{f_t^{(l)}\right\}, \cdots, \left\{f_t^{(1)}\right\}\right)$, as follows:

$$\begin{aligned} p\left(d^{(l)} \middle| \left\{f_t^{(l)}\right\}_{\mathrm{N}}, \cdots, \left\{f_t^{(1)}\right\}\right) & \cong \int p\left(d^{(l)} \middle| \left\{f_t^{(l)}\right\}_{\mathrm{N}}, \cdots, \left\{f_t^{(1)}\right\}, \Theta\right) \delta\left(\Theta - \hat{\Theta}\right) d\Theta \\ & = p\left(d^{(l)} \middle| \left\{f_t^{(l)}\right\}_{\mathrm{N}}, \cdots, \left\{f_t^{(1)}\right\}, \hat{\Theta}\right) \end{aligned} \tag{38}$$

5.3 Parameter variance estimation using Supplemented EM technique

In order to evaluate Eq. 34, i.e. Eq. 37, for each layer l, the variance-covariance matrix of Θ at the previous layer l -1 is required. However, this matrix cannot be estimated directly by the MAP-EM algorithm. Usually used naive approximate method is evaluating

$\left(-\partial^2 \ln p\left(\Theta\middle|\left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right)/\partial\Theta^2\right)^{-1}$ numerically at Θ_{MAP} of layer l -1. In this study, we aim to compute the variance-covariance matrix by a stable and efficient scheme, and an application of the Supplemented EM (SEM) algorithm (Meng & Rubin, 1991) is examined and proposed. Using the SEM algorithm, the asymptotic variance-covariance of the parameters can be estimated using only the code for the EM algorithm and the code for computing the complete data asymptotic variance-covariance matrix.

Since, in this study, the priors of the parameters should be considered at each layer l, the usage of the MAP-EM algorithm is appropriate. Hence, the S-MAP-EM algorithm, which is the MAP-EM algorithm with the supplemented procedures described below in detail, is actually used instead of the SEM algorithm. Let $\mathbf{V}_{\hat{\Theta}}$ denote the "observed" asymptotic variance-covariance matrix, which is evaluated for the converging value $\hat{\Theta}$ without expectation operation, and it is used as an estimate. The important equation for the S-MAP-EM algorithm is

$$\mathbf{V}_{\hat{\Theta}} = \mathbf{V}_{c\hat{\Theta}} + \delta\mathbf{V}, \tag{39}$$

$$\mathbf{V}_{c\hat{\Theta}} = \left\{ -\frac{\partial^2}{\partial\Theta\partial\Theta^{\mathrm{T}}} \left[\ln p\left(\left\{f_t^{(l)}\right\}, \left\{d^{(l)}\right\} \middle| \left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\},\Theta \right) + \ln p\left(\Theta \middle| \left\{f_t^{(l-1)}\right\},\cdots,\left\{f_t^{(1)}\right\}\right) \right]_{\Theta=\hat{\Theta}} \right\}^{-1}, \tag{40}$$

$$\delta\mathbf{V} = \left\{\mathbf{I} - \mathbf{DM}_{\hat{\Theta}}\right\}^{-1} \mathbf{DM}_{\hat{\Theta}} \mathbf{V}_{c\hat{\Theta}}. \tag{41}$$

In Eq. 41, $\mathbf{I}$ indicates a unit matrix, and $\mathbf{DM}$ is the Jacobian matrix for the mapping $M:\Theta^k \to \Theta^{k+1}$ implicitly defined by the MAP-EM algorithm, where k indicates the iteration number. In the same way as a wide class of problems where probability of the complete-data belongs to the exponential family, $\mathbf{V}_{c\hat{\Theta}}$ can be derived analytically in this study. On the other hand, $\mathbf{DM}_{\hat{\Theta}}$ has to be estimated by a stable computation.

If Θ^k converges to some value $\hat{\Theta}$ and $M(\Theta)$ is continuous, $\hat{\Theta}$ is a fixed point of the iteration and satisfies $\hat{\Theta} = M\left(\hat{\Theta}\right)$. By a Taylor series expansion of $\Theta^{k+1} = M\left(\Theta^k\right)$ at $\hat{\Theta}$, we can have the first approximation

$$\Theta^{k+1} - \hat{\Theta} \approx \mathbf{DM}_{\hat{\Theta}}\left(\Theta^k - \hat{\Theta}\right). \tag{42}$$

From this equation, the $d \times d$ matrix $\mathbf{DM}_{\hat{\Theta}}$ is often referred to as the rate of convergence, where d is the degree of freedom of Θ and in this study $d = 7$. It is convenient to estimate $\mathbf{DM}_{\hat{\Theta}}$ by numerical evaluation using an iteration procedure, for example the following formulation

$$\lim_{k\to\infty} \frac{\Theta_i^{k+1} - \Theta_i^k}{\Theta_i^k - \Theta_i^{k-1}},\ i = 1,\cdots,d. \tag{43}$$

However, this provides only a few eigenvalues of $\mathbf{DM}_{\hat{\Theta}}$, and the matrix itself cannot be computed. On the other hand, Meng and Rubin (1991) explained that each element of $\mathbf{DM}_{\hat{\Theta}}$ is the component-wise rate of convergence of a "forced EM" for the S-MAP-EM algorithm. Let r_{ij} be the (i, j)th element of $\mathbf{DM}_{\hat{\Theta}}$, and we define

$$\Theta_{(j)}^{k} \equiv \left[\hat{\Theta}_1, \cdots, \hat{\Theta}_{j-1}, \Theta_j^k, \hat{\Theta}_{j+1}, \cdots, \hat{\Theta}_d\right]^{\mathrm{T}}, \tag{44}$$

where Θ_j^k is the value of Θ_j on the kth iteration of the MAP-EM algorithm and $\hat{\Theta}_i$ indicates the ith component of the converging value $\hat{\Theta}$. By the definition of r_{ij}, the following equation holds

$$\begin{aligned} r_{ij} &= \frac{\partial M_i\left(\hat{\Theta}\right)}{\partial \Theta_j} \\ &= \lim_{\Theta_j \to \hat{\Theta}_j} \frac{M_i\left(\hat{\Theta}_1, \cdots, \hat{\Theta}_{j-1}, \Theta_j, \hat{\Theta}_{j+1}, \cdots, \hat{\Theta}_d\right) - M_i\left(\hat{\Theta}\right)}{\Theta_j - \hat{\Theta}_j} \\ &= \lim_{k \to \infty} \frac{M_i\left(\Theta_{(j)}^k\right) - \hat{\Theta}_i}{\Theta_j^k - \hat{\Theta}_j} \\ &\equiv \lim_{k \to \infty} r_{ij}^k . \end{aligned} \tag{45}$$

From this, for example, the following procedure can be introduced to obtain $\mathbf{V}_{c\hat{\Theta}}$.

S-MAP-EM algorithm

(i) Run the MAP-EM algorithm in order to obtain $\hat{\Theta}$, and save all the values $\{\Theta^k\}$ at each iteration.

(ii) For each Θ^k, compute $\Theta_{(j)}^k$ using Eq. 44, and run one iteration of the MAP-EM algorithm using $\Theta_{(j)}^k$ as a current estimate to obtain $M_i(\Theta_{(j)}^k)$, and subsequently the ratio r_{ij}^k in Eq. 45 for each i (i = 1, …, d). This is done for each j (j = 1, …, d).

(iii) Check the convergence of r_{ij}^k using $\left|r_{ij}^k - r_{ij}^{k+1}\right| < \xi$ with a certain threshold value ξ, and determine r_{ij} (i, j =1, … , d) and hence, $\mathbf{DM}_{\hat{\Theta}}$.

(iv) Compute $\mathbf{V}_{c\hat{\Theta}}$ by Eq. 40, and then using Eqs. 39 and 41, evaluate the observed asymptotic variance-covariance matrix $\mathbf{V}_{\hat{\Theta}}$.

At step (i) in the above procedure, we assume that the updating sequence of the parameters by the MAP-EM algorithm is stored to save computational time. However, to save extra storage, at step (ii) one-iteration of the MAP-EM algorithm to obtain Θ^k from Θ^{k-1} is done firstly instead of saving all the values of Θ^k. Using this Θ^k, one-iteration of the S-MAP-EM at step (ii) is realized.

Note that $\delta\mathbf{V}$ is a symmetric matrix, and if it seems to be quite asymmetric, there has been a programming error in either MAP-EM or S-MAP-EM, or convergences of both have not been sufficient.

6. Implementation

6.1 Image decomposition into multi-scale images

Ideally, we have to decompose images into multi-scale images using spatio-temporal filtering. In this study, we will examine an algorithm using only two successive frames. Therefore, temporal filter can not be used and hence only spatial filter is discussed.

We assume that input images have 256×256 pixels and the number of resolution layers is 4. For each resolution *l*, the size N_l of a local region N_l, where depth is constant, has to be defined. The resolution *l* and the size N_l can be treated independently, but here, we simply define N_l so that intensity pattern has a slow slope in N_l. The set of N_l values and the corresponding spatial wavelengths, which components are extracted by an ideal band-pass filter, are shown in Table 1.

layer number *l*	N_l [pixels]	spatial wavelength [pixels]
1	32*32	DC – 64 (DC – 4 cycles)
2	16*16	64 – 32 (4 cycles – 8 cycles)
3	8*8	32 – 16 (8 cycles – 16 cycles)
4	8*8	16 – 8 (16 cycles – 32 cycles)

Table 1. Decomposition parameters

6.2 Derivative filter

To get the accurate estimates of the spatial gradients of intensity, the choice of convolution kernels of derivative is important. Directly applying simple first-order differences produces poor estimates, especially in highly textured region.

By assuming no spatial alias occurs in images, the derivative of the continuous sampling function is the best kernel, but the resulting kernel needs to be quite large to estimate high accurate gradients. Hence, a lot of computer vision researchers have used sampled Gaussian derivatives that have better properties than simple differences, but are less computationally expensive than sampling function.

On the other hand, Farid and Simoncelli have proposed a simple design procedure for matched pairs of 1-D kernels, which consists of a interpolator and a differentiator, suitable for gradient estimation (Farid & Simoncelli, 1997). Let $\hat{B}(\mathbf{k})$ be the frequency domain representation of the interpolator, and $\hat{D}(\mathbf{k})$ be that of the differentiator. The kernel pairs determined by their procedure can have the following properties:

1. The derivative filters are good approximations to the derivative of the interpolator. This means that, for a derivative along the x-axis, $jk_x\hat{B}(\mathbf{k}) \approx \hat{D}(\mathbf{k})$ holds, where k_x is the component of the frequency coordinate in the x direction;
2. The interpolator is symmetric with $\hat{B}(\mathbf{0}) = 1$;

3. Both kernels are separable, and hence the design problem is reduced to one dimensional, for computational efficiency and ease of design; and
4. The design algorithm includes a model for signal and noise statistics.

In this research, we use the kernels derived by this procedure, and concretely, the five-tap kernel $[-0.108415, -0.280353, 0, 0.280353, 0.108415]$ is applied to all the resolution layers as a differentiator.

As a temporal derivative, we compute simply the finite difference using two successive frames, hence the temporal derivative may be corrupted by large error.

6.3 Concrete procedures for M-step

At the M-step in the MAP-EM algorithm, we have to minimize Eq. 34 for each iteration. As described in Sec.5.2, the generalized MAP-EM algorithm can be adopted for estimating only the parameters values. However, the variances of the parameters are also required to be estimated, and therefore S-MAP-EM is performed. For the conventional S-MAP-EM algorithm denoted in Sec.5.3, it is assumed that the Q function is maximized at each M-step. If we use the generalized MAP-EM scheme, certain extension of S-MAP-EM has to be achieved. Hence, we will give up applying generalized MAP-EM and completely minimize Eq. 34 at the M-step.

Since Eq. 34 is not a quadratic form with respect to the parameter vector, we will use the steepest descent method to minimize it. We introduce the gradient vector, i.e. the steepest descent direction, of Eq. 34. By defining $\mathbf{m} \equiv [u_x, u_y, u_z, \mathbf{r}^{\mathrm{T}}]^{\mathrm{T}}$, partial derivative with respect to $\mathbf{m}$ can be written as follows:

$$\frac{\partial J(\Theta)}{\partial \mathbf{m}} = \frac{2}{\sigma_1^2}\left(\frac{\partial \mathbf{m}_0}{\partial \mathbf{m}}\right)^{\mathrm{T}}\left(\mathbf{A}\mathbf{m}_0 - \mathbf{b}\right) + 2\tilde{\mathbf{V}}_m^{-1}\left(\mathbf{m} - \tilde{\mathbf{m}}\right). \tag{46}$$

The matrix $\mathbf{A}$ and vector $\mathbf{b}$ in Eq. 46 are shown in Sec. 10.1. Additionally, partial derivatives with respect to σ_0^2 and σ_1^2 can be introduced as follows:

$$\frac{\partial J(\Theta)}{\partial \sigma_0^2} = \sum_{i=1}^{M^{(l)}} \frac{1}{\mathrm{I}^{(l-1)2}\left[\tilde{\sigma}_d^{2(l-1)}\right]_i + \sigma_0^2} - \sum_{i=1}^{M^{(l)}} \frac{\gamma_i}{\left(\mathrm{I}^{(l-1)2}\left[\tilde{\sigma}_d^{2(l-1)}\right]_i + \sigma_0^2\right)^2} + \frac{2\left(\sigma_0^2 - \tilde{\sigma}_0^2\right)}{\tilde{\sigma}_{\sigma 0}^2}, \tag{47}$$

$$\frac{\partial J(\Theta)}{\partial \sigma_1^2} = \frac{N^{(l)}M^{(l)}}{\sigma_1^2} - \frac{\chi(\mathbf{m})}{\sigma_1^4} + \frac{2\left(\sigma_1^2 - \tilde{\sigma}_1^2\right)}{\tilde{\sigma}_{\sigma 1}^2}, \tag{48}$$

where γ_i and $\chi(\mathbf{m})$ are defined as $J_d = \sum \gamma_i / (\mathrm{I}^{(l-1)^2}[\tilde{\sigma}_d^{2(l-1)}]_i + \sigma_0^2)$ and $J_f = \chi(\Theta)/\sigma_1^2$.

By evaluating the values of Eqs. 46, 47 and 48 at the current values of parameters, we can know the steepest descent direction and perform numerical search for the minimization parameters of $J(\Theta)$ defined in Eq. 34.

6.4 Parameter variance with complete data

In order to estimate the variance-covariance matrix of the parameters using the SEM scheme, we have to know the variance-covariance matrix of the complete data $\mathbf{V}_{c\hat{\Theta}}$ in Eqs. 39 and 41. We will compute analytically the 2nd derivative according to Eq. 40. Let $L(\Theta)$ be the conditional log likelihood function in Eq. 40 and $\Theta_0 \equiv [\mathbf{m}_0, \sigma_0^2, \sigma_1^2]^{\mathrm{T}}$, and we can use $\partial L / \partial \Theta \big|_{\Theta=\hat{\Theta}} = \mathbf{0}$, hence the following equation can be derived

$$-\frac{\partial^2 L}{\partial \Theta \partial \Theta^{\mathrm{T}}}\bigg|_{\Theta=\hat{\Theta}} = -\left(\frac{\partial \Theta_0}{\partial \Theta}\right)^{\mathrm{T}}_{\Theta=\hat{\Theta}} \left(\frac{\partial^2 L}{\partial \Theta_0 \partial \Theta_0^{\mathrm{T}}}\right)_{\Theta=\hat{\Theta}} \left(\frac{\partial \Theta_0}{\partial \Theta}\right)_{\Theta=\hat{\Theta}} . \tag{49}$$

In the right hand side of Eq. 49, $\partial \Theta_0 / \partial \Theta$ is shown as follows:

$$\frac{\partial \Theta_0}{\partial \Theta} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -u_x \Big/ \sqrt{1-u_x^2-u_y^2} & -u_y \Big/ \sqrt{1-u_x^2-u_y^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{50}$$

Additionally, $\partial^2 L / \partial \Theta_0 \partial \Theta_0^{\mathrm{T}}$ in Eq. 49 consists of the components shown in Eq. A5. By evaluating Eqs. 50 and A5 at $\Theta = \hat{\Theta}$, we can know $-\partial^2 L / \partial \Theta \partial \Theta^{\mathrm{T}} \big|_{\hat{\Theta}}$ from Eq. 49. On the other hand, log of the prior density $\ln p(\Theta \big| \{f_t^{(l-1)}\}, \cdots, \{f_t^{(1)}\})$ is assumed to be

$$\ln p\left(\Theta \Big| \left\{f_t^{(l)}\right\}, \cdots, \left\{f_t^{(1)}\right\}\right) = -\frac{1}{2}(\mathbf{m}-\tilde{\mathbf{m}})^{\mathrm{T}} \tilde{\mathbf{V}}_m^{-1} (\mathbf{m}-\tilde{\mathbf{m}}) - \frac{\left(\sigma_0^2 - \tilde{\sigma}_0^2\right)^2}{2\tilde{\sigma}_{\sigma 0}^2} - \frac{\left(\sigma_1^2 - \tilde{\sigma}_1^2\right)^2}{2\tilde{\sigma}_{\sigma 1}^2} + \text{Const.}, \tag{51}$$

hence,

$$-\frac{\partial^2}{\partial \Theta \partial \Theta^{\mathrm{T}}} \ln p\left(\Theta \Big| \left\{f_t^{(l)}\right\}, \cdots, \left\{f_t^{(1)}\right\}\right)\bigg|_{\Theta=\hat{\Theta}} = \begin{bmatrix} \tilde{\mathbf{V}}_m^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1/\tilde{\sigma}_{\sigma 0}^2 & 0 \\ \mathbf{0}^{\mathrm{T}} & 0 & 1/\tilde{\sigma}_{\sigma 1}^2 \end{bmatrix} . \tag{52}$$

Using Eqs. 49 and 52, $\mathbf{V}_{c\hat{\Theta}}$ can be computed as follows:

$$\mathbf{V}_{c\hat{\Theta}} = \left\{ -\left(\frac{\partial \Theta_0}{\partial \Theta}\right)^{\mathrm{T}}_{\Theta=\hat{\Theta}} \left(\frac{\partial^2 L}{\partial \Theta_0 \partial \Theta_0^{\mathrm{T}}}\right)_{\Theta=\hat{\Theta}} \left(\frac{\partial \Theta_0}{\partial \Theta}\right)_{\Theta=\hat{\Theta}} + \begin{bmatrix} \tilde{\mathbf{V}}_m^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1/\tilde{\sigma}_{\sigma 0}^2 & 0 \\ \mathbf{0}^{\mathrm{T}} & 0 & 1/\tilde{\sigma}_{\sigma 1}^2 \end{bmatrix} \right\}^{-1} . \tag{53}$$

7. Examples

To confirm the effectiveness of the proposed method, we conducted numerical experiments using artificial images. Figure 3(a) shows the first image, with 256×256 pixels generated by a computer graphics (CG) technique using the depth map shown in Fig. 3(b) and a random texture. The second successive image was generated at a different viewpoint, which was assumed to move with $\mathbf{u} = [0.1, 0.0, 0.0]^{\mathrm{T}}$ and $\mathbf{r} = [0.0, 0.0, 0.0]^{\mathrm{T}}$. In this situation, the theoretically calculated norm of the optical flow between the two successive images was approximately two pixels on average for the whole image. These images were decomposed into four layers with different resolutions in accordance with the method and the parameters described in Sec.6.1. The decomposed images are shown in Fig. 4.

The estimated depth maps are shown in Fig. 5. As mentioned in Sec. 2.2, $\|\mathbf{u}\|$ and $|d|$ can not be uniquely determined, and hence the scale of the depth in Fig. 5 is adjusted so that the value of the estimated $\|\mathbf{u}\|$ can be regarded as a true value. The mesh size in Fig. 5 is denoted by N_l, and for example, $N_1 = 32 \times 32$ pixels. In the experiments, the variances of d and Θ in the priors to $l = 1$ were set to be sufficiently large. The result obtained using all the observed information corresponding to the four layers at the same time without BP is shown in Fig. 6. For this result, the local region size was 8×8 pixels. From these results, we can confirm that stable recovery of the depth map is achieved by the proposed method.

The above results were derived for noise-free images. Therefore, $n_1^{(l)}$ in Eq. 11 corresponds to the 1st approximation error of the gradient equation. We confirmed through experiments that, for the Gaussian image noise with a standard deviation of 5% with respect to the dynamic range of the image intensity, the proposed method has almost the same performance as that shown in Fig. 5. Additionally, we omitted BP of Θ, and found that the root mean square error (RMSE) of the depth is one and a half times larger than that using BP for Θ. This result is due to the estimation bias.

8. Conclusions

We introduced a scheme and an explicit algorithm for stably recovering object shape as a depth map. This scheme is based on the multi-scale Bayesian network and the approximate BP using the EM algorithm. Especially, in this study to estimate the variance-covariance matrix in a stable way, the Supplemented MAP-EM algorithm is applied. The effectiveness and the applicability of the proposed algorithm were shown through numerical examples. In the future, the performance for real image sequences needs to be examined, and a quantitative evaluation of the accuracy is required. Additionally, we are very interested in a temporal expansion of this scheme, and it has been considered in our current work.

9. Acknowledgments

This work has been supported by the research funds from Tokyo Metropolitan University. The authors are thankful to Dr. Todorka Alexandrova for her cooperation.

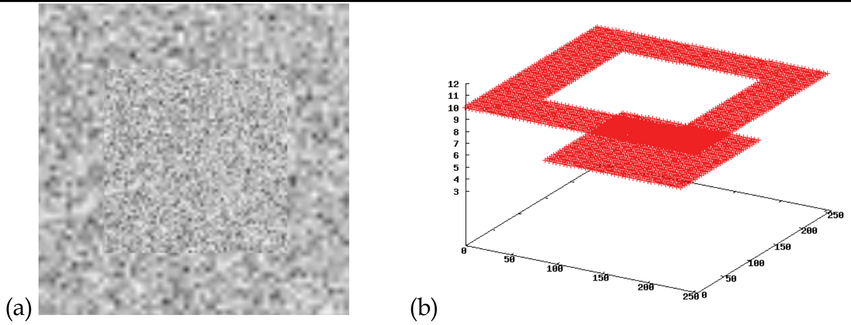


Fig. 3. Data used in the experiments: (a) artificial image; (b) true depth map

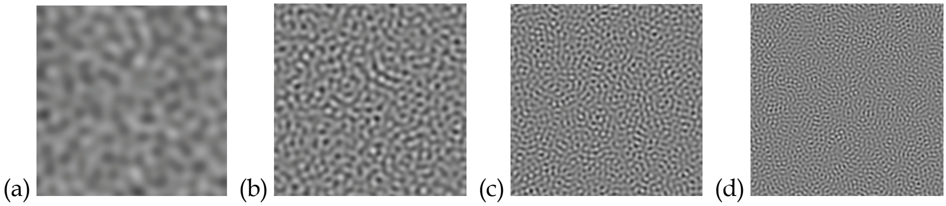


Fig. 4. Decomposed images: (a) l=1; (b) l=2; (c) l=3; (d) l=4

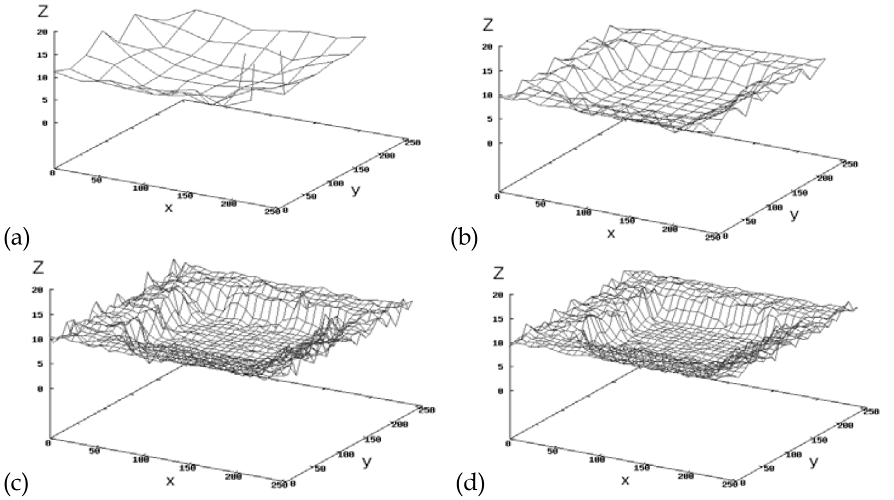


Fig. 5. Estimated depth map with BP; (a) l=1; (b) l=2; (c) l=3; (d) l=4

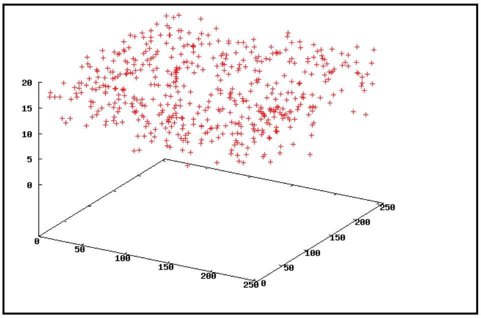


Fig. 6. Estimated depth map without BP

10. References

Adiv, G. (1985). Determinig three-dimensional motion and structure from optical flow generated by several moving objects, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 7, No. 4, pp. 384-401

Brox, T., Bruhn, A., Papenberg, N., Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping, *Proceedings of ECCV*, Vol. 4, pp. 171-177

Bruhn, A. & Weickert, J. (2005). Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods, *Int. J. Comput. Vision*, Vol. 61, No. 3, pp. 211-231

Daniilidis, K. & Nagel, H.H. (1990). Analytical results on error sensitivity of motion estimation from two views, *Image and Vision Computing*, Vol. 8, pp. 297-303

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood from incomplete data, *J. Roy. Statist. Soc. B*, Vol. 39, pp. 1-38

van Dyk, D.A., Meng, X.L., Rubin, D.B. (1995). Maximum likelihood estimation via the ECM algorithm: Computing the asymptotic variance, *Statistica Sinica*, Vol. 5, pp. 55-75

Farid, H. & Simoncelli, E.P. (1997). Optimally rotation-equivariant directional derivative kernels, *Proceedings of Int. Conf. Computer Analysis of Image and Patterns*, pp. 207-214

Farnebäck, G. (2001). Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field, *Proceedings of ICCV*, pp. 171-177

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 6, No. 6, pp. 721-741

Han, M & Kanade, T. (2002). A perspective factorization method for Euclidean reconstruction with uncalibrated cameras, *J. of Visualization and Computer Animation*, Vol. 13, No. 4, pp. 211-223

Huang, T.S. & Faugeras, O.D. (1989). Some properties of the *E* matrix in two-view motion estimation, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 11, No. 12, pp. 1310-1312

Huang, T.S. & Ho, H.T. (1999). A Kalman filter approach to direct depth estimation incorporating surface structure, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 21, No. 6, pp. 570-575

Horn, B.K.P. & Schunk, B. (1981). Determining optical flow, *Artif. Intell.*, Vol. 17, pp. 185-203

Horn, B.K.P. & Weldon Jr, E.J. (1988). Direct methods for recovering motion, *Int. J. Comput. Vision*, Vol. 2, pp. 51-76

Kanatani, K. (1993). *Geometric Computation for Machine Vision,* Oxford University Press, Oxford

Kanatani, K. (1996). *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam

Ke, Q. & Kanade, T. (2005). Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming, *Proceedings of CVPR*, pp. 739-746

Kearney, J.K., Thompson, W.B., Boley, D.L. (1987). Optical flow estimation : An error analysis of gradient-based methods with local optimization, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 9, No. 2, pp. 229-244

Longuet-Higgins, H.C. (1981). A computer algorithm for reconstructing a scene from two projections, *Nature*, Vol. 293, pp. 133-135

Lucas, B.D. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision, *Proceedings of Imaging Understanding Workshop*, pp. 121-130

Maki, A., Watanabe, M., Wiles, C. (2002). Geotensity : Combining motion and lighting for 3D surface reconstruction, *Int. J. Comput. Vision*, Vol. 48, No. 2, pp. 75-90

Matthies, L., Kanade, T., Szeliski, R. (1989). Kalman filter-based algorithm for estimating depth from image sequences, *Int. J. Comput. Vision*, Vol. 3, No. 3, pp. 209-238

Maybank, S. (1990). Ambiguity in reconstruction from image correspondences, *Proceedings of ECCV*, pp. 177-186

Meng, X.L., Rubin, D.B. (1991). Using EM to obtain asymptotic variance-covariance matrix : The SEM algorithm, *J. of the American Statist. Assoc.*, Vol. 86, No. 416, pp.899-909

Nestares, O., Fleet, D.J., Heeger, D.J. (2000). Likelihood functions and confidence bounds for total-least-squares problems, *Proceedings of CVPR*, pp. 760-767

Simoncelli, E.P. (1999). Bayesian multi-scale differential optical flow, In : *Handbook of Computer Vision and Applications,* Jähne, B., Haussecker, H, Geissler, P., (Eds.), pp. 397-422, Academic Press, San Diego

Stein, G.P. & Shashua, A. (1997). Model-based brightness constraints : On direct estimation of structure and motion, *Proceedings of CVPR*, pp. 400-406

Tagawa, N., Toriu, T., Endoh, T. (1993). Un-biased linear algorithm for recovering three-dimensional motion from optical flow, *IEICE Trans. Inf. & Syst.*, Vol. E76-D, No. 10, pp. 1263-1275

Tagawa, N., Toriu, T., Endoh, T. (1994). An objective function for 3-D motion estimation from optical flow with lower error variance than maximum likelihood estimator, *Proceedings of IEEE Int. Conf. on Image Processing*, pp. 252-256

Tagawa, N., Moriya, T. (1995). Computing 2-D motion field with multi-resolution images and cooperation of gradient-based and matching-based schemes, *IEICE Trans. Fundamentals*, Vol. E78-A, No. 6, pp. 685-692

Tagawa, N., Toriu, T., Endoh, T. (1996). 3-D motion estimation from optical flow with low computational cost and small variance, *IEICE Trans. Inf. & Syst.*, Vol. E79-D, No. 3, pp. 230-241

Tagawa, N., Kawaguchi, J., Naganuma, S., Okubo, K. (2008). Direct 3-D shape recovery from image sequence based on multi-scale Bayesian network, *Proceedings of ICPR*, CD

Tsai, R.Y. & Huang, T.S. (1984). Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surface, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 6, No. 6, pp. 13-27

Weiss, Y. & Fleet, D.J. (2001). Velocity likelihoods in biological and machine vision, In : *Probabilistic Models of the Brain : Perception and Neural Function*, Rao, R.P.N., Olshausen, B.A., Lewicki, M.S., Jahne, (Eds.), pp. 81-100, MIT Press, Cambridge

Zhuang, X., Huang, T.S., Ahuja, N., Haralick, R.M. (1988). A simplified linear optical flow-motion algorithm, *Comput. Vision, Graphics, Image Processing*, Vol. 42, pp. 334-344

11. Appendix

11.1 Definitions of matrix A and vector b in Eq. 46

The symmetric matrix **A** and vector **b** in Eq. 46 are explicitly defined as follows:

$$\begin{aligned}
&A_{11} = \sum_{i=1}^{M^{(l)}} \alpha_i \sum_{\mathrm{N}_i} f_x^{(l)2},\ A_{12} = \sum_{i=1}^{M^{(l)}} \alpha_i \sum_{\mathrm{N}_i} f_x^{(l)} f_y^{(l)},\ A_{13} = -\sum_{i=1}^{M^{(l)}} \alpha_i \sum_{\mathrm{N}_i} f_x^{(l)} \left(f_x^{(l)}x + f_y^{(l)}y\right), A_{14} = -\sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} f_x^{(l)} \left\{f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right\}, \\
&A_{15} = \sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} f_x^{(l)} \left\{f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right\},\ A_{16} = -\sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} f_x^{(l)} \left(f_x^{(l)}y - f_y^{(l)}x\right), A_{22} = \sum_{i=1}^{M^{(l)}} \alpha_i \sum_{\mathrm{N}_i} f_y^{(l)2},\ A_{23} = -\sum_{i=1}^{M^{(l)}} \alpha_i \sum_{\mathrm{N}_i} f_y^{(l)} \left(f_x^{(l)}x + f_y^{(l)}y\right), \\
&A_{24} = -\sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} f_y^{(l)} \left\{f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right\}, A_{25} = \sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} f_y^{(l)} \left\{f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right\},\ A_{26} = -\sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} f_y^{(l)} \left(f_x^{(l)}y - f_y^{(l)}x\right), \\
&A_{33} = \sum_{i=1}^{M^{(l)}} \alpha_i \sum_{\mathrm{N}_i} \left(f_x^{(l)}x + f_y^{(l)}y\right)^2,\ A_{34} = \sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} \left(f_x^{(l)}x + f_y^{(l)}y\right)\left\{f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right\},\ A_{35} = -\sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} \left(f_x^{(l)}x + f_y^{(l)}y\right)\left\{f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right\}, \\
&A_{36} = \sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} \left(f_x^{(l)}x + f_y^{(l)}y\right)\left(f_x^{(l)}y - f_y^{(l)}x\right), A_{44} = \sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i} \left\{f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)^2\right\},\ A_{45} = -\sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i} \left\{f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right\}\left\{f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right\}, \\
&A_{46} = \sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i} \left\{f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right\}\left(f_x^{(l)}y - f_y^{(l)}x\right), A_{55} = \sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i} \left\{f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right\}^2,\ A_{56} = -\sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i} \left\{f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right\}\left(f_x^{(l)}y - f_y^{(l)}x\right), \\
&A_{66} = \sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i} \left(f_x^{(l)}y - f_y^{(l)}x\right)^2,
\end{aligned} \tag{A1}$$

$$\begin{aligned}
&b_1 = \sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} f_t^{(l)} f_x^{(l)},\ b_2 = \sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} f_t^{(l)} f_y^{(l)},\ b_3 = -\sum_{i=1}^{M^{(l)}} \beta_i \sum_{\mathrm{N}_i} f_t^{(l)} \left(f_x^{(l)}x + f_y^{(l)}y\right),\ b_4 = -\sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i} f_t^{(l)} \left\{f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right\}, \\
&b_5 = \sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i} f_t^{(l)} \left\{f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right\},\ b_6 = -\sum_{i=1}^{M^{(l)}} \sum_{\mathrm{N}_i} f_t^{(l)} \left(f_x^{(l)}y - f_y^{(l)}x\right),
\end{aligned} \tag{A2}$$

where the coefficients α_i and β_i depending on the depth estimation are also defined

$$\alpha_i = \hat{\tilde{d}}_i^{(l)2} + \hat{\tilde{\sigma}}_{di}^{2\,(l)},\ \beta_i = \hat{\tilde{d}}_i^{(l)}, \tag{A3}$$

and the Jacobian $\partial \mathbf{m}_0 / \partial \mathbf{m}$ is represented as follows:

$$\frac{\partial \mathbf{m}_0}{\partial \mathbf{m}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -u_x \Big/ \sqrt{1-u_x^2-u_y^2} & -u_y \Big/ \sqrt{1-u_x^2-u_y^2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A4}$$

11.2 Definitions of matrix $\partial^2 L / \partial\Theta_0 \partial\Theta_0^{\mathrm{T}}$ in Eq. 53

The components of $\partial^2 L / \partial\Theta_0 \partial\Theta_0^{\mathrm{T}}$ can be written as follows:

$$\frac{\partial^2 L}{\partial u_x^2} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} f_x^{(l)2} \tilde{d}_i^{(l)2}}{\sigma_1^2},\ \frac{\partial^2 L}{\partial u_y^2} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} f_y^{(l)2} \tilde{d}_i^{(l)2}}{\sigma_1^2},\ \frac{\partial^2 L}{\partial u_z^2} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left(f_x^{(l)}x + f_y^{(l)}y\right)^2 \tilde{d}_i^{(l)2}}{\sigma_1^2},$$

$$\frac{\partial^2 L}{\partial u_x \partial u_y} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} f_x^{(l)} f_y^{(l)} \tilde{d}_i^{(l)2}}{\sigma_1^2},\ \frac{\partial^2 L}{\partial u_x \partial u_z} = \frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left(f_x^{(l)}x + f_y^{(l)}y\right) f_x^{(l)} \tilde{d}_i^{(l)2}}{\sigma_1^2},\ \frac{\partial^2 L}{\partial u_y \partial u_z} = \frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left(f_x^{(l)}x + f_y^{(l)}y\right) f_y^{(l)} \tilde{d}_i^{2}}{\sigma_1^2},$$

$$\frac{\partial^2 L}{\partial r_x^2} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right]^2}{\sigma_1^2},\ \frac{\partial^2 L}{\partial r_y^2} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}\left(1+x^2\right)xy + f_y^{(l)}xy\right]^2}{\sigma_1^2},\ \frac{\partial^2 L}{\partial r_z^2} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left(f_x^{(l)}y - f_y^{(l)}x\right)^2}{\sigma_1^2},$$

$$\frac{\partial^2 L}{\partial r_x \partial r_y} = \frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right]\left[f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right]}{\sigma_1^2},\ \frac{\partial^2 L}{\partial r_x \partial r_z} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right]\left(f_x^{(l)}y - f_y^{(l)}x\right)}{\sigma_1^2},$$

$$\frac{\partial^2 L}{\partial r_y \partial r_z} = \frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right]\left(f_x^{(l)}y - f_y^{(l)}x\right)}{\sigma_1^2},\ \frac{\partial^2 L}{\partial u_x \partial r_x} = \frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right] f_x^{(l)} \tilde{d}_i^{(l)}}{\sigma_1^2},$$

$$\frac{\partial^2 L}{\partial u_x \partial r_y} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right] f_x^{(l)} \tilde{d}_i^{(l)}}{\sigma_1^2},\ \frac{\partial^2 L}{\partial u_x \partial r_z} = \frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left(f_x^{(l)}y - f_y^{(l)}x\right) f_x^{(l)} \tilde{d}_i^{(l)}}{\sigma_1^2},$$

$$\frac{\partial^2 L}{\partial u_y \partial r_x} = \frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right] f_y^{(l)} \tilde{d}_i^{(l)}}{\sigma_1^2},\ \frac{\partial^2 L}{\partial u_y \partial r_y} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right] f_y^{(l)} \tilde{d}_i^{(l)}}{\sigma_1^2},$$

$$\frac{\partial^2 L}{\partial u_y \partial r_z} = \frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left(f_x^{(l)}y - f_y^{(l)}x\right) f_y^{(l)} \tilde{d}_i^{(l)}}{\sigma_1^2},\ \frac{\partial^2 L}{\partial u_z \partial r_x} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right]\left(f_x^{(l)}x + f_y^{(l)}y\right) \tilde{d}_i^{(l)}}{\sigma_1^2},$$

$$\frac{\partial^2 L}{\partial u_z \partial r_y} = \frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right]\left(f_x^{(l)}x + f_y^{(l)}y\right) \tilde{d}_i^{(l)}}{\sigma_1^2},\ \frac{\partial^2 L}{\partial u_z \partial r_z} = -\frac{\sum_{M^{(l)}} \sum_{N_i^{(l)}} \left(f_x^{(l)}y - f_y^{(l)}x\right)\left(f_x^{(l)}x + f_y^{(l)}y\right) \tilde{d}_i^{(l)}}{\sigma_1^2},$$

$$\frac{\partial^2 L}{\partial\left(\sigma_0^2\right)^2} = -\sum_{M^{(l)}} \frac{\left(\tilde{d}_i^{(l)} - \mathrm{I}^{(l-1)}\left[\tilde{d}^{(l-1)}\right]_i\right)^2}{\left(\mathrm{I}^{(l-1)2}\left[\tilde{\sigma}_d^{2(l-1)}\right]_i + \sigma_0^2\right)^3} + \frac{1}{2}\sum_{M^{(l)}} \frac{1}{\left(\mathrm{I}^{(l-1)2}\left[\tilde{\sigma}_d^{2(l-1)}\right]_i + \sigma_0^2\right)^2},$$

$$\frac{\partial^2 L}{\partial\left(\sigma_1^2\right)^2} = \frac{M^{(l)}N^{(l)}}{2\left(\sigma_1^2\right)^2} - \frac{1}{\left(\sigma_1^2\right)^3} \sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_t^{(l)} - f_x^{(l)}\tilde{d}_i^{(l)}u_x - f_y^{(l)}\tilde{d}_i^{(l)}u_y + \left(f_x^{(l)}x + f_y^{(l)}y\right)\tilde{d}_i^{(l)}u_z + \left(f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right)r_x - \left(f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right)r_y + \left(f_x^{(l)}y - f_y^{(l)}x\right)r_z\right]^2,$$

$$\frac{\partial^2 L}{\partial\sigma_0^2 \partial\sigma_1^2} = \frac{\partial^2 L}{\partial\sigma_0^2 \partial u_x} = \frac{\partial^2 L}{\partial\sigma_0^2 \partial u_y} = \frac{\partial^2 L}{\partial\sigma_0^2 \partial u_z} = \frac{\partial^2 L}{\partial\sigma_0^2 \partial r_x} = \frac{\partial^2 L}{\partial\sigma_0^2 \partial r_y} = \frac{\partial^2 L}{\partial\sigma_0^2 \partial r_z} = 0,$$

$$\frac{\partial^2 L}{\partial\sigma_1^2 \partial u_x} = -\frac{1}{\left(\sigma_1^2\right)^2} \sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_t^{(l)} - f_x^{(l)}\tilde{d}_i^{(l)}u_x - f_y^{(l)}\tilde{d}_i^{(l)}u_y + \left(f_x^{(l)}x + f_y^{(l)}y\right)\tilde{d}_i^{(l)}u_z + \left(f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right)r_x - \left(f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right)r_y + \left(f_x^{(l)}y - f_y^{(l)}x\right)r_z\right] f_x^{(l)}\tilde{d}_i^{(l)},$$

$$\frac{\partial^2 L}{\partial\sigma_1^2 \partial u_y} = -\frac{1}{\left(\sigma_1^2\right)^2} \sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_t^{(l)} - f_x^{(l)}\tilde{d}_i^{(l)}u_x - f_y^{(l)}\tilde{d}_i^{(l)}u_y + \left(f_x^{(l)}x + f_y^{(l)}y\right)\tilde{d}_i^{(l)}u_z + \left(f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right)r_x - \left(f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right)r_y + \left(f_x^{(l)}y - f_y^{(l)}x\right)r_z\right] f_y^{(l)}\tilde{d}_i^{(l)},$$

$$\frac{\partial^2 L}{\partial\sigma_1^2 \partial u_z} = \frac{1}{\left(\sigma_1^2\right)^2} \sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_t^{(l)} - f_x^{(l)}\tilde{d}_i^{(l)}u_x - f_y^{(l)}\tilde{d}_i^{(l)}u_y + \left(f_x^{(l)}x + f_y^{(l)}y\right)\tilde{d}_i^{(l)}u_z + \left(f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right)r_x - \left(f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right)r_y + \left(f_x^{(l)}y - f_y^{(l)}x\right)r_z\right]\left(f_x^{(l)}x + f_y^{(l)}y\right)\tilde{d}_i^{(l)},$$

$$\frac{\partial^2 L}{\partial\sigma_1^2 \partial r_x} = \frac{1}{\left(\sigma_1^2\right)^2} \sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_t^{(l)} - f_x^{(l)}\tilde{d}_i^{(l)}u_x - f_y^{(l)}\tilde{d}_i^{(l)}u_y + \left(f_x^{(l)}x + f_y^{(l)}y\right)\tilde{d}_i^{(l)}u_z + \left(f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right)r_x - \left(f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right)r_y + \left(f_x^{(l)}y - f_y^{(l)}x\right)r_z\right]\left(f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right),$$

$$\frac{\partial^2 L}{\partial\sigma_1^2 \partial r_y} = -\frac{1}{\left(\sigma_1^2\right)^2} \sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_t^{(l)} - f_x^{(l)}\tilde{d}_i^{(l)}u_x - f_y^{(l)}\tilde{d}_i^{(l)}u_y + \left(f_x^{(l)}x + f_y^{(l)}y\right)\tilde{d}_i^{(l)}u_z + \left(f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right)r_x - \left(f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right)r_y + \left(f_x^{(l)}y - f_y^{(l)}x\right)r_z\right]\left(f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right),$$

$$\frac{\partial^2 L}{\partial\sigma_1^2 \partial r_z} = \frac{1}{\left(\sigma_1^2\right)^2} \sum_{M^{(l)}} \sum_{N_i^{(l)}} \left[f_t^{(l)} - f_x^{(l)}\tilde{d}_i^{(l)}u_x - f_y^{(l)}\tilde{d}_i^{(l)}u_y + \left(f_x^{(l)}x + f_y^{(l)}y\right)\tilde{d}_i^{(l)}u_z + \left(f_x^{(l)}xy + f_y^{(l)}\left(1+y^2\right)\right)r_x - \left(f_x^{(l)}\left(1+x^2\right) + f_y^{(l)}xy\right)r_y + \left(f_x^{(l)}y - f_y^{(l)}x\right)r_z\right]\left(f_x^{(l)}y - f_y^{(l)}x\right),$$

(A5)

A Robust Iterative Multiframe SRR based on Hampel Stochastic Estimation with Hampel-Tikhonov Regularization

Vorapoj Patanavijit
Faculty of Engineering, Assumption University, Bangkok, 10240
Email: Patanavijit@yahoo.com
Thailand

Abstract

Typically, Super Resolution Reconstruction (SRR) is the process by which additional information is incorporated to enhance a noisy low resolution image hence producing a high resolution image. Although many such SRR algorithms have been proposed in the last two decades, almost SRR estimations are based on L1 or L2 statistical norm estimation therefore these SRR algorithms are usually very sensitive to their assumed model of data and noise that limits their utility. Unfortunately, the real noise models that corrupt the measure sequence are unknown; consequently, SRR algorithm using L1 or L2 norm may degrade the image sequence rather than enhance it. This paper proposes a novel SRR algorithm based on the stochastic regularization technique of Bayesian MAP estimation by minimizing a cost function. The Hampel norm is used for measuring the difference between the projected estimate of the high-resolution image and each low resolution image in order to remove outliers in the data. Moreover, Tikhonov regularization and Hampel-Tikhonov regularization are used to remove artifacts from the final answer and improve the rate of convergence. Finally, the efficiency of the proposed algorithm is demonstrated here in the experimental results using the Lena (Standard Image) and the Susie (40th Frame: Standard Sequence) in both subjective and objective measurement. The numbers of experimental results confirm the effectiveness of our method and demonstrate its superiority to other super-resolution algorithms based on L1 and L2 norm for a several noise models (such as noiseless, AWGN, Poisson, Salt & Pepper Noise and Speckle Noise) and several noise power.

1. Introduction

Super Resolution Reconstruction (SRR) traditionally allows the recovery of a high-resolution (HR) image from several low-resolution (LR) images that are noisy, blurred, and down sampled. Thus, SRR have a variety of applications in remote sensing, video frame freezing, medical diagnostics and military information acquisition. Consequently, SRR has emerged as an alternative for producing one or a set of HR images from a sequence of LR images.

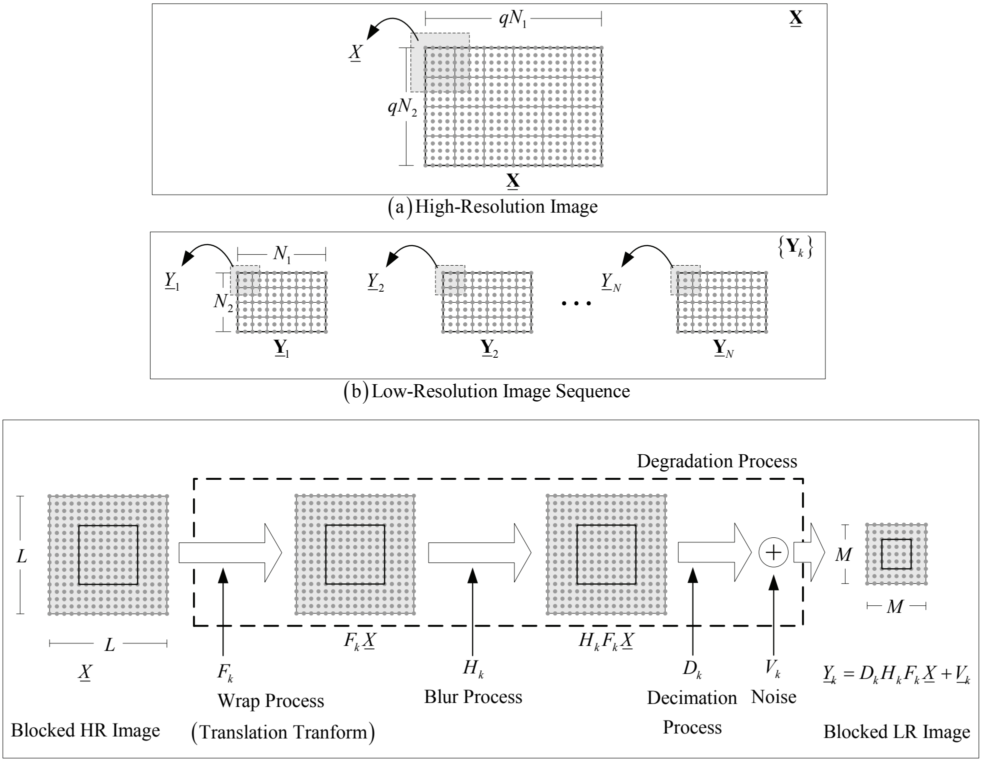
In the section, we will concentrate on the regularized reconstruction point of view therefore the estimation is one of the most important parts of the SRR algorithms and directly affect to the SRR performance. R. R. Schultz et al. (Schultz, R. R. and Stevenson R. L. 1994; Schultz, R. R. and Stevenson R. L. 1996) proposed the SRR algorithm using ML estimator (L2 Norm) with HMRF Regularization in 1996. In 1997, M. Elad et al. (Elad, M. and Feuer, A. 1997) proposed the SRR algorithm using the ML estimator (L2 Norm) with nonellipsoid constraints. Next, M. Elad et al. (Elad, M. and Feuer, A. 1999a; Elad, M. and Feuer, A. 1999c) proposed the SRR algorithm using R-SD and R-LMS (L2 Norm) in 1999. M. Elad et al. (Elad, M. and Hecov Hel-Or, Y. 2001) proposed the fast SRR algorithm ML estimator (L2 Norm) for restoration the warps are pure translations, the blur is space invariant and the same for all the images, and the noise is i.i.d. Gaussian in 2001. A. J. Patti et al. proposed (Patti, A. J. and Altunbasak, Y. 2001) a SRR algorithm using ML (L2 Norm) estimator with POCS-based regularization in 2001 and Y. Altunbasak et al. (Altunbasak, Y., Patti, A. J. and Mersereau, R. M. 2002) proposed a SRR algorithm using ML (L2 Norm) estimator for the MPEG sequences in 2002. D. Rajan et al. (Rajan, D., Chaudhuri, S. and Joshi, M. V. 2003, Rajan, D. and Chaudhuri, S. 2003) proposed SRR using ML (L2 Norm) with MRF regularization to simultaneously estimate the depth map and the focused image of a scene in 2003. S. Farsiu et al. (Farsiu, S., Robinson, M. D., Elad, M., Milanfar, P. 2004; Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P. 2004) proposed SRR algorithm ML estimator (L1 Norm) with BTV Regularization in 2004. Next, they propose a fast SRR of color images (Farsiu, S., Elad, M. and Milanfar, P. 2006) using ML estimator (L1 Norm) with BTV and Tikhonov Regularization in 2006. Y. He et at. (He, Y., Yap, K., Chen, L. and Lap-Pui 2007) proposed SRR algorithm to integrate image registration into SRR estimation (L2 Norm) in 2007. For the data fidelity cost function, all the above SRR methods are based on the simple estimation techniques such as L1 Norm or L2 Norm Minimization. For normally distributed data, the L1 norm produces estimates with higher variance than the optimal L2 (quadratic) norm but the L2 norm is very sensitive to outliers because the influence function increases linearly and without bound. From the robust statistical estimation (Black, M. J. and Rangarajan, A. 1996), Hampel Norm is designed to be more robust than L1 and L2. Hampel norm is designed to be robustness and reject outliers, the norm must be more forgiving about outliers; that is, it should increase less rapidly than L2. This paper proposes a robust iterative SRR algorithm using Hampel norm for the data fidelity cost function with Tikhonov Regularization and Hampel-Tikhonov Regularization. While the former is responsible for robustness and edge preservation, the latter seeks robustness with respect to blur, outliers, and other kinds of errors not explicitly modeled in the fused images. This experimental results demonstrate that our method's performance is superior to what was proposed earlier in this previous reviews.

The organization of this paper is as follows. Section 2 briefly introduces the main concepts of estimation technique in SRR frameworks based on L1 and L2 norm minimization. Section 3 presents the proposed SRR based on Hampel norm minimization with Tikhonov Regularization and Hampel-Tikhonov Regularization. Section 4 outlines the proposed solution and presents the comparative experimental results obtained by using the proposed Hampel norm method and by using the L1 and L2 norm method. Finally, Section 5 provides the summary and conclusion.

2. Introduction of SRR algorithms

For SRR framework (Elad, M. and Feuer, A. 1999b, Elad, M. and Hecov Hel-Or, Y. 2001), Assume that low-resolution frames of images are $\{\mathbf{Y}(t)\}$ as our measured data and each frame contains $N_1 \times N_2$ pixels. A high-resolution frame $\mathbf{X}(t)$ is to be estimated from the N low-resolution images and each frame contains $qN_1 \times qN_2$ pixels, where q is an integer-valued interpolation factor in both the horizontal and vertical directions. To reduce the computational complexity, each frame is separated into overlapping blocks. For convenience of notation, all overlapping blocked frames will be presented as vector, ordered column-wise lexicographically. Namely, the overlapping blocked LR frame is $\underline{Y}_k \in \mathbb{R}^{M^2}$ $(M^2 \times 1)$ and the overlapping blocked HR frame is $\underline{X} \in \mathbb{R}^{q^2M^2}$ $(L^2 \times 1$ or $q^2M^2 \times 1)$. We assume that the two images are related via the following equation

$$\underline{Y}_k = D_k H_k F_k \underline{X} + \underline{V}_k \quad ; k = 1, 2, \ldots, N \tag{1.1}$$



(a) High-Resolution Image

(b) Low-Resolution Image Sequence

(c) The Relation between Overlapping Blocked HR Image
and Overlapping Blocked LR Image Sequence
(SRR Observation Model)

Fig. 1. The Classical SRR Observation Model

where

- $\underline{X}$ (vector format) is the original high-resolution blocked image.

- $\underline{Y}_k(t)$ (vector format) is the blurred, decimated, down sampled and noisy blocked image
- F_k ($F \in \mathbb{R}^{q^2M^2 \times q^2M^2}$ and matrix format) stands for the geometric warp (Typically, Translational Motion) between the images $\underline{X}$ and $\underline{Y}_k$.
- H_k ($H_k \in \mathbb{R}^{q^2M^2 \times q^2M^2}$ and matrix format)is the blur matrix which is a space and time invariant.
- D_k ($D_k \in \mathbb{R}^{M^2 \times q^2M^2}$ and matrix format) is the decimation matrix assumed constant.
- $\underline{V}_k$ ($\underline{V}_k \in \mathbb{R}^{M^2}$ and vector format) is a system noise.

A popular family of estimators is the ML-type estimators (M estimators) (Elad, M. and Feuer, A. 1999c). We rewrite the definition of these estimators in the super resolution reconstruction framework as the following minimization problem:

$$\hat{\underline{X}} = \underset{\underline{X}}{\text{ArgMin}} \left\{ \sum_{k=1}^{N} \rho\left(D_k H_k F_k \underline{X} - \underline{Y}_k \right) \right\} \tag{1.2}$$

where $\rho(\cdot)$ is a norm estimation. To minimize (1.2), the intensity at each pixel of the expected image must be close to those of the original image.

SRR (Super-Resolution Reconstruction) is an ill-posed problem (Elad, M. and Feuer, A. 1997; Elad, M. and Feuer, A. 1999a; Elad, M. and Feuer, A. 1999b; Elad, M. and Hecov Hel-Or, Y. 2001; Elad, M. and Feuer, A. 1999c). For the under-determined cases (i.e., when fewer than required frames are available), there exist an infinite number of solutions which satisfy (1.2). The solution for squared and over-determined cases is not stable, which means small amounts of noise in measurements will result in large perturbations in the final solution. Therefore, considering regularization in SRR algorithm as a mean for picking a stable solution is very useful, if not necessary. Also, regularization can help the algorithm to remove artifacts from the final answer and improve the rate of convergence. A regularization term compensates the missing measurement information with some general prior information about the desirable HR solution, and is usually implemented as a penalty factor in the generalized minimization cost function. Unfortunately, certain types of regularization cost functions work efficiently for some special types of images but are not suitable for general images.

2.1 L1 Norm with Tikhonov Regularization

A popular family of estimators is the L1 Norm estimators that are used in SRR problem (Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P. 2004; Farsiu, S., Elad, M. and Milanfar, P. 2006). Due to ill-posed problem of SRR, a regularization term compensates the missing measurement information with some general prior information about the desirable HR solution, and is usually implemented as a penalty factor in the generalized minimization cost function. The most classical and simplest Tikhonov regularization cost functions is the

Laplacian regularization (Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P. 2004) therefore we rewrite the definition of these estimators in the SRR context as the following minimization problem:

$$\underline{X} = \underset{\underline{X}}{\text{ArgMin}} \left\{ \sum_{k=1}^{N} \left\| D_k H_k F_k \underline{X} - \underline{Y}_k \right\| + \lambda \cdot \left(\Gamma \underline{X} \right)^2 \right\} \tag{2}$$

where the Laplacian kernel (Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P. 2004) is defined as

$$\Gamma = 1/8 \left[1 \quad 1 \quad 1 \quad ; \quad 1 \quad -8 \quad 1 \quad ; \quad 1 \quad 1 \quad 1 \right] \tag{3}$$

By the steepest descent method, the solution is:

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n + \beta \cdot \left\{ \begin{array}{l} \left(\sum_{k=1}^{N} F_k^T H_k^T D_k^T \, \text{sign} \left(D_k H_k F_k \underline{\hat{X}}_n - \underline{Y}_k \right) \right) \\ - \left(\lambda \cdot \left(\Gamma^T \Gamma \right) \underline{\hat{X}}_n \right) \end{array} \right\} \tag{4}$$

where β is the step size in the gradient direction.

2.2 L2 Norm with Tikhonov Regularization

Another popular family of estimators is the L2 Norm estimators that are used in SRR problem (Schultz, R. R. and Stevenson R. L. 1994; Schultz, R. R. and Stevenson R. L. 1997). We rewrite the definition of these estimators in the SRR context that is combined the Laplacian regularization as the following minimization problem:

$$\underline{X} = \underset{\underline{X}}{\text{ArgMin}} \left\{ \sum_{k=1}^{N} \left\| D_k H_k F_k \underline{X} - \underline{Y}_k \right\|_2^2 + \lambda \cdot \left(\Gamma \underline{X} \right)^2 \right\} \tag{5}$$

By the steepest descent method, the solution is:

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n + \beta \cdot \left\{ \begin{array}{l} \sum_{k=1}^{N} F_k^T H_k^T D_k^T \left(\underline{Y}_k - D_k H_k F_k \underline{\hat{X}}_n \right) \\ - \left(\lambda \cdot \left(\Gamma^T \Gamma \right) \underline{\hat{X}}_n \right) \end{array} \right\} \tag{6}$$

3. The Proposed Robust SRR Algorithm

The success of SRR algorithm is highly dependent on the accuracy of the imaging process model. Unfortunately, these models are not supposed to be exactly true, as they are merely mathematically convenient formulations of some general prior information. When the data or noise model assumptions do not faithfully describe the measure data, the estimator performance degrades. Furthermore, existence of outliers defined as data points with different distributional characteristics than the assumed model will produce erroneous estimates. Almost all noise models used in SRR algorithms are based on Additive White Gaussian Noise (AWGN) model; therefore, SRR algorithms can effectively apply only on the image sequence that is corrupted by AWGN. Due to this noise model, L1 norm or L2 norm error are effectively used in SRR algorithm. Unfortunately, the real noise models that corrupt the measure sequence are unknown therefore SRR algorithm using L1 norm or L2 norm may degrade the image sequence rather than enhance it. The robust norm error is necessary for SRR algorithm applicable to several noise models. For normally distributed data, the L1 norm produces estimates with higher variance than the optimal L2 (quadratic) norm but the L2 norm is very sensitive to outliers because the influence function increases linearly and without bound. From the robust statistical estimation (Black, M. J. and Rangarajan, A. 1996), Hampel Norm is designed to be more robust than L1 and L2. While these robust norms are designed to reject outliers, these norms must be more forgiving about the remaining outliers; that is, it should increase less rapidly than L2.

A robust estimation is estimated technique that is resistance to such outliers. In SRR framework, outliers are measured images or corrupted images that are highly inconsistent with the high resolution original image. Outliers may arise from several reasons such as procedural measurement error, noise or inaccurate mathematical model. Outliers should be investigated carefully; therefore, we need to analyze the outlier in a way which minimizes their effect on the estimated model. L2 norm estimation is highly susceptible to even a small number of discordant observations or outliers. For L2 norm estimation, the influence of the outlier is much larger than the other measured data because L2 norm estimation weights the error quadraticly. Consequently, the robustness of L2 norm estimation is poor.

Hampel's norm (Black, M. J. and Rangarajan, A. 1996) is one of error norm from the robust statistic literature. It is equivalent to the L1 norm for large value. But, for normally distributed data, the L1 norm produces estimates with higher variance than the optimal L2 (quadratic) norm, so Hampel's norm is designed to be quadratic for small values and its influence does not descend all the way to zero. The Hampel norm function ($\rho(\cdot)$) and its influence function ($\rho'(\cdot)$) are shown in Figure 2.1 (a) and Figure 2.1 (b), respectively

We rewrite the definition of these estimators in the super resolution context as the following minimization problem:

$$\underline{X} = \underset{\underline{X}}{\text{ArgMin}} \left\{ \sum_{k=1}^{N} f_{HAMPEL}\left(D_k H_k F_k \underline{X} - \underline{Y}_k \right) \right\} \tag{7}$$

By the steepest descent method, the solution is:

$$f_{HAMPEL}(x) = \begin{cases} x^2 & ;|x| \le T \\ 2T|x| - T^2 & ;T < |x| \le 2T \\ 4T^2 - \left(3T - |x|\right)^2 & ;2T < |x| \le 3T \\ 4T^2 & ;|x| > 3T \end{cases} \tag{8}$$

where T is Hampel constant parameter.

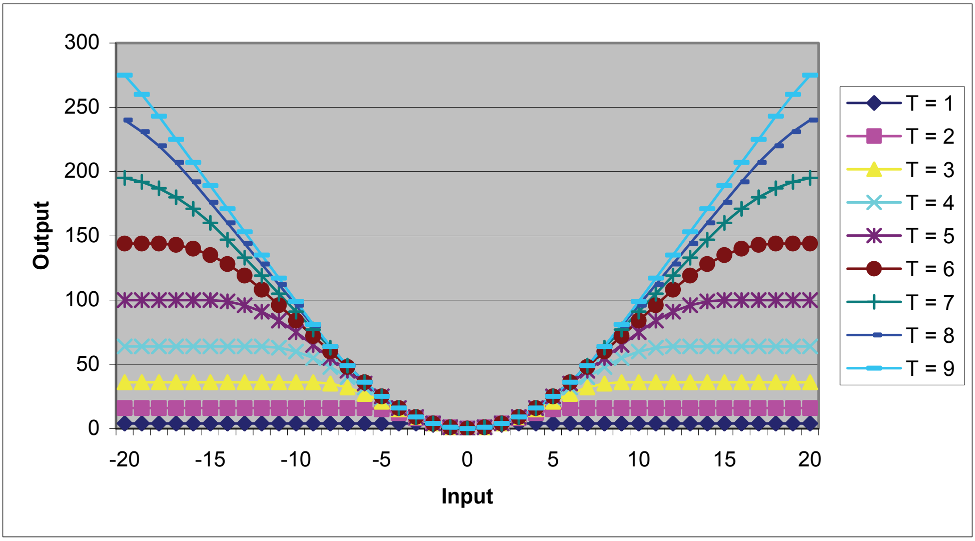


Fig. 2(a). The Hampel Norm function

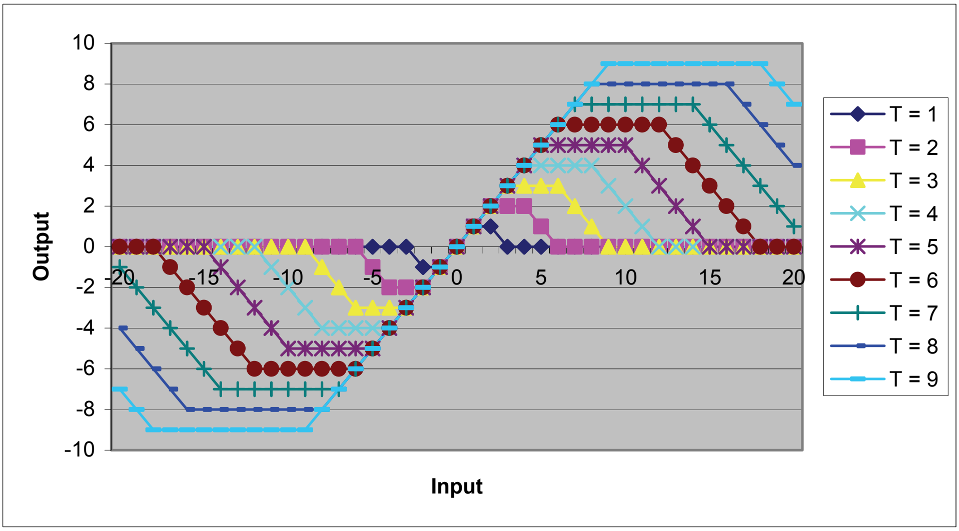


Fig. 2(b). The Influence function of Hampel Norm

3.1 Hampel Norm with Tikhonov Regularization

The most classical and simplest Tikhonov regularization cost functions is the Laplacian regularization (Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P. 2004) therefore we rewrite the definition of these estimators in the SRR context as the following minimization problem:

$$\underline{X} = \underset{\underline{X}}{\operatorname{ArgMin}} \left\{ \sum_{k=1}^{N} f_{HAMPEL} \left(D_k H_k F_k \underline{X} - \underline{Y}_k \right) + \lambda \cdot \left(\Gamma \underline{X} \right)^2 \right\} \tag{9}$$

By the steepest descent method, the solution is:

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n + \beta \cdot \left\{ \begin{array}{l} \sum_{k=1}^{N} F_k^T H_k^T D_k^T \cdot \psi_{HAMPEL} \left(\underline{Y}_k - D_k H_k F_k \underline{\hat{X}}_n \right) \\ -\left(\lambda \cdot \left(\Gamma^T \Gamma \right) \underline{\hat{X}}_n \right) \end{array} \right\} \tag{10}$$

$$\psi_{HAMPEL}(x) = f'_{HAMPEL}(x) = \begin{cases} 2x & ; |x| \le T \\ 2T \operatorname{sign}(x) & ; T < |x| \le 2T \\ 2\left(3T - |x|\right) \operatorname{sign}(x) & ; 2T < |x| \le 3T \\ 0 & ; |x| > 3T \end{cases} \tag{11}$$

3.2 Hampel Norm with Hampel-Tikhonov Regularization

This paper proposes an alternative robust regularization function, so called Hampel-Tikhonov regularization, for incorporating in the SRR algorithm. Consequently, we rewrite the definition of these estimators in the SRR context combining with the Hampel-Laplacian regularization as the following minimization problem:

$$\underline{X} = \underset{\underline{X}}{\operatorname{ArgMin}} \left\{ \sum_{k=1}^{N} f_{HAMPEL} \left(D_k H_k F_k \underline{X} - \underline{Y}_k \right) + \lambda \cdot g_{HAMPEL} \left(\Gamma \underline{X} \right) \right\} \tag{12}$$

$$g_{HAMPEL}(x) = \begin{cases} x^2 & ; |x| \le T_g \\ 2T_g |x| - T_g{}^2 & ; T_g < |x| \le 2T_g \\ 4T_g{}^2 - \left(3T_g - |x|\right)^2 & ; 2T_g < |x| \le 3T_g \\ 4T_g{}^2 & ; |x| > 3T_g \end{cases} \tag{13}$$

By the steepest descent method, the solution is:

$$\underline{\hat{X}}_{n+1} = \underline{\hat{X}}_n + \beta \cdot \left\{ \begin{array}{l} \sum_{k=1}^{N} F_k^T H_k^T D_k^T \cdot \psi_{HAMPEL} \left(\underline{Y}_k - D_k H_k F_k \underline{\hat{X}}_n \right) \\ -\left(\lambda \cdot \Gamma^T \cdot \zeta_{HAMPEL} \left(\Gamma \underline{\hat{X}}_n \right) \right) \end{array} \right\} \tag{14}$$

$$\zeta_{HAMPEL}(x) = g'_{HAMPEL}(x) = \begin{cases} 2x & ; |x| \le T_g \\ 2T_g \operatorname{sign}(x) & ; T_g < |x| \le 2T_g \\ 2\left(3T_g - |x|\right) \operatorname{sign}(x) & ; 2T_g < |x| \le 3T_g \\ 0 & ; |x| > 3T_g \end{cases} \tag{15}$$

4. Experimental Result

This section presents the experiments and results obtained by the proposed robust SRR methods using Hampel norm with Tikhonov regularization and with Hampel- Tikhonov regularization that are calculated by (10-11) and (14-15) respectively. To demonstrate the proposed robust SRR performance, the results of L1 norm SRR (Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P. 2004; Farsiu, S., Elad, M. and Milanfar, P. 2006) with Laplacian regularization that is calculated by (4) and the results of L2 norm SRR (Schultz, R. R. and Stevenson R. L. 1994; Schultz, R. R. and Stevenson R. L. 1997) with Laplacian regularization that is calculated by (6) are presented in order to compare the performance.

These experiments are implemented in MATLAB and the block size is fixed at 8x8 (16x16 for overlapping block). In this experiment, we create a sequence of LR frames by using the Lena (Standard Image) and Susie (40th Frame: Standard Sequence). First, we shifted this HR image by a pixel in the vertical direction. Then, to simulate the effect of camera PSF, this shifted image was convolved with a symmetric Gaussian low-pass filter of size 3x3 with standard deviation equal to one. The resulting image was subsampled by the factor of 2 in each direction. The same approach with different motion vectors (shifts) in vertical and horizontal directions was used to produce 4 LR images from the original scene. We added difference noise model to the resulting LR frames. Next, we use 4 LR frames to generate the high resolution image by the different SRR methods.

The criterion for parameter selection in this paper was to choose parameters which produce both most visually appealing results and highest PSNR. Therefore, to ensure fairness, each experiment was repeated several times with different parameters and the best result of each experiment was chosen (Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P. 2004; Farsiu, S., Elad, M. and Milanfar, P. 2006).

For objective or PSNR measurement of the Lena (Standard image) and Susie (40th Frame) are shown in Table I and Table II respectively. For subjective or virtual measurement of the Lena (Standard image) and Susie (40th Frame) are shown in figure 3 and figure 4 respectively.

4.1 Noiseless

For objective or PSNR measurement, the result of the Lena (Standard image) and Susie (40th Frame) are shown in Table I and II respectively. The result of SRR based on Hampel estimator with Laplacian and Hampel-Laplacian Regularization gives outstandingly higher PSNR than L1 and L2 norm estimator about 1-3 dB.

For subjective or virtual measurement of Lena (Standard image), the original HR image is shown in Fig. 3 (a-1) and one of corrupted LR images is shown in Fig. 3 (a-2). Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 3 (a-3) – 3 (a-6) respectively.

For subjective or virtual measurement of Susie (40th Frame), the original HR image is shown in Fig. 4 (a-1) and one of corrupted LR images is shown in Fig. 4 (a-2). Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 4 (a-3) – 4 (a-6) respectively.

Noise Model	The PSNR of SRR Image (dB)				
	LR Image	L1 with Reg	L2 with Reg	Hamp. with Reg	Hamp. with H-Reg
Noiseless	28.8634	28.8634	30.8553	31.4877 T=19	31.4877 T=19 Tg=19
AWGN (dB):					
SNR=25	27.8884	27.949	29.6579	29.7453 T=19	29.7453 T=19 Tg=5
SNR=22.5	27.2417	27.4918	29.1611	29.1916 T=19	29.1923 T=19 Tg=9
SNR=20	26.2188	26.7854	28.6024	28.6089 T=19	28.6095 T=19 Tg=15
SNR=17.5	24.9598	26.0348	27.8153	27.8186 T=19	27.8186 T=19 Tg=19
SNR=15	23.3549	25.1488	26.6406	26.6117 T=19	26.6117 T=19 Tg=19
Poisson	26.5116	26.9604	28.719	28.713 T=19	28.7142 T=19 Tg=9
Salt&Pepper:					
D=0.005	26.8577	27.1149	28.8495	30.9745 T=9	30.9745 T=9 Tg=5
D=0.010	25.2677	26.0569	28.0346	30.9721 T=9	30.9721 T=15 Tg=19
D=0.015	24.219	25.3534	27.3188	30.9652 T=9	30.9652 T=15 Tg=19
Speckle:					
V=0.03	23.5294	25.3133	26.6956	26.1051 T=19	26.1051 T=19 Tg=19
V=0.05	21.7994	24.4215	25.3165	25.2729 T=1	25.3542 T=1 Tg=19

Table 1. The experimental Result of Proposed Method (Lena Image)

4.2 AWGN (Additive White Gaussian Noise)

For objective or PSNR measurement, the result of the Lena (Standard image) and Susie (40th Frame) are shown in Table I and II respectively. For the Lena image, the result of SRR based on Hampel estimator with Laplacian and Hampel-Laplacian Regularization gives the higher PSNR than L1 and L2 norm estimator. For the Susie image, the result of SRR based on Hampel estimator with Laplacian and Hampel-Laplacian Regularization and L2 estimator gives the higher PSNR than L1 norm estimator.

For subjective or virtual measurement of Lena (Standard image) at 5 AWGN cases, the original HR image is shown in Fig. 3 (b-1) - 3 (f-1) respectively and one of corrupted LR images is shown in Fig. 3 (b-2) - 3 (f-2) respectively. Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 3 (b-3) – 3 (b-6), Figs. 3(c-3) – 3 (c-6), Figs. 3 (d-3) – 3 (d-6), Figs. 3 (e-3) – 3 (e-6) and Figs. 3 (f-3) – 3 (f-6) respectively.

For subjective or virtual measurement of Susie (40th Frame) at 3 AWGN cases, the original HR image is shown in Fig. 4 (b-1) - 4 (f-1) respectively and one of corrupted LR images is shown in Fig. 4 (b-2) - 4 (f-2) respectively. Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 4 (b-3) – 4 (b-6), Figs. 4 (c-3) – 4 (c-6), Figs. 4 (d-3) – 4 (d-6), Figs. 4 (e-3) – 4 (e-6) and Figs. 4 (f-3) – 4 (f-6) respectively.

Noise Model	The PSNR of SRR Image (dB)				
	LR Image	L1 with Reg	**L2 with Reg**	**Hamp. with Reg**	**Hamp. with H-Reg**
Noiseless	32.1687	32.1687	34.2	34.747 T=19	34.747 T=19
AWGN (dB):					
SNR=20	27.5316	28.7003	30.6898	30.6642 T=19	30.6655 T=19 Tg=19
SNR=17.5	25.7322	27.5771	29.3375	29.3112 T=19	29.3112 T=19 Tg=19
SNR=15	23.7086	26.2641	27.6671	27.6565 T=1	27.6565 T=1 Tg=19
Poisson	27.9071	28.9197	30.7634	30.7853 T=19	30.7859 T=19 Tg=19
Salt&Pepper:					
D=0.005	29.0649	29.5041	31.5021	34.4785 T=9	34.4785 T=9 Tg=19

D=0.010	26.4446	27.7593	29.8395	34.4803 T=9	34.4803 T=9 Tg=19
D=0.015	25.276	26.9247	28.7614	34.4483 T=9	34.4483 T=9 Tg=19
Speckle:					
V=0.01	27.6166	28.8289	30.6139	30.415 T=19	30.4293 T=19 Tg=9
V=0.03	24.0403	26.8165	27.7654	27.9409 T=1	28.0189 T=1 Tg=9

Table 2. The experimental Result of Proposed Method (Susie 40th Frame)

4.3 Poisson Noise

For objective or PSNR measurement, the result of the Lena (Standard image) and Susie (40th Frame) are shown in Table I and II respectively. The result of SRR based on Hampel estimator with Laplacian and Hampel-Laplacian Regularization and L2 estimator gives the higher PSNR than L1 norm estimator.

For subjective or virtual measurement of Lena (Standard image), the original HR image is shown in Fig. 3 (g-1) and one of corrupted LR images is shown in Fig. 3 (g-2). Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 3 (g-3) – 3 (g-6) respectively.

For subjective or virtual measurement of Susie (40th Frame), the original HR image is shown in Fig. 4 (g-1) and one of corrupted LR images is shown in Fig. 4 (g-2). Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 4 (g-3) – 4 (g-6) respectively

4.4 Salt&Pepper Noise

For objective or PSNR measurement, this experiment is a 3 Salt&Pepper Noise cases at D=0.005, D=0.010 and D=0.015 respectively (D is the noise density for Salt&Pepper noise model). The result of the Lena (Standard image) and Susie (40th Frame) are shown in Table I and II respectively. The result of SRR based on Hampel estimator with Laplacian and Hampel-Laplacian Regularization gives dramatically higher PSNR than L1 and L2 norm estimator about 4-5 dB.

For subjective or virtual measurement of Lena (Standard image) at 3 Salt&Pepper Noise cases, the original HR image is shown in Fig. 3 (h-1) - 3 (j-1) respectively and one of corrupted LR images is shown in Fig. 3 (h-2) - 3 (j-2) respectively. Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 3 (h-3) – 3 (h-6), Figs. 3 (i-3) – 3 (i-6) and Figs. 3 (j-3) – 3 (j-6) respectively.

For subjective or virtual measurement of Susie (40th Frame) at 3 Salt&Pepper Noise cases, the original HR image is shown in Fig. 4 (h-1) - 4 (j-1) respectively and one of corrupted LR images is shown in Fig. 4 (h-2) - 4 (j-2) respectively. Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 4 (h-3) – 4 (h-6), Figs. 4 (i-3) – 4 (i-6) and Figs. 4 (j-3) – 4 (j-6) respectively.

4.5 Speckle Noise

For objective or PSNR measurement, the result of the Lena (Standard image) and Susie (40th Frame) are shown in Table I and II respectively. (V is the noise variance for Speckle noise model) The result of SRR based on Hampel estimator with Laplacian and Hampel-Laplacian Regularization and L2 estimator gives the higher PSNR than L1 norm estimator.

For subjective or virtual measurement of Lena (Standard image) at 2 Speckle cases, the original HR image is shown in Fig. 3 (k-1) - 3 (l-1) respectively and one of corrupted LR images is shown in Fig. 3 (k-2) - 3 (l-2) respectively. Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 3 (k-3) – 3 (k-6) and Figs. 3 (l-3) – 3 (l-6) respectively.

For subjective or virtual measurement of Susie (40th Frame) at 2 Speckle cases, the original HR image is shown in Fig. 4 (k-1) - 4 (m-1) respectively and one of corrupted LR images is shown in Fig. 4 (k-2) - 4 (m-2) respectively. Next, the result of implementing the SRR algorithm using L1 estimator with Laplacian Regularization, L2 estimator with Laplacian Regularization, Hampel estimator with Laplacian Regularization and Hampel estimator with Hampel-Laplacian Regularization are shown in Figs. 4 (k-3) – 4 (k-6), Figs. 4 (l-3) – 4 (l-6) and Figs. 4 (m-3) – 4 (m-6) respectively.

From the number of experimental results, the T parameter is low (like L1 norm) such as T=1 to T=5 for high noise power and is high for low noise power (like L2 norm) such as T=15 to T=19. Moreover, the T_g parameter is medium (like L1-Tikhonov regularization) for high noise power and is high for low noise power (like classical Tikhonov regularization).

The computation cost of the proposed algorithm slightly higher than the SRR algorithm based on L1 and L2.

From all experimental results of both Susie (40th Frame) and Lena (The Standard Image), all comparatively experimental results are concluded as follow:

1. For AWGN case, the L2 estimator usually gives the best reconstruction because noise distribution is a quadratic similar to L2.
2. For Salt&Pepper Noise cases, the Hampel estimator gives the far better reconstruction than L1 and L2 estimator because these robust estimators are designed to be robust and reject outliers. The norms are more forgiving on outliers; that is, they should increase less rapidly than L2.
3. The SRR algorithm using L1 norm with the proposed registration gives the lowest PSRN because the L1 norm is excessively robust against the outliers.

5. Conclusion

In this paper, we propose an alternate approach using a novel robust estimation norm function (based on Hampel norm function) for SRR framework with Tikhonov and Hampel-Tikhonov Regularization. The proposed robust SRR can be effectively applied on the images that are corrupted by various noise models. Experimental results conducted clearly that the proposed robust algorithm can well be applied on the any noise models (such as Noiseless, AWGN, Poisson Noise, Salt&Pepper Noise and Speckle Noise) at different noise power and the proposed algorithm can obviously improve the result in using both subjective and objective measurement.

6. Acknowledgement

This research work is a partial part of "VIDEO ENHANCEMENT USING AN ITERATIVE SRR BASED ON A ROBUST STOCHASTIC ESTIMATION WITH AN IMPROVED OBSERVATION MODEL" that has been supported by Research Grant for New Scholar (MRG5180263) from TRF (Thai Research Fund) and CHE (Commission on Higher Education) under Assumption University (Thailand).

7. References

Altunbasak, Y.; Patti, A.J.; and Mersereau, R.M. 2002. Super-resolution still and video reconstruction from MPEG-coded video. IEEE Trans. on Circuits and Systems for Video Technology 12(4): 217-26.

Black, M. J. and Rangarajan, A. (1996). On The Unification Of Line Processes, Outlier Rejection and Robust Statistics with Applications in Early Vision, International Journal of Computer Vision 19, 1 (July 1996): 57-92.

Elad, M. and Feuer, A. (1997). Restoration of a Single Superresolution Image from Several Blurred, Noisy and Undersampled Measured Images, IEEE Trans. on IP., 1997.

Elad, M. and Feuer, A. (1999a), Superresolution Restoration of an Image Sequence: Adaptive Filtering Approach, IEEE Trans. on IP., 1999.

Elad, M. and Feuer, A. (1999b). Super-Resolution Reconstruction of Image Sequences, IEEE Trans. on PAMI., vol. 21, Sep. 1999.

Elad, M. and Feuer, A. (1999c). Super-Resolution Restoration of Continuous Image Sequence – Adaptive Filtering Approach, Technical Report, The Technion, The Electrical Engineering Faculty, Israel Institute of Technology, Haifa, pp. 1-12.

Elad, M. and Hecov Hel-Or, Y. (2001). A Fast Super-Resolution Reconstruction Algorithm for Pure Translational Motion and Common Space-Invariant Blur, IEEE Trans. on IP., 2001.

Farsiu, S., Robinson, M. D., Elad, M., Milanfar, P. (2004), Advances and Challenges in Super-Resolution, Wiley Periodicals, Inc., 2004

Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P. (2004). Fast and Robust Multiframe Super Resolution, IEEE Transactions on Image Processing, Oct. 2004. pp. 1327-1344.

Farsiu, S., Elad, M. and Milanfar, P. (2006), Multiframe Demosaicing and Super-Resolution of Color Images, IEEE Transactions on Image Processing, Vol. 15, Jan. 2006, pp. 141-159.

He, Y., Yap, K., Chen, L. and Lap-Pui (2007). A Nonlinear Least Square Technique for Simultaneous Image Registration and Super-Resolution, IEEE Trans. on IP., Nov. 2007.

Kang, M. G. and Chaudhuri, S. (2003). Super-Resolution Image Reconstruction, IEEE Signal Processing Magazine, May. 2003.

Ng, M. K. and Bose, N. K. (2003), Mathematical analysis of super-resolution methodology, IEEE Signal Processing Mag., 2003.

Park, S. C., Park, M. K. and Kang, M. G. (2003). Super-Resolution Image Reconstruction : A Technical Overview, IEEE Signal Processing Magazine, Vol. 20, May 2003, pp 21-36.

Patti, A. J. and Altunbasak, Y. (2001). Artifact Reduction for Set Theoretic Super Resolution Image Reconstruction with Edge Constraints and Higher-Order Interpolation, *IEEE Trans. on Image Processing*, Jan. 2001

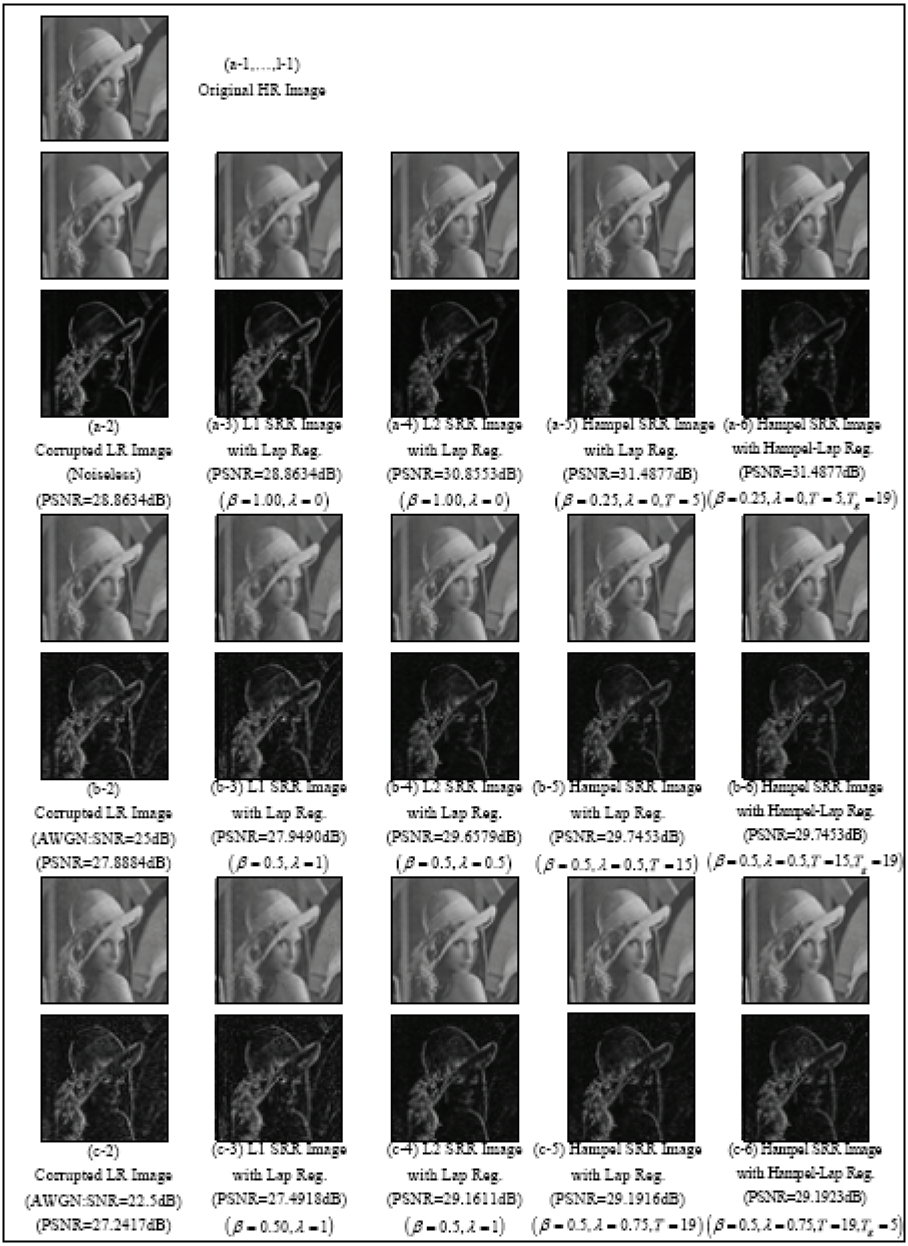
Rajan, D., Chaudhuri, S. and Joshi, M. V. (2003), Multi-objective super resolution concepts and examples, IEEE Signal Processing Magazine, Vol. 20, Issue 3, May. 2003, pp. 49-61

Rajan, D. and Chaudhuri, S. (2003), Simultaneous Estimation of Super-Resolution Scene and Depth Map from Low Resolution Defocuses Observations, IEEE Trans. PAMI., Sep. 2003

Schultz, R. R. and Stevenson R. L. (1994). A Bayesian Approach to Image Expansion for Improved Definition, IEEE Transactions on Image Processing, vol. 3, no. 3, May 1994, pp. 233-242.

Schultz, R. R. and Stevenson R. L. (1996). Extraction of High-Resolution Frames from Video Sequences, IEEE Transactions on Image Processing, vol. 5, no. 6, June 1996, pp. 996-1011.

Vorapoj Patanavijit received the B.Eng., M.Eng. and Ph.D. degrees from the Department of Electrical Engineering at the Chulalongkorn University, Bangkok, Thailand, in 1994, 1997 and 2007 respectively.
He has served as a full-time lecturer at Department of Computer and Network Engineering, Faculty of Engineering, Assumption University since 1998. He works in the field of signal processing and multidimensional signal processing, specializing, in particular, on Image/Video Reconstruction, SRR (Super-Resolution Reconstruction), Enhancement, Fusion, Denoising, Inverse Problems, Motion Estimation and Registration.



(a-1,…,l-1)
Original HR Image

(a-2)
Corrupted LR Image
(Noiseless)
(PSNR=28.8634dB)

(a-3) L1 SRR Image
with Lap Reg.
(PSNR=28.8634dB)
$(\beta = 1.00, \lambda = 0)$

(a-4) L2 SRR Image
with Lap Reg.
(PSNR=30.8553dB)
$(\beta = 1.00, \lambda = 0)$

(a-5) Hampel SRR Image
with Lap Reg.
(PSNR=31.4877dB)
$(\beta = 0.25, \lambda = 0, T = 5)$

(a-6) Hampel SRR Image
with Hampel-Lap Reg.
(PSNR=31.4877dB)
$(\beta = 0.25, \lambda = 0, T = 5, T_g = 19)$

(b-2)
Corrupted LR Image
(AWGN:SNR=25dB)
(PSNR=27.8884dB)

(b-3) L1 SRR Image
with Lap Reg.
(PSNR=27.9490dB)
$(\beta = 0.5, \lambda = 1)$

(b-4) L2 SRR Image
with Lap Reg.
(PSNR=29.6579dB)
$(\beta = 0.5, \lambda = 0.5)$

(b-5) Hampel SRR Image
with Lap Reg.
(PSNR=29.7453dB)
$(\beta = 0.5, \lambda = 0.5, T = 15)$

(b-6) Hampel SRR Image
with Hampel-Lap Reg.
(PSNR=29.7453dB)
$(\beta = 0.5, \lambda = 0.5, T = 15, T_g = 19)$

(c-2)
Corrupted LR Image
(AWGN:SNR=22.5dB)
(PSNR=27.2417dB)

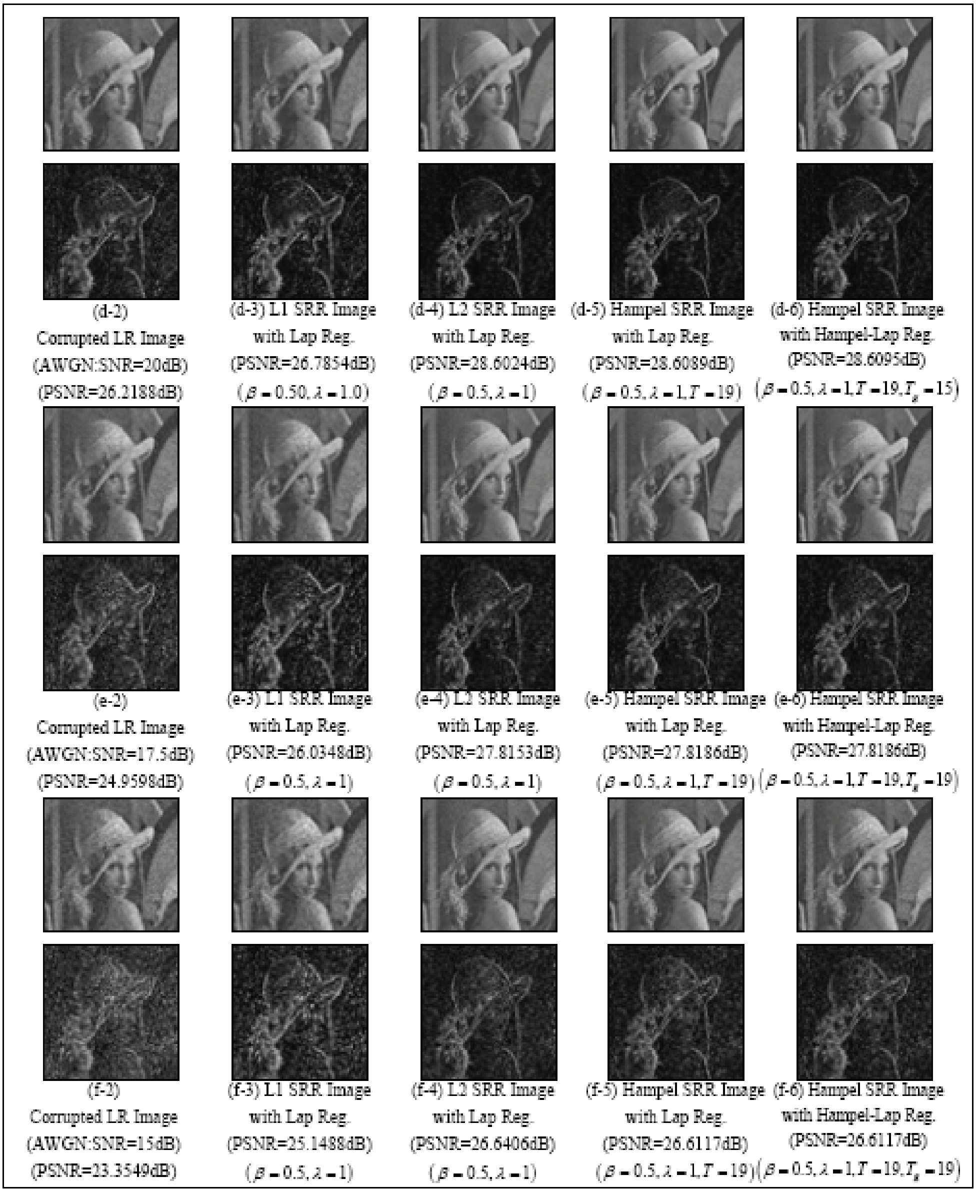
(c-3) L1 SRR Image
with Lap Reg.
(PSNR=27.4918dB)
$(\beta = 0.50, \lambda = 1)$

(c-4) L2 SRR Image
with Lap Reg.
(PSNR=29.1611dB)
$(\beta = 0.5, \lambda = 1)$

(c-5) Hampel SRR Image
with Lap Reg.
(PSNR=29.1916dB)
$(\beta = 0.5, \lambda = 0.75, T = 19)$

(c-6) Hampel SRR Image
with Hampel-Lap Reg.
(PSNR=29.1923dB)
$(\beta = 0.5, \lambda = 0.75, T = 19, T_g = 5)$

Fig. 3. The Experimental Result of Proposed SRR Algorithm: Lena
(The bottom image on our experiment result of each subfigure is the absolute difference between it's correspond top image to the original HR image. The difference is magnified by 5.)



(d-2) Corrupted LR Image (AWGN:SNR=20dB) (PSNR=26.2188dB)

(d-3) L1 SRR Image with Lap Reg. (PSNR=26.7854dB) $(\beta = 0.50, \lambda = 1.0)$

(d-4) L2 SRR Image with Lap Reg. (PSNR=28.6024dB) $(\beta = 0.5, \lambda = 1)$

(d-5) Hampel SRR Image with Lap Reg. (PSNR=28.6089dB) $(\beta = 0.5, \lambda = 1, T = 19)$

(d-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=28.6095dB) $(\beta = 0.5, \lambda = 1, T = 19, T_g = 15)$

(e-2) Corrupted LR Image (AWGN:SNR=17.5dB) (PSNR=24.9598dB)

(e-3) L1 SRR Image with Lap Reg. (PSNR=26.0348dB) $(\beta = 0.5, \lambda = 1)$

(e-4) L2 SRR Image with Lap Reg. (PSNR=27.8153dB) $(\beta = 0.5, \lambda = 1)$

(e-5) Hampel SRR Image with Lap Reg. (PSNR=27.8186dB) $(\beta = 0.5, \lambda = 1, T = 19)$

(e-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=27.8186dB) $(\beta = 0.5, \lambda = 1, T = 19, T_g = 19)$

(f-2) Corrupted LR Image (AWGN:SNR=15dB) (PSNR=23.3549dB)

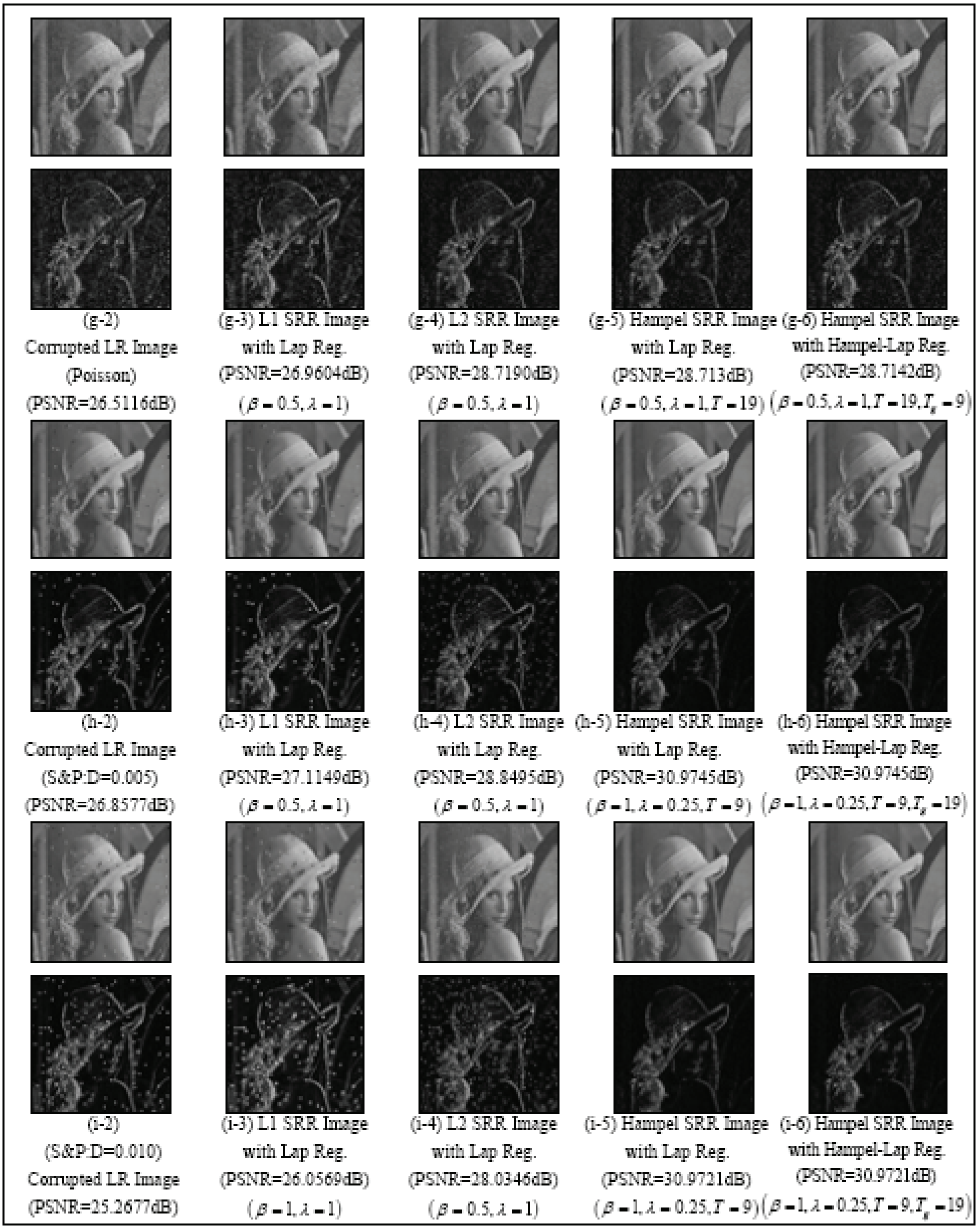
(f-3) L1 SRR Image with Lap Reg. (PSNR=25.1488dB) $(\beta = 0.5, \lambda = 1)$

(f-4) L2 SRR Image with Lap Reg. (PSNR=26.6406dB) $(\beta = 0.5, \lambda = 1)$

(f-5) Hampel SRR Image with Lap Reg. (PSNR=26.6117dB) $(\beta = 0.5, \lambda = 1, T = 19)$

(f-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=26.6117dB) $(\beta = 0.5, \lambda = 1, T = 19, T_g = 19)$

Fig. 3. The Experimental Result of Proposed SRR Algorithm: Lena (Cont.)
(The bottom image on our experiment result of each subfigure is the absolute difference between it's correspond top image to the original HR image. The difference is magnified by 5.)



(g-2) Corrupted LR Image (Poisson) (PSNR=26.5116dB)

(g-3) L1 SRR Image with Lap Reg. (PSNR=26.9604dB) $(\beta = 0.5, \lambda = 1)$

(g-4) L2 SRR Image with Lap Reg. (PSNR=28.7190dB) $(\beta = 0.5, \lambda = 1)$

(g-5) Hampel SRR Image with Lap Reg. (PSNR=28.713dB) $(\beta = 0.5, \lambda = 1, T = 19)$

(g-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=28.7142dB) $(\beta = 0.5, \lambda = 1, T = 19, T_g = 9)$

(h-2) Corrupted LR Image (S&P:D=0.005) (PSNR=26.8577dB)

(h-3) L1 SRR Image with Lap Reg. (PSNR=27.1149dB) $(\beta = 0.5, \lambda = 1)$

(h-4) L2 SRR Image with Lap Reg. (PSNR=28.8495dB) $(\beta = 0.5, \lambda = 1)$

(h-5) Hampel SRR Image with Lap Reg. (PSNR=30.9745dB) $(\beta = 1, \lambda = 0.25, T = 9)$

(h-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=30.9745dB) $(\beta = 1, \lambda = 0.25, T = 9, T_g = 19)$

(i-2) (S&P:D=0.010) Corrupted LR Image (PSNR=25.2677dB)

(i-3) L1 SRR Image with Lap Reg. (PSNR=26.0569dB) $(\beta = 1, \lambda = 1)$

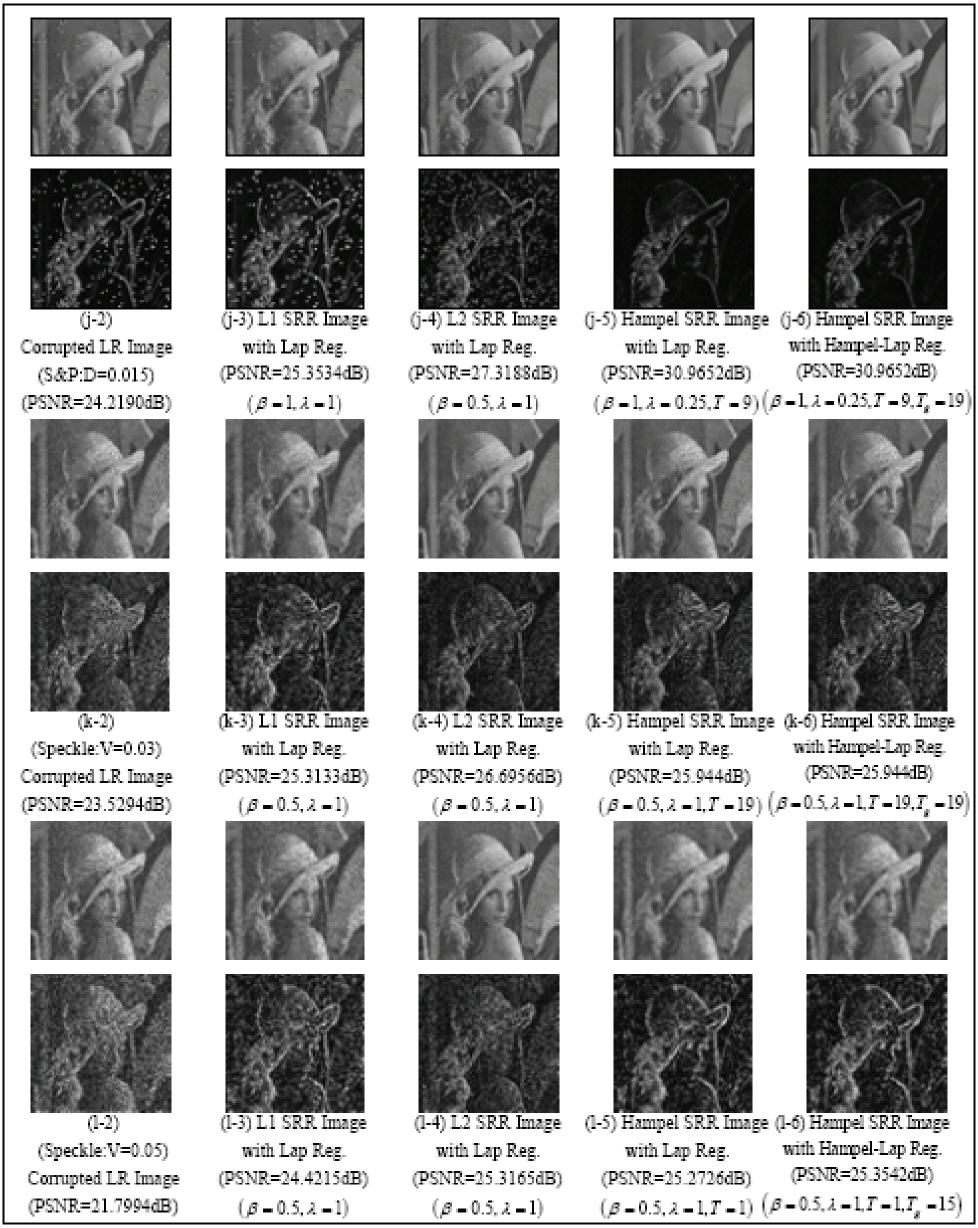
(i-4) L2 SRR Image with Lap Reg. (PSNR=28.0346dB) $(\beta = 0.5, \lambda = 1)$

(i-5) Hampel SRR Image with Lap Reg. (PSNR=30.9721dB) $(\beta = 1, \lambda = 0.25, T = 9)$

(i-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=30.9721dB) $(\beta = 1, \lambda = 0.25, T = 9, T_g = 19)$

Fig. 3. The Experimental Result of Proposed SRR Algorithm: Lena (Cont.)

(The bottom image on our experiment result of each subfigure is the absolute difference between it's correspond top image to the original HR image. The difference is magnified by 5.)



(j-2)
Corrupted LR Image
(S&P:D=0.015)
(PSNR=24.2190dB)

(j-3) L1 SRR Image
with Lap Reg.
(PSNR=25.3534dB)
$(\beta = 1, \lambda = 1)$

(j-4) L2 SRR Image
with Lap Reg.
(PSNR=27.3188dB)
$(\beta = 0.5, \lambda = 1)$

(j-5) Hampel SRR Image
with Lap Reg.
(PSNR=30.9652dB)
$(\beta = 1, \lambda = 0.25, T = 9)$

(j-6) Hampel SRR Image
with Hampel-Lap Reg.
(PSNR=30.9652dB)
$(\beta = 1, \lambda = 0.25, T = 9, T_g = 19)$

(k-2)
(Speckle:V=0.03)
Corrupted LR Image
(PSNR=23.5294dB)

(k-3) L1 SRR Image
with Lap Reg.
(PSNR=25.3133dB)
$(\beta = 0.5, \lambda = 1)$

(k-4) L2 SRR Image
with Lap Reg.
(PSNR=26.6956dB)
$(\beta = 0.5, \lambda = 1)$

(k-5) Hampel SRR Image
with Lap Reg.
(PSNR=25.944dB)
$(\beta = 0.5, \lambda = 1, T = 19)$

(k-6) Hampel SRR Image
with Hampel-Lap Reg.
(PSNR=25.944dB)
$(\beta = 0.5, \lambda = 1, T = 19, T_g = 19)$

(l-2)
(Speckle:V=0.05)
Corrupted LR Image
(PSNR=21.7994dB)

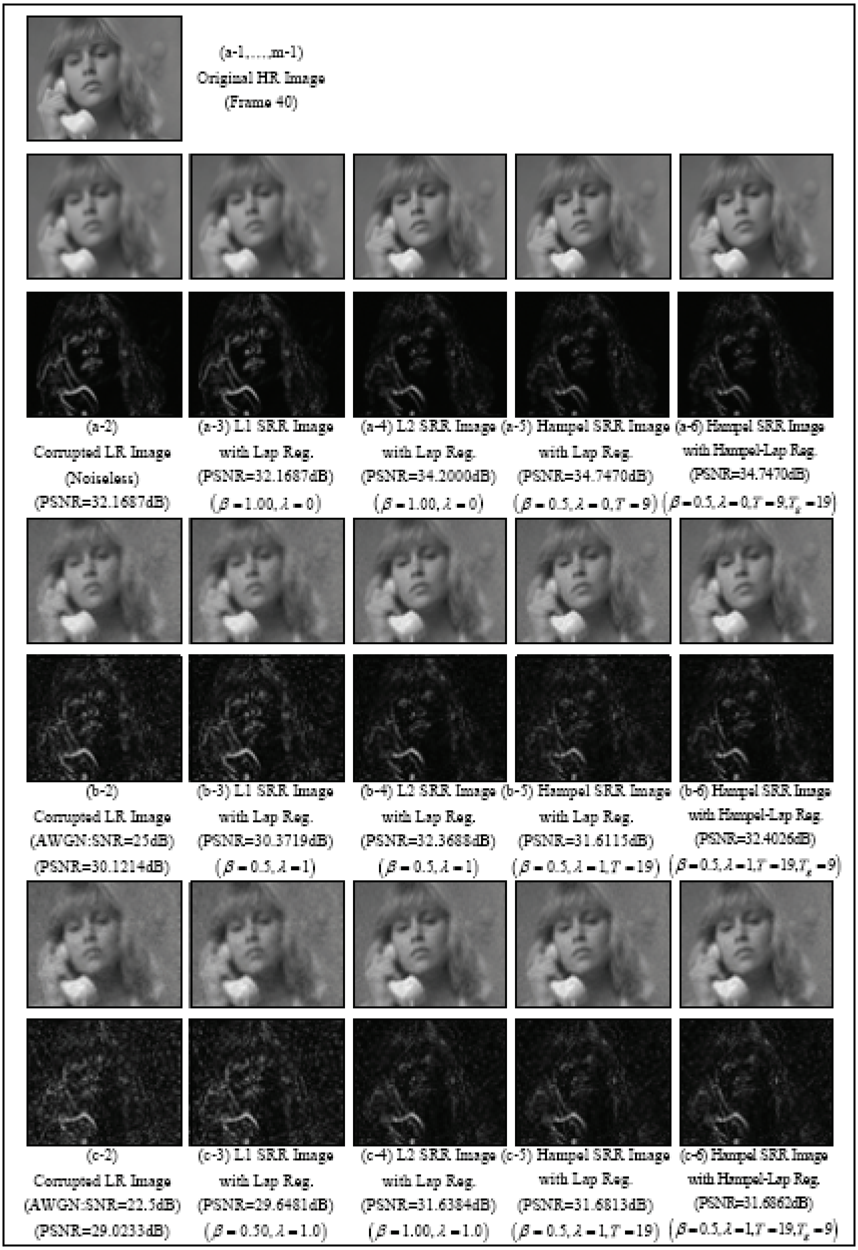
(l-3) L1 SRR Image
with Lap Reg.
(PSNR=24.4215dB)
$(\beta = 0.5, \lambda = 1)$

(l-4) L2 SRR Image
with Lap Reg.
(PSNR=25.3165dB)
$(\beta = 0.5, \lambda = 1)$

(l-5) Hampel SRR Image
with Lap Reg.
(PSNR=25.2726dB)
$(\beta = 0.5, \lambda = 1, T = 1)$

(l-6) Hampel SRR Image
with Hampel-Lap Reg.
(PSNR=25.3542dB)
$(\beta = 0.5, \lambda = 1, T = 1, T_g = 15)$

Fig. 3. The Experimental Result of Proposed SRR Algorithm: Lena (Cont.)
(The bottom image on our experiment result of each subfigure is the absolute difference between it's correspond top image to the original HR image. The difference is magnified by 5.)



(a-1,…,m-1)
Original HR Image
(Frame 40)

(a-2)
Corrupted LR Image
(Noiseless)
(PSNR=32.1687dB)

(a-3) L1 SRR Image
with Lap Reg.
(PSNR=32.1687dB)
$(\beta = 1.00, \lambda = 0)$

(a-4) L2 SRR Image
with Lap Reg.
(PSNR=34.2000dB)
$(\beta = 1.00, \lambda = 0)$

(a-5) Hampel SRR Image
with Lap Reg.
(PSNR=34.7470dB)
$(\beta = 0.5, \lambda = 0, T = 9)$

(a-6) Hampel SRR Image
with Hampel-Lap Reg.
(PSNR=34.7470dB)
$(\beta = 0.5, \lambda = 0, T = 9, T_g = 19)$

(b-2)
Corrupted LR Image
(AWGN:SNR=25dB)
(PSNR=30.1214dB)

(b-3) L1 SRR Image
with Lap Reg.
(PSNR=30.3719dB)
$(\beta = 0.5, \lambda = 1)$

(b-4) L2 SRR Image
with Lap Reg.
(PSNR=32.3688dB)
$(\beta = 0.5, \lambda = 1)$

(b-5) Hampel SRR Image
with Lap Reg.
(PSNR=31.6115dB)
$(\beta = 0.5, \lambda = 1, T = 19)$

(b-6) Hampel SRR Image
with Hampel-Lap Reg.
(PSNR=32.4026dB)
$(\beta = 0.5, \lambda = 1, T = 19, T_g = 9)$

(c-2)
Corrupted LR Image
(AWGN:SNR=22.5dB)
(PSNR=29.0233dB)

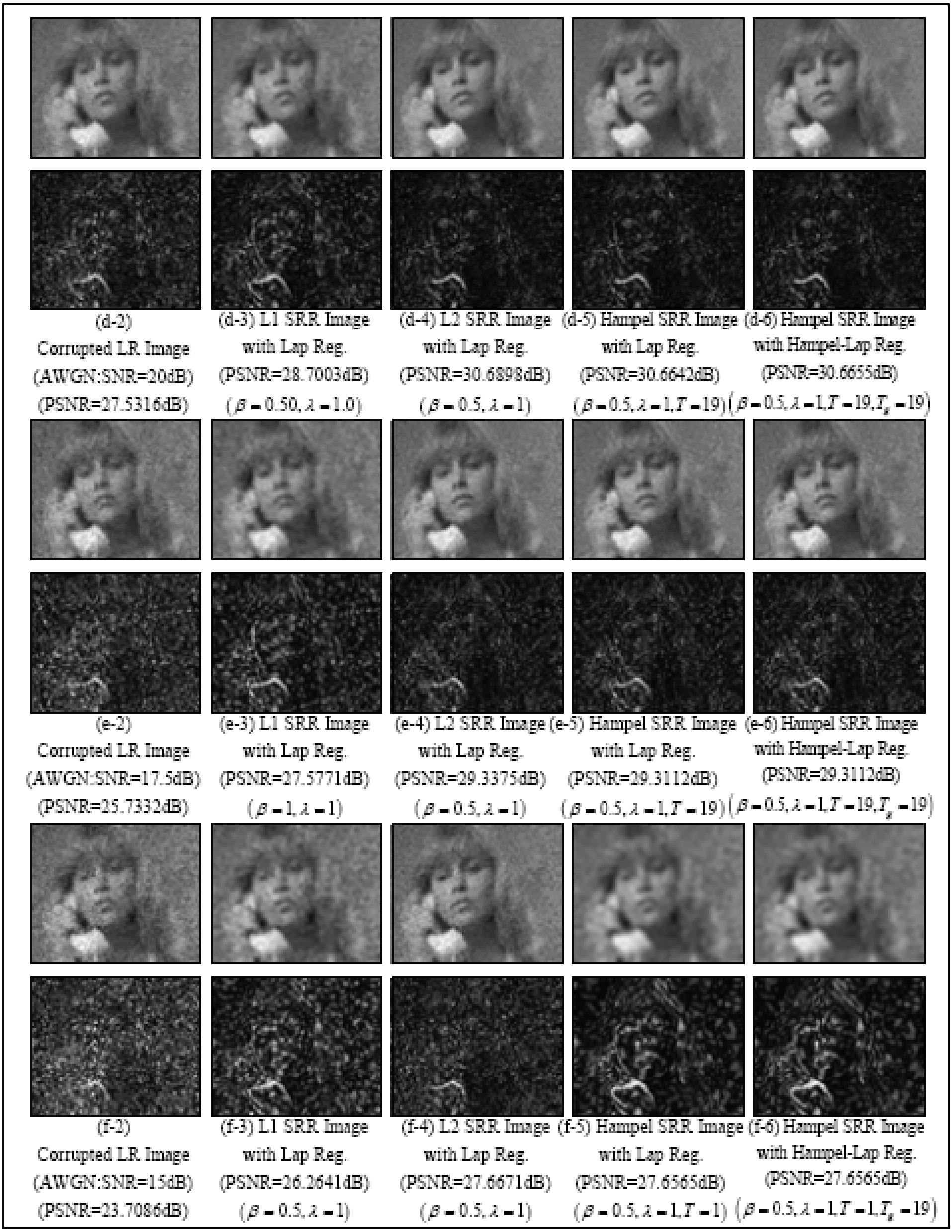
(c-3) L1 SRR Image
with Lap Reg.
(PSNR=29.6481dB)
$(\beta = 0.50, \lambda = 1.0)$

(c-4) L2 SRR Image
with Lap Reg.
(PSNR=31.6384dB)
$(\beta = 1.00, \lambda = 1.0)$

(c-5) Hampel SRR Image
with Lap Reg.
(PSNR=31.6813dB)
$(\beta = 0.5, \lambda = 1, T = 19)$

(c-6) Hampel SRR Image
with Hampel-Lap Reg.
(PSNR=31.6862dB)
$(\beta = 0.5, \lambda = 1, T = 19, T_g = 9)$

Fig. 4. The Experimental Result of Proposed SRR Algorithm: Susie (40th Frame)
(The bottom image on our experiment result of each subfigure is the absolute difference between it's correspond top image to the original HR image. The difference is magnified by 5.)



(d-2) Corrupted LR Image (AWGN:SNR=20dB) (PSNR=27.5316dB)

(d-3) L1 SRR Image with Lap Reg. (PSNR=28.7003dB) $(\beta = 0.50, \lambda = 1.0)$

(d-4) L2 SRR Image with Lap Reg. (PSNR=30.6898dB) $(\beta = 0.5, \lambda = 1)$

(d-5) Hampel SRR Image with Lap Reg. (PSNR=30.6642dB) $(\beta = 0.5, \lambda = 1, T = 19)$

(d-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=30.6655dB) $(\beta = 0.5, \lambda = 1, T = 19, T_g = 19)$

(e-2) Corrupted LR Image (AWGN:SNR=17.5dB) (PSNR=25.7332dB)

(e-3) L1 SRR Image with Lap Reg. (PSNR=27.5771dB) $(\beta = 1, \lambda = 1)$

(e-4) L2 SRR Image with Lap Reg. (PSNR=29.3375dB) $(\beta = 0.5, \lambda = 1)$

(e-5) Hampel SRR Image with Lap Reg. (PSNR=29.3112dB) $(\beta = 0.5, \lambda = 1, T = 19)$

(e-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=29.3112dB) $(\beta = 0.5, \lambda = 1, T = 19, T_g = 19)$

(f-2) Corrupted LR Image (AWGN:SNR=15dB) (PSNR=23.7086dB)

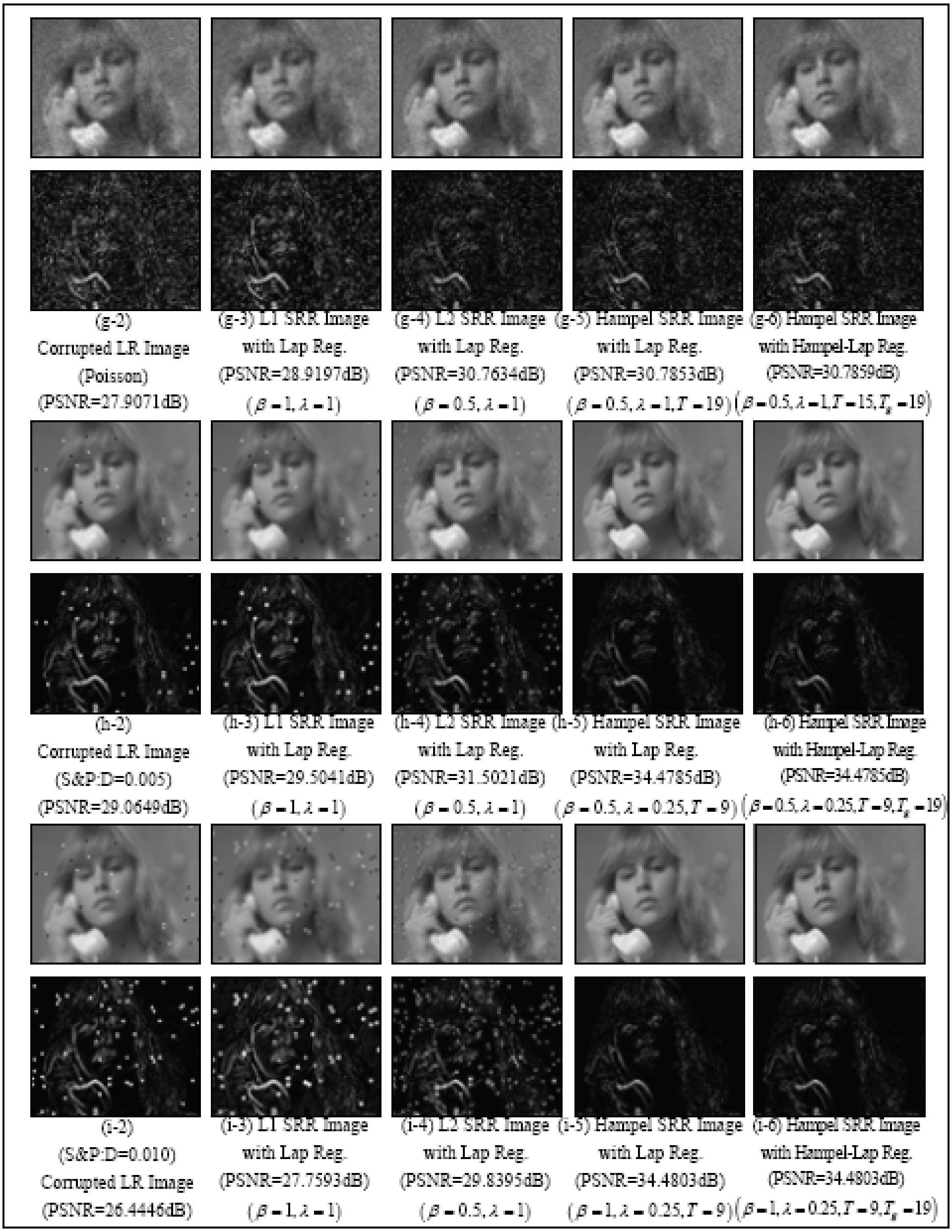
(f-3) L1 SRR Image with Lap Reg. (PSNR=26.2641dB) $(\beta = 0.5, \lambda = 1)$

(f-4) L2 SRR Image with Lap Reg. (PSNR=27.6671dB) $(\beta = 0.5, \lambda = 1)$

(f-5) Hampel SRR Image with Lap Reg. (PSNR=27.6565dB) $(\beta = 0.5, \lambda = 1, T = 1)$

(f-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=27.6565dB) $(\beta = 0.5, \lambda = 1, T = 1, T_g = 19)$

Fig. 4. The Experimental Result of Proposed SRR Algorithm: Susie (40th Frame) (Cont.)
(The bottom image on our experiment result of each subfigure is the absolute difference between it's correspond top image to the original HR image. The difference is magnified by 5.)



(g-2) Corrupted LR Image (Poisson) (PSNR=27.9071dB)

(g-3) L1 SRR Image with Lap Reg. (PSNR=28.9197dB) $(\beta = 1, \lambda = 1)$

(g-4) L2 SRR Image with Lap Reg. (PSNR=30.7634dB) $(\beta = 0.5, \lambda = 1)$

(g-5) Hampel SRR Image with Lap Reg. (PSNR=30.7853dB) $(\beta = 0.5, \lambda = 1, T = 19)$

(g-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=30.7859dB) $(\beta = 0.5, \lambda = 1, T = 15, T_g = 19)$

(h-2) Corrupted LR Image (S&P:D=0.005) (PSNR=29.0649dB)

(h-3) L1 SRR Image with Lap Reg. (PSNR=29.5041dB) $(\beta = 1, \lambda = 1)$

(h-4) L2 SRR Image with Lap Reg. (PSNR=31.5021dB) $(\beta = 0.5, \lambda = 1)$

(h-5) Hampel SRR Image with Lap Reg. (PSNR=34.4785dB) $(\beta = 0.5, \lambda = 0.25, T = 9)$

(h-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=34.4785dB) $(\beta = 0.5, \lambda = 0.25, T = 9, T_g = 19)$

(i-2) (S&P:D=0.010) Corrupted LR Image (PSNR=26.4446dB)

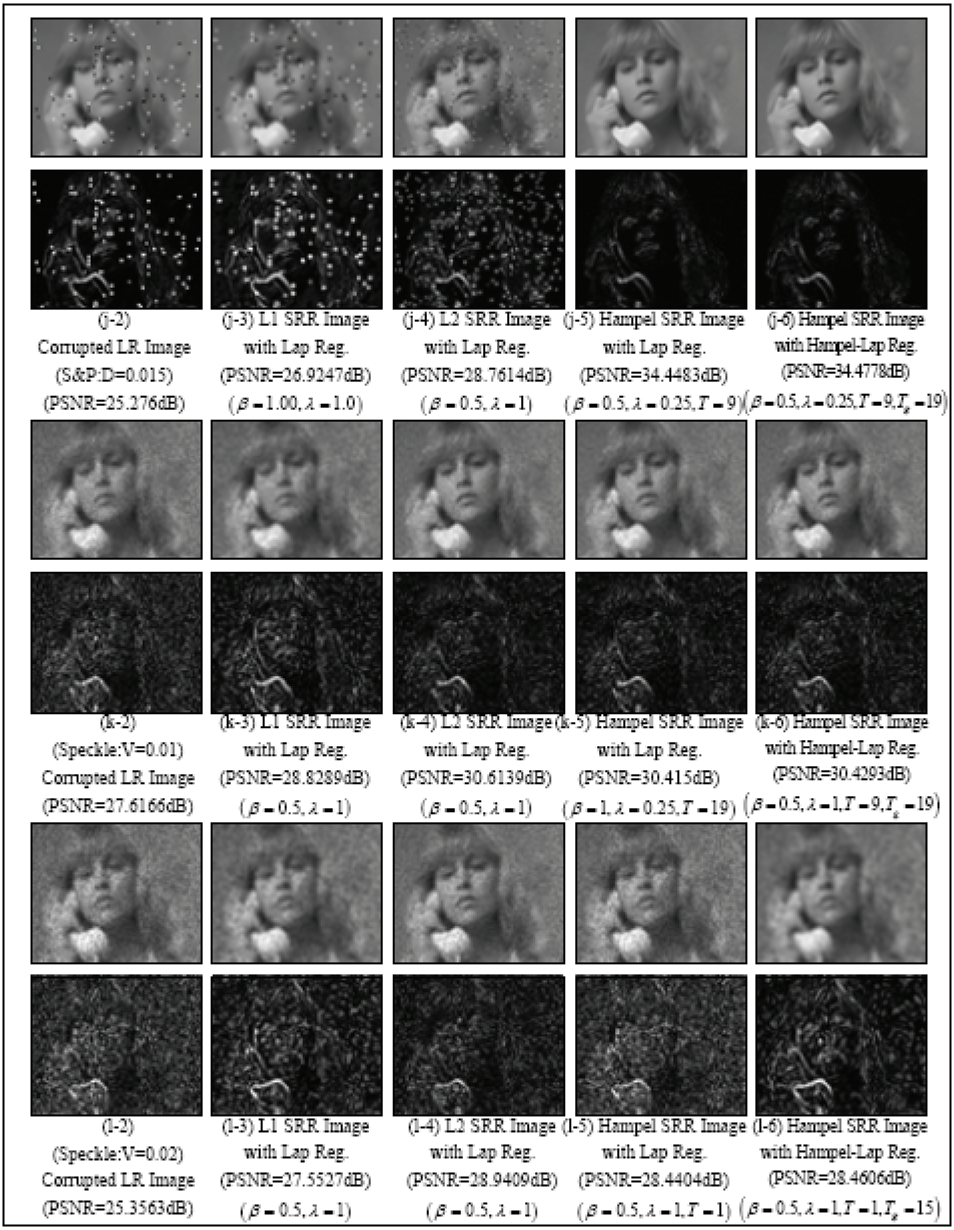
(i-3) L1 SRR Image with Lap Reg. (PSNR=27.7593dB) $(\beta = 1, \lambda = 1)$

(i-4) L2 SRR Image with Lap Reg. (PSNR=29.8395dB) $(\beta = 0.5, \lambda = 1)$

(i-5) Hampel SRR Image with Lap Reg. (PSNR=34.4803dB) $(\beta = 1, \lambda = 0.25, T = 9)$

(i-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=34.4803dB) $(\beta = 1, \lambda = 0.25, T = 9, T_g = 19)$

Fig. 4. The Experimental Result of Proposed SRR Algorithm: Susie (40th Frame) (Cont.)
(The bottom image on our experiment result of each subfigure is the absolute difference between it's correspond top image to the original HR image. The difference is magnified by 5.)



(j-2) Corrupted LR Image (S&P:D=0.015) (PSNR=25.276dB)

(j-3) L1 SRR Image with Lap Reg. (PSNR=26.9247dB) $(\beta = 1.00, \lambda = 1.0)$

(j-4) L2 SRR Image with Lap Reg. (PSNR=28.7614dB) $(\beta = 0.5, \lambda = 1)$

(j-5) Hampel SRR Image with Lap Reg. (PSNR=34.4483dB) $(\beta = 0.5, \lambda = 0.25, T = 9)$

(j-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=34.4778dB) $(\beta = 0.5, \lambda = 0.25, T = 9, T_R = 19)$

(k-2) (Speckle:V=0.01) Corrupted LR Image (PSNR=27.6166dB)

(k-3) L1 SRR Image with Lap Reg. (PSNR=28.8289dB) $(\beta = 0.5, \lambda = 1)$

(k-4) L2 SRR Image with Lap Reg. (PSNR=30.6139dB) $(\beta = 0.5, \lambda = 1)$

(k-5) Hampel SRR Image with Lap Reg. (PSNR=30.415dB) $(\beta = 1, \lambda = 0.25, T = 19)$

(k-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=30.4293dB) $(\beta = 0.5, \lambda = 1, T = 9, T_R = 19)$

(l-2) (Speckle:V=0.02) Corrupted LR Image (PSNR=25.3563dB)

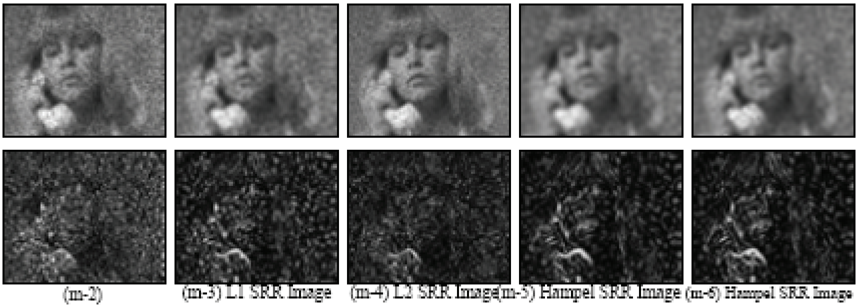
(l-3) L1 SRR Image with Lap Reg. (PSNR=27.5527dB) $(\beta = 0.5, \lambda = 1)$

(l-4) L2 SRR Image with Lap Reg. (PSNR=28.9409dB) $(\beta = 0.5, \lambda = 1)$

(l-5) Hampel SRR Image with Lap Reg. (PSNR=28.4404dB) $(\beta = 0.5, \lambda = 1, T = 1)$

(l-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=28.4606dB) $(\beta = 0.5, \lambda = 1, T = 1, T_R = 15)$

Fig. 4. The Experimental Result of Proposed SRR Algorithm: Susie (40th Frame) (Cont.) (The bottom image on our experiment result of each subfigure is the absolute difference between it's correspond top image to the original HR image. The difference is magnified by 5.)



(m-2) (Speckle:V=0.03) Corrupted LR Image (PSNR=24.0403dB)

(m-3) L1 SRR Image with Lap Reg. (PSNR=26.8165dB) $(\beta = 0.5, \lambda = 1)$

(m-4) L2 SRR Image with Lap Reg. (PSNR=27.7654dB) $(\beta = 0.5, \lambda = 1)$

(m-5) Hampel SRR Image with Lap Reg. (PSNR=27.9409dB) $(\beta = 0.5, \lambda = 1, T = 1)$

(m-6) Hampel SRR Image with Hampel-Lap Reg. (PSNR=28.0189dB) $(\beta = 0.5, \lambda = 1, T = 1, T_R = 9)$

Fig. 4. The Experimental Result of Proposed SRR Algorithm: Susie (40th Frame) (Cont.) (The bottom image on our experiment result of each subfigure is the absolute difference between it's correspond top image to the original HR image. The difference is magnified by 5.)

7

Novelty Detection: An Approach to Foreground Detection in Videos

Alireza Tavakkoli
University of Nevada, Reno
USA

1. Introduction

Classification is an important mechanism in many pattern recognition applications. In many of these application, such as object recognition, there are several classes from which the data originates. In such cases many traditional classification methods such as Artificial Neural Networks or Support Vector Machines are used. However, in some applications the training data may belong to only one class. In this case, the classification is performed by finding whether a test sample belongs to the known class or not. The main criteria in single-class classification (also known as novelty detection) is to perform the classification without any information about other classes.

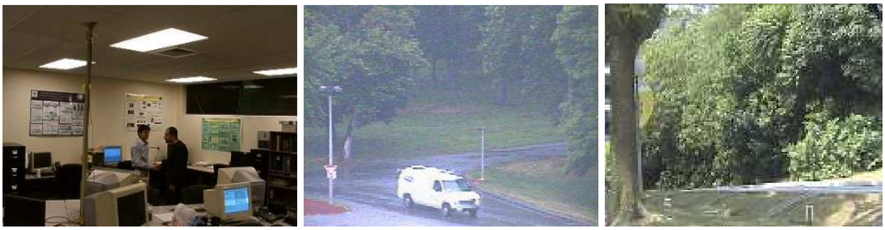
This chapter presents a classic problem in video processing applications and addresses the issues through novelty detection techniques. The problem at hand is to detect foreground objects in a video with quasi-stationary background. The video background is called quasi-stationary if the camera is static but the background itself changes due to waving tree branches, flags, water surfaces, etc. Detection of foreground region in such scenarios requires a pixel-wise background model for each pixel in the scene. Once the pixel models are built, there should be a mechanism to decide whether pixels in new frames belong to their corresponding background model or not. The generation of pixel models from their history and the decision making mechanism is a novelty detection problem.

In order to address the foreground detection problem, two main approaches to novelty detection, namely statistical and analytical, are presented in this chapter. The advantage and disadvantages of these approaches are discussed. Moreover, the suitability of each approach to specific scenarios in video processing applications are evaluated.

2. Foreground Detection

Detecting foreground regions in videos is one of the most important tasks in high-level video processing applications. One of the major issues in detecting foreground regions using background subtraction techniques is that because of inherent changes in the background, such as fluctuations in monitors and lights, waving flags and trees, water surfaces, the background may not be completely stationary. These difficult situations are illustrated in Fig. 1.

In the presence of these types of backgrounds, referred to as quasi-stationary, a single background frame is not enough to accurately detect moving regions. Therefore the background pixels of the video have to be modeled in order to detect foreground regions while allowing



(a) (b) (c)

Fig. 1. Examples of challenges in quasi-stationary backgrounds: (a) Fluctuating monitors. (b) Rain/Snow. (c) Waving tree branches.

for the changes in the background. The scenarios in which the background modeling techniques are used to detect foreground regions are very diverse. Applications vary from indoor scenes to outdoor, from completely stationary to dynamic backgrounds, from high quality videos to low contrast scenes and so on. Therefore, a single system capable of addressing all possible situations while being time and memory efficient is yet to be devised.

(Pless et al., 2003) evaluated different models for dynamic backgrounds. Typically, background models are defined independently on each pixel, and depending on the complexity of the problem employ the expected pixel features (i.e. colors), (Elgammal et al., 2002), or consistent motion, (Pless et al., 2000). They also may employ pixel-wise information, (Wern et al., 1997), or regional models of the features, (Toyama et al., 1999). To improve robustness to spatio-temporal features, (Li et al., 2004), may be used.

In (Wern et al., 1997) a single 3-D Gaussian model for each pixel in the scene is built, where the mean and covariance of the model are learned in each frame. This system tried to model the noise and used a background subtraction technique to detect those pixels whose probabilities are smaller than a threshold. However, the system fails to label a pixel as foreground or background when it has more than one modality due to fluctuations in its values, such as a pixel belonging to a fluctuating monitor.

A mixture of Gaussians modeling technique was proposed in (Stauffer & Grimson, 2000); (Stauffer & Grimson, 1999) to address the multi-modality of the underlying background. In this modeling technique background pixels are modeled by a mixture of a number of Gaussian functions. During the training stage, parameters of each Gaussian are trained and used in the background subtraction, where the probability of each pixel is generated. Each pixel is labeled as foreground or background based on its probability.

There are several shortcomings for mixture learning methods. First, the number of Gaussians needs to be specified. Second, this method does not explicitly handle spatial dependencies. Even with the use of incremental-EM, the parameter estimation and its convergence is noticeably slow where the Gaussians adapt to a new cluster. The convergence speed can be improved by sacrificing memory as proposed in (McKenna et al., 1998), limiting its applications where mixture modeling is pixel-based and over long temporal windows.

In (Elgammal et al., 2002), a non-parametric kernel density estimation method (KDE) for pixel-wise background modeling is proposed without making any assumption on its probability distribution. Therefore, this method can easily deal with multi-modality in background pixel distributions without specifying the number of modes in the background. However, there are several issues to be addressed using non-parametric kernel density estimation. These methods are memory and time consuming since the system has to compute the average of all kernels centered at each training sample for each pixel in each frame. Also the size of temporal window used as the background model is critical. In order to adapt the model a sliding window is used in (Mittal & Paragios, 2004). However, the model convergence is problematic in situations where the illumination suddenly changes.

In the traditional approaches for foreground detection presented above, the problem is addressed by reformatting a bi-class classification methodology to fit into the novelty detection approach. For example in the Mixture of Gaussian approach, changes in each pixel are modeled by a number of Gaussian functions. For new pixels a probability is calculated using the pixel model. Then a heuristically selected threshold is used to determine whether the pixel belongs to background or foreground based on its probability.

The major drawback of such approaches is the threshold choice. In these statistical approaches such as the mixture of Gaussians or the KDE, the pixel model is its probability distribution function belonging to the background. Since the background is quasi-stationary and natural, pixels in different locations undergo different amount of changes. Since the probability density functions are normalized, the pixels with less changes will have narrow but tall probability density functions while the pixels with more changes are represented by wider but shorter density functions. Therefore, finding a global threshold that works well for the majority of the background pixels and in a diverse range of applications is practically untractable.

In this chapter, two approaches based on novelty detection to address the single class classification, inherent to background modeling, are investigated. The statistical approach is based on a recursive modeling of the background pixels. This technique is called the RM, (Tavakkoli et al., 2006c). As an alternative to this statistical approach an analytical counter part to the RM technique is presented and is based on the Support Vector Data Description, (Tax & Duin, 2004). This technique is called Support Vector Data Description Modeling (SVDDM) and looks at modeling the pixels as an analytical description boundary, (Tavakkoli, Kelley, King, Nicolescu, Nicolescu & Bebis, 2007). An incremental version of the SVDDM technique is presented in (Tavakkoli, Nicolescu & Bebis, 2008).

The rest of this chapter is organized as follows. In Section 3 the theory behind the RM technique is presented. Section 4 gives a detailed algorithm of the support vector data description method in detecting foreground regions in video sequences. Performances of the proposed methods are evaluated in Section 5. Section 6 presents a comparison between the performance of these techniques and other existing methods on real videos as well as synthetic data and a comparison summary is drawn in this section. Finally, Section 7 concludes the chapter and gives future direction for research.

3. The Recursive Modeling

This section describes a technique called Recursive Modeling (RM) for foreground region detection in videos. The theory behind this approach is to generate a histogram of the data samples, with the hope that when a large number of training samples are processed, the histogram estimates the actual probability of the underlying data. System details and its theory are explained in the following, (Tavakkoli et al., 2006a), (Tavakkoli et al., 2006c), and (Tavakkoli, Nicolescu, Bebis & Nicolesu, 2008).

3.1 The theory

Let x_t be the the intensity value of a pixel at time t. The non-parametric estimation of the background model that accurately follows its multi-modal distribution can be reformulated in terms of recursive filtering, (Tavakkoli, Nicolescu, Bebis & Nicolesu, 2008):

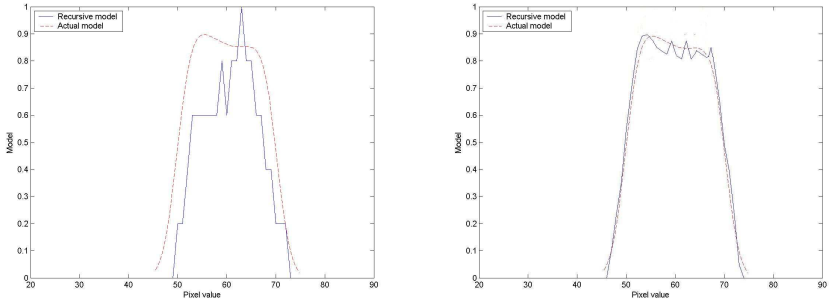
$$\hat{\theta}_t^B(x) = [1 - \beta_t] \cdot \theta_{t-1}^B(x) + \alpha_t \cdot H_\Delta(x - x_t) \qquad \forall x \in [0{,}255] \tag{1}$$

$$\sum_{x=0}^{255} \theta_t^B(x) = 1 \tag{2}$$

where θ_t^B is the background pixel model at time t, normalized according to (2). $\hat{\theta}_t^B$ is updated by the local kernel $H(\cdot)$ with bandwidth Δ centered at x_t. Parameters α_t and β_t are the learning rate and forgetting rate schedules, respectively. The kernel H should satisfy the following:

$$\begin{aligned} \sum_x H_\Delta(x) &= 1 \\ \sum_x x \times H_\Delta(x) &= 0 \end{aligned} \tag{3}$$

These conditions should be satisfied to ensure that the kernel is normalized, symmetric and positive definite in case of multivariate kernels. Note that in this context there is no need to specify the number of modalities of the background representation at each pixel. In our implementation of the RM method we use a Gaussian kernel which satisfies the above conditions.



(a) Model after 10 frames

(b) Model after 200 frames

Fig. 2. Recursive modeling convergence to the actual probability density function over time.

Figure 2 shows the updating process using our proposed recursive modeling technique. It can be seen that the trained model (solid line) converges to the actual one (dashed line) as new samples are introduced. The actual model is the probability density function of a sample population and the trained model is generated by using the recursive formula in (1).

In existing non-parametric kernel density estimation methods, the learning rate α is selected to be constant and has small values. This makes the pixel model convergence slow and keeps its history in the recent temporal window of size $L = 1/\alpha$. The window size in non-parametric models is important as the system has to cover all possible fluctuations in the background model. That is, pixel intensity changes may not be periodic or regular and consequently do not fit in a small temporal window. In such cases larger windows are needed, resulting in higher memory and computational requirements to achieve accurate, real-time modeling.

Another issue in non-parametric density estimation techniques is that the window size is fixed and is the same for all pixels in the scene. However, some pixels may have less fluctuations and therefore need smaller windows to be accurately modeled, while others may need a much longer history to cover their fluctuations.

3.1.1 Scheduled learning

In order to speed up the modeling convergence and recovery we use a schedule for learning the background model at each pixel based on its history. This schedule makes the adaptive learning process converge faster, without compromising the stability and memory requirements of the system. The learning rate changes according to the schedule:

$$\alpha_t = \frac{1-\alpha_0}{h(t)} + \alpha_0 \tag{4}$$

where α_t is the learning rate at time t and α_0 is a small target rate which is:

$$\alpha_0 = 1/256 \times \sigma_\theta \tag{5}$$

where σ_θ is the model variance. The function $h(t)$ is a monotonically increasing function:

$$h(t) = t - t_0 + 1 \tag{6}$$

where t_0 is the time at which a sudden global change is detected. At early stages the learning occurs faster ($\alpha_t = 1$), then it monotonically decreases and converges to the target rate ($\alpha_t \to \alpha_0$). When a global change is detected $h(t)$ resets to 1. The effect of this schedule on improving the convergence and recovery speed are discussed later.

The forgetting rate schedule is used to account for removing those values that have occurred long time ago and no longer exist in the background. In the current implementation we assume that the forgetting rate is a portion of the learning rate $\beta_t = l \cdot \alpha_t$, where $l \leq 1$. In the current implementation $l = 0.5$ is employed in all experiments. This accounts for those foreground objects that cover some parts of the background but after a sufficiently small period move. This keeps the history of the covered background in short-term.

3.1.2 Incorporating color information

The recursive learning scheme in 1-D has been explained in the previous section. The background and foreground models are updated using the intensity value of pixels at each frame. To extend the modeling to higher dimensions and incorporate color information, one may consider each pixel as a 3 dimensional feature vector in $[0,255]^3$. The kernel H in this space is a multivariate kernel H_Σ. In this case, instead of using a diagonal matrix H_Σ a full multivariate kernel can be used. The kernel bandwidth matrix Σ is a symmetric positive definite 3×3 matrix. Given N pixels, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$, labeled as background, their successive deviation matrix is a matrix Δ_X whose columns are:

$$\left[\mathbf{x}_i - \mathbf{x}_{i-1}\right]^T \quad \text{with} \quad i = 2,3,\cdots,N \tag{7}$$

The bandwidth matrix is defined so that it represents temporal scatter of the training data:

$$\Sigma = \mathrm{cov}(\Delta_X) \tag{8}$$

In order to decrease the memory requirements of the system we assumed that the two chrominance values are independent. Making this assumption results in a significant decrease in memory requirements while the accuracy of the model does not decay drastically. The red/green chrominance values can be quantized into 256 discrete values.

```
1. Initialization; $\Delta, \alpha_0, \beta, \kappa$ and $th$
2. For each frame
   For each pixel
    2.1. Training stage
        - Update $\alpha_t = \frac{1-\alpha_0}{h(t)} + \alpha_0$ and $\Delta$
        - Update $\theta_t^B = (1-\beta_t)\theta_{t-1}^B + \alpha_t \cdot H_\Delta$
        - If $\theta_t^B \leq th$ then update $\theta_t^F = (1-\beta_t)\theta_{t-1}^F + \alpha_t \cdot H_\Delta$
    2.2. Classification stage
        - If $ln\big(\mathrm{med}(\theta_t^F)/\mathrm{med}(\theta_t^B)\big) \geq \kappa$ then label pixel as foreground.
    2.3. Update stage
        - Update $\kappa$ and $th$
```

Fig. 3. The RM algorithm.

3.2 The algorithm

The proposed method, in pseudo-code, is shown in Figure 3. There are three major steps in the RM method: training, classification and update stages, respectively. The role and results of each stage along with its details are presented in the following.

3.2.1 The Training Stage

Before new objects appear in the scene, at each pixel all the intensity values have the same probability of being foreground. However, in each new frame the pixel background models are updated according to equation (1), resulting in larger model values (θ^B) at the pixel intensity value x_t. In essence, the value of the background pixel model at each intensity x is:

$$\theta_t^B(x) = P(\mathrm{Bg}|x) \qquad x \in [0,255] \tag{9}$$

In order to achieve better detection accuracy we introduce the foreground model which in the classification stage is compared to the background model to make the decision on whether the pixel belongs to background or foreground. This foreground model represents all other unseen intensity/color values for each pixel that does not follow the background history and is defined by:

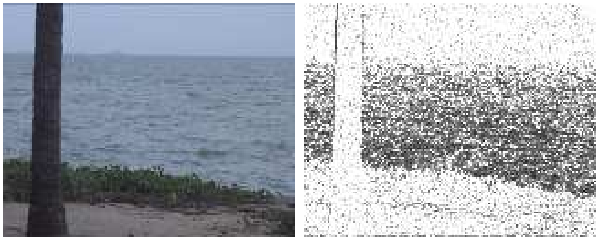
$$\hat{\theta}_t^F(x) = [1-\beta_t^F] \cdot \theta_{t-1}^F(x) + \alpha_t^F \cdot H_\Delta(x - x_t) \qquad \forall x \in [0,255] \tag{10}$$

$$\sum_{x=0}^{255} \theta_t^F(x) = 1 \tag{11}$$

Once the background model is updated, it is compared to its corresponding threshold th. This threshold is automatically maintained for each pixel through the update stage which is described in details later. If the pixel probability is less than this threshold the foreground model for that pixel value is updated according to (10) and (11).

3.2.2 The Classification stage

For each pixel at time t we use a function θ_t^B for the background model and θ_t^F for the foreground. The domain of these functions is $[0,255]^N$, where N is the dimensionality of the pixel feature vector. For simplicity assume the one dimensional case again, where θ_t is the background/foreground model whose domain is $[0,255]$. From equation (10), each model ranges between 0 to 1 and its value shows the amount of evidence accumulated in the updating process (i.e., the estimated probability). For each new intensity value x_t we have the evidence



(a) An arbitrary frame (b) The threshold map

Fig. 4. Adaptive threshold map: different pixels need different thresholds.

of each model as $\theta_t^B(x_t)$ and $\theta_t^F(x_t)$. The classification uses a *maximum a posteriori* criterion to label the pixel as foreground:

$$ln\left(\frac{\theta_t^B}{\theta_t^F}\right) \leq \kappa \tag{12}$$

3.2.3 The Update stage

In order for the RM technique to address the single class classification problem at hand there is a need for an adaptive classification criteria. Because not all pixels in the scene follow the same changes, the decision threshold, θ and κ should be adaptive and independent for each pixel and has to be derived from the history of that pixel. Figure 4 explains this issue.

For each pixel its threshold value (*th*) is selected such that its classifier results in 5% false reject rate. That is, 95% of the time the pixel is correctly classified as belonging to background. Therefore, The Thresholds *th* for each pixel should adapt to a value where:

$$\sum_{x:\theta_t^B(x)\geq th} \theta_t^B(x) \geq 0.95 \tag{13}$$

This can be seen in Figure 4, where (a) shows an arbitrary frame of a sequence containing a water surface and (b) shows the trained threshold map for this frame. Darker pixels in Figure 4(b) represent smaller threshold values and lighter pixels correspond to larger threshold values. As it can be observed, the thresholds in the areas that tend to change more, such as the water surface, are lower than in those areas with less amount of change, such as the sky. This is because for pixels which change all the time, the certainty about the background probability values is less.

For the other set of thresholds κ, we similarly use a measure of changes in the intensity at each pixel position. Therefore the threshold κ is proportional to the logarithm of the background model variance:

$$\kappa \approx \ln\left\{\sum_{x=0}^{255}\left(\theta_t^B(x) - \text{mean}[\theta^B(x)]\right)\right\} \tag{14}$$

This ensures that for pixels with more changes, higher threshold values are chosen for classification, while for those pixels with fewer changes smaller thresholds are employed. It should be mentioned that in the current implementation of the algorithm, the thresholds are updated every 30 frames (kept as the background buffer and used to perform the adaptation process).

More in depth evaluation of the RM technique for novelty detection and its experimental results on synthetic data and real videos will be presented in the future sections. The RM is also compared intensively with the SVDDM as well as the traditional background modeling approaches.

4. The Support Vector Data Description Modeling

In this section a powerful technique in describing the background pixel intensities, called Support Vector Data Description Modeling is presented, (Tavakkoli, Nicolescu & Bebis, 2007). Single-class classifiers, also known as novelty detectors are investigated in the literature, (Bishop, 1994). Our method trains single class classifiers for each pixel in the scene as their background model. The backbone of the proposed method is based on describing a data set using their support vectors, (Tax & Duin, 2004). In the following, details of the SVDDM and the algorithm which detects foreground regions based on this technique are presented.

4.1 The theory

A normal data description gives a closed boundary around the data which can be represented by a hyper-sphere (i.e. $F(R,a)$) with center a and radius R, whose volume should be minimized. To allow the possibility of outliers in the training set, slack variables $\epsilon_i \geq 0$ are introduced. The error function to be minimized is:

$$F(R,a) = R^2 + C\sum_i \epsilon_i \, \|x_i - a\|^2 \leq R^2 + \epsilon_i \tag{15}$$

subject to:

$$\|x_i - a\|^2 \leq R^2 + \epsilon_i \quad \forall i. \tag{16}$$

In order to have a flexible data description kernel functions $K(x_i,x_j) = \Phi(x_i) \cdot \Phi(x_j)$ are used. After applying the kernel and using Lagrange optimization the SVDD function becomes:

$$L = \sum_i \alpha_i K(x_i,x_i) - \sum_{i,j} \alpha_i \alpha_j K(x_i,x_j) \tag{17}$$

$$\forall \alpha_i : 0 \leq \alpha_i \leq C$$

Only data points with non-zero α_i are needed in the description of the data set, therefore they are called *support vectors* of the description. After optimizing (17) the Lagrange multipliers should satisfy the normalization constraint $\sum_i \alpha_i = 1$.

Optimizing equation (17) is a Quadratic Programming (QP) problem. Generally the SVDD is used to describe large data sets. In such applications optimization via standard QP techniques becomes intractable. To address this issue several algorithms have been proposed which employ faster solutions to the above QP problem.

4.2 The algorithm

The methodology described in section 4.1 is used in our technique to build a descriptive boundary for each pixel in the background training frames to generate its model for the background. Then these boundaries are used to classify their corresponding pixels in new frames as background and novel (foreground) pixels. There are several advantages in using the Support Vector Data Description (SVDD) method in detecting foreground regions:

- Unlike existing statistical modeling techniques, the proposed method explicitly addresses the single-class classification problem.

```
1. Initialization; $C$, Trn_No, $\sigma$
2. For each frame t
   For each pixel $\mathbf{x}$(i,j)
      2.1. **Training stage** % OC(i,j) = 1- class classifier for pixel (i,j)
           SVD(i,j)$\leftarrow$ Incrementally train($\mathbf{x}_t(i,j)$) % SVD: The Description
      2.2. **Classification stage** % Desc(i,j) = classification values
           Desc(i,j)$\leftarrow$ Test($\mathbf{x}_t(i,j)$,OC(i,j))
           Label pixel based on Desc(i,j).
      2.3. **Update stage**
           Re-train classifiers every 30 frames
```

Fig. 5. The SVDDM algorithm.

- The proposed method has less memory requirements compared to non-parametric density estimation techniques, in which all the training samples for the background need to be stored in order to estimate the probability of each pixel in new frames. The proposed technique only requires a very small portion of the training samples, *support vectors*.
- The accuracy of this method is not limited to the accuracy of the estimated probability density functions for each pixel.
- The efficiency of our method can be explicitly measured in terms of false reject rates. The proposed method considers a goal for false positive rates, and generates the description of the data by fixing the false positive tolerance of the system.

Figure 5 shows the proposed algorithm in pseudo-code format[1]. The only critical parameter is the number of training frames (Trn_No) that needs to be initialized. The support vector data description confidence parameter C is the target false reject rate of the system. This is not a critical parameter and accounts for the system's tolerance. Finally the Gaussian kernel bandwidth, σ does not have a particular effect on the detection rate as long as it is not set to be less than one, since features used in our method are normalized pixel chrominance values. For all of our experiments we set $C = 0.1$ and $\sigma = 5$. The optimal value for these parameters can be estimated by a cross-validation stage.

4.2.1 The Training Stage

In order to generate the background model for each pixel the SVDDM method uses a number of training frames. The background model in this technique is the description of the data samples (color and/or intensity values). The data description is generated in the training stage of the algorithm. In this stage, for each pixel a SVDD classifier is trained using the training frames, detecting support vectors and the values of Lagrange multipliers.

The support vectors and their corresponding Lagrange multipliers are stored as the classifier information for each pixel. This information is used for the classification step of the algorithm. The training stage can be performed off-line in cases where there are not global changes in the illumination or can be performed in parallel to the classification to achieve efficient results.

4.2.2 The Incremental SVDD Training Algorithm

Our incremental training algorithm is based on the theorem proposed by Osuna et al. in Osuna et al. (1997). According to Osuna a large QP problem can be broken into series of

[1] The proposed method is implemented in MATLAB 6.5, using Data Description toolbox (Tax, 2005).

smaller sub-problems. The optimization converges as long as at least one sample violates the KKT conditions.

In the incremental learning scheme, at each step we add one sample to the training working set consisting of only support vectors. Assume we have a working set which minimizes the current SVDD objective function for the current data set. The KKT conditions do not hold for samples which do not belong to the description. Thus, the SVDD converges only for the set which includes a sample outside the description boundary.

The smallest possible sub-problem consists of only two samples (Platt, 1998b). Since only the new sample violates the KKT conditions at every step, our algorithm chooses one sample from the working set along with the new sample and solves the optimization. Solving the QP problem for two Lagrange multipliers can be done analytically. Because there are only two multipliers at each step, the minimization constraint can be displayed in 2-D. The two Lagrange multipliers should satisfy the inequality in (17) and the linear equality in the normalization constraint.

We first compute the constraints on each of the two multipliers. The two Lagrange multipliers should lie on a diagonal line in 2-D (equality constraint) within a rectangular box (inequality constraint). Without loss of generality we consider that the algorithm starts with finding the upper and lower bounds on α_2 which are $H = min(C, \alpha_1^{old} + \alpha_2^{old})$ and $L = max(0, \alpha_1^{old} + \alpha_2^{old})$, respectively. The new value for α_2^{new} is computed by finding the maximum along the direction given by the linear equality constraint:

$$\alpha_2^{new} = \alpha_2^{old} + \frac{E_1 - E_2}{K(x_2, x_2) + K(x_1, x_1) - 2K(x_2, x_1)} \tag{18}$$

where E_i is the error in evaluation of each multiplier. The denominator in (18) is a step size (second derivative of objective function along the linear equality constraint). If the new value for α_2^{new} exceeds the bounds it will be clipped ($\hat{\alpha}_2^{new}$). Finally, the new value for α_1 is computed using the linear equality constraint:

$$\alpha_1^{new} = \alpha_1^{old} + \alpha_2^{old} - \alpha_2^{new} \tag{19}$$

4.2.3 The Classification Stage

In this stage for each frame, its pixels are used and evaluated by their corresponding classifier to label them as background or foreground. To test each pixel $\mathbf{z}_t$, the distance to the center of the description hyper-sphere is calculated:

$$\|\mathbf{z}_t - \mathbf{a}\|^2 = (\mathbf{z}_t \cdot \mathbf{z}_t) - 2\sum_i \alpha_i (\mathbf{z}_t \cdot \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{20}$$

A pixel is classified as a background pixel if its distance to the center of the hyper-sphere is less than or equal to R:

$$\|\mathbf{z}_t - \mathbf{a}\|^2 \leq R^2 \tag{21}$$

R is the radius of the description. Therefore, it is equal to the distance of each support vector from the center of the hyper-sphere:

$$R^2 = (\mathbf{x}_k \cdot \mathbf{x}_k) - 2\sum_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_k) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{22}$$

Note that in the implementation of the algorithm, since the boundaries of the data description are more complicated than a hyper-sphere, a kernel is used to map the training samples into a

Memory Req.	Intensity	Chrominance	Intensity+Chrominance
Bytes per pixel	1024	2048	3072

Table 1. Per-pixel memory requirements for the RM method.

higher dimension. As the result the mapped samples in the higher dimension can be described by a high dimensional hyper-sphere and the above discussion can be used.

5. Performance Evaluation

This section presents an evaluation of the performance of the RM as well as the SVDDM techniques in terms of memory requirements, speed, and other relevant parameters.

5.1 The RM Evaluation

In this section the RM method performance is evaluated. As it will be discussed later the RM method memory requirements and computation cost are independent of the number of training samples. This property makes the RM method a suitable candidate to be used in scenarios where the background changes are very slow.

5.1.1 Parameters

In the RM method there are 5 parameters: the learning and forgetting rate α and β, thresholds th and κ, and the bandwidth Σ. As described earlier in this chapter these parameters are trained and estimated from the data to generate an accurate and robust model. The reason that the RM technique is robust is that it uses most of the information in the data set and there is no limit on the number of training samples. With all parameters being automatically updated, the system performance does not require manually chose values for these parameters.

5.1.2 Memory requirements

- *Using only intensity values.*

  Since the model is a 1-D function representing the probability mass function of the pixel,it only needs 256×4 bytes per pixel to be stored. Notice that in this case, for each pixel the intensity values are integer numbers. If the memory of the system is scarce larger bin sized can be used by quantizing the intensity values.

- *Using chrominance values.*

  In this case the model is 2-D and needs $256^2 \times 4$ bytes in memory. The current implementation of the RM method uses a simple assumption of independence between color features which results in 8×256 bytes memory requirements (Tavakkoli et al., 2006c).

Table 1 shows the memory requirements in bytes per pixel for the RM method, using intensity, chrominance values and their combinations, respectively. In conclusion the asymptotic memory requirement of the RM algorithm is large but constant $O(1)$.

5.1.3 Computation cost

- *Using only intensity values.*

  If we only use pixel intensity values for pixels according to equation (1) we need 256 addition and 2×256 multiplication operations. Both the kernel and the model range from 0 to 255.

Operations	Addition	Multiplication	Asymptotic
Intensity	256	512	$O(1)$
Chrominance	512	1024	$O(1)$

Table 2. Per-pixel computational cost for the RM method.

Memory Req.	Intensity	Chrominance	both	asymptotic
Bytes per pixel	$f(C,\sigma) \times 5 \geq 10$	$f(C,\sigma) \times 8 \geq 24$	$f(C,\sigma) \geq 32$	$O(1)$
No. of SVs	$f(C,\sigma) \geq 2$	$f(C,\sigma) \geq 3$	$f(C,\sigma) \geq 4$	$O(1)$

Table 3. Per-pixel memory requirements for the SVDDM method.

- *Using chrominance values.*

  Similarly, if we use 2-D chrominance values as pixel features and use the independence assumption discussed earlier, the system requires only 2×256 addition and 4×256 multiplication operations to update the model.

Table 2 summarizes the per-pixel computational cost of the RM algorithm using only intensity values or red/green chrominance values for each pixel. The asymptotic computation cost for this system is constant, $O(1)$, since the updating process merely consists of adding two functions. Note that this technique does not need to compute the exponential function and acts as an incremental process. The algorithm is inherently fast and an efficient implementation runs in real-time reaching frame rates of 15 frames per second (fps).

5.2 The SVDDM Evaluation

In this section the SVDDM performance in terms of memory requirements and computation cost is discussed. The key to evaluate the performance of this technique is to analyze the optimization problem solved by the system to find support vectors.

5.2.1 Parameters

In order to generate the data description, a hyper-sphere of minimum size containing most of the training samples is constructed to represents the boundary of the known class. The training has three parameters including the number of training samples N, the trade off factor C and the Gaussian kernel bandwidth σ. As mentioned in Section 4.2 for all of the experiments the values for C and σ are taken 0.10 and 5, respectively. This leaves the system with only the number of frames as a scene-dependent parameter.

5.2.2 Memory requirements

It is not easy to answer how many data samples are required to find an accurate description of a target class boundary. It not only depends on the complexity of the data itself but also on the distribution of the outlier (unknown) class. However, there is a trade-off between the number of support vectors and the description accuracy. In that sense, a lower limit can be found for the number of samples required to describe the coarsest distribution boundary.

In theory, only $d+1$ support vectors in d dimensions are sufficient to construct a hyper-sphere. The center of the sphere lies within the convex hull of these support vectors.

- *Using only intensity values.*

  Since by using intensity for each pixel there is only one feature value, the support vectors are 1-D and therefore the minimum number of support vectors required to describe

Training Set Size	Incremental[1] SVDD	Online[2] SVDD	Canonical[3] SVDD
100	0.66	0.73	1.00
200	1.19	1.31	8.57
500	2.19	2.51	149.03
1000	4.20	6.93	1697.2
2000	8.06	20.1	NA
n	$O(1)$	$\Omega(1)$	$O(n)$

1- (Tavakkoli, Nicolescu, M., Nicolescu & Bebis, 2008)
2- (Tax & Laskov, 2003)
3- (Tax & Duin, 2004)

Table 4. Speed comparison of the incremental, online and canonical SVDD.

the data will be 2. For each support vector 2 bytes are required to store the intensity and 8 bytes to store the Lagrange multipliers, requiring at least 10 bits per pixel.

- *Using chrominance values.*

  By using red and green chrominance values, c_r and c_g, the minimum of 3 support vectors are needed to be used. This requires at least 24 bytes per pixel.

The above reasoning provides a lower limit on the number of support vectors. In practical applications this lower limit is far from being useful for implementation. However, notice that the number of support vectors required to sufficiently describe a data set is related to the target description accuracy. Therefore, the memory requirement of the SVDDM method is independent of the number of training frames. Table 3 shows memory requirements in bytes per pixel for the SVDDM method using intensity, chrominance values and their combinations, respectively. The asymptotic memory requirement of the SVDDM algorithm is $O(1)$.

5.2.3 Computation cost

Training the SVDDM system for each pixel needs to solve a quadratic programming (QP) optimization problem. The most common technique to solve the above QP is the Sequential Minimal Optimization (Platt, 1998c); (Platt, 1998a), runing in polynomial time $O(n^k)$.

In order to show the performance of the proposed incremental training method and its efficiency we compare the results obtained by our technique with those of the online SVDD (Tax & Laskov, 2003) and canonical SVDD (Tax & Duin, 2004).

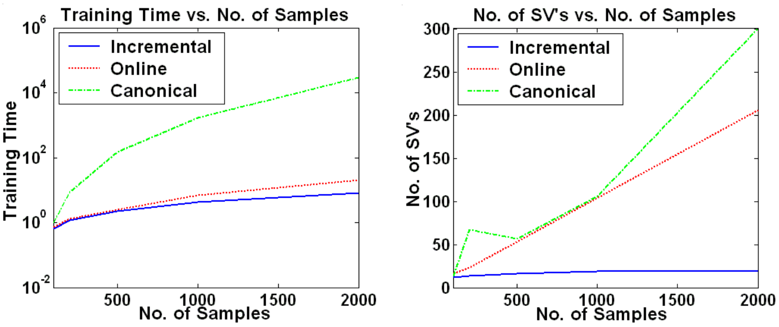
The SVVD Training Speed. In this section we compare the speed of incremental SVDD against its online and canonical counterparts. The experiments are conducted in Matlab 6.5 on a P4 Core Duo processor with 1GB RAM. The reported training times are in seconds. Table 4 Shows the training speed of the incremental SVDD, online and canonical versions on a data set of various sizes. The proposed SVDD training technique runs faster than both canonical and online algorithms and its asymptotic speed is linear with the data set size. As expected, both online and our SVDD training methods are considerably faster than the canonical training of the classifier. Notice that the training time of a canonical SVDD for 2000 training points is not available because of its slow speed.

Number of Support Vectors. A comparison of the number of retained support vectors for our technique, canonical, and online SVDD learning methods is presented in Table 5. Both online and canonical SVDD training algorithm increase the number of support vectors as the size

Training Set Size	Incremental[1] No. of SV's	Online[2] No. of SV's	Canonical[3] No. of SV's
100	12	16	14
200	14	23	67
500	16	53	57
1000	19	104	106
2000	20	206	NA
n	$O(1)$	$O(n)$	$O(n)$

1- (Tavakkoli, Nicolescu, M., Nicolescu & Bebis, 2008)
2- (Tax & Laskov, 2003)
3- (Tax & Duin, 2004)

Table 5. The number of support vectors retained.



(a) training speed (b) number of support vectors

Fig. 6. Speed and the number of support vectors comparison between the canonical learning (·— curve), the online learning (—— curve), and the incremental method (— line).

of the data set increases. However, our method keeps almost a constant number of support vectors. This can be interpreted as mapping to the same higher dimensional feature space for any given number of samples in the data set.

Notice that by increasing the number of training samples the proposed SVDD training algorithm requires less memory than both online and canonical algorithms. This makes the proposed algorithm suitable for applications in which the number of training samples increase by time. Since the number of support vectors is inversely proportional to the classification speed of the system, the incremental SVDD classification time is constant with respect to the number of samples compared with the canonical and the online methods. Figure 6 (a) and (b) shows the training speed and the number of retained support vectors, respectively.

6. Experimental Results and Comparison

In this section the performances of our approaches on a number of challenging videos are discussed and their results are compared with those of existing methods in the literature. A number of challenging scenarios are presented to the algorithms and their ability to handle issues are evaluated. The comparisons are performed both qualitatively and quantitatively.

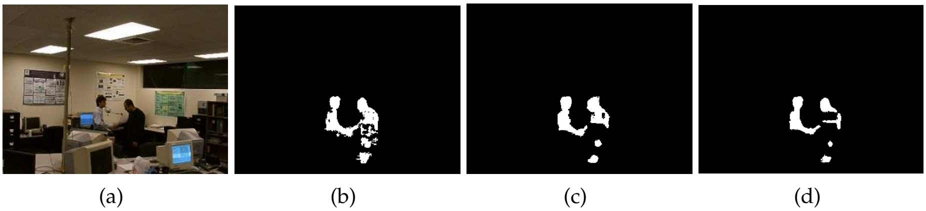


Fig. 7. Rapidly fluctuating background: (a) *Handshake* video sequence. Detected foreground regions using (b) AKDE. (c) RM. (d) SVDDM.

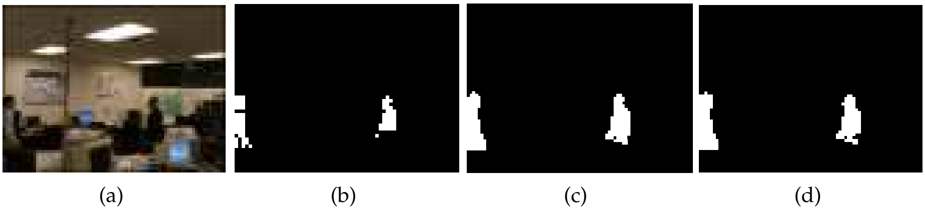


Fig. 8. Low contrast videos: (a) *Handshake* video sequence. Detected foreground regions using (b) AKDE. (b) RM. (d) SVDDM.

6.1 Foreground Detection in Videos

This section compares the performance of the proposed techniques using several real video sequences that pose significant challenges. Their performances are also compared with the mixture of Gaussians method (Stauffer & Grimson, 2000), the spatio-temporal modeling presented in (Li et al., 2004) and the simple KDE method (Elgammal et al., 2002). We use different scenarios to test the performance of the proposed techniques and to discuss where each method is suitable. In order to have a unified comparison and evaluation we use a baseline system based on Adaptive Kernel Density Estimation (AKDE) (Tavakkoli et al., 2006b). The following a several scenarios which the comparisons and evaluations are performed on.

6.1.1 Rapidly fluctuating backgrounds

Our experiments showed that for videos where possible fluctuations in the background occur in about 10 seconds, the AKDE technique needs less memory and works faster compared to the RM and SVDDM.

Figure 7 shows the detection results of the AKDE, RM and the SVDDM algorithms on the *Handshake* video sequence. From this figure the AKDE performs better than both the RM and the SVDDM. Note that in this particular frame the color of foreground objects is very close to the background in some regions. The SVDDM technique results in very smooth and reliable foreground regions but may result in missing some parts of the foreground which are very similar to the background. Moreover, all methods successfully modeled the fluctuations seen on monitors as a part of the background.

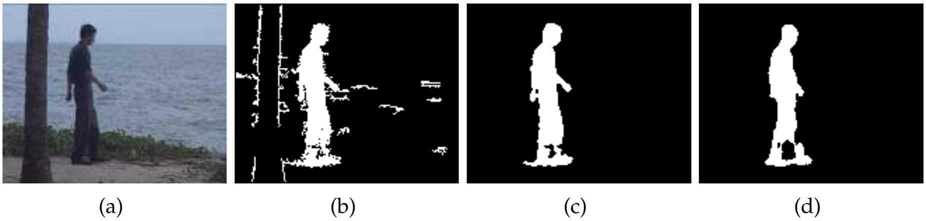


Fig. 9. Slowly changing background: (a) *Water* video sequence. Detected foreground region using (b) AKDE. (c) RM. (d) SVDDM.

6.1.2 Low contrast videos

To evaluate the accuracy of the SVDDM technique in low contrast video sequences and to compare it with the AKDE technique, the experiment is performed on the *Handshake* video sequence. Figure 8 shows a frame where the background and foreground colors are different. In this experiment the quality of the images in video sequence are decreased by blurring the video. The accuracy of the foreground regions detected using the SVDDM technique is clearly better than those of the AKDE method. The reason is that the SVDDM fixes the false reject rate of the classifier. This produces a description without estimating the probability density function of the background.

6.1.3 Slowly changing backgrounds

In videos with slowly changing backgrounds the AKDE requires more training frames to generate a good background model. Therefore the system memory requirements is increased resulting in drastic decrease in its speed. In these situations the RM technique is a very good alternative, since its performance is independent of the number of training frames.

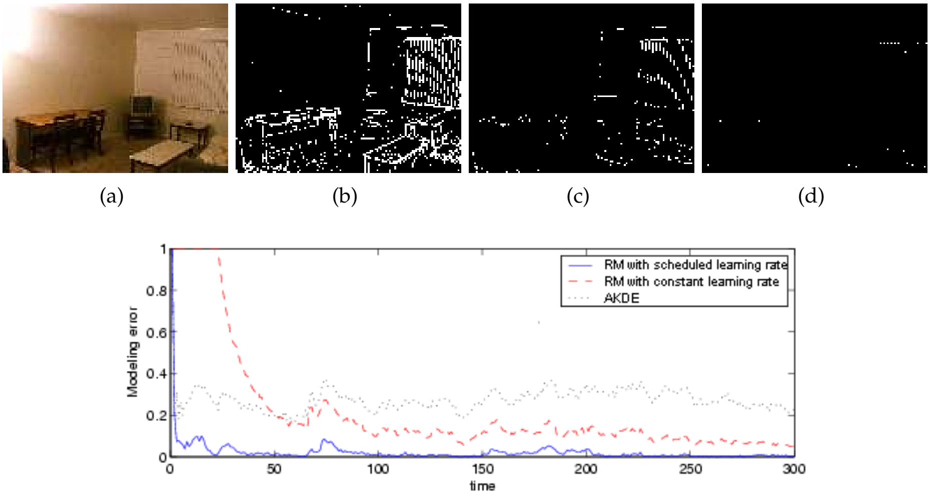
Figure 9(a) shows an arbitrary frame of the *Water* video sequence. This example is particularly difficult because waves do not follow a regular motion pattern and their motion is slow. From Figure 9, the AKDE without any post-processing results in many false positives while the detection results of the RM and the SVDDM which uses more training sample are far better.

We can conclude that the RM method has a better performance compared to both the AKDE and the SVDDM in situations in which the background has slow and irregular motion. The AKDE employs a sliding window of limited size which may not cover all changes in the background. The model is continuously updated in the RM method therefore keeping most of the changes that occurred in the past. The SVDDM method performs better than the AKDE technique in this scenario because the model that the SVDDM builds automatically generates the decision boundaries of the background class instead.

6.1.4 Hand-held camera

In situations when the camera is not completely stationary, such as the case of a hand-held camera, the AKDE and the current batch implementation of the SVDDM methods are not suitable. In these situations there is a consistent, slow and irregular global motion in the scene. These changes can not be modeled by a limited size sliding window of training frames. In such cases the RM method outperforms other tecniques.

Figure 10 shows the modeling error of the RM method in the *Room* video sequence. In Figure 10(a) an arbitrary frame of this video is shown. Figure 10(b)-(d) show the false positives de-



(a) (b) (c) (d)

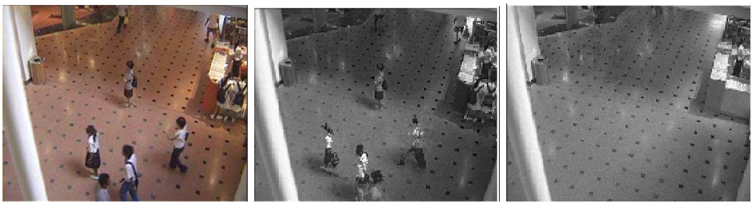
(e)

Fig. 10. Hand-held camera: (a) *Room* video sequence. False positives after (b) 2 frames, (c) 32 frames, ans (d) 247 frames using the AKDE method (e) Modeling error in a hand-held camera situation using different methods.

tected as foreground regions using the RM method. As expected early into the video the RM models are not very accurate resulting in a lot of false positives. However, as more and more frames are processed the model becomes more and more accurate (Figure 10(d)). Figure 10(e) compares the modeling error of the RM with and without scheduling as well as the AKDE (constant window size).

6.1.5 Non-empty backgrounds

In situations in which the background of the video is not empty (that is, there is no clear background at any time in the video sequence), the AKDE and SVDDM methods fail to accurately detect the foreground regions. In these situations the RM technique has to be used.



(a) (b) (c)

Fig. 11. Non-empty background: (a) *Mall* video sequence. (b) Background model after 5 frames using the RM method. (c) Background model after 95 frames using the RM method.

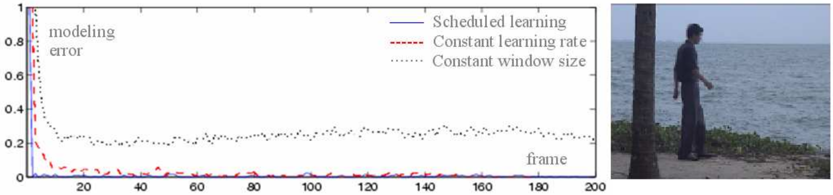


Fig. 12. Convergence speed.

Figure 11 shows the background model in the *Mall* video sequence in which the background is almost never empty. In the RM method however, the background model is updated every frame from the beginning of the video. When an object moves, the new pixel information is used to update the background model and converges to the new one. Figure 11(b) shows the background model after 5 frames from the beginning of the video and Figure 11(c) shows the model after 95 frames into the scene. The model converges to the empty background since each background pixel is covered by moving people only a short time compared to the length of the time it is not covered.

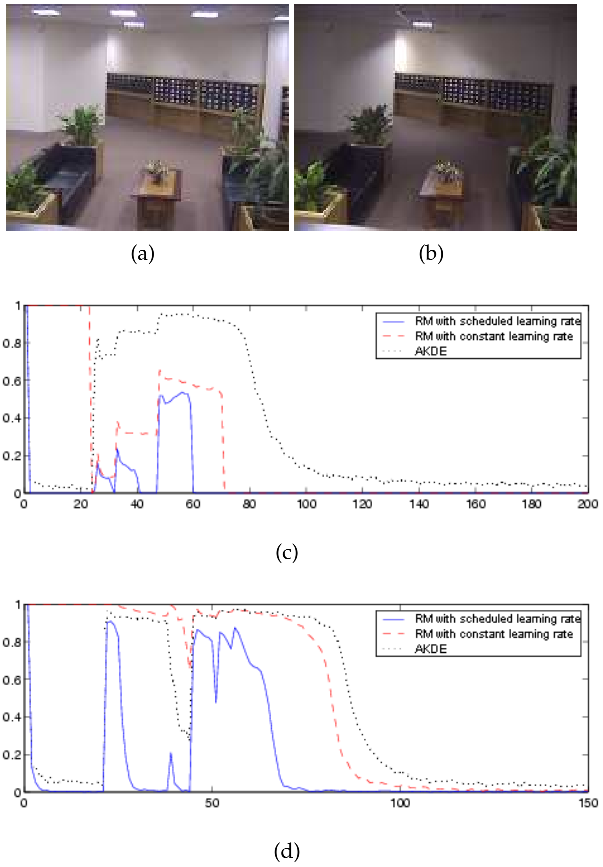
6.1.6 The RM convergence speed

An important issue in the recursive learning is the convergence speed of the system (how fast the model converges to the actual background). Figure 12 illustrates the convergence speed of the RM with scheduled learning rate, compared to constant learning and kernel density estimation with constant window size. In this figure the modeling error of the RM with scheduled learning and constant learning rate as well as the AKDE modeling error are plotted against frame number. From Figure 12, the AKDE modeling error (the black $(-\cdot)$ curve) drops to about 20% after about 20 frames – the training window size. The modeling error for this technique does not converge to 0 since the constant window size does not cover all of the slow changes in the background. In contrast, the error for an RM approach decreases as more frames are processed. This is due to the recursive nature of this algorithm and the fact that every frame contributes to the generation and update of the background model. The effect of the scheduled learning proposed in section 3 can be observed in Figure 12.

6.1.7 Sudden global changes

In situations where the video background suddenly changes, such as lights on/off, the proposed RM technique with scheduled learning recovers faster than the AKDE method. Generally, with the same speed and memory requirements, the RM method results in faster convergence and lower modeling error.

Figure 13 shows the comparison of the recovery speed from an expired background model to the new one. This happens in the *Lobby* video sequence when the lights go off (Figure 13(a)) or they go on (Figure 13(b)). In our example, lights go from on to off through three global, sudden illumination changes at frames 23, 31 and 47 (Figure 13(c)). The Figure shows that the scheduled learning RM method (solid curve) recovers the background model after these changes faster than non-scheduled RM and AKDE with constant window size. The constant, large learning rate recovers much slower (dashed curve) and the AKDE technique (dotted curve) is not able to recover even after 150 frames. A similar situation with lights going from off to on through three global, sudden illumination changes is shown in Figure 13(d).



(a) (b)

(c)

(d)

Fig. 13. Sudden global changes in the background: (a) the *Lobby* video sequence with lights on. (b) Lights off. (c) Recovery speed comparison in lights turned off scenario. (d) Recovery speed comparison in lights turned on scenario.

6.1.8 Other difficult examples

Figure 14 shows three video sequences with challenging backgrounds. In column (a) the original frames are shown; while columns (b), (c), and (d) show the results of the AKDE, the RM and the SVDDM methods, respectively. In this figure, from top row to the bottom; heavy rain, waving tree branches, and the water fountain pose significant difficulties in detecting accurate foreground regions.

6.2 Quantitative Evaluation

Performances of our proposed methods, RM and SVDDM are evaluated quantitatively on randomly selected samples from different video sequences, taken from (Li et al., 2004).

To evaluate the performance of each method a value "called similarity" measure is used. The similarity measure between two regions $\mathcal{A}$ (detected foreground regions) and $\mathcal{B}$ (ground

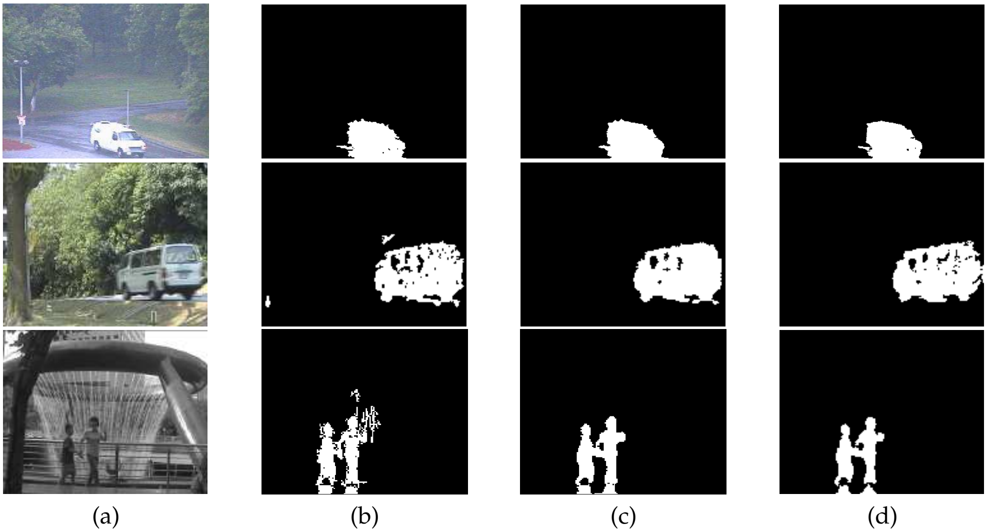


Fig. 14. Other difficult examples: (a) Original frame. Detected foreground region using (b) AKDE. (c) RM. (d) SVDDM.

truth) is defined by (Li et al., 2004):

$$\mathcal{S}(\mathcal{A},\mathcal{B}) = \frac{\mathcal{A} \cap \mathcal{B}}{\mathcal{A} \cup \mathcal{B}} \tag{23}$$

This measure increases monotonically with the similarity between detected masks and the ground truth, ranging between 0 and 1. By using this measure we report the performance of the AKDE method, the RM method, the SVDDM, the spatio-temporal technique presented in (Li et al., 2004), and the mixture of Gaussians (MoG) in (Stauffer & Grimson, 2000). By comparing the average of the similarity measure over different video sequences in Table 6, we observed that the RM and the SVDDM methods outperform other techniques. This also shows that the AKDE, RM and SVDDM methods work consistently well on a wide range of video sequences. The reason for such desirable behavior lies under the fact that these techniques automatically deal with the novelty detection problem and do not need their parameters to be fine-tuned for each scenario.

However, from this table one might argue that AKDE does not perform better than the method presented in (Li et al., 2004). The reason is that in (Li et al., 2004) the authors used a morphological post-processing stage to refine their detected foreground regions while the results shown for the AKDE are the raw detected regions. By performing a morphological post-processing on the results obtained by the AKDE it is expected that the average similarity measure increase.

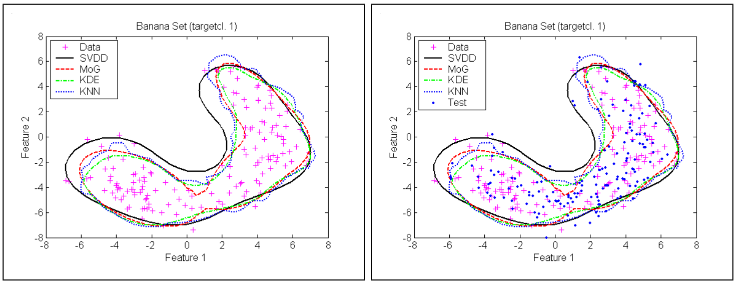
6.3 Synthetic Data Sets

We used a synthetic data set, which represents randomly distributed training samples with an unknown distribution function (*Banana* data set). Figure 15 shows a comparison between different classifiers. This experiment is performed on 150 training samples using the support

Video Method	MR	LB	CAM	SW	WAT	FT	Avg. $\mathcal{S}(\mathcal{A},\mathcal{B})$
RM	0.92	0.87	0.75	0.72	0.89	0.87	0.84
SVDDM	0.84	0.78	0.70	0.65	0.87	0.80	0.77
Spatio-Temp[1]	0.91	0.71	0.69	0.57	0.85	0.67	0.74
MoG [2]	0.44	0.42	0.48	0.36	0.54	0.66	0.49
AKDE[3]	0.74	0.66	0.55	0.52	0.84	0.51	0.64

1: (Li et al., 2004)
2: (Stauffer & Grimson, 2000)
3: (Tavakkoli et al., 2006b)

Table 6. Quantitative evaluation and comparison. The sequences are *Meeting Room, Lobby, Campus, Side Walk, Water* and *Fountain*, from left to right from (Li et al., 2004).



(a) (b)

Fig. 15. Comparison between different classifiers on a synthetic data set: (a) Decision boundaries of different classifiers after training. (b) Data points (blue dots) outside decision boundaries are false rejects.

vector data description (SVDDM), the mixture of Gaussians (MoG), the kernel density estimation (AKDE) and a k-nearest neighbors (KNN).

Parameters of these classifiers are manually determined to give a good performance. For all classifiers the confidence parameter is set to be 0.1. In MoG, we used 3 Gaussians. Gaussian kernel bandwidth in the AKDE classifier is considered $\sigma = 1$. For the KNN we used 5 nearest neighbors. In the SVDDM classifier the Gaussian kernel bandwidth is chosen to be 5.

Figure 15(a) shows the decision boundaries of different classifiers on 150 training samples from the *Banana* data set. As it can be seen from Figure 15(b), SVDDM generalizes better than the other three classifiers and classifies the test data more accurately. In this figure the test data is composed of 150 samples drawn from the same probability distribution function as the training data. Therefore this should be classified as the known class.

Method	SVDDM	MoG	AKDE	KNN
FRR	0.1067	0.1400	0.1667	0.1333
RR	0.8933	0.8600	0.8333	0.8667

Table 7. Comparison of False Reject Rate and Recall Rate for different classifiers.

We need to define the False Reject Rate (FRR) and Recall Rate (RR) for a quantitative evaluation. By definition, FRR is the percentage of missed targets, and RR is the percentage of correct prediction (True Positive rate). These quantities are given by:

$$\mathrm{FRR} = \frac{\#\text{Missed targets}}{\#\text{Samples}} \qquad \mathrm{RR} = \frac{\#\text{Correct predictions}}{\#\text{Samples}} \tag{24}$$

Table 7 shows a quantitative comparison between different classifiers. In this table, FRR and RR of classifiers are compared after training them on 150 data points drawn from an arbitrary probability function and tested on the same number of samples drawn from the same distribution. From the above example, the FRR for SVDDM is less than that of the other three classifiers, while its RR is higher. This proves the superiority of this classifier for the purpose of novelty detection.

Method	SVDDM	MoG	AKDE	KNN	RM
Memory needs (bytes)	1064	384	4824	4840	1024

Table 8. Comparison of memory requirements for different classifiers.

Table 8 shows memory requirements for each classifier. Since in SVDDM we do not need to store all the training data, as can be seen from the table, it requires much less memory than the KNN and KDE methods. Only the MoG and the RM methods need less memory than the SVDDM technique. However, the low memory requirements of the RM are achieved by coarse quantization of the intensity value.

6.3.1 Classification comparison

Table 9 compares the classification error, the F_1 measure, as well as the training and the classification asymptotic time for various classifiers. The incremental training of the SVDD reaches good classification rates compared to the other methods. The trade-off parameter is set to be $C = 0.1$ in SVDD. Kernel bandwidth for the three SVDD methods and the Parzen window is $\sigma = 3.8$. $K = 3$ is selected for the number of Gaussians in the MoG and number of nearest neighbors in the K-NN method. The F_1 measure combines both the recall and the precision rates of a classifier:

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{25}$$

Classifier	Error	F_1	Training	Classification
Proposed	0.015	0.992	$O(1)$	$O(1)$
Batch SVD	0.100	0.947	$O(N)$	$O(N)$
Online SVD	0.103	0.945	$O(N)$	$O(N)$
KDE(Parzen)	0.114	0.940	$O(N)$	$O(N)$
MoG	0.143	0.923	$O(1)$	$O(1)$
K-means	0.150	0.919	$O(1)$	$O(1)$

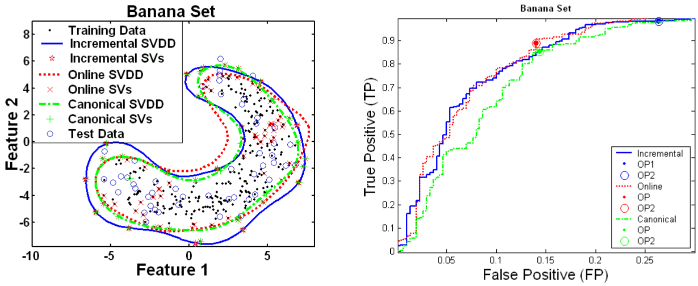
Table 9. Comparison of the classification error, F_1 measure, and asymptotic speeds with various classifiers on a *complex data set* of size 1000.

Training	Data Set	Error	F_1	No. SV's	Time
Banana	Proposed	0.005	0.997	19	4.2
	Online	0.075	0.961	104	6.9
	Canonical	0.085	0.956	106	1697
Ellipse	Proposed	0.013	0.993	6	3.72
	Online	0.100	0.947	105	4.1
	Canonical	0.110	0.994	108	2314
Egg	Proposed	0.065	0.966	8	3.85
	Online	0.095	0.950	101	3.7
	Canonical	0.128	0.932	87	1581

Table 10. Comparison of the incremental SVDD training algorithm with, online and batch methods on *Banana*, *Ellipse* and *Egg* data sets of size 1000.

6.3.2 Error evaluation

Table 10 compares the classification error, the F_1 measure, the number of the support vectors, and the learning time for the three learning methods. The experiments are performed on three data sets ('*Banana*', '*Ellipse*', '*Egg*') with 1000 training samples and 1000 test samples.

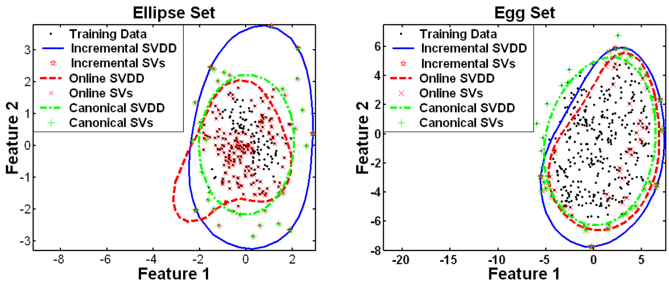


(a) (b)

Fig. 16. Comparison of incremental with canonical and online SVDD: (a) Classification boundaries . (b) Receiver Operating Curve (ROC).

6.3.3 Classification boundaries and Receiver Operating Curves

In Figure 16 (a) the classification boundaries of the three SVDD training algorithms are shown. In this figure the blue dots are the training samples drawn from the *Banana* data set and the circles represent the test data set drawn from the same probability distribution function.
The $\star$, $\times$, and $+$ symbols are the support vectors of the Incremental, Online and Canonical SVDD training algorithms, respectively. The proposed incremental learning had fewer support vectors compared to both online and canonical training algorithms. From Figure 16 (a) the decision boundaries of the classifier trained using the Incremental algorithm (solid curve) is objectively more accurate than those trained by Online (dotted curve) and Canonical (dashed curve) methods.
Figure 16 (b) shows the comparison between the Receiver Operating Curve (ROC) of the three algorithms. The solid curve is the ROC of the Incremental learning while dotted and dashed



(a) (b)

Fig. 17. Comparison of incremental with online and canonical SVDD: (a) Normal data set. (b) Complex (egg) data set.

	AKDE	RM	SVDDM	Spatio-temporal[1]	MoG[2]	Wallflower[3]
Automated	Yes	Yes	Yes	No	No	No
Classifier	Bayes	MAP	SVD	Bayes	Bayes	K-means
Memory req.*	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$
Comp. cost*	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$

$*$: Per-pixel memory requirements or computational cost
n: number of training frames or training features used per pixel
1 : (Li et al., 2004)
2 : (Stauffer & Grimson, 2000)
3 : (Toyama et al., 1999)

Table 11. Comparison between the proposed methods and the traditional techniques.

curves represent the Online and the Canonical learning algorithms, respectively. In this figure the operating point (OP) of the three ROC's (for the given trade-off value) are represented by the circle and the dot symbols. The true positive rate for the incremental SVDD is higher than the others. Therefore, the proposed method – under the same conditions – has higher precision and recall rates.

Figure 17 shows a comparison of the classification boundaries, and the support vectors between the three SVDD training algorithms. The classification boundaries on a 2-D normal distribution (Figure 17(a)) and a more complex distribution function in 2-D (Figure 17 (b)) are extracted using the three SVDD algorithms. From the figure the incremental SVDD results in more accurate classification boundaries than both online and canonical versions.

6.4 Comparison Summary

Table 11 provides a comparison between different traditional methods for background modeling in the literature and our methods. The SVDDM explicitly deals with the single-class classification. Other methods shown in the table – except the RM – use a binary classification scheme and use heuristics or a more sophisticated training scheme to make it useful for the single-class classification problem of background modeling. The RM method which has the adaptive threshold updating mechanism solves this issue and acts as a novelty detector.

	AKDE	RM	SVDDM	Spatio-temp	MoG	Wallflower
Low contrast	S*	NS**	S	NS	NS	NS
Slow changes	NS	S	S	S	S	S
Rapid changes	S	S	S	S	NS	S
Global changes	NS	S	NS	S	S	NS
Non-empty	NS	S	NS	S	S	S
Hand-held camera	NS	S	NS	NS	NS	NS

$*$: Suitable

$**$: Not suitable

Table 12. Scenarios where each method appears to be particularly suitable.

Table 12 shows different scenarios and illustrates where each method is suitable for foreground region detection. As expected the RM method is suitable for a wide range of applications except when the contrast of images in the video is low. From this table, the only method suitable for the hand-held camera scenario is the RM. The other methods fail to build a very long term model for the background because of the fact that their cost grows with the number of training background frames.

7. Conclusion

In this chapter the idea of applying a novelty detection approach to detect foreground regions in videos with quasi-stationary is investigated. In order to detect foreground regions in such videos the changes of the background pixel values should be modeled for each pixel or a groups of pixels. In the traditional approaches the pixel models are generally statistical probabilities of the pixels belonging to the background. In order to find the foreground regions the probability of each pixel in new frames being a background pixel is calculated from its model. A heuristically selected threshold is employed to detect the pixels with low probabilities.

In this chapter two approaches are presented to deal with the single class classification problem inherent to foreground detection. By employing the single class classification (novelty detection) approach the issue of heuristically finding a suitable threshold in a diverse range of scenarios and applications is addressed. These approaches presented in this chapter are also extensively evaluated. Quantitative and qualitative comparisons are conducted between the proposed approaches and the state-of-the-art, employing synthetic data as well as real videos. The proposed novelty detection mechanisms have their own strengths and weaknesses. However, the experiments show that these techniques could be used as complimentary to one another. The establishment of a universal novelty detection mechanism which incorporates the strengths of both approaches can be considered as a potential future direction in this area.

8. References

Bishop, C. (1994). Novelty detection and neural network validation., *In IEE proceedings on Vision, Image and Signal Processing. Special Issue on Application of Neural Networks.* **141**(4): 217–222.

Elgammal, A., Duraiswami, R., Harwood, D. & Davis, L. (2002). Background and foreground modeling using nonparametric kernel density estimation for visual surveillance., *In proceedings of the IEEE* **90**: 1151–1163.

Li, L., Huang, W., Gu, I. & Tian, Q. (2004). Statistical modeling of complex backgrounds for foreground object detection., *IEEE Transactions on Image Processing.* **13**(11): 1459–1472.

McKenna, S., Raja, Y. & Gong, S. (1998). Object tracking using adaptive color mixture models., *In proceedings of Asian Conference on Computer Vision* **1**: 615–622.

Mittal, A. & Paragios, N. (2004). Motion-based background subtraction using adaptive kernel density estimation., *In proceedings of CVPR* **2**: 302–309.

Osuna, E., Freund, R. & Girosi, F. (1997). Improved training algorithm for support vector machines, *In Proc. Neural Networks in Signal Processing* .

Platt, J. (1998a). *Advances in Kernel Methods - Support Vector Learning.*, Editors: B. Schlkopf, C. Burges, A. J. Smola, MIT Press.

Platt, J. (1998b). Fast training of support vector machines using sequential minimal optimization, *Advances in Kernel Methods - Support Vector Learning.* **MIT Press**: 185–208.

Platt, J. (1998c). Sequential minimal optimization: A fast algorithm for training support vector machines, *Microsoft Research Technical Report MSR-TR-98-14* .

Pless, R., Brodsky, T. & Aloimonos, Y. (2000). Detecting independent motion: The statistics of temporal continuity., *IEEE Transactions on PAMI* **22**(8): 68–73.

Pless, R., Larson, J., Siebers, S. & Westover, B. (2003). Evaluation of local models of dynamic backgrounds., *In proceedings of the CVPR* **2**: 73–78.

Stauffer, C. & Grimson, W. (1999). Adaptive background mixture models for real-time tracking., *In proceedings of CVPR* **2**: 246–252.

Stauffer, C. & Grimson, W. (2000). Learning patterns of activity using real-time tracking., *IEEE Transactions on PAMI* **22**(8): 747–757.

Tavakkoli, A., Kelley, R., King, C., Nicolescu, M., Nicolescu, M. & Bebis, G. (2007). A vision-based approach for intent recognition, *In Proceedings of the 3rd International Symposium on Visual Computing* .

Tavakkoli, A., Nicolescu, M. & Bebis, G. (2006a). An adaptive recursive learning technique for robust foreground object detection., *In proceedings of the International Workshop on Statistical Methods in Multi-image and Video Processing (in conjunction with ECCV06)* .

Tavakkoli, A., Nicolescu, M. & Bebis, G. (2006b). Automatic statistical object detection for visual surveillance., *In proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation* pp. 144–148.

Tavakkoli, A., Nicolescu, M. & Bebis, G. (2006c). Robust recursive learning for foreground region detection in videos with quasi-stationary backgrounds., *In proceedings of 18th International Conference on Pattern Recognition* .

Tavakkoli, A., Nicolescu, M. & Bebis, G. (2007). A support vector data description approach for background modeling in videos with quasi-stationary backgrounds., *to appear in the International Journal on Artificial Intelligence Tools* .

Tavakkoli, A., Nicolescu, M. & Bebis, G. (2008). Efficient background modeling through incremental support vector data description, *In Proceedings of the 19th International Conference on Pattern Recognition* .

Tavakkoli, A., Nicolescu, M., Bebis, G. & Nicolesu, M. (2008). Non-parametric statistical background modeling for efficient foreground region detection, *International Journal of Machine Vision and Applications* pp. 1–16.

Tavakkoli, A., Nicolescu, M., M., Nicolescu & Bebis, G. (2008). Incremental svdd training: Improving efficiency of background modeling in videos, *In Proceedings of the 10th IASTED Conference on Signal and Image Processing* .

Tax, D. (2005). Ddtools, the data description toolbox for matlab. version 1.11.

Tax, D. & Duin, R. (2004). Support vector data description., *Machine Learning* **54**(1): 45–66.

Tax, D. & Laskov, P. (2003). Online svm learning: from classification and data description and back., *Neural Networks and Signal Processing* (1): 499–508.

Toyama, K., Krumm, J., Brumitt, B. & Meyers, B. (1999). Wallflower: principles and practice of background maintenance., *In proceedings of ICCV* **1**: 255–261.

Wern, C., Azarbayejani, A., Darrel, T. & Pentland, A. (1997). Pfinder: real-time tracking of human body., *IEEE Transactions on PAMI* **19**(7): 780–785.

8

Application of the Wrapper Framework for Robust Image Segmentation For Object Detection and Recognition

Michael E. Farmer
University of Michigan-Flint
USA

1. Introduction

Traditional methods for object detection and classification in images involve either matched filter detectors which are convolved over the image, or else strong image segmentation followed by classification of the resultant segmented regions in the image. Neither of these have lived up to their potential due to (i) the inflexibility of the first approach in detecting objects of varying scale and orientation in varying collection conditions, and (ii) the inherent semantic gap between segmentation and classification in the second approach. Existing segmentation algorithms are built upon the following two common underlying assumptions; (i) the object homogeneity with respect to some characteristic, and (ii) difference between adjacent regions. In this chapter, we propose improving the segmentation process by infusing semantic knowledge into the segmentation process by combining the problems of segmentation and classification through a wrapper framework. Li et al. have noted, "it is often difficult....to determine which regions....should be used for the final segmentation" (Li et al., 2000). The goal of the wrapper framework is to directly address this problem by integrating segmentation processing with a classification process to provide the required semantic information needed to identify the regions of interest.

The key to integrating semantics into the low-level segmentation is to utilize additional feature information of the objects of interest to provide the additional needed contextual information. The features available for classifying an object for image retrieval include texture, color, shape and structure (Safar, 2000). Since texture and color are used as low-level cues, shape and structure are the remaining features to provide additional semantic content. Using the structure of the objects of interest requires associating regions in the image with key structures of the object of interest, and then combining these semantically meaningful regions to provide the complete semantics of the desired object. These regions can either be semantically meaningful on their own, for example, head, limbs, torso for recognizing people, or they can be shape fragments that consistently occur on an object, for example using critical object boundary characteristics. A region combination algorithm then uses a shape template of the object of interest to guide the assembly of these fragments.

We believe a fundamental flaw of these structure-based approaches is requiring the identification of critical shapes and even semantically meaningful sub-shapes within the image. In images with complex natural illumination shadows and bright regions can be created which obfuscate sub-structures. In order to not require mapping of image regions with sub-structures the wrapper framework uses the *overall* shape of the object of interest as the source of semantic information and does not rely on sub-structures. The approach performs a low-level segmentation of the image, and then, irrespective of the shape of the labeled regions in this segmentation, applies an algorithm to combine regions based on knowledge of the shape of the desired object of interest. The proposed approach has been validated through successful demonstrations on a wide range of image applications including automotive occupant sensing, breast cancer detection in mamograms, and wide area disaster surveillance using aerial imagery.

2. Related Work

There is an abundance of literature on image segmentation, due to its importance in serving as the foundation for applications such as image understanding, object detection, and content-based image retrieval. Unfortunately, mechanisms to improve the results to provide strong segmentation where the objects of interest are reliably isolated from the background has continued to elude researchers. Early methods for improving segmentation involved pixel-level post-processing of the initial segmentation to further regularize the segmentation output. This approach has often relied on mathematical models such as Markov Random Fields (Bouman& Shapiro , 1994) (Kim, et al., 2000), or other models such as the harmonic oscillator model by Shi and Malik (Shi& Malik, 2000). More recently Luo and Guo proposed regularization at a region level rather than a pixel level, and they apply a Markov Random Field to combine regions using a non-purposive grouping approach that combines regions based on a defined characteristics of a 'good' segmentation rather than relying on any model of the desired object of interest (Luo & Guo, 2003).

There has also been a significant amount of research in adaptive image segmentation, where the control parameters of the underlying segmentation algorithm are modified, based on some general figures of merit of the output segmentation (Bhanu & Fonder, 2000). More recent low-level segmentation approaches proposed a continuously executing algorithm where the user stops the algorithm when the resultant segmentation appears acceptable (Tu & Zhu, 2002). These methods still relied on the assumption that the pixels belonging to the object of interest share a common set of low-level image attributes, thereby allowing the object to be extracted as a single entity. Unfortunately even relatively simple objects of interest can be composed of multiple regions of differing texture or color which would cause the object to be oversegmented and hence divided into multiple regions. The results of these approaches had limited generalized performance and demonstrated the need to devise a means for integrating additional semantic information into the segmentation process.

One of the earlier approaches to integrating segmentation with classification for infusing semantic information, involved adjusting the segmenter control parameters of the underlying segmentation algorithm based on the classification of the binary (foreground-background) segmentation (Bhanu & Peng, 2000). Unfortunately, this approach still assumed the object of interest is homogeneous in the segmentation feature, and finding it was a matter of discovering the correct control parameter via the classification results.

Integrating semantics into the segmentation have found some early success in very focused domains, such as the work by by Tu, et al. (Tu et al., 2003) which performs simultaneous human face and word segmentation from an image for a system for assisting the blind. This method directly relies on the fact that the objects of interest can be completely defined by their texture (text) or color (human face). Another approach for integrating classification information into the segmentation process was proposed by Sifakis, et al. where they provide context by providing a set of two coarse object contours, one ensured to be outside of the object of interest, and the other designed to be inside the object of interest, in a manner similar to the marker-based approaches to Watershed processing (Sifakis et al., 2002). While this method clearly provides strong context, it still is based on two key assumptions; (i) the object of interest "should be uniform and homogeneous with respect to some characteristic", and (ii) "adjacent regions should be differing significantly" (Bhanu et al., 1995). Additionally these methods operate at the pixel level and hence are computationally intensive.

One key development in image segmentation has been the developing interest in operating at the region level of images rather than at the pixel level. Some of the earliest work in region-based analysis is by Belongie et al, in their 'Blob world' system, where images were grouped into regions based on color and texture and then the user defined regions of interest based on these parameters for that to search databases (Belongie et al., 1997). Unfortunately, this approach still relies on regions being of understandable interest to the user . Li et al. have relaxed the limitation of identifiable sub-regions, by using properties such as color and texture of *all* of the regions in the image to attempt to allow image retrieval systems to bypass the segmentation process (Li et al., 2000). One drawback of this approach is that it compares not only the foreground, but also the background regions in the two images to derive the similarity, which can be particularly limiting if the object of interest is considerably smaller than the field of view of the image, or if the object of interest may be present in a wide variety of backgrounds. Jing, et al. have also recognized that region analysis is essential for effective retrieval, but their approach uses region color rather than shape for the retrieval feature (Jing et al., 2004). More recently Athanasiadis, et al. have proposed a region-based simultanous segmentation and detection scheme which relies on two low-level features defined by MPEG-7, namely homogenous texture and dominant color to perform low-level segmentaiton. Based on semantic models of objects of interest, these low-level regions are then merged together based on fuzzy relations associated with semantic information regarding the objects of interest (Athanasiadis et al., 2007).

The research highlighted to this point were based on low-level image attributes, such as color, grayscale, or texture, and clearly these failed to provide adequate semantic content for strong segmentation. Clearly, additional features are required for successful segmentation, and these can be found by refering to the body of research from content-based image retrieval where the spectrum of features available for retrieval have been defined and highlighted in Fig. 1 (Safar et al., 2000). Based on this taxonomy integrated segmentation-classification methods have been recently directed at developing structural models of the desired object and using either tree or graph theory-based techniques to assemble detected regions in the image that may correspond to sub-structures in the object of interest (Yu, et al., 2002) (Borenstein & Ullman, 2002) (Lee & Cohen, 2004) and most recently by Cours and Shi (Cour & Shi, 2007). For example, Borenstein and Ullman have developed an approach which searches for object 'fragments' within the image, where these fragments correspond to key identifiable regions of the object of interest but not necessarily semantically meaningful structures (Borenstein & Ullman, 2002). The fragments are found using a correlation detector approach. Borenstein and Malik developed a top-down mechanism to augment the traditional bottom-up segmentation algorithms similar to how the proposed wrapper framework operates. Their top-down approach first integrates the low-level regions into semantically meaningful parts using shape templates, and then further integrates these components to form the desired object of interest (Borenstein & Makik, 2006). Good results are possible with these approaches when applied to relatively well-formed images with relatively simple backgrounds, however, there are two underlying assumptions of these methods that can limit their broader applicability, namely: (i) they define a particular form for the classification problem, namely using tree or graph distances, and (ii) they build the segmentation using specific identifiable sub-regions in the images (e.g., head, arms, torso, etc. for human segmentation), and then rely on the known syntactic structure of the object of interest to assemble these components. One key drawback to these approaches is that syntactic methods can be sensitive to errors in the low-level segmentation which was concisely state by Datta: "extracting semantically meaningful coherent regions is… very challenging" (Datta et al. 2008). This was also demonstrated by Lee, et al. where the algorithm had problems recognizing human poses if the subjects wore gloves thereby hiding the skin color (Lee & Cohen, 2004).

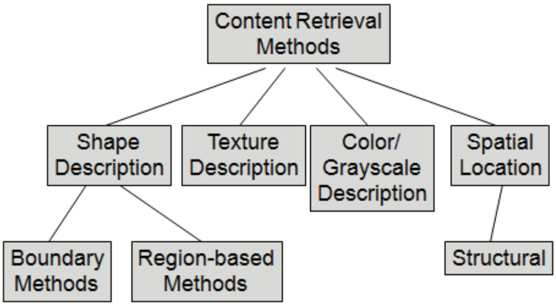
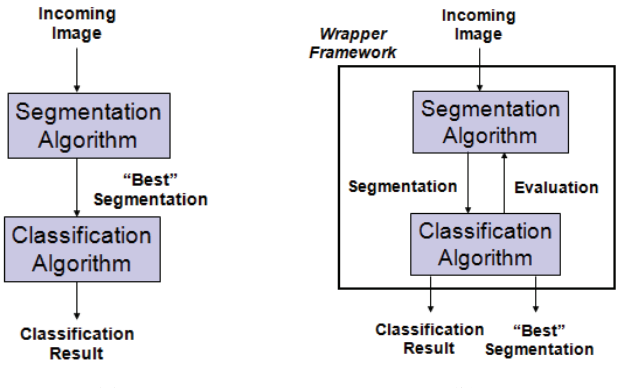


Fig. 1. Taxonomy of feature-based description techniques for image classification.

The approach taken by the wrapepr framework is to use *shape* rather than structure to provide context to the image segmentation problem. Other researchers such as Ko and Byun have added shape rather than structure to their region-based search by adding a small set of shape features to each region (Ko & Byun, 2005). The user then selects from a sample image a number of regions they consider important for query, and the system then computes a combined search distance based on the collection of regions found in the database of images. Like Ko and Byun, the proposed wrapper approach computes shape featues for each region in the image, however rather than relying on region-to-region comparisons, our wrapper approach uses the image classification in a more global scheme. The image classification is used to *assemble* regions derived from the traditional segmentation algorithms rather than simply searching for instances of individual regions. The wrapper approach has numerous advantages over the various methodes described above. One advantage is that by using shape over structure we do not require identifiable sub-structures to be segmented from the background image. Additionally, since we do not rely on particular constructs to represent the problem, such as trees and graphs, it can incorporate any classification algorithm. Also, unlike the fragment approach of Borenstien and Ullman and Borenstien and Malik the wrapper approach is a more organic approach where it builds the desired object from the regions provided in the image rather than *a priori* defined representative sub-images that are searched for in the image. This has an important consequence in that the wrapper approach can utilize any existing image segmentation algorithm to create the regions with which it then operates. Thus rather than being considered another segmentation algorithm, the wrapper approach is actually a *framework* within which any segmenter and classifier can be considered for integration to address the particular problem being addressed.

3. Wrapper Approach to Integrating Segmentation & Classification

We derive the motivation for our approach from the domain of feature selection in pattern recognition, where there are two common mechanisms for selection, namely the filter method and the wrapper method (Dash & Liu, 1997). Filter methods analyze features independently of the classifier and use some 'goodness' metric to decide which features should be kept. Wrapper methods, on the other hand, use a specific classifier, and its resultant probability of error, to select the features. Hence in the wrapper method, the feature selection algorithm is *wrapped* inside the classifier. Based on this, we propose a new paradigm for image segmentation that follows the wrapper methods of feature selection, where we *wrap* the segmentation and the classification together, and use the classifier as the metric for selecting the best segmentation. Fig. 2 compares the traditional image segmentation approaches with our proposed wrapper-based segmentation approach. The classification algorithm provides both the *semantic context* for the segmentation, as well as a *figure of merit* for the resultant segmentation, based on the classification accuracy for the pattern class under consideration.



(a) (b)

Fig. 2. Approaches to image segmentation, (a) conventional methods and (b) proposed wrapper method.

The general processing flow for the entire wrapper framework can be seen in Fig. 3. The processing is divided two distinct phases, (i) conventional (context-free) segmentation, and (ii) wrapper-based (semantic) segmentation. In the conventional segmentation phase no contextual or semantic information is used and the image is segmented based on traditional low-level homogeneity metrics, typically greyscale or color depending on the application. The second phase is the wrapper segmentation phase where the critical semantic information is integrated via the classifier.

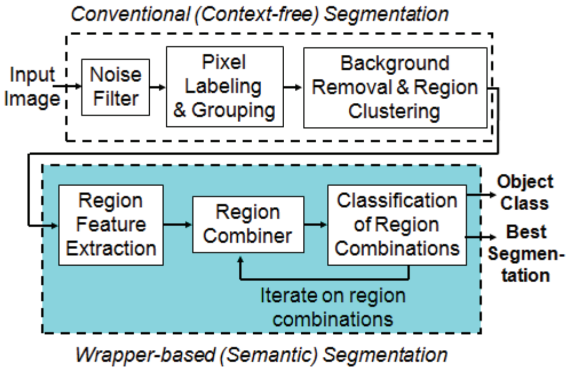

Conventional (Context-free) Segmentation
Input Image
Noise Filter
Pixel Labeling & Grouping
Background Removal & Region Clustering
Region Feature Extraction
Region Combiner
Classification of Region Combinations
Object Class
Best Segmen-tation
Iterate on region combinations
Wrapper-based (Semantic) Segmentation

Fig. 3. Processing flow for wrapper-based image segmentation.

3.1 Conventional Segmentation Processing

The conventional segmentation processing flow begins with a low-level order statistic filter such as a median filter for removing high frequency image speckle. Order statistic filters are most attractive since they are edge preserving which prevents degradation of subsequent region labeling due to edge blurring. This step is optional and depends on the quality of the images being processed by the system. The Pixel Labeling and Grouping task in Figure 3 performs conventional low-level weak segmentation. It converts the pixel values into labels, and then groups these labeled pixels into contiguous regions. The series of sub-tasks that comprise the processing of this stage are provided in Fig. 4. The first sub-task is Compute Pixel Data Parameters which is responsible for determining the parameters to be used to determine the low-level pixel labeling based on some common characteristic of the pixel values such as color, grayscale, or texture. There are many mechanisms proposed for defining the 'common characteristics', such as Expectation Maximization (EM), normalized cuts, relaxation methods, region growing methods, and split-and-merge methods, and finally DDMCMC which provides a framework for unifying many of these approaches (Belongie et al., 1997) (Tu et al., 2003). The output of all of these methods is a labeling of the incoming image into a small number of regions. We selected the EM algorithm for the region labeling algorithm was based on its relative ease of use, its flexibility, and its suitability for real-time operation. It fits a mixture of Gaussians that best matches the histogram of the grayscale values. The EM algorithm is attractive because it can easily be extended to use multiple features, such as texture depending on the application.

The Label Pixels task then uses the mixtures defined by EM to label each pixel with its appropriate mixture membership, with typical results being shown in Fig. 5 (b). This labeled image is then mode filtered to further remove isolated pixels. Other regularization algorithms such as the Markov Random Fields methods discussed in Section 2 may be used rather than the mode filter, however, the mode filter is easy to implement, imposes a relatively low processing burden, and has previously been shown to be effective (Farmer & Jain, 2005) (Rabiei et al., 2007).

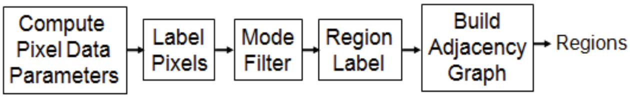


Fig. 4. Processing flow for Pixel Labeling and Grouping**.**

The third sub-task of the Pixel Labeling and Region Grouping Task is Region Label in which the pixels are grouped together into regions of common labeling based on an 8-way connected components algorithm. The Region Label sub-task then removes regions that fall below a user-defined threshold in order to minimize the total number of regions, with the particular threshold value being dependent on the application and the particular input image size. At this point the image has been completely divided into regions of low-level homogeneity (grayscale or color for the applications shown in this chapter). The final sub-task of the Pixel Labeling and Grouping is Build Adjacency Graph. In this sub-task an adjacency graph is constructed to define the relative adjacency of all of the regions in the image. This adjacency graph will play a critical role in subsequent processing for cluster detection and to limit the combinatorial complexity of the region combining algorithm within the wrapper portion of the segmentation process.

Recall from Fig. 3, the next stage in the conventional segmentation processing is the Background Removal and Blob Clustering task. Obviously, the goal of this stage is not to remove then entire background but rather to remove as much background as possible based on simple structural knowledge of images. There will still most likely be significant amounts of background connected to the object of interest, and this remaining background will be removed during region combining. The background in an image is defined as the larger regions and regions along the periphery of the image that typically are not of interest. The size of the background regions is independently defined by two characteristics, the area and the length. Thus regions of large area or large regional extent (such as roads and rivers in surveillance applications) can be ignored. Removing of the background, as shown in Fig. 5 (c), allows the algorithm to now focus on more interesting regions, in a similar manner to human perception where known background regions are ignored while more interesting or ambiguous regions are analyzed further. Once the background is removed, of clusters of regions can readily be detected using the adjacency graph. Clusters are defined by collections of regions that are adjacent to each other, as can be seen in Fig. 5 (d). The wrapper segmentation processing then analyzes each of these region clusters to determine if any objects if interest may be present. The ability of the wrapper framework to process clusters of regions, rather than all the regions in an image, is critical for performance, since the number of possible region region combinations rapidly suffers combinatorial explosion .

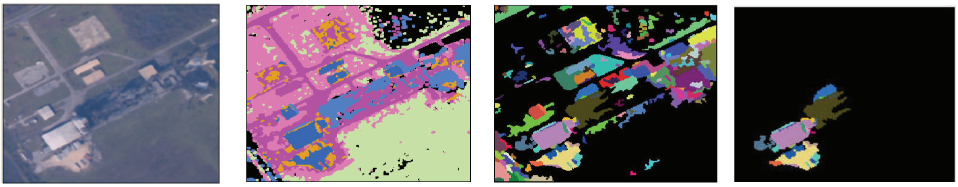

(a) (b) (c) (d)

Fig. 5. Conventional segmentation processing results: (a) example incoming surveillance image, (b) labeled image after mode filter and small region removal, (c) labeled image after background removal, and (d) cluster from lower left region in (c).

3.2 Wrapper Segmentation Processing

Up to this point the input image has been over-segmented to try to maximize the likelihood that the object of interest is not connected to a background region. In order maximize this likiehood, the image is intentionally over-segmented which means the object of interest is most likely sub-divided into multiple regions. The wrapper framework processes each cluster of regions independently and only tests combinations of the regions within each of these clusters, thereby significantly reducing the combinatorial explosion. Clusters consisting of individual region are tested first against a training database to see if any of them may match an object of interest since they require no combining. Then the remaining more complex clusters are processed, where a variety of combinations of region are attempted to see if any of these combinations may match any objects in the training database.

3.2.1 Region Feature Extraction

Recall from Fig. 3, the first task in the wrapper processing is the feature extraction for each region. Fig. 1 demonstrated there are four possible feature spaces for image retrival and classification. The wrapper framework incorporates shape as its semantically rich feature. Shape may be defined by either region or boundary descriptions (Veltkamp & Hagedorn, 2001). While either method can be used to capture the shape of the regions that have been defined as comprising our image, the research to date with the wrapper framework has employed moments to describe these shapes. The geometric moments of an image are defined by (Teague, 1980):

$$M_{lk} = \sum_{j=0}^{N} \sum_{i=0}^{M} I(i,j) i^l j^k , \quad (1)$$

where *I(i,j)* is the value of the image at pixel *(i,j)* and *N* and *M* are the numbers of rows and columns in the image, respectively.

Computing the moments features on every region combination would be computationally prohibitive since many combinations will be generated for every region as will be shown in Section 3.2.2. Fortunately, due to the non-overlapping nature of the regions that comprise the image labeling, the basic geometric moment features can be calculated for each region prior to the subsequent region combining and classification stages of processing. Then during the region combining processing, the moments of the combined regions is simply the sum of the moments of the individual regions, which implies all pixel-level processing need only be performed once, and all subsequent processing is performed at the region level.

This speedup mechanism is related to the concept of Borel sets and the calculation of measures on these sets. A value μ is a measure if it assigns a non-negative number to each subset, which can be seen to be true from Equation (1) since *I(i,j), i,* and *j* are never negative. One important property of these measures is: "if a set is decomposed into a countable number of disjoint Borel sets then the total measure of the pieces equals the measure of the whole", which can be mathematically stated as (Falconer, 2004):

$$\mu\left(\bigcup_{i=0}^{\infty} A_i\right) = \sum_{i=0}^{\infty} \mu\left(A_i\right), \tag{2}$$

where A_i is the ith subset and μ is the measure. This fact was originally exploited by Spiliotis and Mertzios, where the computation is decomposed into a summation over a set of non-overlapping rectangular homogenous blocks (Spiliotis & Mertzios, 1998). We abstract this approach and, rather than decomposing the image into non-overlapping rectangles, we propose a more natural decomposition of the image into the collection of regions defined by the image labeling. Using the notion of disjoint Borel sets allows representing the image as:

$$I(i,j) = \bigcup_{k=0}^{K} I^{(k)}(i,j)' \tag{3}$$

where $I^{(k)}(i,j)$ is the portion of the image corresponding to region *k*. From this we can now rewrite the geometric moment equation as:

$$M_{lk} = \sum_{j=0}^{N}\sum_{i=0}^{M}\left(\bigcup_{k=0}^{K} I^{(k)}(i,j)\right)\cdot i^l j^k\ . \tag{4}$$

From Equation (2) we can now replace the measure over the union of subsets as a summation over the individual measures of each of the subsets and obtain:

$$M_{lm} = \sum_{k=0}^{K}\left[\sum_{j=1}^{Nrows}\sum_{i=1}^{Ncols} I^{(k)}(i,j)\cdot i^l \cdot j^m\right]\ . \tag{5}$$

The term in the square brackets, $\sum_{j=1}^{Nrows}\sum_{i=1}^{Ncols} I^{(k)}(i,j)\cdot i^l \cdot j^m$, is the moment calculation for the moment of order *(l+m)* for the *k*th region of the image, $I^{(k)}(i,j)$, and we define this moment to be M_{lm}^{k} . We can then rewrite Equation (5) according to:

$$M_{lk} = \sum_{k=0}^{K} M_{mn}^{k}\ , \tag{6}$$

where we have reversed the order of the summations, and M_{mn}^{k} is the moment of order *(m+n)* corresponding to the *k*th region. Thus, the geometric moments for the entire image are merely a sum of the geometric moments computed for each region.

Now we can pre-compute the moments for each region, which allows us to add the feature vectors from each region together to compute the moments for any region combination. The ability to pre-compute features can provide a considerable benefit, since, as Yoshitaka and Ichikawa state: "[feature extraction] processing is one of the most time consuming parts in content-based retrieval. Improving the [feature extraction] processing therefore improves the overall performance... (Yoshitake & Ichikawa, 1998)." We will see in Section 3.2.3 that from this point in the processing all operations will be performed on regions rather than pixels which greatly reduces the overall processing complexity of the wrapper framework.

There are many forms of image moments that can be used for image classifcation, including central, scale invariant, rotationally invariant, legendre, zernicke, et cetera (Teague, 1980). For most image object detection, classification, and retrieval applications central moments are always required since they provide translational invariance within the image.

3.2.2 Region Combining

For Region Combining processing an algorithm is required which will combine subsets of the regions in the image together while assuming a particular object class is present in the image. The wrapper framework operates by assuming a pattern class *C* to be the true class, and computes the classification distance of candidate segmentations to that class. The specifics of the classification algorithm upon which the wrapper relies are provided in Section 3.3. The region combining task is performed for every class, *C*, and at the completion of the processing of all the candidate classes, the class *C* that provides the highest membership probability, $P(\{X_k\}|C)$, is selected, where the set $\{X_k\}$ defines the subset of regions that comprise the best segmentation for iteration *k* of the algorithm. Likewise, the set of regions $\{X_k\}$ that produces this best classification probability corresponds the the best strong segmentation of the image.

It is this conditioning of the probability on a particular object class that provides the semantic content to direct the segmentation processs. The classification distance is then used as a quality metric for the segmentation that corresponds to that region combination. If the probabilities of membership, $P(\{X_k\}|C)$, for every class, *C* are too low, then the image is 'rejected', which implies the object of interest is not in the image.

The selection of these regions which will be combined into the final segmentation is analogous to feature selection in pattern recognition. There are a number of feature selection methods that can be adapted for region selection. The taxonomy for feature selection methods, shown in Fig. 6, divides these methods into three primary categories: (i) complete, (ii) heuristic, and (iii) random (Dash & Liu, 1997). These methods are the results of an extraordinary amount of research in the pattern recognition community, and are backed by both considerable empirical results as well as strong theoretical underpinnings. There is still no consensus as to which method is the best, since there is such a strong dependence of the performance of the algorithm on the data sets being analyzed (Dash & Liu, 1997). We have developed an approach in each of the major categories: an exhaustive search in the complete category, a Genetic Algorithm in the random category, and a forward sequential search in the heuristic category.

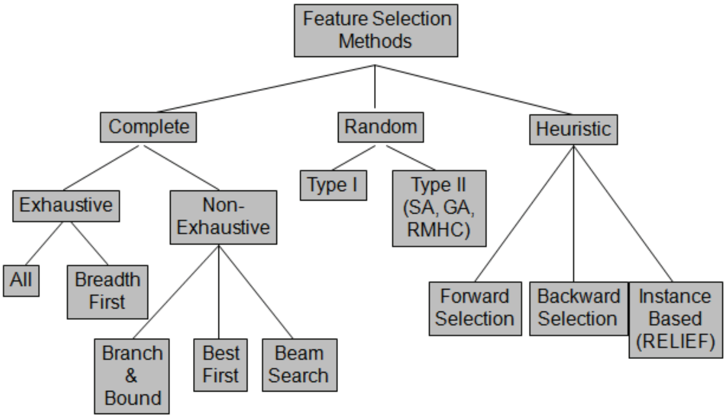


Fig. 6. Taxonomy of available feature selection methods.

The simplest algorithm is an exhaustive search through which all the possible region combinations are created and tested to find the best combination, which is feasible only when there are a limited number of regions in a cluster. For an exhaustive search, if there are *N* regions then the number of region combinations can rapidly become intractable since we must test the following number of combinations:

$$N_{combinations} = \sum_{k=1}^{N} \binom{N}{k}, \text{ where } \binom{N}{k} = \frac{n!}{k!(n-k)!}. \tag{7}$$

The summation over the number of regions is due to the fact that we do not know how many regions will be required to produce the best combination of blobs, and therefore every combination of every possible number of regions must be tested. For each of these possible number of regions, *k*, there are *N choose k* possible ways to select these regions from the complete set of *N* regions.

For all of the search methods, particlarly for the exhaustive search, the number of possible region combinations can quickly become intractible, where for clusters consisting of as few as twenty regions a brute force search of every possible combination would require roughly one million combinations, and if the number of regions only increased modestly to twenty-five, the number of combinations would exceed 33 million. Fortunately, the total number of possible region combinations that must be explored is considerably less than this value which is actually an upper limit based on complete connectivity of all regions in the image. In reality the regions in an image are only *locally* connected which can easily be visualized using an adjacency graph. For region combiniing, only region combinations which satisfy an adjacency constraint must be tested. Fig. 7 (b) shows the adjacency graph for the cluster on Fig. 7 (a), and originally shown in Fig. 5. Here there are 22 regions in the cluster which for an exhasutive search of all possible combinations of all regions would result in 4.2 million combinations, however, the relatively sparse connectivity of the adjacency graph allows the sequential search algorithm to complete the analysis of this cluster with testing only 200 region combinations and correctly extracting the building of interest.

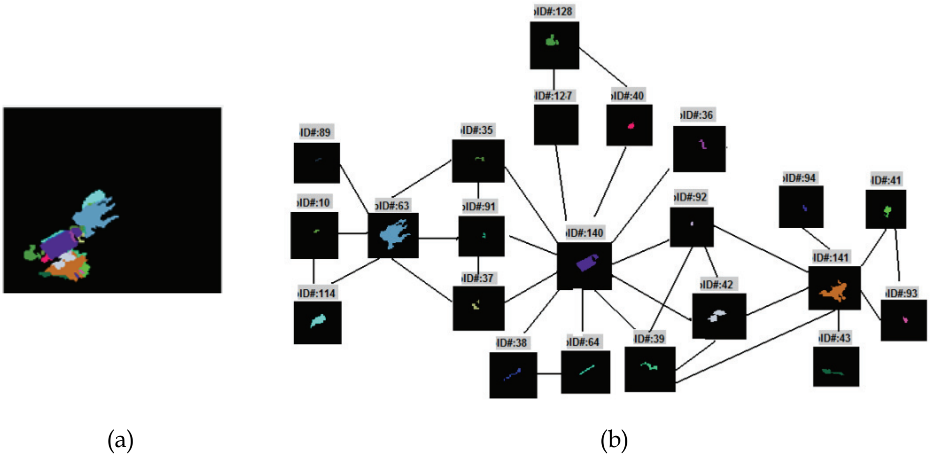


Fig. 7. Graph representation of cluster from Fig. 5 (d): (a) image of cluster, and (b) graph representation of adjacency of cluster regions**.**

The adjacency of the regions creates a local connectivity in the graph, and it is this median local connectivity of the graph which determines the number of possible region combinations that must be explored. There are *n* posible selections for the first region to be selected, but then rather than *(n-1)* options for the second region, there are only *m* which is equal to the local connectivity of the first region. The number of possible regions for the third region then varies as *(m-1)*, etc. For the fourth region, the number of possible regions varies from *2*(m-1)* when the first three regions form a chain to *(m-2)* when they are all in a tight cluster adjacent to the first region. While the actual number of possible region selections cannot be calculated in closed form, the average number of region combinations which must be tested for each pass of the algorithm is:

$$\binom{N}{k}_{RAG} = n\left\langle \begin{array}{l} (m)(m-1)(m-1)\ldots(m-1), \\ (m)(m-1)(m-1)\ldots(m-k-1), \\ \ldots, \\ (m)(m-1)(m-2)\ldots(m-k) \end{array} \right\rangle, \tag{8}$$

Which is clearly significantly smaller than that estimated by *N-choose-k.*

Genetic Algorithm-based Region Combining

Genetic algorithms are a natural candidate for wrapper-based segmentation, since GAs can "successfully deal with combinatorial problems" (Kim, et al., 2000). Three key design issues must be addressed when using GAs: (i) the representation of the problem into a chromosome, (ii) the definition of a fitness function, and (iii) the the selection of cross-over and mutation strategies (Goldberg, 1989). Since we are using the GA for region selection, only the fact that a region is to be used in the segmentation must be encoded, which greatly simplifies the use of a GA for the wrapper-based segmenter versus other low-level segmentation approaches such as that described in (Bhandarkar & Zhang, 1999). The resultant encoding is a simple binary representation where the gene is set to one if the region is used, and set to zero if the region is not used.

The output from every iteration *i* of the GA-based region combiner algorithm is the set of regions for each member *k* of the population, $\{X_{ik}\}$ and the associated probability of correct classification for that member of the population, $P(\{X_{ik}\} \mid C)$ conditioned on the given pattern class, *C*.

The genetic algorithm uses this probability of correct classification as the fitness function for evaluating the population members for possible reproduction using a fitness-proportional selection scheme (Goldberg, 1989). The wrapper uses the fourth or second power of the probability of correct classification as the fitness proportionality measure, *Fitness(k)*, according to:

$$Fitness(k) = \frac{P(\{X_{ik}\} \mid C)^4}{\sum_{j=1}^{N} P(\{X_{ij}\} \mid C)^4} \text{ or } Fitness(k) = \frac{P(\{X_{ik}\} \mid C)^2}{\sum_{j=1}^{N} P(\{X_{ij}\} \mid C)^2}, \tag{9}$$

where *N* is the size of the population. The fourth power is employed if the variance of the distances is below a threshold, and the second power is employed as the population becomes more varied, to slightly reduce the dominating effects of a single highly fit parent. Not raising the classification distances to a higher power resulted in inadequate variation in the proportionality factors, thus leading to a nearly random selection scheme.

Pairs of parents are selected for mating using the roulette wheel selection mechanism, where each set of parents then has a probability $P_{crossover}$ of executing a cross-over to generate the children; otherwise the parents proceed intact, where we set $P_{crossover} = 0.85$. For the applications for which the wrapper has been employed, the region-labeling algorithm generates on average 20 regions, resulting in the chromosome having only 20 individual genes. Due to this relatively short sequence, a simple single-point cross-over scheme for the genetic operator has proven adequate. For each mating pair of parents chosen for cross-over, the cross-over point is randomly selected.

After the children are produced via the cross-over processing, the children experience mutation with a probability of any gene mutating being $P_{mutate} = 0.05$. Lastly, an elitist selection strategy is employed where the fittest 10% of the population prior to mating (this corresponds to the parents with highest fitness) are retained in the population (Back, 1996). In order to ensure a diverse population of segmentation candidates is maintained two additional schemes are used to increase the diversity of the population. In the first scheme, if the variance of the fitness of the population falls below a threshold (i.e. the members of the population are becoming too similar), an additional mutation event is applied on the entire population, where this time $P_{mutate} = 0.25$. In the second scheme, if the fitness of the best member of the population has not improved in the last *N* iterations, where *N=25*, an additional mutation event is applied on the entire population with $P_{mutate} = 0.25$.

Sequential Search-based Region Combining

The sequential feature selection methods can be implemented in either a forward selection mode or a backward selection mode as can be seen from Fig. 6 under the heuristic methods. The forward selection mode begins with the empty set (an empty image) and then adds regions until the classification accuracy is maximized. The backward selection mode, on the other hand, begins with the complete image and removes regions until the classification accuracy is maximized. For the wrapper segmentation framework the forward selection is employed since the objects of interest are visually a fraction of the entire image. The forward selection algorithm that has been implemented is called the *plus-L-minus-R* algorithm, which has been identified as one of the more powerful heuristic methods for feature selection (Kudo & Sklansky, 2000). It begins with an initial set of regions, $\{X_0\}$ and then adds up to *L* regions per iteration and then after adding these *L* regions, tries region combinations where it subtracts up to *R* regions. The complete addition and then removal of regions is one iteration of the algorithm. The details of the algorithm are shown in Table 1. For the *plus-L-minus-R* implementation of the forward sequential search algorithm, the selection of *L* and *R* depends on the specific application and characteristics of the objects of interest within the images, for the airbag application where there were many regions that comprised the image we employed *L=5* and *R=3*, while for the tumor and the detection applications we employed *L=3* and *R=2*. The intitial number of regions to use is also an open parameter, which was five for the airbag application and two for the other two applications since the objects being processed were much smaller than the size of the image.

1) For a given class *C*, create an initial set of regions $X_0 = \phi$, the empty set.
2) Region Addition: At each stage *k* in the processing, test each region in the set of unselected regions and add region x_l if $P(\{X_k\} + x_l \mid C) \geq P(\{X_k\} \mid C)$, where $P(\{X_k\} \mid C)$ is the classification accuracy for the region set $\{X_k\}$, given class C. The output of this stage is a new subset of regions $\{X_{k+1}\} = \{\{X_k\}, x_{k+1}^{(1)}, x_{k+1}^{(2)} \ldots x_{k+1}^{(L)}\}$, where $x_{k+1}^{(i)}$ is the region with the i$^{\text{th}}$ best improvement in classification accuracy up to *L* regions.
3) Region Removal: Test each region in the current selected region set, $\{X_{k+1}\}$, and remove each region x_r from the set if $P(\{X_{k+1}\} - x_r \mid C) \geq P(\{X_k\} \mid C)$, where $P(\{X_k\} \mid C)$ is the classification accuracy for the region set $\{X_k\}$given class *C*. Continue testing and removing regions until all the regions in the current subset $\{X_k\}$ are tested, or until *R* regions have been removed.
4) Record $P(\{X_k\} \mid C)$, and the corresponding subset of regions$\{X_k\}$, and return to step (2) unless the last region has been processed.

Table 1. *Plus-L- minus-R* forward sequential search algorithm for region combining.

3.2.3 Classification

Every possible combination of regions must be classified based on the class of interest to determine the goodness of the segmentation, however, prior to each classification, the features for the region combination must be computed. Recall to this point only the geometric moments have been employed to allow the features for each region combination to be quickly computed by adding or subtracting the moments for each region included in the combination. Recall from Equation (6), the geometric moments feature vector for a region combination is simply the sum of the feature vectors for every region that comprises the combination. This raw geometric moment vector is then converted to the desired invariant moments prior to classification. As a minimum the central moments must be used to make the object search translation invariant across the images and are computed by (Teague, 1980):

$$\mu_{mn} = \sum_{r=0}^{m} \sum_{s=0}^{n} \binom{j}{r}\binom{k}{s} \left(\bar{i}\right)^{j-r} \left(\bar{j}\right)^{k-s} M_{rs}, \tag{10}$$

where

$$\bar{i} = \sum_{j=1}^{Nrows} \sum_{i=1}^{Ncols} I(i,j) \cdot i \text{ and } \bar{j} = \sum_{j=1}^{Nrows} \sum_{i=1}^{Ncols} I(i,j) \cdot j \tag{11}$$

Depending on the desires of the user, the features of the object can also be made central scale invariant, rotation invariant, affine invariant and even projection (perspective) invariant (Suk & Flusser, 2004). More complex invariances, however, require more complex processing which impacts the throughput of the system. Also the more invariant the measures, the less discriminating the moments features can often be (Suk & Flusser, 2004). For the airbag suppression and the tumor applications, the sizes of the tumors were critical information so only central (translational invariant) moments were used. However, for the aerial surveillance application, the range to the objects of interest varied, and hence their size varied, which required central scale invariant moments. It is also important to note that not all objects require all invariances, for example when searching for bears, buildings, etc. as there is not a need to be fully rotationally invariant. Also the author has found that rotational invariance can be accomplished more cost effectively by adding a rotation generation function when creating the training database to create rotated examples of the training samples.

One key decision that must be made when employing moments is to decide the order of the moments being retained. The applications to be highlighted in Section 4 have varied from only fifth order for an aerial surviellance application designed to detect buildings to up to twenty-fifth order for detecting occupants in an automotive airbag application. The tumor application was in the middle of this range with tenth order.

While it is possible to use any of a number of possible classifiers in our wrapper method that provides a real-valued measure of the classification accuracy (or inversely classification distance), the k-nearest neighbor classifier has been used for the following reasons: (i) ease of implementation, (ii) non-parametric nature, (iii) demonstrated performance over a broad class of problems, and (iv) asymptotic convergence to the Bayes error rate (Jain et al., 2000) (Duda et al., 2000). The best results typically occur when the nearest neighbor classifier (*k=1)* is used. Additionally all of the moment values are used in the classification process (i.e. no feature selection is employed), since the complete set of features appears to be required to provide a good representation of the object shape. One last decision to be made for classification regards whether the features are normalized or not. The effects of normalization also varied with application, where we found that for the airbag suppression, un-normalized moments worked best, while the tumor and surveillance applications performed best when the features were normalized. This may be due to the fact that for the airbag suppression, the shapes were more complex and un-normalized features more fully captured and preserved the shape information that is providing the semantic information to the segmentation process.

4. Results

The wrapper framework has been demonstrated on three distinct applications, a vision system for automotive safety, an MRI analysis tool for automated breast cancer detection, and an aerial surveillance application. There has been considerable attention paid to developing 'smart' airbags that can determine not only if they should be deployed in a crash event, but also with what force they should be deployed. In May 2001 the U.S National Highway Transportation and Safety Administration (NHTSA) defined the Federal Motor Vehicle Safety Standard (FMVSS) 208 that mandated automatic airbag suppression when an infant is in the passenger seat. The detection of an infant in the seat defines a 2-class recognition problem where the classes are: (i) infant, and (ii) adults. An example of an input adult image, the preliminary segmentation after background removal and the resultant labeled image are provided in Fig. 8 (Farmer & Jain, 2005). For this particular application the background removal occured prior to low-level labeling since there was contextual information available regarding the knowledge of the empty vehicle which facilitated the removal of significant amounts of background information except the occupant and the seat. This system was tested using both the heuristic *plus-L-minus-R* algorithm and the Genetic Algorithm. For both algorithms, the central moments are converted to central-Legendre moment of order *(n+m)* for each region combination using (Teague, 1980):

$$L_{mn} = \frac{(2m+1)(2n+1)}{4} \sum_{l=0}^{m} \sum_{k=0}^{n} C_{ml} C_{nk} \mu_{lk}, \tag{12}$$

where C_{ml} are the coefficients defined by the Legendre polynomial generating function and μ_{lk} is the central moment of order *(l+k)*. Note for this application, the moments cannot be scale invariant since the size of the object is a critical factor in determining its class.

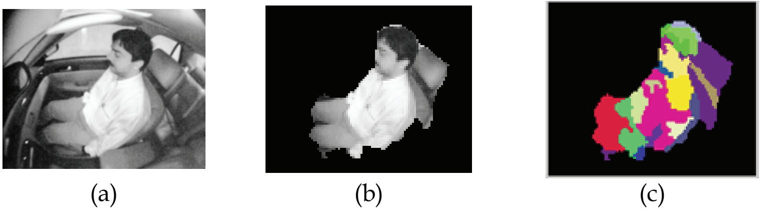

(a) (b) (c)

Fig. 8. Preliminary segmentation results for adult occupant, (a) adult image, (b) preliminary segmented adult, and (c) region labeled adult image (Farmer & Jain, 2005).

The classification results are provided in the confusion matrix in Table 2 for the *plus-L-minus-R* and in Table 3 for the genetic algorithm. The overall accuracy of the system provided by the *plus-L-minus-R* algorithm is $P_{correct}$(overall) is 91% with $P_{correct}$(infant) being 98.8% and $P_{correct}$(adult) being only 53.2% and with examples of correct segmentations shown in . Notice without shape information, accurate segmentation of these objects from the images would have been impossible since there is no low-level homogeniety constraint to differentiatethe object of interest from the background. Unfortunately, for the *plus-L-minus-R* algorithm the adult results are disappointing due to the high variability of the test images, which can be seen from an example incorrect segmentation in , where the occupant was moving forward and hence was not in the standard seating position. The *plus-L-minus-R* had trouble converging to the right answer on these conditions, but the testing of the GA on similar dataset improved the adult performance at a slight cost to the infant classification accuracy, as shown by its performance highlighted in Table 3. In summary $P_{correct}$(overall) is 88% with $P_{correct}$(infant) being 89.2% and $P_{correct}$(adult) being a much improved 83.4% (Farmer & Shugars, 2006). This performance improvement is due to the fact that the region selection space is a complicated search space with many local optima, and genetic algorithms have been shown to be more effective in these spaces (Kudo & Sklansky, 2000).

	True Infant	**True Adult**
Classified as Infant	1631	19
Classified as Adult	166	189

Table 2. Confusion matrix for the two-class suppression problem using *plus-L-minus-R* (Farmer & Jain, 2005).

	True Infant	**True Adult**
Classified as Infant	793	34
Classified as Adult	96	171

Table 3. Confusion matrix for the two-class suppression problem using a Genetic Algorithm (Farmer & Shugars, 2006).

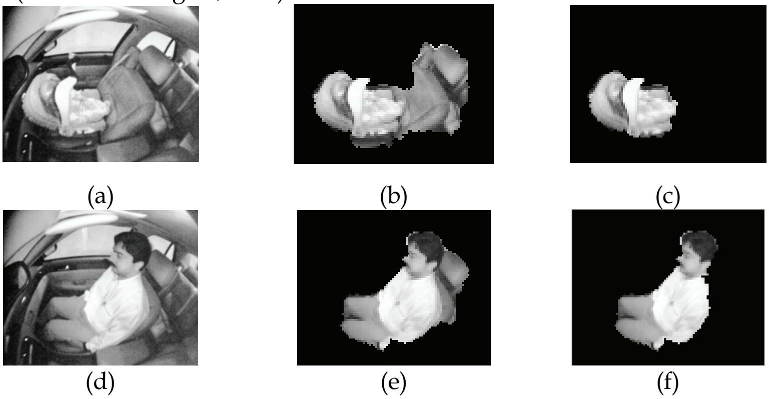


Fig. 9. Segmentation of an occupant images: (a) infant image, (b) preliminary infant segmentation, (c) final wrapper-based infant segmentation, (d) adult image, (e) preliminary adult segmentation, and (f) final wrapper-based adult segmentation (Farmer & Jain, 2005).

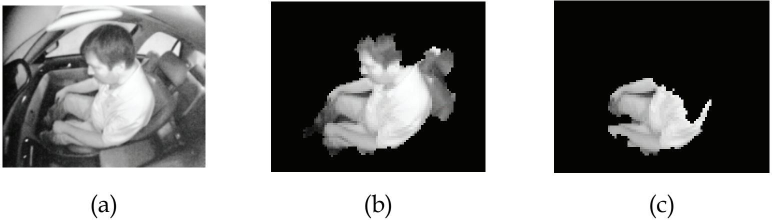


Fig. 10. Incorrect segmentation of an adult image: (a) adult image, (b) preliminary segmentation, and (c) final wrapper-based segmentation (Farmer & Jain, 2005).

The wrapper framework has also been applied to breast tumor detection (Rabei et al. 2007). Dynamic contrast-enhanced magnetic resonance imaging (DCE-MRI) has been identified as a valuable complementary technique for breast imaging. Unfortunately, while these multi-temporal image sequences provide new information, integrating and evaluating the much wider range of information is challenging task for human observers. The wrapper framework was used to direct the segmentation based on the underlying shape and temporal characteristics of the object of interest (Rabei et al. 2007). Examination of temporal kinetic patterns as measured for small regions of interest is a common method for characterizing lesion masses. These dynamic parameters cannot be computed for each pixel in every breast slice, due to processing complexity. Traditionally, these measures are computed by sampling pixels within a grid superimposed on the image, which can reduce sensitivity to detection of small tumors since much of the tissue within the grid cell is normal. The wrapper approach utilized the regions selected by the region combining and computed the dynamic parameters for each of these groupings, as can be seen in Fig. 11 (Rabei et al. 2007). These values are then used with the region shape information for tumor detection. The overall accuracy of the system is roughly 92% with the false positive diagnoses rate for normal patients as having either malignant or benign tumors of 4.5%, and a misdiagnosis rate for normal patients as either having malignant, benign, or suspicious growths of 7.5% as shown in Table 4 (Rabei et al. 2007).

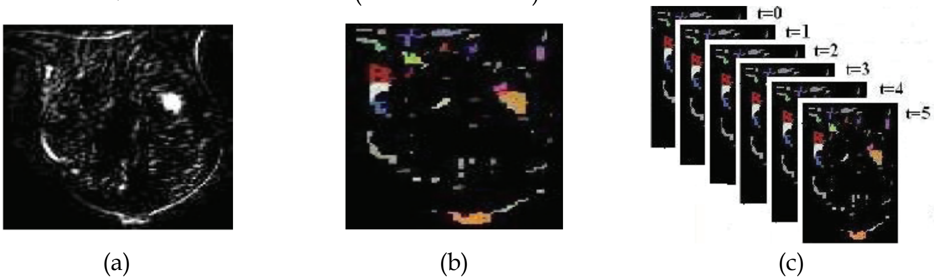


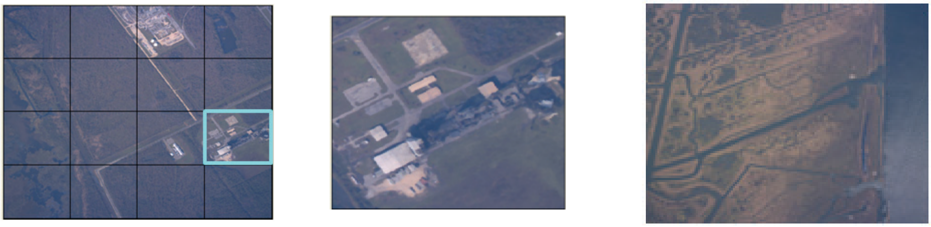
Fig. 11. Intermediate breast tumor processing results, (a) input image, (b) labled image with background removed showing regions for combination, and (c) sequence of region combined images for dynamic analysis (Rabei et al. 2007).

Thus far, the wrapper has been demonstrated on a two-class problem for the airbag control system and a four-class problem for the tumor recognition system. The last application for which the wrapper framework has been applied is an aerial surveillance application addressing wide-area surveillance of disaster areas such as during hurricane Katrina, with an example wide-area image shown in Fig. 12. This is an object detection problem, which can be considered a single-class problem. The goal of the system is to detect manmade structures in wide-area imagery to ease the workload of image analysts who are searching for possibly stranded people in very remote rural areas. In this application, due to the immense sizes of the images, the first step in processing is a mosaicing process that divides the incoming image into a *4x4* grid, and each mosaic in the grid is then processed in parallel to reduce the processing time allowing it to benefit from multi-core architectures.

	True Normal	True Benign	True Suspicious	True Malignant
True Normal	925	23	31	21
True Benign	4	205	12	4
True Suspicious	7	19	343	6
True Malignant	3	2	4	91

Table 4. Results of wrapper framework applied to breast tumor detection (Rabei et al. 2007).

One other difference in processing is that the mode filtering step shown in Fig. 4 is bypassed since the objects tend to be relatively small in these massive images and the mode filtering distorted the shape characteristics of the objects of interest. The detection results for the wrapper on the image shown in Fig. 12 (b) are provided in Table 5, where the detection results are quite respectable. The quality of the segmentations and detections can be seen beginning with typical initial clusters and the resultant detections are provided in Fig. 13 and Fig. 14. These figures show the detected clusters in (a), the resultant combinations of regions that define the segmentations in (b), the region in the color image showing the object detection in (c), and the training sample that was used for the detection in (d).



(a) (b) (c)

Fig. 12. Surveillance images, (a) original wide-area image with buildings ,(b) zoom of highlighted region, and (c) image with no buildings.

	Combinations Detected as Objects	Combinations Detected as non-Objects
Actual Objects	5	0
Actual Clusters not Containing Object	2	7

Table 5. Wrapper Results on Image in Fig. 12(b).

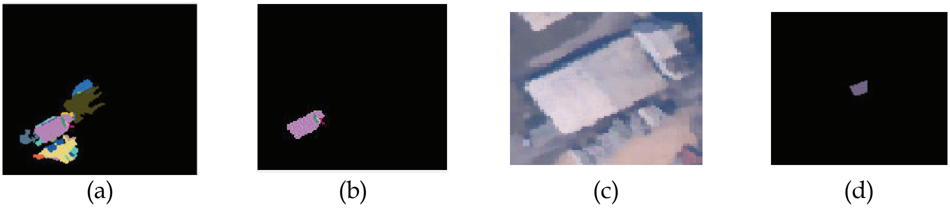


Fig. 13. Best match for cluster: (a) Original blob cluster, (b) final blob combination, (c) image region and (d) best training sample.

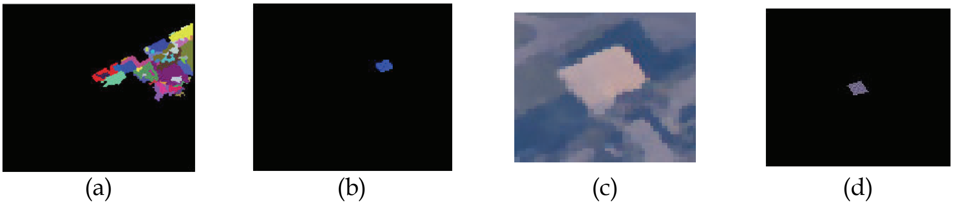


Fig. 14. Best match for cluster: (a) Original blob cluster, (b) final blob combination, (c) image region and (d) best training sample.

We also processed the entire image in Fig. 12 (a) and registered detections of buildings simply within each mosaic, so for each image there would be a total of sixteen possible detections. For analysis of this wide-area problem we quantify system performance in terms of recall and precision which are defined as:

$$\begin{aligned} \text{Recall} &= \left(\frac{\text{correct detections}}{\text{correct detections} + \text{missed detections}} \right) \\ \text{Precision} &= \left(\frac{\text{correct detections}}{\text{correct detections} + \text{false alarms}} \right) \end{aligned} \tag{13}$$

Unfortunately, the basic performance was not very impressive, with seven regions falsely having buildings detected, three with positive detections, and one missed detection, resulting in a Recall = .75 and Precision = 0.3.

There are two characteristics of the image segments where the wrapper framework had false detections. The first is where there are manmade entities such as parking lots and intersections of multiple roadways, which since the goal of the application is to detect manmade structures can only partially be considered false detections. The second cause of false detections occurs when the initial segmentation is severely over-segmented (we term this *hyper-segmentation*) which occurs when the image of interest has strong texture characteristics as shown in Fig. 15. For example, in these hyper-segmented regions there were on the order of 10^{100} to 10^{200} possible region combinations which are extraordinary. This high region count, and hence high number of region combinations explains the high false detection rate, since Borenstein and Mailk (Borenstein & Malik, 2006) pointed out that the segmented regions cannot be too small or else any object is possible to create. In this application, the hyper-segmentation can be avoided by implementing a texture and color-based low-level segmentation, which is beyond the scope of this paper. These conditions are easy to detect since they result in significant numbers of regions (typically over 400-600 where the normal number is less than 200). Thus the wrapper framework can also provide quality feedback regarding the initial segmentation, and redirect either the parameter selection or in this case the actual low-level feature set to use for labeling. When the hyper-segmented regions were removed from the calculation, the results are: Recall = 0.75, Precision = 0.6. This performance is more reasonable for a system that is designed to reduce operator workload.

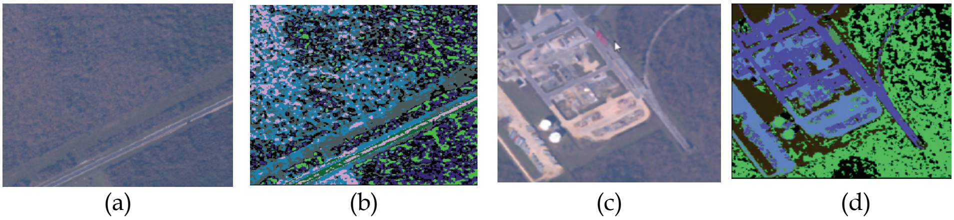


Fig. 15. Problematic low-level segmentations, (a) hyper-segmentation due to significant texture from Mosaic (2, 4) from the Fig. 12 (a), showing texture, (b) the resultant color-based labelling using EM, (c) original image for showing under-segmentation, and (d) under-segmentation due to similarity in color of objects and the background roadway.

5. Conclusion

This chapter proposes an alternative paradigm for object segmentation that follows the wrapper methods of feature selection, where in this case the segmentation and the classification are *wrapped* together, and the classifier provides the metric for selecting the best segmentation. Rather than considering this method as yet another segmentation algorithm, the wrapper method is actually an alternative image segmentation framework, within which existing image segmentation algorithms may be executed. Unlike previous work in image segmentation, the proposed system makes no assumptions regarding the homogeneity of the object of interest. It attempts to bridge the semantic gap in image segmentation by considering the shape of the desired object, rather than relying on lower level features such as color or texture. The approach has been implemented with two different region selection algorithms, the heuristic *Plus-L Minus-R* algorithm and a genetic algorithm, while for small region combinations an exhaustive search is applied. The wrapper framework has been demonstrated on three very different applications, a vision-based automotive occupant sensing system, a breast tumor recognition system using MRI, and an aerial surveillance application for disaster assessment. In all cases, the resultant segmentations were often of high quality and would have been impossible without the semantic information provided by the shape of the object of interest. In the surveillance application, the results were more dependent on the low-level segmentation, caused by hyper-segmentation due to high texture images. Future work will address integrating more powerful low-level segmenters other than the EM algorithm. Current research work is directed at developing a complete content-based image query system using the wrapper framework to support the search through an image database of user defined shapes of interest. Shape-based Content-based Image Retrieval (CBIR) is currently a very active area of research and the wrapper framework may provide an effective means for integrating shape information into the search process.

6. References

Athanasiadis, T., Mylonas, P. Avrithis, Y., & Kollias, A. (2007). "Semantic image segmentation and image labelling", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 3, 2007.

Back, T. (1996). *Evolutionary Algorithms in Theory and Practice*, Oxford Press.

Belongie, S.; Carson, C.; Greenspan, H. & Malik, J. (1997). "Color and texture-based image segmentation using EM and its application to content-based image retrieval," *Proc. International Conference on Computer Vision (ICCV)*, pp. 675-683, 1997.

Bhandarkar, S. M. & Zhang, H. (1999). "Image segmentation using evolutionary computation", *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 1, 1999.

Bhanu, B.; Lee, S. & Das, S. (1995). "Adaptive image segmentation using genetic and hybrid search methods", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 4, pp. 1268-1290, Oct. 1995.

Bhanu, B. & Fonder, S. (2000). "Learning-based interactive image segmentation", *Proc. of IEEE International Conference on Pattern Recognition*, pp. 1299-1302, 2000.

Bhanu, B. & Peng, J. (2000). "Adaptive integrated image recognition and segmentation", *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 30, no. 4, pp. 427-441, Nov. 2000.

Borenstein, E. & Ullman, S. (2002). "Class-Specific, Top-Down Segmentation", In: *Lecture Notes in Computer Science*, vol. 2351, Springer-Verlag, pp. 109-122, 2002.

Borenstein, E. & Malik, J. (2006), "Shape Guided Object Segmentation", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp**.** 969 – 976, 2006.

Bouman, C.A. & Shapiro, M. (1994). "A multi-scale random field model for bayesian image segmentation," *IEEE Transactions on Image Processing*, vol. 3, no. 2, pp. 162-177, 1994.

Cour, T. & Shi, J. (2007). "Recognizing objects by piecing together the segmentation puzzle", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

Dash, M. & Liu, H. (1997). "Feature selection for classification", *Intelligent Data Analysis*, vol. 1, pp. 131-156, 1997.

Datta, R.; Joshi, D.; Limm, J. & Wang, J.Z. (2008). "Image retrieval: Ideas, influences, and trends of the new age", *ACM Computing Surveys,* vol. 40, no. 2, pp. 5:1-5:60, 2008.

Duda, R.O.; Hart, P.E. & Stork, D.G. (2000). *Pattern Classification*, 2nd edition, Wiley, New York, 2000.

Duin, R.P.W. (1996). "A note on comparing classifiers," *Pattern Recognition Letters*, vol. 17, no. 5, pp. 529-536, 1996.

Falconer, K. (2004). *Fractal Geometry Mathematical Foundations and Applications*, 2nd edition, Wiley, New York, 2004.

Farmer, M. & Jain, A. (2005). "Wrapped based approach to image segmentation and classification", *IEEE Transactions in Image Processing*, vol. 14 no.12, pp. 2060-2072, 2005.

Farmer, M. & Shugars, D. (2006). "Application of Genetic Algorithms for Wrapper-based Image Segmentation and Classification", *Proc. of IEEE World Congress on Evolutionary Computation*, pp. XXX, 2006.

Farmer, M. (2008). "Application of the wrapper framework for object detection", *Proc. of the IEEE International Conference on Pattern Recognition*, pp. XXX, Tampa, Florida, 2008.

Flusser, J. & Suk, T. (1999). “On the calculation of image moments”, *Research Report #1946, Institute for Information Theory and Automation, Academy of Sciences of the Czech Republic*, 1999.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

Jain, A.K., Duin, R.P.W. & Mao, J. (2000). “Statistical pattern recognition: A review”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, 2000.

Jing, F.; Li, M.; Zhang, H.-J. & Zhang, B. (2004). “An efficient and effective region-based image retrieval framework”, *IEEE Transactions on Image Processing*, vol. 13, no. 5, 2004.

Kim, E.Y.; Park, S.H. & Kim, H.J. (2000). “A genetic algorithm-based segmentation of Markov random field modeled images”, *IEEE Signal Processing Letters*, vol. 7, no. 11, 2000.

Ko, B. & Byun, H. (2005). “FRIP: A region-based image retrieval tool using automatic segmentation and stepwise Boolean AND matching”, *IEEE Transactions on Multimedia*, vol. 7, no. 1, 2005.

Kohavi, R. & John, G.H. (1998). “The wrapper approach”, In: *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Kluwer Academic, pp. 33-50, 1998.

Kudo, M. & Sklansky, J. (2000). “Comparison of algorithms that select features for pattern classifiers”, *Pattern Recognition*, vol. 33, pp.25-41, 2000.

Lee, M.W. & Cohen, I. (2004). “Proposal maps driven MCMC for estimating human body pose in static images” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.334-341, 2004.

Li, J.; Wang, J.Z. & Wiederhold, G. (2000). “IRM: Integrated region matching for image retrieval”, *Proc. 8th ACM Intl Conf. on Multimedia*, pp. 147-156, 2000.

Luo, J. & Guo, C. (2003). “Perceptual grouping of segmented regions in color images”, *Pattern Recognition*, vol. 36, pp. 2781-2792, 2003.

National Oceanic and Atmospheric Administration website, http://ngs.woc.noaa.gov/katrina/KATRINA0000.HTM.

Pal, N. R. & Pal, S. K. (1993). "A review on image segmentation techniques", *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294, 1993.

Rabiei, H.; Mahloojifar, A. & Farmer, M. (2007). “Providing context for tumor recognition using the wrapper framework”, *IEEE International Symposium on Biomedical Imaging*, 2007.

Safar, M.; Shababi, C. & Sun, X.(2000). “Image retrieval by shape: A comparative study”, *Proc. IEEE International Conference on Multimedia and Exposition*, pp. 141-144, 2000.

Shi, J. & Malik, J. (2000). “Normalized cuts and image segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.

Sifakis, E., Garcia, C. & Tziritas , G. (2002). “Bayesian Level Sets for Image Segmentation”, *Journal of Visual Communication and Image Representation*, vol. 12, no. 1/2, pp. 44-64, 2002.

Smeulders, A.W.M.; Worring, A.; Santini, S.; Gupta, A. & Jain, R. (2000). “Content-based image retrieval at the end of the early years”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349-1380, Mar. 2000.

Spiliotis, I.M. & Mertzios, B.G. (1998). “Real-time computation of two-dimensional moments on binary images using image block representation”, *IEEE Transactions Image Processing*, vol. 7, no. 11, pp. 1609-1615, 1998.

Suk, T. & Flusser, J. (2004). “Projective moment invariants”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, 2004.

Teague, M.R. (1980). “Image analysis via the general theory of moments”, *J. Opt. Soc. Amer*. vol. 70, no. 8, pp. 920-930, 1980.

Tu, Z. & Zhu, S.-C. (2002). “Image segmentation by data-driven Markov chain monte carlo” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 657-673, May 2002.

Tu, Z.; Chen, X.; Yuille, A.L. & Zhu, S.-C. (2003). “Image parsing: Unifying segmentation, detection, and recognition”, *Proc. IEEE International Conference on Computer Vision*, pp. 18-25, 2003.

Veltkamp, R.C. & Hagedorn, M. (2001). “State-of-the-art in shape matching”, In *Principles of Visual Information Retrieval*. pp. 87-119, Springer, 2001.

Yang, J. & Honavar, V. (1998). “Feature subset selection using a genetic algorithm”, *IEEE Intelligent Systems*, March-April, pp. 44-49, 1998.

Yoshitake, A. & Ichikawa, T. (1998). “A survey on content-based retrieval for multimedia databases”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 81-93, 1999.

Yu, S.X.; Gross, R. & Shi, J. (2002). "Concurrent object recognition and segmentation by graph partitioning“, In: *Advances in Neural Information Processing Systems 15*, 2002.

Projective Registration with Manifold Optimization

Guangwei Li [1) 2)], Yunpeng Liu [1)], Yin Jian[3)] and Zelin Shi [1)]
*1) Shenyang Institue of Automation, CAS
P.R.China
2) Qingdao University
P.R.China
3）The Research Institute on General Development and Argumentation of Air Force
Beijing, P.R.China*

1. Introduction

Image registration is of interest to scientists and engineers of various fields: computer vision, pattern recognition, and robotics (Trucco & Plakas, 2006). One of the remarkable problems both in theory and in technique is how to cope with the dynamic geometric-warps. For small objects and large camera-to-scene, i.e., the background appearance of the target is enough far, the projective transformation effect of the target is negligible. A satisfactory tracking result can be achieved by means of approximating the geometric warps with the affine transformation. However, in many special and practical applications, such as image mosaics in computer graphics and vision guidance in the military field, the projective transformation should be considered. It is well known that the projective transformation exactly models the motion relationship between images of the identical planar object scene, and can describe the pan and the tilt of the camera, which the affine transformation cannot do (Mann & Picard, 1997). The projective map model has eight independent parameters which have widely varying sensitivities and the transformation is highly nonlinear (Michael Gleicher, 1997), all of which affect to design the registration algorithms with efficiency, accuracy and robustness.

Lucas-Kanade image registration method was first proposed in 1981. Within the classical space transformation-based tracking framework proposed by Hager et al (Hager & Belhumeur, 1998), a projective image registration approach based on the matrix parameterization was presented (Buenaposada & Baumela, 2002) in terms of the forward-addition algorithm on the vector space, which is called the VECTOR-GN algorithm in this paper . The inverse-composition algorithm was proposed (Baker & Matthews, 2004) not only to compute the Hessian matrix and the gradient matrix offline but also to improve the efficiency by improving the iterative structure. However, these strategies can not utilize the projective Lie group structure sufficiently and leave room to improve the performance of the image registration algorithms.

The geometric optimization algorithm based on the manifolds, a novel approach to solve the constrained problem, was proposed in the 1970s-1980s (Gabay, 1982). The fundamental idea of this approach is to regard the constrained sets as one underlying manifold and to exploit the geometry of the underlying parameter space. That is to develop a strategy which views constrained problems as to be equivalent to unconstrained problems posed on the constraint sets. Philosophically, this approach is geometrically more intuitive, conceptually simpler and mathematically more elegant. Various problem have been solved by applying the optimization algorithms on various manifolds. Especially, the optimization algorithms based on Lie Groups (Owren & Welfert, 1996) and Riemannian manifolds (Yaguang, 1999) have already been applied to robot control and machine learning. Recent years have also witnessed the rich achievements in the fields of signal processing, computer vision and pattern recognition. Smith generalized some optimization algorithms on the vector space to Riemannian manifolds and studied the adaptive filter problem of nonlinear signal (Smith, 1993). Yean considered the optimization algorithm on SE(n, R) about the 2D-3D pose estimation in computer vision (Yean, 2005). Grenander proposed his famous General Pattern theory, of which the deformable template idea (Grenander, et al., 1998) is that the object is represented by the template and the infinite varieties of the pose and location associated with its occurrences are represented via transformations which act on the template. These transformations form one transitive group acting on the space of all possible transformed templates, which becomes a Lie group orbit. Hence, the problem on the automated target recognition and tracking switches into the parameter optimization problem on the Lie Group manifolds.

Exploiting the deep connection between the Lie group and its associated Lie algebra which is called the Lie group exponential map, the geometric optimization approach based on Lie Groups theory switches the constrained nonlinear problems into equivalently unconstrained problems, thereby significantly reducing the computational complexity. A novel homograhy-based target image registration and tracking approach based on the Lie algebra parameterization which is called the LEXP-GN algorithm in this paper was proposed (Eduardo & Jaime, 2007). The performance, such as the tracking precision and rate, is better than that of the tracking method based on the matrix parameterization.

Noticeably, there exists a bi-invariant Riemannian metric on a compact Lie group (such as SO(n,R)) and the geodesic through the identity element of group is one-parameter group. Hence, the Lie group exponential map agrees with the Riemannian exponential map. However, a noncompact Lie group (such as $SE(n,R)$, SL(n,R) and GA(n,R)) has not a bi-invariant Riemannian metric and the Riemannian exponential map based on the geodesic is usually different from the exponential map based on the Lie group structure. Therefore, the geometric optimization algorithms on the noncompact Lie groups based on the Lie group exponential map have its limitations. To our knowledge, it seems that there is not very much research on the noncompact Lie group optimization. Mahony and Manton provided an instructive interpretation of the Newton optimization method on the noncompact Lie groups from the Cartan-Schouten connection views of Riemannian geometry (Mahony & Manton, 2002). However, the Newton method needs to compute the complicated Hessian matrix and is usually not feasible to be applied to the real time application.

The core of our registration algorithm is the optimization problem on the special linear group SL(3,R). Recently, Seok, et al studied the optimization algorithm on SL(3,R) about the medical images registration problems (Seok, et al, 2007). Based on the Riemannian exponential map obtained from the geodesic equation, we propose a second-order efficient target tracking algorithm within the intrinsic geometric optimization framework. The comparative experiments with VECTOR-GN and LEXP-GN indicate the improvement on the tracking rate and precision.

The rest of the chapter is organized as follows. After a brief introduction to the Lie group exponential map and the Riemannian exponential map, the connection between them is studied on section 2, with the geometric optimization framework. Section 3 investigates the second-order projective image registration algorithm based on the Riemannian exponential map within the intrinsic optimization framework. Some comparative results are shown for illustration and verification in section 4. Finally, section 5 concludes the investigation and proposes some further work. Some necessary supplementary material will be given in section 6.

2. Mathematical background

The exponential map and the intrinsic geometric optimization algorithm build the basis for our efficient projective registration method. The tools used here come primly from Lie group and Riemannian geometry. To enable further discussion, we need to take a small detour into geometry on them. Further information can be found in the famous textbooks (Helgason, 1978; Berger, 2003).

2.1 Lie group exponential map

A Lie group is a group endowed with the smooth manifold structure, and its group multiplicative operation is denoted by $\circ$. The tangent space at the identity element e of Lie group M is denoted by T_eM . Let assume $m \in M, X_e \in T_eM$. The left-invariant vector field X , which determines a left-invariant flow $\varphi_x(t,m) = X_t(m)$, can be obtained by the left-translation $X_m = (L_m)_* X_e$. The one-parameter group, an integral curve at e , is denoted by $\gamma_X(t)$. The vector space $(T_eM,[\cdot,\cdot])$ equipped with a bilinear bracket operation is a Lie algebra denoted by $\Lambda(M)$. It is known that the left-invariant vector field, the left-invariant flow, the tangent space at the identity element, one-parameter group and the Lie algebra are equivalent in essence.

Definition 1. Lie group exponential map $\mathrm{Lexp} : \Lambda(M) \times R \to M, (X, t) \to \mathrm{Lexp}(tX) = \gamma_X(t)$

For convenience, we usually define the Lie group exponential map, $\mathrm{Lexp} : \Lambda(M) \to M$, as follows

$$\mathrm{Lexp}(X) = \gamma_X(1) \tag{1}$$

Lemma 1. There exists an open neighborhood W of 0 in Lie algebra $\Lambda(M)$ and an open neighborhood U of e in M such that Lexp is an analytic diffeomorphism of W onto U .

From Lemma 1, we can define its inverse function known as the logarithm map which returns $X = \log y$ such that Lexp(X)=y . (See Fig. 1).

The space of all $n \times n$ nonsingular real matrices forms a Lie group, called the general linear group denoted by $\mathrm{GL}(n, R)$. Its algebra is usually denoted by $\mathrm{gl}(n, R)$, the set of all real square matrices. Being a sub-group of $\mathrm{GL}(n,R)$, the special linear group $\mathrm{SL}(n,R)$ is the space of all real $n\times n$ matrices H satisfying $\det H = 1$. Its Lie algebra denoted by $\mathrm{sl}(n,R)$ consists of the real matrices of trace zero. What we concern in this paper is $\mathrm{SL}(3,R)$ whose Lie algebra is $\mathrm{sl}(3,R)$ with the following basis vectors.

$$e_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad e_4 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$e_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad e_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad e_7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad e_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

For matrix Lie groups, the group operation is matrix multiplication. The Lie bracket operation is $[A,B] = AB - BA$ and the Lie exponential map of a matrix $A \in \mathrm{gl}(n,R)$ is computed by the formula

$$\mathrm{L}\exp A = \exp A = \sum_{n=0}^{\infty} \frac{A^n}{n!} \tag{2}$$

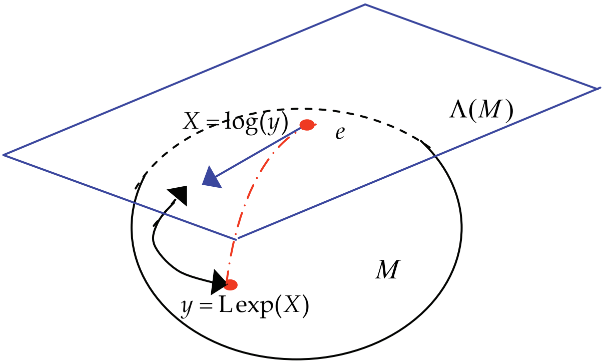


Fig. 1. Lie group exponential map and its inverse map

2.2 Riemannian exponential map

Let M be a smooth manifold of m dimensions. For every point $p \in M$, if an Euclidean inner product $g(p) = <,>: T_pM \times T_pM \to R$ is assigned on its tangent space, (M,g) is called a Riemannian manifold of m dimensions and g is called its Riemannian metric

Let M be a smooth manifold of m dimensions. For every point $p \in M$, if an Euclidean inner product $g(p) = <,>: T_pM \times T_pM \to R$ is assigned on its tangent space, (M,g) is called a Riemannian manifold of m dimensions and g is called its Riemannian metric.

Let $\gamma:[a,b]\to M$ be a smooth curve in M. Its length is defined as

$$L(\gamma)=\int_a^b\left|\gamma^{'}(t)\right|dt=\int_a^b\left|\sqrt{<\gamma^{'}(t),\gamma^{'}(t)>}\right|dt \tag{3}$$

A smooth curve $\gamma(t)$ is called a geodesic on Riemannian manifold if the family of tangent vector field $\gamma'(t)$ is parallel with respect to $\gamma(t)$. The geodesic has an important property that it is the minimal length curve of the following energy function

$$E(\gamma(t))=\int_a^b\left|\gamma^{'}(t)\right|^2 dt \tag{4}$$

Let $\gamma(t)=\gamma(t;p,v)$, $t\in[0,1]$ is a geodesic on Riemannian manifold satisfying $\gamma(0)=p$, $\gamma'(0)=v$, and $B_p(\varepsilon)$ be an open ball on $T_p(M)$ whose center is origin and radius is ε . The Riemannian exponential map at p is defined as follows.

Definition 2. Riemannian exponential map: $\mathrm{Rexp}_p:B_p(\varepsilon)\to M$, $\mathrm{Rexp}_p(tv)=\gamma(t;p,v)$.

In what follows, we denote the map Rexp_e , the Riemannian exponential map from the tangent space T_eM at the identity element e of the Lie group M , by Rexpp ,

$$\mathrm{Rexpp}(v)=\mathrm{Rexp}_e(v)=\gamma(1) \tag{5}$$

Lemma 2. Let (M,g) be a Riemannian manifold of dimensions. For any point $p\in M$, there exists an open neighborhood V of origin such that Rexp is an analytic diffeomorphism of V onto $U=\mathrm{Rexpp}(V)$.

From Lemma 2, if we define $\mathrm{Rexp}_p(v)=q$, its inverse function known as the logarithm, can be defined as v=$\mathrm{Rlog}_{\,p}(q)$ such that, $\gamma(0)=p,\gamma(1)=q$ and $\dot{\gamma}(0)=v$. (See Fig. 2)

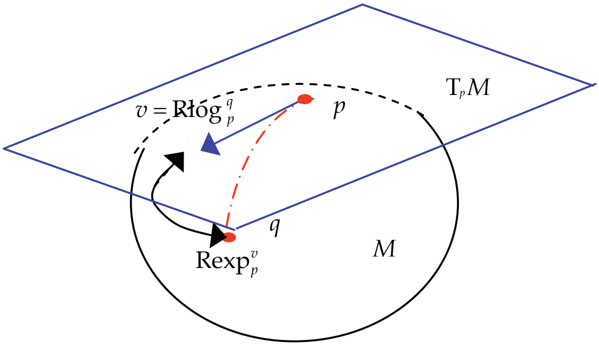


Fig. 2. Riemannian exponential map and its inverse map

For $\mathrm{SL(n,R)}$, $\mathrm{Rexpp}(v)$ is one-to-one on $\|V\|<\dfrac{1}{2}$ (Begelfor & Werman, 2005).

2.3 Relationship between Lie group exponential and Riemannian map

Definition 3: Let G be a Lie group, g be a Riemannian metric on G, g is a left-invariant (right-invariant) Riemannian metric if its left-translation L_g (right-translation R_g) is an isometric transformation on Riemannian manifolds. A metric is bi-invariant if and only if the metric is both left-invariant and right-invariant.

Lemma 3. There exists a bi-invariant Riemannian metric on a compact Lie group and the geodesics at the identity element e are one-parameter subgroups.

Lemma 3 shows that the Riemannian exponential map at the identity defined by the bi-invariant Riemannian metric agrees with the Lie group exponential map, that is, for any tangent $v \in \Lambda(M)$

$$\text{Lexp}(v) = \text{Rexpp}(v) \tag{6}$$

Let us consider the general linear group GL(n,R). For every point p on the tangent space T_pM, inner product $< \cdot >$ is defined as follows

$$< A, B >= Tr(AB^T) \tag{7}$$

where $A, B \in T_pM$, and $Tr(\cdot)$ is the trace of a matrix. Hence, the length of a tangent vector is defined

$$\|v\|^2 = Tr(vv^T) \tag{8}$$

Now, we begin to consider the Riemannian exponential map on a noncompact Lie group. First, we propose the following theorem on the minimal geodesics with respect to the right - invariant Riemannian metric on general linear group manifolds. Please pay attention to the point g on the Riemannian manifold, which is not the Riemannian metric.

Theorem 1. Let $g(t),\ t \in [0,1]$ be a minimal geodesic connecting $g, h \in \text{GL}(n, R)$. The tangent vectors minimizing the energy function $\int_0^1 \|v(t)\|^2 dt$ and satisfying $dg(t)/dt = v(t)g(t), g(0) = g, g(1) = h$ are the solutions of the following matrix differential equation

$$\begin{aligned} dv(t)/dt &= v(t)v^T(t) - v^T(t)v(t) \\ &= [v(t), v^T(t)] \end{aligned} \tag{9}$$

A proof of Theorem 1 is given in 6.1.

From Theorem 1, it follows that the equation of the minimal geodesic on $\text{GL}(n, R)$ is

$$g(t) = \exp((v(0) + v(0)^T)t)\exp(-v(0)^T t)g \tag{10}$$

Therefore, the Riemannian exponential map, the equation of the minimal geodesic equation on $GL(n,R)$ which emanates from the identity with a velocity v is expressed

$$\mathrm{Rexpp}(v) = \exp((v^T + v))\exp(-v^T) \tag{11}$$

Now, we consider the exponential map on the subgroups of $\mathrm{GL}(n,R)$. The special orthogon al matrix $\mathrm{SO}(n,R)$ is a compact group, and its algebra $\mathrm{so}(n,R)$ are skew symmetric matrice s with zero trace. It follows that, for every $v \in \mathrm{so}(n,R)$, we have $v = -v^T$, yielding

$$\mathrm{Rexpp}(v) = \exp((v^T + v))\exp(-v^T) = \exp(v) \tag{12}$$

We can see that the Riemannian exponential map is the same as the Lie group exponential map for the compact groups, which completely consists with the Lemma 3. For the noncompact group $\mathrm{SE}(n,R)$, its geodesics can be obtained by lifting the geodesics from $\mathrm{SO}(n,R)$ and R^n (Zefran & Kumar, 1998). However, for the noncompact group $\mathrm{SL}(n, R)$, every $v \in \mathrm{sl}(n,R)$, $\mathrm{Rexpp}(v) \neq \mathrm{Lexp}(v)$. We construct the optimization algorithm with the Riemannian exponential map $\mathrm{Rexpp}(v) = \exp(v^T + v)\exp(-v^T)$, $v \in \mathrm{sl}(3,R)$ in this paper

2.4 Framework for geometric optimization

If a Lie group is embedded in Euclidean space to be a sub-manifold, the optimization problem on it can often become a classical constrained optimization. The conventional approach of dealing with the structure of the group is to use Lagrange multipliers. Based on the geometric optimization theory, we use local canonical coordinates to represent parameters and intrinsically take care of the geometric structure of Lie Groups to allow the use of unconstrained optimization routines (Vercauteren & Malis, 2007).

Let x be a point in the neighborhood of $t \in M$. From Theorem 1, there exists $\omega = \sum_i^n v_i e_i \in \Lambda(M)$ such that

$$x = t \circ \mathrm{Lexp}(\omega) = t \circ \mathrm{Lexp}(\textstyle\sum_i^n v_i e_i) \tag{13}$$

where $v = (v_1, v_2, \cdots, v_n)^T$; $e_i (i = 1, \cdots, n)$ is the basis of Lie algebra $\Lambda(M)$ .Then, the Taylor series of a smooth function $\varphi(\cdot)$ on Lie group M is obtained

$$\varphi(t \circ \mathrm{Lexp}(\omega)) = \varphi(t) + J_t^\varphi v + \frac{1}{2} v^T H_t^\varphi v + O(\|v\|^3) \tag{14}$$

where $\left[J_t^\varphi\right]_i = \frac{\partial}{\partial v_i}\varphi(t \circ \mathrm{Lexp}(\omega))|_{v=0}$ and $\left[H_t^\varphi\right]_{ij} = \frac{\partial}{\partial v_i v_j}\varphi(t \circ \mathrm{Lexp}(\omega))|_{v=0}$

The Taylor series (14) allows us to construct various optimization algorithms on Lie groups by generalizing algorithms on vector space. For example, the classical Newton-Raphson method adopts the following intrinsic update step

$$t \leftarrow t \circ \mathrm{Lexp}(\omega) \tag{15}$$

where v solves $H_t^{\varphi} v = -\left[J_t^{\varphi}\right]^T \varphi(t)$.

Unfortunately, in many cases, the Hessian matrix H_t^{φ} is often difficult or impossible to compute. Even worse is that the convergence problem may arise when it is not definite positive. Hence, Benhimane et al constructed the intrinsic Gauss-Newton algorithm by preserving the linear part and discarding the quadric item of Taylor series (Benhimane & Malis 2007). Motivated by the second-order minimization method based on the Lie algebra parameterization, we take place of the Lie group exponential map with the Riemannian exponential map to construct an efficient second-order minimization algorithm based on the geodesics on manifolds, which is called REXPP-ESM algorithm in this paper. This algorithm can also find back the Hessian matrix information discarded by LEXP-GN algorithm within the intrinsic optimization framework to further improve the registration performance. The performance of the REXPP-ESM will be explained in 4.2.

Lemma 4. Any manifold of dimension d can be embedded in E^{2d+1} (Berger, 2003).

From Lemma 4, we know that for a Lie group, there naturally exists an embedding map $\pi : G \to R^n$, $t \to \pi(t)$ such that it is a sub-manifold in Euclidean space. Especially, the spatial transform groups (e.g. rigid body, affine, projective.) used in the target recognition and tracking are often represented and computed by common matrixes in homogeneous coordinates.

3. Projective registration with manifold optimization

3.1 Problem Statement

Suppose the camera is not be calibrated and the tracked object has a flat appearance. When the target is moving in the space, the relation between images can be described by projective transformation. The projective transform group is the group of the matrices of the form $T = \begin{bmatrix} R & t \\ v & 1 \end{bmatrix}$, where R is a 2×2 nonsingular matrix, t is a column vector for the translation and $(v,1)^T$ is the projection of the line at infinity. We choose the scale factor to normalize the projective group matrices T such that the determinants of T are equal to 1. Then the matrices T belong to the special linear group $\mathrm{SL}(3,R)$. This normalization cannot change the degree of parametric freedom and is reasonable in real applications [21].

From Lemma 4, we can suppose the homogeneous coordinate of point p be $(x,y,1)^T$ and the embedding map in Euclidean space of $\mathrm{SL}(3,R)$ be $\pi : t \to \pi(t)$. Define a group action from $\mathrm{SL}(3,R)$ on p : $w : \mathrm{SL}(3,R) \times p \to p$. The projective transformation is represented as follows

$$w(\pi(t))(p) = \frac{1}{a_{31}x + a_{32}y + a_{33}} \begin{bmatrix} a_{11}x + a_{21}y + a_{13} \\ a_{21}x + a_{22}y + a_{23} \\ a_{31}x + a_{32}y + a_{33} \end{bmatrix} \tag{16}$$

Let $I(p)$ be the brightness value of the template and $I(w(t)(p))$ be the intensity of projective-transformed target in the input image. The algorithm assumes the gray value is invariable at the same target position in two consecutive frames and calculates the projective transformation parameters to know the current position where the target is in the current image by solving the following function

$$\arg\min\left\|I(w(\pi(t))(p))-I(p)\right\|^2 \tag{17}$$

3.2 Optimum parameters for projective registration

To solve the projective parameters is in fact to perform optimization on $\mathrm{SL}(3,R)$. Based on the Lie algebra parameterization, $\omega=\sum_{i=1}^{8}v_ie_i$, where $v=(v_1,\cdots,v_8)^T$ is the incremental motion parameter vector, we take place of the Lie group exponential map with the Riemannian exponential map and the nonlinear least-squared optimization problem can switch into

$$\arg\min\left\|I\circ t\circ \mathrm{Rexpp}(\textstyle\sum_{i=1}^{n}v_ie_i)(p)-I(p)\right\|^2 \tag{18}$$

Let $f_p(t\circ \mathrm{expp}(\omega))=I\circ t\circ \mathrm{expp}(\omega)(p)-I(p)$ .we get

$$\begin{aligned}\left\|f_p(t\circ \mathrm{Rexpp}(\omega))\right\|^2 &=\left\|I\circ t\circ \mathrm{Rexpp}(\omega)(p)-I(p)\right\|^2\\ &\approx\left\|I\circ t(p)-I(p)+J_t^{f_p}\,|_{v=0}\,v+\frac{1}{2}v^TH_t^{f_p}v+O(\|v\|^3)\right\|^2\end{aligned} \tag{19}$$

We pay attention to fact that when the images are aligned with the optimal spatial transformation in target tracking, the template and the warped image as well as their gradient should be very close to each other, i.e. $\nabla_p I\circ t^*\approx\nabla_p I$. An efficient tracking algorithm will be constructed by utilizing this information to recover the information discarded with Gauss-Newton method by means of expanding the Jacobian matrix at the optimal transformation t^* , hence avoiding computing the Hessian matrix at the same time. A first-order Taylor series around 0 of $J_t^{f_p}$ in (20) can lead to

$$v^TH_t^{f_p}=J_t^{f_p}(v)-J_t^{f_p}(0)+O(\|v\|^2) \tag{20}$$

Incorporating this expression into (19), we can get a true second-order approximation

$$\left\|f_p(t\circ \mathrm{Rexpp}(\omega))\right\|^2\approx\left\|I\circ t(p)-I(p)+\frac{1}{2}(J_t^{f_p}(v)+J_t^{f_p}(0))v+O(\|v\|^3)\right\|^2 \tag{21}$$

The following is to compute the Jacobian matrix $J_t^{f_p}(0)$ and $J_t^{f_p}(v)$ corresponding to the derivative of at 0 and v .

$$\begin{aligned} J_t^{f_p}(0) &= \frac{\partial I \circ t \circ \mathrm{Rexpp}(\sum_{i=1}^{n} v_i e_i)}{\partial v^T}\Big|_{v=0} \\ &= \frac{\partial I \circ t(q)}{\partial q^T}\Big|_{q=p} \cdot \frac{\partial w(t,p)}{\partial t^T}\Big|_{t=\pi(Id)} \cdot \Big[\frac{\partial \pi(\mathrm{Rexpp}(\sum_{i=1}^{8} v_i e_i))}{\partial v_1}\Big|_{v_1=0}, \cdots, \frac{\partial \pi(\mathrm{Rexpp}(\sum_{i=1}^{8} v_i e_i))}{\partial v_8}\Big|_{v_8=0}\Big] \\ &= \nabla_p^T (I \circ t) \frac{\partial w(t,p)}{\partial t^T}\Big|_{t=\pi(Id)} \cdot \big[\pi(e_1) \quad \cdots \quad \pi(e_8)\big] \\ &= \nabla_p^T (I \circ t) J^{w_p} e_\pi \end{aligned} \tag{22}$$

For every line of the matrix $J_t^{f_p}(0)$, $\left[\nabla_p^T (I \circ t\right]_{1\times 3}$ is corresponding to the spatial derivative of the current warped image using the projective transformation t; $\left[J^{w_p}\right]_{3\times 9}$ is the Jacobian matrix for projective transformation (13), and $[e_\pi]_{9\times 8}$ is the Jacobian matrix where $\pi(e_i)$ is the matrix e_i reshaped as a vector (the entries are picked line per line). Id is identical transformation. The two Jacobians J^{w_p} and e_π are constants to be computed once and for all while the Jacobian $J_t^{f_p}(0)$ has to be computed at each iteration since it depends on the updated value of projective parameters.

However, the Jacobian matrix $J_t^{f_p}(v)$ is complicated and usually depends on t. Hence, we do not directly compute $J_t^{f_p}(v)$. If replacing the gradient of the optimally warped image $I \circ t^* = I \circ t \circ \mathrm{expp}(v_t^*)$ by its equivalent gradient of the template image, we can get a simple linear approximation of $J_t^f(v_t^*) \cdot v_t^*$ as follows

$$J_t^{f_p}(v_t^*) \cdot v_t^* \approx \nabla_p^T I \cdot J^{w_p} \cdot e_\pi \cdot v_t^* \tag{23}$$

A proof of（23）is give in 6.2.
Let J_t be the following matrix

$$J_t = -\frac{1}{2}(\nabla_p^T I + \nabla_p^T (I \circ t)) J^{w_p} e_\pi \tag{24}$$

By incorporating (24) into (21), we have

$$\left\| f_p(t \circ \mathrm{Rexpp}(\omega_v^*)) \right\|^2 \approx \left\| I \circ t(p) - I(p) + J_t v_t^* + O\left\| v_t^* \right\|^3 \right\|^2 \tag{25}$$

This cost function has a local or global minimum at v

$$v_t^* = J_t^+ (I \circ t(p) - I(p)) \tag{26}$$

where J_t^+ is the pseudo-inverse of J_t. Hence, the intrinsic iterative update is

$$t = t \circ \mathrm{Rexpp}(\textstyle\sum_{i=1}^{8} v_i e_i) \tag{27}$$

4. Experimental Results and Analysis

4.1 Experimental Results

To validate the feasibility and efficiency of our algorithm, we compare our REXPP-ESM algorithm with VECTOR-GN algorithm and LEXP-GN algorithm. All the algorithms are implemented in matlab and tested in the computer with Intel PIV 2.4GHZ and 512 Memory. Since the 8 parameters in the projective warp have different units, we compute the RMS (root-mean-square) error of the corresponding points between the template and target image rather than the RMS of parameters. In addition, it should be emphasized that neither preliminary image filtering nor multi-scale pyramid implementations nor other robust techniques has been used for this evaluation.

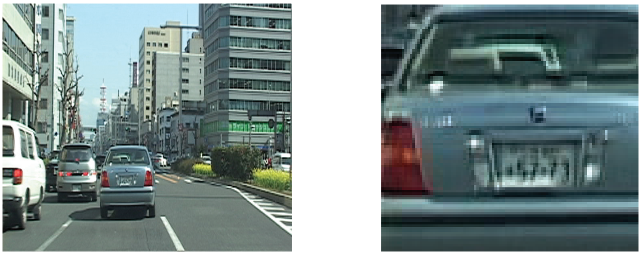
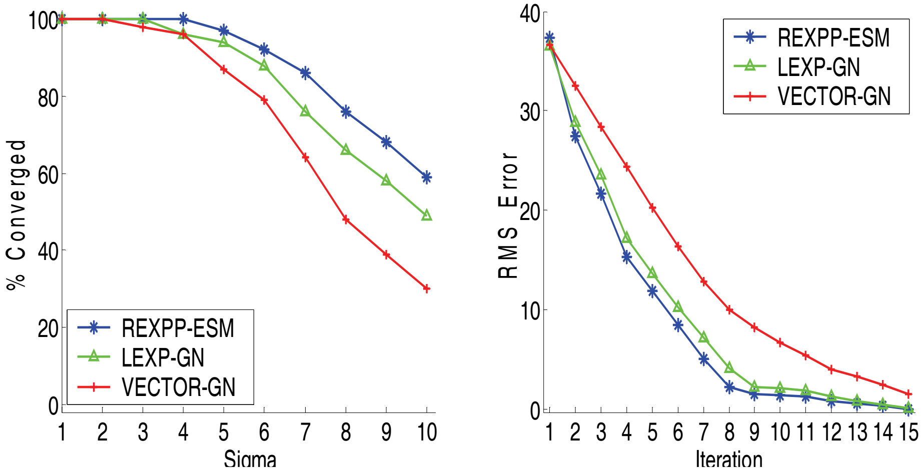

(a) (b)

Fig. 3. Input image and template. (a) Input image. (b) Template.

Experiment 1: We utilize the experiment data provided by Baker etc. in CMU and the same experiment setting to compare the three algorithms (http://www.ri.cmu.edu/people/bakers_simon.html). We experimented with the image in Fig. 3(a) and manually selected a 100×100 pixel template (see Fig. 3(b)) in the center of the image. We randomly perturbed the four corner points of the template 1000 times with additive white Gaussian noise of a certain standard variance σ from one pixel to ten pixels and fitted for the projective warp parameters that these perturbed points define (for each standard variance, we generated 100 randomly inputs). We say that an algorithm converged if the RMS error in the canonical point locations is less than 3.0 pixels after 15 iterations. We computed the percentage of times that each algorithm converged for each standard variance. The results are shown in Fig. 4(a) that shows when the perturbation to the canonical point locations is less than about 3.0 pixels, all the three algorithms converge almost always. With the increase of the σ, the frequency of convergence for LEXP-GN algorithm rapidly decreases. While $\sigma = 10$, the frequency of convergence for VECTOR-GN algorithm, LEXP-GN algorithm and our REXPP-ESM algorithm is 30%, 49% and 60% respectively. For 100 times experiments of $\sigma = 6$, all experiment test data are shown in Fig. 4(b). Our REXPP-ESM algorithm requires 8 iterations to coverage while LEXP-GN requires 9 iterations and VECTOR-GN requires 14 iterations.



(a) (b)

Fig. 4. Comparison between REXPP-ESM and other two methods. (a) Frequency of convergence. (b) Average converge rates

Experiment 2: We show three experiment results for three typical image sequences using the three algorithms. The size of each frame in the first sequence is 512 × 512 （http://esm.gforge.inria.fr/ESMdownloads.html）and the size of the tracked region is 150×150. The tracked region shows projective transformation. The VECTOR-GN algorithm, LEXP-GN algorithm and REXPP-ESM algorithm converge after 11, 7 and 6 iterations respectively. Fig. 5 shows some tracking results of them. The 161st and 187th frames have larger deformation. The VECTOR-GN algorithm cannot converge and the tracker slides off the tracking region. LEXP-GN algorithms cannot lock the tracking region at 161st frame. However, our REXPP-ESM algorithm can be implemented very well on all the frames. The second virtual house sequence includes one hundred still frames (http://vasc.ri.cmu.edu/idb/html/motion/index.htm). The size of each frame is 512×480. The tracked target is the window of the house and the size of template is 52×40. The tracked region shows larger projective transformation. The VECTOR-GN algorithm, LEXP-GN algorithm and REXPP-ESM algorithm converge after 7, 5 and 4 iterations respectively. Fig. 6 shows some tracking results of them. The sequences after 80th frame have larger deformation. The VECTOR-GN algorithm cannot converge and LEXP-GN algorithms cannot lock the tracking region at 91st frame either. However, our EXPP-ESM algorithm can be implemented very well on all the frames. The third car sequence contains 150 frames .The size of each frame is 768×576. The tracked target is the back of the running car and the tracked region shows larger enlargement warp. The VECTOR-GN algorithm, LEXP-GN algorithm and REXPP-ESM algorithm converge after 6, 4 and 3 iterations respectively. Fig. 7 show some tracking results of them. The sequences after 130th frame have larger deformation. The VECTOR-GN algorithm cannot converge and the tracker suffers form the drifts from the tracking region. LEXP-GN algorithm cannot lock the tracking region at 150th frame. However, our REXPP-ESM algorithm can be implemented very well on all the frames. The table 1 summarizes the comparative performance of the three algorithms.

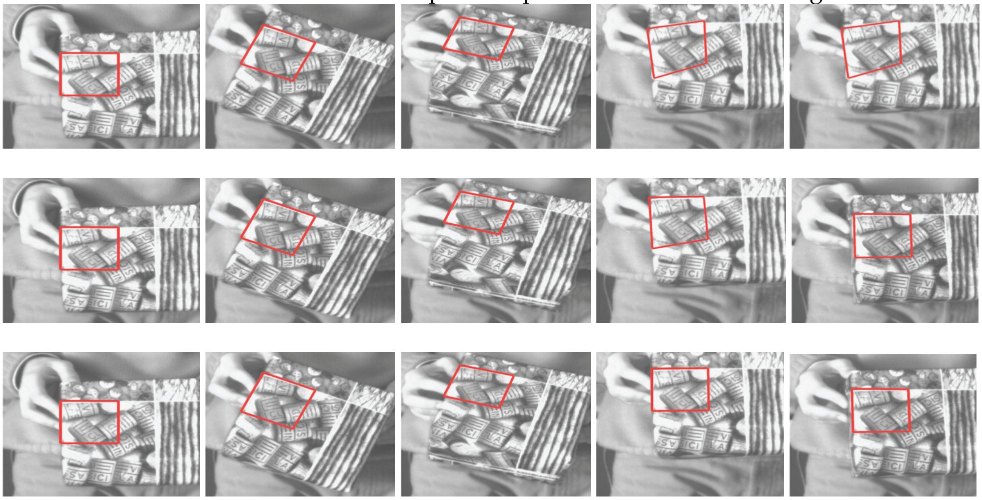


Fig. 5. Comparison of the VECTOR-GN (*first row*), LEXP-GN (*second row*) and REXPP-ESM (*third row*). The sequences contain 200 frames. From left to right in each column is No. 1, 50, 100, 161, 187 frame in wooden box sequences. See text for details.

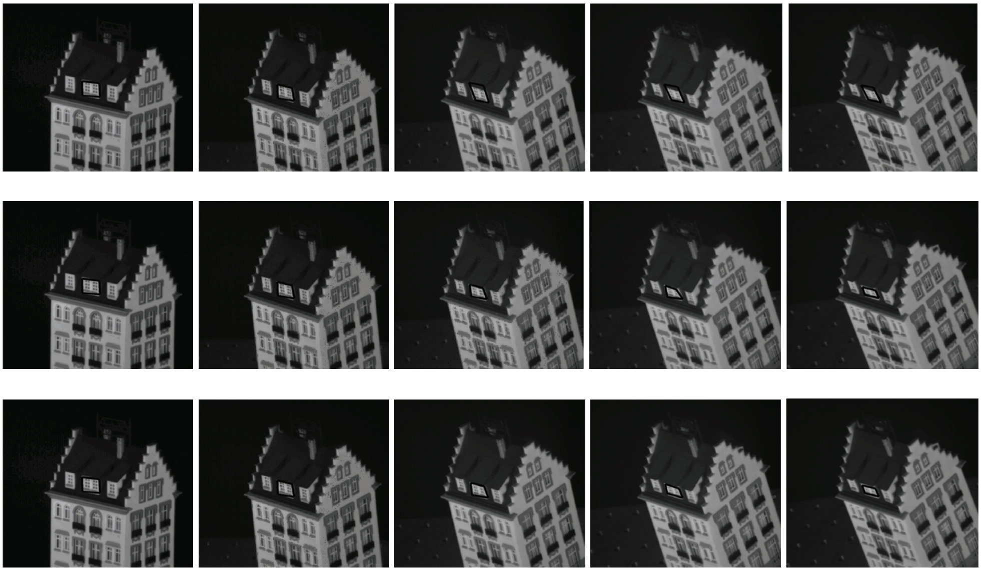


Fig. 6. Comparison of the VECTOR-GN (*first row*), LEXP-GN (*second row*) and REXPP-ESM (*third row*). The sequences contain 100 frames. From left to right in each column is No. 31, 60, 80, 90, 100 frame in virtual houses box sequences. See text for details.

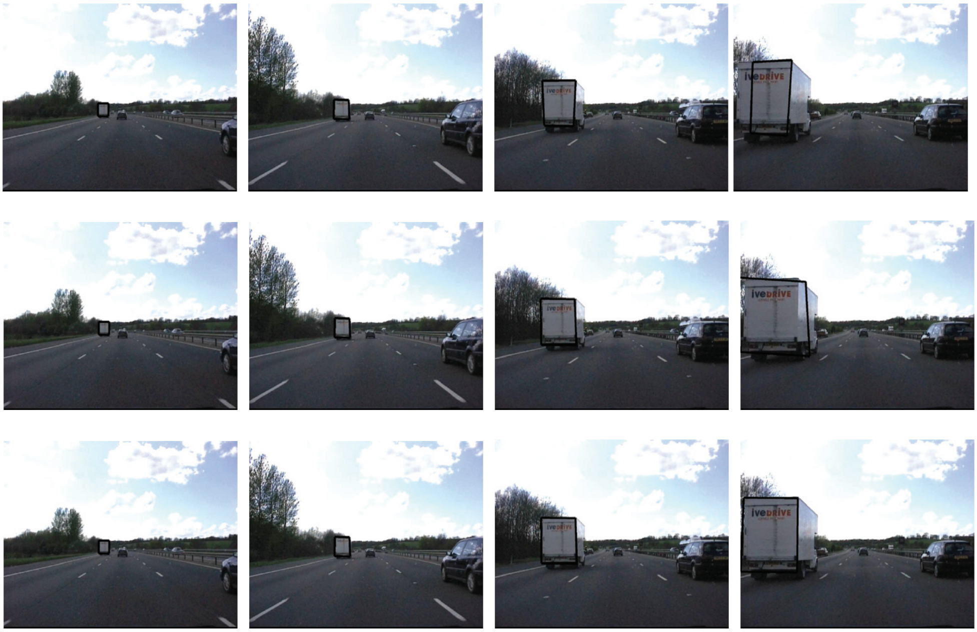


Fig. 7. Comparison of the VECTOR-GN (*first row*), LEXP-GN (*second row*) and REXPP-ESM (*third row*). The sequences contain 150 frames. From left to right in each column is No. 1, 60, 130, 150 frame in virtual houses box sequences. See text for details.

4.2 Analysis

From the results of the two experiments, we conclude that our algorithm utilizes the minimum geodesics and avoids computing Hessian matrix can make our REXPP-ESM algorithm is much superior to the VECTOR-GN algorithm and is slightly better than the LEXP-GN algorithm in the convergent frequency and convergence rate. Firstly, it is evident that the VECTOR-GN algorithm performs not well because it can not exploit the projective parameters intrinsic manifold structure. Secondly, it should be noted that when the distance between the two points are much close to the identity element, the REXPP-ESM performance is almost identical with the LEXP-GN because now the geodesic on SL(3,R) can be replaced by Lie exponential map. This can be obtained from the Fig. 4(a) of the first experiment where its data are synthesized when the perturbation is very small, namely, the projective warp is not remarkable. Although we adopt the second-order optimization, our REXPP-ESM doesn’t perform much better than LEXP-GN. In real video sequences, although the deformation of the two continuous frames is usually not big, the drawbacks of the VECTOR-GN algorithm make it very easy to get local minimum while LEXP-GN and our REXPP-ESM perform well similarly. When the warps are bigger on some frames, our REXPP-ESM performs better than LEXP-GN, especially on the coverage rate. The reason for this is that our REXPP-ESM algorithm marches along the shortest distance during the optimization process than that of LEXP-GN. We confirm that some deep theory on Riemannian geometry should be introduced to explain it and leave it to work in the future.

sequence	algorithm	number of average iteration	Average coverage rate
wooden box	VECTOR-GN	11	89%
	LEXP-GN	7	96%
	REXPP-ESM	6	99.5%
virtual house	VECTOR-GN	7	90%
	LEXP-GN	5	95%
	REXPP-ESM	4	99.5%
car	VECTOR-GN	6	86%
	LEXP-GN	4	98%
	REXPP-ESM	3	100%

Table 1. Comparative performance of the three algorithms

5. Conclusion

We have presented the Riemannian exponential map on the noncompact Lie groups based on the minimum geodesics and constructed the registration and tracking algorithm without computing the Hessian matrix. The experimental results compared with the classical vector space algorithm and the Gauss-Newton optimization algorithm based on the Lie group exponential map show that the accuracy and the convergent rate demonstrate some evident improvements. It is emphasized that both Lie group exponential map and the Riemannian exponential map are based on the local linearization and are easy to diverge if the initial value and the iterative step size are not chosen improperly. Besides, we also investigate that the representative methods of the projective parameters have important effect on the experiment results and should be considered seriously.

6. Appendix

6.1 Proof of theorem 1

Assume $\eta(t)\in gl(n,R)$. Perturbing the shortest length curve $g(t)$ with $\exp(\varepsilon\eta(t))$ along $\eta(t)$ leads to

$$g(t,\varepsilon)=\exp(\varepsilon\eta(t))g(t)=(\mathrm{id}+\varepsilon\eta(t))g(t)+o(\varepsilon) \tag{28}$$

where the perturbation has the property that $\eta(0)=0,\ \eta(1)=0$ and that the boundary conditions remain satisfying at $g(0,\varepsilon)=g, g(1,\varepsilon)=h$.

From (9), we can draw（29）and（30）. On one side

$$d(g(t)+\varepsilon\eta(t)g(t))/dt=v(t)g(t)+\varepsilon\dot{\eta}(t)g(t)+\varepsilon\eta(t)v(t)g(t) \tag{29}$$

on the other side

$$d(g(t)+\varepsilon\eta(t)g(t))/dt=v(t)g(t)+\varepsilon v(t)\eta(t)g(t)+\varepsilon\, dv(t,\varepsilon)/d\varepsilon\, g(t)+o(\varepsilon) \tag{30}$$

Comparing（29）with（30）gives the equation

$$dv(t,\varepsilon)/d\varepsilon = \dot{\eta}(t) - [v(t),\eta(t)] \tag{31}$$

The minimal geodesic $g(t)$ should satisfy to minimize the energy function $\int_0^1 \|v(t)\|^2 dt$

$$\begin{aligned}
&\frac{d}{d\varepsilon}\int_0^1 \|v(t)+\varepsilon \cdot dv(t,\varepsilon)/dt\|^2 dt \,|_{\varepsilon=0} \\
&= 2\int_0^1 < v(t), dv(t,\varepsilon)/dt > dt \,|_{\varepsilon=0} \\
&= 2\int_0^1 < v(t), \dot{\eta}(t) > dt + 2\int_0^1 < [v(t), v^T(t)], \eta(t) > dt \\
&= 2\int_0^1 < -\dot{v}(t) + [v(t), v^T(t)], \eta(t) > dt = 0
\end{aligned} \tag{32}$$

with using integration by parts and the boundary conditions on the perturbation $\eta(0)=0$ and $\eta(1)=0$. Since this is zero over all perturbations $\eta(t)$ ， then we have

$$\begin{aligned}
\dot{v}(t) &= v(t)v^T(t) - v^T(t)v(t) \\
&= [v(t), v^T(t)]
\end{aligned} \tag{33}$$

End

6.2 Proof of (23)

Paying attention to the definition of the directional derivative and using compound derivative chain rule, we have

$$\begin{aligned}
&J_t^{f_p}(v_t^*) \cdot v_t^* \\
&= \partial(I \circ t \circ \mathrm{Rexpp}(v_t^*)(q)) \big/ \partial q^T \,|_{q=\mathrm{Rexpp}(-v_t^*)\circ \mathrm{Rexpp}(v_t^*)(p)} \\
&\quad \cdot \partial w(s,p) \big/ \partial s^T \,|_{s=\pi(\mathrm{Rexpp}(-v_t^*)\circ \mathrm{Rexpp}(v_t^*))} \cdot \partial \pi(\mathrm{Rexpp}(-v_t^*) \cdot \mathrm{Rexpp}(v)) \big/ \partial v^T \,|_{v=v_t^*} \cdot v_t^* \\
&= \nabla_p^T (I \circ t^*) \cdot \partial w(s,p) \big/ \partial s^T \,|_{s=\pi(\mathrm{Rexpp}(-v_t^*)\circ \mathrm{Rexpp}(v_t^*))} \cdot \partial \pi(\mathrm{Rexpp}(-v_t^*) \cdot \mathrm{Rexpp}(v_t^* + t \cdot v_t^*)) \big/ \partial v^T \,|_{t=0} \\
&= (\nabla_p^T I + \varepsilon) \cdot J^{w_p} \cdot \partial \pi(\mathrm{Rexpp}(t v_t^*)) \big/ \partial t \,|_{t=0} \\
&\approx \nabla_p^T I \cdot J^{w_p} \cdot \mathrm{e}_\pi \cdot v_t^*
\end{aligned} \tag{34}$$

where ε is a noise term of image to be discarded.

7. References

Buenaposada, J. M. & Baumela, L. (2002). Real-time tracking and estimation of plane pose. In: *Proceedings of 16th International Conference on Pattern Recognition*. Canada: IEEE Computer Society Press, 2: 697－700.

Baker, S & Matthews, M. (2004). Lucas-Kanade 20 years on: a unifying framework. *International Journal of Computer Vision*, 56(3):221—255.

Berger, M. (2003). *A panoramic view of Riemannian geometry*. Berlin: Springer.

Begelfor, E. & Werman, M. (2005). M. How to put probabilities on homographies. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 27(10):1666—1670.

Benhimane S, & Malis E. (2007). Homography-based 2D visual tracking and servoing. *The International Journal of Robotics Research*, 26(7): 661—676.

Eduardo, Jaime O A. (2007). Lie algebra approach for tracking and 3D motion estimation using monocular vision. *Image and Vision Computing*, 25(6): 907—921

Gabay, D. (1982). Minimizing a differentiable function over a differential manifold. *Journal of optimization theory and applications*. 37(2): 117—217.

Grenander, Ulf.; Miller, M. I. & Srivastava A. (1998). Hilbert—Schmidt lower bounds for estimators on matrix Lie groups for ATR. IEEE Trans on Pattern Analysis and Machine Intelligence, 20(8): 790—802.

Hager, G. D. & Belhumeur, P. N. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 20(10):1025—1039

Helgason, S. (1978). *Differential geometry, Lie Groups, and symmetric spaces.* Academic Press, 98—104.

Mann, S & Picard, W. (1997). Video orbits of the projective group: a simple approach to featureless estimation of parameters. *IEEE Trans on Image Processing*, 6(9):1281—1295

Michael Gleicher. (1997). Projective registration with difference decomposition. In: Proceedings of *IEEE Computer Society Conference on Computer Vision and pattern Recognition.*

Mahony, R. & Manton, J. H. (2002). The geometry of the Newton method on noncompact Lie Group. *Journal of Global Optimization*, 23(3-4): 309—327.

Owren, B. & Welfert, B. (1996). The Newton iterations on Lie Groups. Technical Report Numerics.

Smith, S. T. (1993). Geometric Optimization methods for adaptive filtering, *PhD Thesis*, Cambridge: University of Harvard.

Seok, L. Minseok, C & Kim Hyungmin et al. Geometric direct search algorithms for image registration. *IEEE Trans. On Image Processing*, 2007, 16(9): 2215—2227.

Trucco, E. & Plakas, K. (2006). Video tracking: a concise survey. *IEEE Trans on Oceanic Engineering*, 31(2): 520—529.

Vercauteren, T.; Pennec, X., & Malis, E., etal. (2007). Insight into efficient image registration techniques and the demons algorithm. Proc. IPMI'07. Volume 4584 of LNCS. (July 2007) 495–506.

Wang, H.C. (1969). Discrete nilpotent subgroups of Lie groups. *Journal of Differential Geometry* 3, pp. 481-492.

Yaguang, Y. (1999). Optimization on Riemannian manifold. *Proceedings of the 38th Conference on Decision & Control*. USA: 1: 888—893.

Yean, L. P. (2005). Geometric optimization for computer vision, *PhD Thesis*, Canberra: Australian National University.

Zefran, M & Kumar, V. & Croke, C. B. (1998). On the generation of smooth three-dimensional rigid body motions. *IEEE Transactions on Robotics and Automation*, 1998, 14(4):576 – 589

10

Learning Pattern Classification Tasks with Imbalanced Data Sets

Giang Hoang Nguyen, Abdesselam Bouzerdoum and Son Lam Phung
University of Wollongong
Australia

1. Introduction

This chapter is concerned with the class imbalance problem, which has been recognized as a crucial problem in machine learning and data mining. The problem occurs when there are significantly fewer training instances of one class compared to another class. Most machine learning algorithms work well with balanced data sets since they aim to optimize the overall classification accuracy or a related measure. For imbalanced data sets, the decision boundary established by standard machine learning algorithms tends to be biased towards the majority class; therefore, the minority class instances are more likely to be misclassified.

There are many problems that arise from learning with imbalanced data sets. The first problem concerns measures of performance. Evaluation metrics are known to play a vital role in machine learning. They are used to guide the learning algorithm towards the desired solution. Therefore, if the evaluation metric does not take the minority class into consideration, the learning algorithm will not be able to cope with class imbalance very well. With standard evaluation metrics, such as the overall classification accuracy, the minority class has less impact compared to the majority class. The second problem is related to lack of data. In an imbalanced training set, a class may have very few samples. As a result, it is difficult to construct accurate decision boundaries between classes. For a class consisting of multiple clusters, some clusters may contain a small number of samples compared to other clusters; therefore, the lack of data can occur within the class itself. The third problem in learning from imbalanced data is noise. Noisy data have a serious impact on minority classes than on majority classes. Furthermore, standard machine learning algorithms tend to treat samples from a minority class as noise.

In this chapter, we review the existing approaches for solving the class imbalance problem, and discuss the various metrics used to evaluate the performance of classifiers. Furthermore, we introduce a new approach to dealing with the class imbalance problem by combining both unsupervised and supervised learning. The rest of the chapter is organized as follows. Section 2 describes the problems caused by class imbalance. Section 3 reviews current state-of-the-art techniques for tackling these problems. Section 4 describes existing classification performance measures for imbalanced data. Section 5 describes our proposed learning approach to handle the class imbalance problem. Section 6 presents experimental results, and Section 7 gives concluding remarks.

2. Class Imbalance Problems

Class imbalance occurs when there are significantly fewer training instances of one class compared to other classes. In some applications, class imbalance is an intrinsic property. For example, in credit card usage data there are very few cases of fraud transactions as compared to the number of normal transactions. However, imbalanced data can also occur in areas that do not have an inherent imbalance problem. Instead, the imbalance is mainly caused by limitations in collecting data, such as cost, privacy, and the large effort required to obtain a representative data set. Class imbalance presents several difficulties in learning, including imbalanced in class distribution, lack of data, and concept complexity. These factors are explained in more detail in the following subsections.

2.1 Imbalance in class distribution

The class imbalance problems can arise either from between classes (inter-class) or within a single class (intra-class). We first discuss issues related with inter-class imbalance, where the number of examples of one class is much larger than the number of examples of another class, namely the minority class. The degree of imbalance can be represented by the ratio of sample size of the minority class to that of the majority class. Most classification techniques such as decision tree, discriminant analysis and neural networks assume that the training samples are evenly distributed amongst different classes. However, in real-world applications, the ratio of minority to majority samples can be as low as 1 to 100, 1 to 1000, or 1 to 10,000 (Chawla et al., 2004). Hence, the standard classifiers are affected by the prevalent classes and tend to ignore or treat the small classes as noise. Weiss and Provost investigated the relationship between the imbalance ratio of training samples in each class and classifier performances, in terms of overall accuracy and area under the ROC curve (AUC) (Weiss and Provost, 2003). They used a decision-tree classifier and tested it on a number of data sets from the UCI Repository (Asuncion and Newman, 2007). Their experimental results indicated that the ratio of samples in each class depends on the evaluation metrics used. When the performance is measured using classification accuracy, the best ratio is near the natural ratio; on the other hand, when the AUC measure is used, the best ratio is near the balanced ratio. Visa and Ralescu also reported similar results using fuzzy classifiers (Visa and Ralescu, 2005). However, we should note that the imbalance ratio between classes is not the only factor that reduces classification performance; other factors such as training size and concept complexity also affect performance.

In tasks that involve learning a concept or detecting an event, data imbalance can appear within a single class. The within-class imbalance problem occurs when a class consists of several sub-clusters or sub-concepts and these sub-clusters do not have the same number of samples (Japkowicz, 2001). The within-class and between-class imbalances together are known as the problem of small disjuncts (Holte et al., 1989), in which classifiers are biased towards recognizing large disjuncts correctly, but overfitting and misclassifying samples represented by small disjucts. In most classification tasks, the presence of within-class imbalance is implicit. It is known to have negative effects on the performance of standard classifiers and increases the complexity of concept learning (Yoon and Kwek, 2007). However, most existing methods for class imbalance focus mainly on rectifying the between-class imbalance, and ignore the case where imbalance occurs within each class.

2.2 Lack of data

One of the primary problem when learning with imbalanced data sets is the associated lack of data where the number of samples is small (Weiss, 2004). In a given classification task, the size of data set has an important role in building a good classifier. Lack of examples, therefore, makes it difficult to uncover regularities within the small classes. Fig. 1 illustrates an example of the problem that can be caused by lack of data. Fig. 1 (a) shows the decision boundary (dashed line) obtained when using sufficient data for training, whereas Fig. 1 (b) shows the result when using a small number of samples. When there is sufficient data, the estimated decision boundary (dashed line) approximates well the true decision boundary (solid line); whereas, if there is a lack of data, the estimated decision boundary can be very far from the true boundary. In fact, it has been shown that as the size of training set increases, the error rate caused by imbalanced training data decreases (Japkowicz and Stephen, 2002). Weiss and Provost conducted experiments on twenty six data sets, taken from the UCI repository, to investigate the relationship between the degree of class imbalance and training set sizes (Weiss and Provost, 2003). They showed that when more training data become available, the classifiers are less sensitive to the level of imbalance between classes. This suggests that with sufficient amount of training data, the classification system may not be affected by high imbalance ratio.

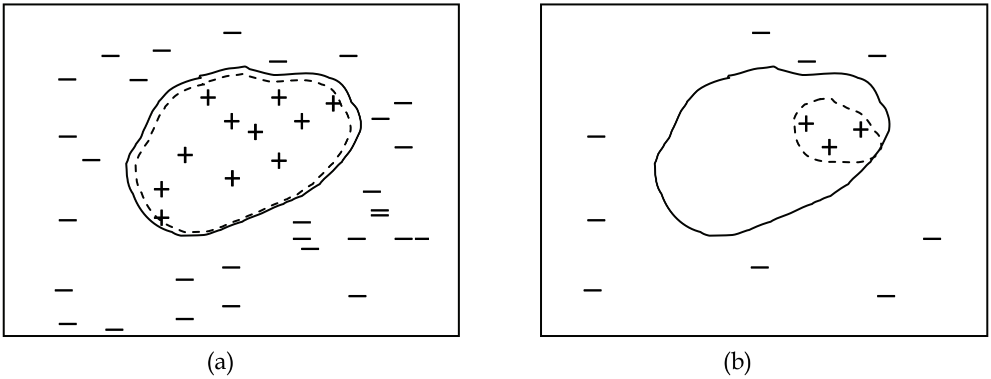


Fig. 1. The effect of lack of data on class imbalance problem; the solid line represents the true decision boundary and dashed line represents the estimated decision boundary.

2.3 Concept complexity

Concept complexity is an important factor in a classifier ability to deal with imbalanced problems. Concept complexity in data corresponds to the level of separability of classes within the data. Japkowicz and Stephen reported that for simple data sets that are linearly separable, classifier performances are not susceptible to any amount of imbalance (Japkowicz and Stephen, 2002). Indeed, as the degree of data complexity increases, the class imbalance factor starts impacting the classifier generalization ability. High complexity refers to inseparable data sets with highly overlapped classes, complex boundaries and high noise level. When samples of different classes overlap in the feature space, finding the optimum class boundary becomes hard. In fact, most accuracy-driven algorithms bias toward the

prevalent class. That is, they improve the overall accuracy by assigning the overlapped area to the majority class, and ignore or treat the small class as noise (Murphey et al., 2007).
The class imbalance problem is more significant when the data sets have a high level of noise. Noise in data sets can emerge from various sources, such as data samples are poorly acquired or incorrectly labeled, or extracted features are not sufficient for classification. It is known that noisy data affect many machine learning algorithms; however, Weiss showed that noise has even more serious impact when learning with imbalanced data (Weiss, 2004). The problem occurs when samples from the small class are mistakenly included in the training data for the majority class, and vice versa. For the prevalent class, noise samples have less impact on the learning process. In contrast, for the small class it takes only a few noise samples to influence the learned sub-concept. For a given data set that is complex and imbalanced, the challenge is how to train a classifier that correctly recognizes samples of different classes with high accuracy.

3. Existing approaches

To address the problems associated with imbalanced data sets, many studies have been conducted to improve traditional learning algorithms. In this section, we review various approaches, which have been proposed both at the data level, such as re-sampling and combinations, and at the algorithmic level, such as recognition-based approach, cost-sensitive learning and boosting.

3.1 Recognition-based approach

As discussed previously, certain discriminative learners such as neural networks, decision trees, support vector machines and fuzzy classifiers tend to recognize the majority class instances since they are trained to achieve the overall accuracy, to which the minority class contributes very little. A recognition-based or one-class learning approach is another alternative solution where the classifier is modeled on the examples of the target class (the small class) in the absence of examples of the non-target class. One of the early systems that utilize this recognition-based approach was proposed in (Japkowicz et al., 1995). It uses neural networks and attempts to learn only from the target class examples and thus recognizing the target concept, rather than to differentiating between majority and minority instances of a concept. One-class learning approach is also applied to autoencoder-based classifiers (Eavis and Japkowicz, 2000), SVMs (Raskutti and Kowalczyk, 2004), and ensemble one-class classifiers (Spinosa and Carvalho, 2005). Here similar patterns from positive instances of a concept are learnt, classifiers are then presented with unseen samples, classification is accomplished by imposing a threshold on the similarity value. A too high threshold will result in misclassifying positive samples, while a too low threshold will include more negative samples. Since threshold draws the boundaries that separate the two classes, choosing an effective threshold is crucial in one-class learning. Japkowicz shows that one-class learning approach to solving the imbalanced class problem is better than discriminative (two-class learning) approach (Japkowicz, 2001). However, recognition-based approach cannot apply to many machine learning algorithms such as, decision tree, Naive Bayes, and associative classifications. These classifiers are not constructed from only samples of one-class.

3.2 Cost-sensitive learning

In many applications such as medical diagnosis, fraud detection, intrusion prevention and risk management, the primary interest is in fact in the small classes. In these applications, it is not only the data distributions that are skewed, but so are the misclassification costs. Most classical learning algorithms assume that all misclassification errors cost equally, and ignore the difference between types of misclassification errors. One practical solution to this problem is to use cost-sensitive learning methods (Elkan, 2001).

A cost-sensitive learning technique takes costs, such as misclassification cost, into consideration during model construction and produces a classifier that has the lowest cost. Let $C(i, j)$ denote the cost of estimating an example from class i as class j. In a two class problem, $C(+, -)$ signifies the cost of misclassifying a positive sample as the negative sample, and $C(-, +)$ denotes the cost of the contrary case. Cost-sensitive learning methods take advantage of the fact that it is more expensive to misclassify a true positive instance than a true negative instance, that is $C(+, -) > C(-, +)$. For a two-class problem, a cost-sensitive learning method assigns a greater cost to false negatives than to false positives, hence resulting in a performance improvement with respect to the positive class.

Existing cost-sensitive learning for dealing with imbalanced data sets can be divided into two different categories. The first category consists of learning algorithms that are designed to optimize a cost-sensitive function directly. One example is cost-sensitive decision tree, proposed in (Ling and Li, 2004) that directly takes costs into model building. The misclassification costs are used to choose the best attribute as a root of the tree. The second category is a collection of existing cost-insensitive learning algorithms that are converted into cost-sensitive ones. This category, also known as cost-sensitive meta-learning, can be further divided into sampling, weighting, thresholding, and ensemble learning. Methods in the weighting group (Alejo et al., 2007), convert sample-dependent costs into sample weights; in other words, they assign heavier weights to the minority training instances. Different weighting strategies have been reported: Nguyen and Ho proposed to weight samples of the minority class based on the local data distributions (Nguyen and Ho, 2005), and others suggested to weight training samples based on posterior probability (Tao et al., 2005). Zhou and Lui conducted a rigorous comparison on the effects of oversampling, and under-sampling, threshold-moving and ensemble classifiers in training cost-sensitive neural networks (Zhou and Liu, 2006). They find that in training cost-sensitive neural networks, threshold-moving and ensemble learning are relatively good choices in both two-class and multi-class tasks. However, like many other solutions, they also have some drawbacks. Cost-sensitive learning approach assumes the misclassification costs are known. In practice, specific cost information is often unavailable because costs often depend on a number of factors that are not easily compared. Moreover, Weiss found that cost-sensitive classifiers may lead to over fitting during training (Weiss, 2004).

3.3 Sampling

One of the common approaches to class imbalance problem is sampling. The key idea is to pre-process training data to minimize any discrepancy between the classes. In other words, sampling methods modify the prior distributions of the majority and minority class in the training set to obtain a more balanced number of instances in each class.

Basic sampling methods. The two basic methods of reducing class imbalance in training data are under-sampling and over-sampling. Under-sampling extracts a smaller set of majority instances while preserving all the minority instances. Under-sampling is suitable for large-scale applications where the number of majority samples is very large and lessening the training instances reduces the training time and storage. However, a drawback with under-sampling is that discarding instances may lead to loss of informative majority class instances and degrade classifier performance.

In contrast, over-sampling increases the number of minority instances by replicating them (Chawla et al., 2002, Japkowicz and Stephen, 2002). The advantage is that no information is lost, all instances are employed. However, over-sampling also has its own drawbacks. By creating additional training instances, over-sampling leads to a higher computational cost. Moreover, if some of the small class samples contain labeling errors, adding them will actually deteriorate the classification performance on the small class (Chawla et al., 2004). Lastly, over-sampling duplicates majority instances rather than introducing new data, so it does not address the underlying lack of data.

Despite the fact that sampling methods are widely used for tackling class imbalance problems, there is no established way to determine the suitable class distribution for a given data set (Weiss and Provost, 2003). The optimal class distribution is dependent on the performance measures and varies from one dataset to another. However, effectively sampling training instances can improve and overcome some of the weaknesses discussed above. Next, we describe some of the advanced sampling methods that are reported to be superior to random over-sampling and under-sampling.

Advanced sampling methods

In advanced sampling, instances are added or removed adaptively. Advanced sampling methods may also combine under-sampling and over-sampling techniques. One of the popular over-sampling approaches is SMOTE (Synthetic Minority Over-sampling TEchnique), which attempts to add information to the training set by introducing new, non-replicated minority class examples (Chawla et al., 2002). Generative over-sampling, proposed in (Liu et al., 2007), is a variation of SMOTE. It creates new data points by learning from available training data. In other words, a probability distribution is selected to model the available minority class examples. Then new data points are generated from this model. A drawback of this method is that when the number of examples of the minority class is not adequate, the probability distribution estimates that model the actual data distributions may not be accurate.

In an under-sampling scheme, instead of eliminating instances randomly, Yu and co-workers proposed a different method to re-sampling the majority class instances (Yu et al., 2007). The authors proposed to use vector quantization, which is a lossy compression method, on the majority class to build a set of representative local models and use them for training the SVM. Another informative re-sampling technique is cluster-based under-sampling (Yen and Lee, 2009). In this technique, clustering is employed for selecting the representative training samples to improve the predictive accuracy for the minority class. Yen and Lee reported that this approach empirically outperforms other under-sampling techniques. Yoon and Kwek also proposed to use clustering to reduce the imbalanced ratio, called Class Purity Maximization (CPM) (Yoon and Kwek, 2005). CPM partitions the data space into clusters, and filters out regions in the data space that consist of high majority class purity. Hence, only regions containing minority samples are used to build a predictive model. CPM reduces the imbalance ratio and makes the learning task more tractable.

Active learning is also another solution to class imbalance problem. Ertekin et al. proposed using active learning to select informative samples of the training set (Ertekin et al., 2007). Similarly to re-sampling, active learning query technique creates balanced training sets at the early stages of the learning process. This technique focuses on query instances near the classification boundary rather than selecting randomly any instance. Active learning gives the learners the ability to select examples adaptively. Furthermore, the risk of losing important information is reduced, compared with the under-sampling approach. Active learning does not create extra data as in oversampling.

3.4 Ensemble-learning methods

Another alternative solution for the class imbalance problem is ensemble-learning, in which multiple classifiers are trained from the original data and their predictions are combined to classify new instances. Boosting (Freund and Schapire 1996) and bagging (Breiman, 1996) are two widely known ensemble-based approaches. Boosting algorithms, such as AdaBoost (Leskovec and Shawe-Taylor, 2003), improve performance of weak classifiers by forcing the learners to focus more on the difficult examples. Boosting algorithms have been adapted to address the problem with small classes. At each boosting iteration, the distribution of training data is altered by updating the weight associated with each sample. Examples of algorithms that use boosting to address the class imbalance problems are SMOTEBoost (Chawla et al., 2002), DataBoost-IM (Guo and Viktor, 2004), and cost-sensitive booting (Sun et al., 2007). Both DataBoost-IM and SMOTEBoost improve boosting by combining data generation and boosting procedures. To avoid over fitting, SMOTEBoost alters the data distribution by adding new minority class samples using the SMOTE algorithm (Chawla et al., 2002).

DataBoot-IM, proposed by (Guo and Viktor, 2004), generates data to balance not only the class distribution but also the total weight within the class. Through experiments on seventeen data sets, the authors showed that DataBoost method does not sacrifice one class over the other but improve the predictive accuracies of both majority and minority classes.

A cost-sensitive booting algorithm for classification of imbalanced data was proposed in (Sun et al., 2007), in which misclassification costs are integrated into AdaBoost learning. The AdaBoost weight-update strategy is altered so that the weights of misclassified samples from the small class increase at a higher rate compared to those of the prevalent class. The weights of correctly classified samples from the small class reduce at a lower rate, compared to those from the prevalent class.

Bagging is one of the ensemble-based meta-learning algorithms. Most current bagging methods use a similar learning procedure: re-sampling subsets from a given training set, building multiple base classifiers on those subsets, and combining their predictions to make final prediction (Breiman, 1996). Several algorithms based on a variety of sampling strategies are proposed, for example roughly balanced (RB) bagging (Hido and Kashima, 2008), underBagging (Liu et al., 2006), overBagging and SMOTEBagging (Wang and Yao, 2009). In underBagging, each subset from the training set is created by under-sampling the majority classes randomly to build a classifier. RB bagging is a variation to underBagging: it makes use of both minority samples and under-sampling majority samples. However, RB bagging uses an effective under-sampling technique based on negative binomial distributions. In comparison, overBagging forms subsets simply by over-sampling the minority classes randomly. SMOTEBagging (Wang and Yao, 2009) differs from underBagging and overBagging in that it involves generating synthetic instances during subset construction. The main advantage of bagging is that it maintains the class distribution of the training set on which bagging is applied. However, bagging relies on a simple strategy that is very limited for dealing with class imbalance problem, except from changing the bag size and sampling step.

4. Classifier performance measures

Evaluation metrics play an important role in machine learning. They are used to evaluate and guide the learning algorithms. If the choices of metrics do not value the minority class, then the learning algorithms will not be able to handle the imbalance problem very well. The commonly used metric for these purposes is the overall classification rate (i.e. accuracy). However, on an imbalanced data set, the overall classification rate is no longer a suitable metric, since the small class has less effect on accuracy as compared to the prevalent class. Weiss and Provost conducted an empirical study on twenty-six data sets, and showed that using the overall accuracy measure leads to poor performance for the minority class (Weiss and Provost, 2003). Therefore, other metrics have been developed to assess classifiers performance for imbalanced data sets. A variety of common metrics are defined based on the *confusion matrix* (also called a contingency table). A two-by-two confusion matrix is shown in Table 1.

		True class	
		Positive	Negative
Prediction class	Positive	True Positives (TP)	False Positives (FP)
	Negative	False Negatives (FN)	True Negatives (TN)

Table 1. Confusion matrix for a two-class classification task.

Among the various evaluation criteria, the measures that are most relevant to imbalanced data are precision, recall, F-measure, sensitivity, specificity, geometric mean, ROC curve, AUC, and precision-recall curve. These metrics share a commonality in that they are all class-independent measures.

Precision, recall and F-measure. These metrics arise from the fields of information retrieval. They are used when performance of positive class (the minority class) is considered, since both precision and recall are defined with respect to the positive class.

- *Precision* of a classifier is the percentage of positive predictions made by the classifier that are correct.

$$\text{Precison} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- *Recall* is the percentage of true positive patterns that are correctly detected by the classifier.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- *F-measure* is defined as the harmonic mean of recall and precision (Fawcett, 2006). A high F-measure value signifies a high value for both precision and recall.

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Sensitivity, Specificity and Geometric mean. These measures are utilized when performance of both classes is concerned and expected to be high simultaneously. The geometric mean (G-mean) metric was suggested in (Kubat and Matwin, 1997) and has been used by several researchers for evaluating classifiers on imbalanced data sets (Ertekin et al., 2007, Karagiannopoulos et al., 2007, Su and Hsiao, 2007). G-mean indicates the balance between classification performances on the majority and minority class. This metric takes into account both the *sensitivity*, (the accuracy on the positive examples) and the *specificity* (the accuracy on the negative examples):

$$\text{Sensitivity} = \text{Recall}$$

$$\text{Specificity} = 1 - \frac{\text{FP}}{\text{Total Negatives}}$$

$$\text{G-means} = \sqrt{\text{Sensitivity} \times \text{Specificity}}$$

ROC and AUC. The receiver operating characteristic (ROC) and the area under the ROC curve (AUC) are the two most common measures for assessing the overall classification performance (Weiss, 2004). The ROC is a graph showing the relationship between benefits (correct detection rate or true positive rate) and costs (false detection rate or false positive rate) as the decision threshold varies. The ROC curve shows that for any classifier, the true positive rate cannot increase without also increasing the false positive rate. The true positive rate is the same as recall, and the false detection rate is equal to

$$\text{FDR} = \frac{\text{FP}}{\text{Total Negatives}} \quad .$$

A ROC curve gives a visual indication if a classifier is superior to another classifier, over a wide range of operating points. However, a single metric is sometimes preferred when comparing different classifiers. The area under the ROC curve (AUC) is employed to summarize the performance of a classifier into a single metric. The AUC does not place more weight on one class over the another. The larger the AUC, the better is the classifier performance.

Precision-Recall (PR) curve. Precision-recall curve is used in information retrieval in a similar fashion as the ROC curve. The PR curve depicts the relationship between precision and recall as the classification threshold varies.

Apart from the above evaluation metrics, a number of new evaluation metrics have been proposed to take small class size into account when evaluating the end result. For imbalanced data sets, not only the class distribution but also the misclassification costs are skewed. Hence, Weng and Poon introduced a new metric, weighted-AUC, that can take into account the cost bias when evaluating classifier performance (Weng and Poon, 2008). Some other authors had suggested that the ROC curve alone is not sufficient, and the effect of imbalance class distribution should be analyzed when comparing different learning algorithms (Landgrebe et al., 2004). Therefore, they proposed to use costs that are dependent on class distribution such as positive fraction together with ROC curve. The positive fraction is defined as the fraction of objects that are positively labeled.

Other metrics such as rank metrics, rank prop and soft ranks are proposed for training and model selection (Caruanan, 2000). These metrics prevent learners from mainly optimizing classification performance on the dominant class. Comparisons of several evaluation metrics were conducted by Liu and Shriberg and found that a single measure such as precision, recall, F-measure, sensitivity, specificity, G-mean or AUC provide limited information, since each measure is designed to assess one particular property or decision point (Liu and Shriberg, 2007). Hence, to analyze and compare learning algorithms involving class imbalance, it is necessary to combine different metrics and performance curves such as ROC and PR.

5. The proposed learning approach for imbalanced data set

In this chapter, we introduce a new learning approach that aim to tackling the class imbalance problem. In our approach, we first propose a new under-sampling method based on clustering. Here, a clustering technique is employed to partition the training instances of each class independently into a smaller set of training prototype patterns. Then a weight is assigned to each training prototype to address the class imbalance problem.. The weighting strategy is introduced in the cost function such that the class distributions become roughly even. In the extreme imbalance cases, where the number of minority instances is small, we apply unsupervised learning to resample only the majority instances, and select cluster centers as prototype samples, and keep all the small class samples.

The proposed learning approach, which combines unsupervised and supervised learning to deal with the class imbalance problem, can be applied on any classifier model. In this chapter, we apply the proposed learning approach to train feed-forward neural networks, which is a classification model that has been used extensively in pattern recognition. Based on the proposed learning approach, we derive and analyze the resilient back-propagation training algorithm for feed-forward neural networks. The algorithm is implemented and tested on some benchmark data sets.

5.1 Under-sampling based on clustering

Suppose that a multi-layer feed-forward neural network is to be trained on a given training data set D of size M

$$D_M = \{(x_i, \mathbf{d}_i) | x_i \in \Re^N, i = 1, 2, \ldots, M\}$$

where x_i is the *i*-th input pattern and $\mathbf{d}_i$ is the corresponding desired output vector. Let $\mathbf{w}$ be a vector consisting of all free network parameters, including weights and biases. The objective of supervised learning is to find a vector $\mathbf{w}_o$ that minimizes a cost function. A common objective function is the *mean square error* (MSE), defined as

$$E(\mathbf{w}) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (y_{ij} - d_{ij})^2, \tag{1}$$

where *N* is the number of neurons in the output layer, and y_{ij} is the network output.

When the numbers of training instances of different classes are uneven, the contribution from each class to the objective function is not equal. In a two-class problem, the majority class has a significant effect in the optimization process. Hence, we propose a more efficient algorithm for training feed-forward neural networks. In this approach, a pre-processing step is introduced to obtain a more balanced number of samples in each class. To this end, unsupervised *clustering* is applied to training samples to extract cluster centers that yield a compact representation of the majority classes.

Here, clustering is applied independently to each class. Therefore, each cluster contains samples from the same class, and each class can have several clusters. We deal with imbalanced data sets by assigning the same number of clusters to each class. When the number of minority samples is small, we only apply unsupervised clustering to resample the majority instances, and retain all the minority samples. After clustering, the data set is reduced to *K* exemplars; each is represented by a cluster *centroid* c_k and cluster *size* z_k. Here, the cluster size z_k is simply the number of training samples in the cluster. Next, we present the resilient back-propagation training algorithm that integrates the cluster centroids and sizes into the learning rule.

5.2 Modified training algorithm

In the supervised learning stage, training samples are replaced by a set of cluster centroids, which is then presented to the network along with the target outputs. To compensate for the information lost during the clustering process, weights for each class are introduced in the cost function, which is modified as follows.

$$E(\mathbf{w}) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ki} - d_{ki})^2 \times p_k \, , \tag{2}$$

where d_{ki} is the *i*-th element of the target or desired output vector $\mathbf{d}_k$, and p_k is the cluster weight. The cluster weight is defined as follow,

$$p_k = \frac{z_k}{\sum_{i=1}^{N_{cl}} w_i \gamma_{ki}}, \qquad k = 1, ..., K \tag{3}$$

where N_{cl} is number of classes in the training set, w_i is the size of class *i*, and γ_{ki} is the degree of membership of cluster *k* in class *i*:

$$\gamma_{ki} = \begin{cases} 1 & \text{if } c_k \in \text{class } i \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Numerous optimization algorithms for minimizing *E* can be derived to train feed-forward neural networks, such as gradient descent (GD), gradient descent with momentum and variable learning rate (GDMV), resilient back-propagation (RPROP), and Levenberg-Marwquardt (LM). We have implemented and analyzed these algorithms based on our proposed learning approach (Nguyen et al., 2008). In this chapter, we only perform the analysis on the resilient back-propagation method (RPROP) to train the feed-forward neural network, refer to (Nguyen et al., 2008) for other training methods. The RPROP training algorithm updates network weights and biases according to $w(t+1) = w(t) + \Delta w(t)$. Because details of the standard RPROP algorithms can be found in (Riedmiller and Braun, 1993), we only summarize its main characteristics here.

Resilient back-propagation: Weight update depends only on the sign of the gradient

$$\Delta w_i(t) = -\text{sign}\left\{\frac{\partial E}{\partial w_i}(t)\right\} \times \Delta_i(t), \tag{1}$$

where $\Delta_i(t)$ is on adaptive step specific to weight $w_i(t)$.

6. Experiments

In this section, we apply the proposed learning approach to four benchmark problems, taken from UCI database repository (Asuncion and Newman, 2007). The benchmarks used are the liver disorder, hepatitis, Wisconsin diagnostic breast cancer, and Pima Indian diabetes data sets. These data sets are summarized in Table 2. Our aim is to study the generalization capability of the proposed approach in tackling the class imbalance problem, compared to the standard approach for training feed-forward neural networks.

The comparison is based on a five-fold cross validation in the classification tasks. For each fold, the data set is partitioned into 60% for training set, 20% for validation set and 20% for test set. Several networks are trained and the best performing network on the validation set is selected to be evaluated on the test set. The average classification rate on the test set, over the five folds, is used as an estimate of generalization performance. Since the overall classification rate is not the most suitable tool for imbalanced data, other measures are also used, including the geometric mean and F-measure.

Data sets	Size	Features	Class distribution	Imbalanced ratio (Majority/Minority)
Liver	345	6	145/200	1.38
Hepatitis	155	19	32/123	3.84
Pima diabetes	768	8	268/500	1.87
Wisconsin Breast cancer	699	10	241/458	1.90

Table 2. Summary of data sets used in the experiments.

The comparison results of different training algorithms over all data sets are shown in Table 3. The modified training (Mod-RPROP) and the standard training (RPROP) achieve almost similar classification rates (CRs). For examples, in the hepatitis data set, CRs of RPROP and Mod-RPROP are 92.00% and 92.67%, respectively. However, the modified algorithm has higher values of G-mean and F-measure than its counterpart. In the hepatitis data set, the G-mean values of RPROP and Mod-RPROP are 90.80% and 91.65%, and the F-measure value of RPROP and Mod-RPROP are 80.48% and 82.57%, respectively. This finding suggests that the modified training algorithm exhibits good classification rates for all classes.

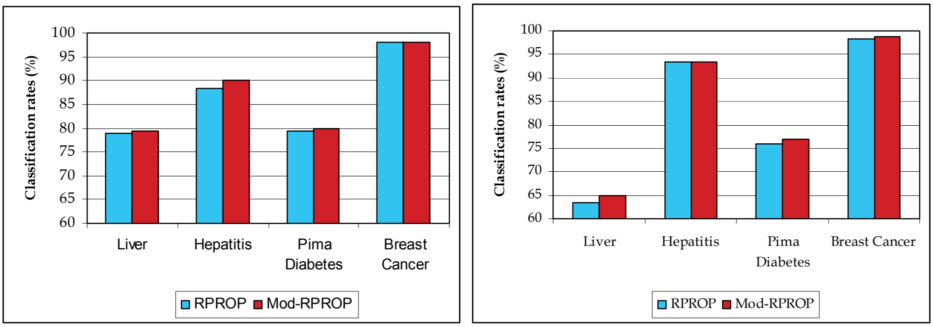
Fig. 2 shows the classification rates of each class over all data sets. The classification rates of positive class (or the sensitivity) as well as classification rates of negative class (or the specificity) are increased. For example, the sensitivity of Mod-RPROP increases by 1.38% in liver data set, and 1.13% in Pima data set, compared to the standard RPROP. In terms of specificity, the Mod-RPROP maintains the accuracies and in some occasions improves them.

Data sets	Overall CR		F-measure		G-means	
	RPROP	Mod-RPROP	RPROP	Mod-RPROP	RPROP	Mod-RPROP
Liver	73.91	74.78	66.53	67.66	70.80	71.79
Hepatitis	92.00	92.67	80.48	82.57	90.80	91.65
Pima Diabetes	80.65	82.22	70.70	72.11	77.60	78.38
Breast Cancer	98.13	98.27	97.35	97.56	98.18	98.39

Table 3. Comparision of standard and reduced training algorithms on benchmark data sets.

Data sets	Specificity		Sensitivity	
	RPROP	Mod-RPROP	RPROP	Mod-RPROP
Liver	79.00	79.50	63.45	64.83
Hepatitis	88.33	90.00	93.33	93.33
Pima Diabetes	79.40	79.80	75.85	76.98
Breast Cancer	98.02	98.02	98.33	98.75

Table 4. Classification rates of each class on benchmark data sets.



(a) (b)

Fig. 2. Comparison of the standard RPROP and Mod-RPROP trainging algorithms on four data sets in terms of (a) classification rates of negative class and (b) classification rates of positive class.

7. Conclusion

In this chapter, we discussed the problems that arise when learning with imbalanced data sets, including between classes imbalance, within-class imbalance, the lack of data, and concept complexity. Then we reviewed various methods and techniques that address the class imbalance problems, both at the data level (re-sampling and combinations) and the algorithmic level (recognition-based approach, cost-sensitive learning and boosting). We also discussed a number of evaluation metrics that have been developed to assess classifier performance on imbalanced data sets. Then we presented a new approach that combines unsupervised clustering and supervised learning to handle imbalanced data set and applied this learning approach for training feed-forward neural networks. The proposed approach can be applied to existing training algorithms. Experimental results show that the proposed approach can effectively improve the classification accuracy of the minority classes, while maintaining the overall classification performance.

8. References

Alejo, R., García, V., Sotoca, J., Mollineda, R. & Sánchez, J. (2007). Improving the Performance of the RBF Neural Networks Trained with Imbalanced Samples, *In Proceedings of Computational and Ambient Intelligence, 9th International Work-Conference on Artificial Neural Networks*, pp. 162-169, San Sebastian, Spain, 2007.

Asuncion, A. & Newman, D. J. (2007) UCI Machine Learning Repository.

Breiman, L. (1996) Bagging Predictors. *Machine learning,* 24**,** 123-140.

Caruanan, R. (2000). Learning from Imbalanced Data: Rank Metrics and Extra Tasks, *The AAAI Workshop on Learning from Imbalanced Data Sets*, pp. 51-57, 2000.

Chawla, N., Japkowicz, N. & Kotez, A. (2004) Editorial: Special Issue on Learning from Imbalanced Data. *Sigkdd Explorations,* 6**,** 1-6.

Chawla, N. V., Bowyer, K., Hall, L. & Kegelmeyer, W. P. (2002) SMOTE: Synthetic Minority Over-sampling TEchnique. *Artificial Intelligence Research,* 16**,** 321-357.

Chong, E. K. P. & Zak, S. H. (1996) *An Introduction to Optimization,* New York, John Wiley and Sons, Inc.

Eavis, T. & Japkowicz, N. (2000). A Recognition-Based Alternative to Discrimination-Based Multi-layer Perceptrons, *The 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, pp. 280-292, London, UK, 2000, Springer-Verlag.

Elkan, C. (2001). The Foundations of Cost-sensitive Learning, *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pp. 73-978, 2001.

Ertekin, S., Huang, J., Bottou, L. & Giles, C. L. (2007). Learning on the Border: Active Learning in Imbalanced Data Classification, *In Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pp. 127-136, Lisbon, Portugal, 2007, ACM Press.

Fawcett, T. (2006) An Introduction to ROC Analysis. *Pattern Recognition Letters,* 27**,** 861-874.

Guo, H. & Viktor, H. L. (2004) Learning from Imbalanced Data Dets with Boosting and Data Generation: the DataBoost-IM Approach. *ACM SIGKDD Explorations Newsletter,* 6**,** 30-39.

Hagan, M. T. & Menhaj, M. B. (1994) Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks,* 5**,** 989-993.

Hido, S. & Kashima, H. (2008). Roughly Balanced Bagging for Imbalanced Data., *In Proceedings of the SIAM International Conference on Data Mining,* pp. 143-152, Atlanta, Georgia, USA, 2008.

Holte, R., Acker, L. & Porter, B. (1989). Concept Learning and the Problem of Small Disjuncts, *In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp., Austin, TX, USA, 1989, Morgan Kaufmann.

Japkowicz, N. (2001). Concept-Learning in the Presence of Between-Class and Within-Class Imbalances, *In Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, pp., London, UK, 2001, Springer-Verlag.

Japkowicz, N., Mayers, C. & Gluck, M. (1995). A Novelty Detection Approach to Classification *In Proceedings of the Fourteenth Joint Conference on Artificial Intelligence*, pp. 518-523, 1995.

Japkowicz, N. & Stephen, S. (2002) The Class Imbalance Problem: A Systematic Study. *Intelligent Data Analysis,* 6**,** 429-449.

Karagiannopoulos, M. G., Anyfantis, D. S., Kotsiantis, S. B. & Pintelas, P. E. (2007). Local Cost Sensitive Learning for Handling Imbalanced Data Sets, *In Proceedings of Mediterranean Conference on Control and Automation*, pp. 1-6, 2007.

Kubat, M. & Matwin, S. (1997). Addressing the Curse of Imbalanced Training Set: One-sides Selection, *In Proceedings of the Fourteenth International Conference on Machine Learning*, pp. 179-186, 1997, Morgan Kaufmann.

Landgrebe, T., Pacli'k, P., Tax, D. J. M., Verzakov, S. & Duin, R. P. W. (2004). Cost-based Classifier Evaluation for Imbalanced Problems, *In Proceedings of The 10th International Workshop on Structural and Syntactic Pattern Recognition and the 5th International Workshop on Statistical Techniques in Pattern Recognition*, pp. 762-770, Lisbon, Portugal, 2004, Springer Verlag, Berlin.

Leskovec, J. & Shawe-Taylor, J. (2003). Linear Programming Boosting for Uneven Datasets, *In Proceedings of The twentieth International Conference on Machine Learning* pp. 456-463, 2003, AAI Press.

Ling, C. X. & Li, C. (2004). Decision Trees with Minimal Costs, *In Proceedings of the Twenty-first International Conference on Machine Learning*, pp. 69, Banff, Alberta, Canada, 2004, ACM.

Liu, A., Ghosh, J. & Martin, C. (2007). Generative Oversampling for Mining Imbalanced Datasets, *In Proceedings of The 2007 International Conference on Data Mining*, pp., Las Vegas, Nevada, USA, 2007, CSREA Press.

Liu, Y., Chawla, N. V., Harper, M., Shriberg, E. & Stolcke , A. (2006) A Study in Machine Learning from Imbalanced Data for Sentence Boundary Detection in Speech. *Computer Speech and Language,* 20**,** 468-494.

Liu, Y. & Shriberg, E. (2007). Comparing Evaluation Metrics for Sentence Boundary Detection, *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 185-188, 2007.

Murphey, Y. L., Wang, H., Ou, G. & Feldkamp, L. (2007). OAHO: and Effective Algorithm for Multi-class Learning from Imbalanced Data, *Proceedings of the International Joint Conference on Neural Networks*, pp. 406-411, Orlando, Florida, USA, 2007, IEEE.

Nguyen, C. & Ho, T. (2005). An Imbalanced Data Rule Learner, *In Proceedings of The 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 617-624, Porto, Portugal, 2005.

Nguyen, G. H., Bouzerdoum, A. & Phung, S. L. (2008). A Supervised Learning Approach for Imbalanced Data Sets, *In Proceeding of International Conference on Pattern Recognition*, pp. 1-4, Florida, USA, 2008.

Raskutti, B. & Kowalczyk, A. (2004) Extreme Re-balancing for SVMs: a Case Study. *ACM Sigkdd Explorations Newsletter,* 6**,** 60-69.

Riedmiller, M. & Braun, H. (1993). A Direct Adaptive Method of Faster Backpropagation Learning: The RPROP Algorithm, *IEEE International Conference on Neural Networks*, pp. 586-591, San Francisco, 1993.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986) Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in The Microstructure of Cognition,* I**,** 318 – 362.

Spinosa, E. J. & Carvalho, A. (2005) Combining one-class classifiers for robust novelty detection in gene expression. *Advances in bioinformatics and computational biology,* 3549**,** 54-64.

Su, C. & Hsiao, Y. (2007) An Evaluation of Robustness of MTS for Imbalanced Data. *IEEE Transaction on Knowledge and Data Engineering,* 19**,** 1321-1332.

Sun, Y., Kamel, M. S., Wong, A. K. C. & Wang, Y. (2007) Cost-sensitive Boosting for Classification of Imbalanced Data. *Pattern Recognition,* 40**,** 3358-3378.

Tao, Q., Gao Wei Wu, Fei Yue Wang & Wang, J. (2005) Posterior Probability Support Vector Machnies for Unbalanced Data. *IEEE Transaction on Neural Networks,* 16**,** 1561-1573.

Visa, S. & Ralescu, A. (2005). The Effect of Imbalanced Data Class Distribution on Fuzzy Classifiers - Experimental Study, *In Proceedings of the IEEE Conference on Fuzzy Systems*, pp. 749-754, 2005.

Wang, S. & Yao, X. (2009). Diversity Analysis on Imbalanced Data Sets by Using Ensemble Models, *In Proceedings of The IEEE Symposium on Computational Intelligence and Data Mining*, pp. 324-331, 2009, IEEE.

Weiss, G. M. (2004) Mining with Rarity: a Unifying Framework. *SIGKDD Explorations and Newsletters,* 6**,** 7-19.

Weiss, G. M. & Provost, F. (2003) Learning When Training Data Are Costly: The Effect of Class Distribution on Tree Induction. *Journal of Artificial Intelligence Research* 19**,** 315-354.

Weng, C. G. & Poon, J. (2008). A New Evaluation Measure for Imbalanced Datasets, *In Proceedings of The Seventh Australasian Data Mining Conference* pp. 27-32, Glenelg, South Australia, 2008, ACS.

Yen, S.-J. & Lee, Y.-S. (2009) Cluster-based Under-sampling Approaches for Imbalanced Data Distributions. *Expert Systems with Applications,* 36**,** 5718-5727.

Yoon, K. & Kwek, S. (2005). An Unsupervised Learning Approach to Resolving the Data Imbalanced Issue in Supervised Learning Problems in Functional Genomics, *In Proceedings of The Fifth International Conference on Hybrid Intelligent Systems*, pp. 303-308, Washington, DC, USA, 2005, IEEE Computer Society.

Yoon, K. & Kwek, S. (2007) A Data Reduction Approach for Resolving The Imbalanced Data Issue in Functional Genomics. *Neural Computing and Applications,* 16**,** 295-306.

Yu, T., Jan, T., Simoff, S. & Debenham, J. (2007) A Hierarchical VQSVM for Imbalanced Data Sets. *In Proceedings of The International Joint Conference on Neural Networks.*

Zhou, Z.-H. & Liu, X.-Y. (2006) Training Cost-sensitive Neural Networks with Methods Addressing The Class Imbalance Problem. *IEEE Transaction on Knowledge and Dada Engineering,* 18**,** 63-77.

Image Kernel for Recognition

Zhu XiaoKai and Li Xiang
National University of Defence Technology
P.R.China

1. Introduction

Kernel-based methods, such as SVM classifier, have been proven to have more predominance of generalization and better performance of classification than traditional methods. As the main point of these technologies, kernel functions can increase the computational power of traditional linear learning machines by projecting the data into a high dimensional feature space, and can transform a non-linear problem into a linear problem (John, S.T. & Nello, C. 2004).

Although kernel functions have been widely used in pattern recognition, they have some weaknesses. Traditional kernel functions only accept one-dimensional vector as their input data. But some real-world data such as image data are often two-dimensional matrices, which can not be directly accepted by the kernel functions unless doing some preprocessing work. One way is to abstract features. One or more features that can denote some information of the image object are combined into a one-dimensional vector, and then a two-dimensional problem becomes a simple one-dimensional problem. This is the common way to do with the image objects. There are many categories of ways to abstract features (Sergios, T. & Konstantinos, K., 2006), including invariant moment, PCA, ICA, statistic analysis, texture analysis, shape analysis, etc, which are not introduced in detail in this paper. But every feature can only be efficient to some special object. How to select the appropriate ones is always a difficult problem. The other way is to decrease the dimensions of data. The simplest method is to treat the image data as a one-dimensional vector. C. Kaynak (1995) divides 32×32 bitmaps of handwritten digits into nonoverlapping blocks of 4 ×4 and counts the number of on-pixels in each block. Then he gets a vector of 64 elements and uses it as the feature vector. This way can only be efficient with data of small size. In fact, many statistic analysis approach of the first way do the same thing. They treat the image as a set of non-relevant pixels and the structural information of two-dimensional data are lost.

In this chapter, we propose a new kind of kernel function that can directly accept image data as input data. Section 2 introduces the traditional RBF kernel function in brief and educes our idea. In Section 3, we describe our kernel function in detail, which is based on the RBF kernel function. In Section 4, the new kernel function is compared with the old approaches on UCI Optical Handwritten Digits dataset and COIL dataset.

2. Traditional Kernel Functions

Our idea comes from the traditional kernel Functions. First, let us see the three types of old ones which are generally used.
Polynomial:

$$K_{poly}(u,v) = (\sigma u \cdot v + r)^d \tag{1}$$

RBF:

$$K_{rbf}(u,v) = \exp(-\frac{\|u-v\|^2}{\sigma^2}) \tag{2}$$

Sigmoid:

$$K_{sig}(u,v) = \tanh(\sigma u \cdot v + r) \tag{3}$$

Here, u and v are one-dimensional vectors, σ and r are parameters. Out purpose is to construct a new form of kernel function where u and v can be two-dimensional data.
In (1) and (3), the operation between u and v is inner product, which can not operate on image data generally. And we don't pay attention to them.
In (2), the operation is $\|u-v\|$, which always indicates the distance between two vectors. And we know that distance between two objects can be considered as the similarity of them. So we get an idea that if the distance or similarity of two image data can be calculated, we can use it to replace the $\|u-v\|$ and the new RBF kernel function can be written as (4).
RBF2D:

$$K_{rbf\,2D}(A,B) = \exp(-\frac{d^2(A,B)}{\sigma^2}) \tag{4}$$

Here, A and B are two-dimensional data of target in images, and $d(A,B)$ is the distance or similarity of them. RBF2D is the new kernel function proposed in this paper that can accept image data.
The particular expression of RBF2D and $d(A,B)$ will be introduced in Section 3.

3. RBF2D

Note that in (4) A and B only appear in the form of $d(A,B)$, so before we get the expression of RBF2D, we should get $d(A,B)$ at first.

3.1 Distance between Image Data

Image data are in the form of matrix. The distance between two matrices can be computed using Frobenius Norm (Horn R.A. & Johnson C.R., 1985) generally.

$$\|A-B\|_F=\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}(a_{i,j}-b_{i,j})^2} \tag{5}$$

Here, A and B are $M \times N$ matrices. And From the formula, we can see that, all elements in A and B are out-of-order, only statistical information is reserved. There is one serious problem with this method. Unlike in a vector or a data set, an element in a matrix has relation to not only the one on the left side and the one on the right side, but also those above it and below it, even those in other directions, as be shown in Figure 1.

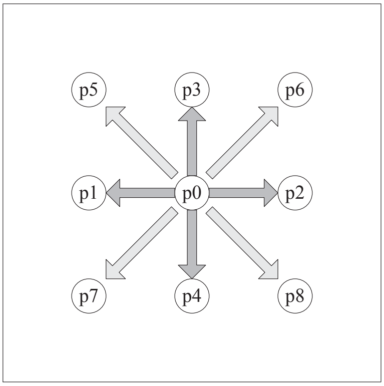


Fig. 1. An element p0 in a matrix and its neighbors p1~p8

Especially for image data, most targets are objects of some shape or structure and cover a region in the image. Structural information is the same important as the statistical one. So we should define a new form of $d(A,B)$ which involves structural information.

Zhou Wang (2004) proposes SSIM. He uses correlation between the two images to quantify the structural similarity. Luminance, contrast, and structure information are included in SSIM. The result shows that SSIM is efficient in quantifying the visibility of differences between a distorted image and a reference image. But the image is treated as a whole entity in SSIM. This will make the algorithm unstable in the following conditions.

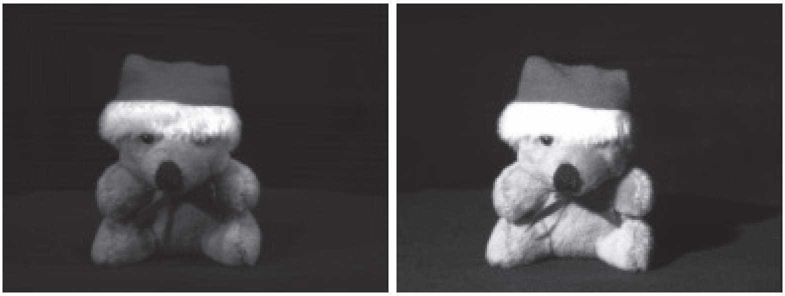


Fig. 2. The same object under different luminance conditions

The two images in Figure 2 are the same object under different luminance conditions. Because the object has an anomaly surface, they look different in luminance, and SSIM doesn't work. For solving this problem, we propose our method which is block-based. Although the two images look different in every block at the corresponding position (in Figure 3), we can increase their similarity after some simple preprocessing work, which will not work on the whole image.

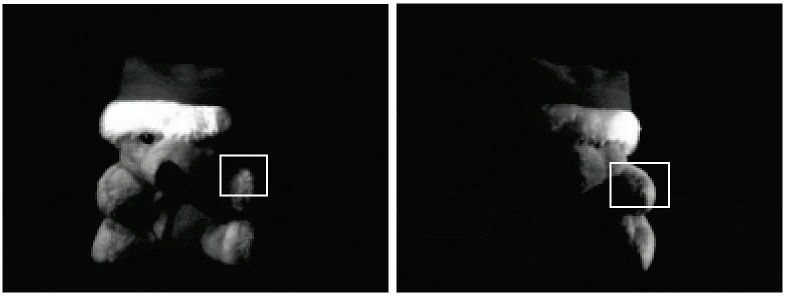


Fig. 3. The same object under different luminance conditions with blocks

So, out definition of $d(A,B)$ based on blocks has the following form, in (6).

$$d^2(A,B) = \sum_{n=1}^{N_{block}} \left\| A_n - B_n \right\|_F^2 \cdot \omega_n \tag{6}$$

Here, A_n and B_n are the data matrices of the nth block. ω_n is the weight of $\left\| A_n - B_n \right\|_F^2$ which can involve the luminance, structure information. This is the main part of our work, and we will discuss it in the following sections. N_{block} is the count of the blocks.

The expression is similar to $\left\| u - v \right\|$, because it will be used in RBF2D which is based on RBF kernel function (in Equation 7). We make $\left\| A_n - B_n \right\|_F^2$ the main part of it and other information as the weight.

$$K_{RBF2D}(A,B) = \exp(-\frac{\sum_{n=1}^{N_{block}} \left\| A_n - B_n \right\|_F^2 \cdot \omega_n}{\sigma^2}) \tag{7}$$

Weight ω_n is the combination of luminance difference weight, content difference weight, self complexity weight, and position weight. The Following sections will discuss each part of $d(A,B)$ in detail.

3.1.1 $\|A_n - B_n\|_F^2$

This part has the same computational formula (in Equation 5) as $\|u - v\|$. For image targets, it indicates the energy difference and it is the base of our distance measure.

3.1.2 Luminance Difference Weight.

We suppose C_n is the luminance difference of A_n and B_n.

$$c_{i,j} = a_{i,j} - b_{i,j} \tag{8}$$

Then we suppose that if the two images are similar, their luminance difference will be a smaller value than that of two images which are not alike. The mean value of C_n

$$\mu_c = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} c_{i,j} \tag{9}$$

can estimate the degree of how much their luminance difference is. If they are equal, $\mu_c = 0$; else, $\mu_c \neq 0$. The bigger the absolute value of μ_c is, the much difference the two images have. Then the weight ω_{lum} is a function of μ_c.

As a weight function, we hope that its value is between 0 and 1, and when the two images have the same luminance level, its value is 0. So we give the following expression

$$\omega_{lum} = 1 - \exp(-\frac{|\mu_c|}{C_1}) \tag{10}$$

where the constant C_1 is included to avoid ω_{lum} increasing too fast. For 8-bit grayscale or 24-bit true-color images, the maximum of each pixel is 255, we choose $C_1 = 10\sim20$ which is calculated using the image data sets on internet.

3.1.3 Content Difference Weight.

For images, content is more important than luminance. The two images in Figure 2 have different luminance level, but obviously they are the same object. We consider that the luminance difference between two images of the same object will be less complex than that of different objects. We suppose C_n is the luminance difference of A_n and B_n. And the standard deviation of C_n

$$\sigma_c = (\frac{1}{M \times N - 1}\sum_{i=1}^{M}\sum_{j=1}^{N}(c_{i,j} - \mu_c)^2)^{\frac{1}{2}} \tag{11}$$

can estimate the degree of how much the difference is.
As in Section 3.1.2, we give the following expression

$$\omega_{con} = 1 - \exp(-\frac{\sigma_c}{C_2}) \tag{12}$$

where the constant C_2 can avoid ω_{con} increasing too fast. For 8-bit grayscale or 24-bit true-color images, we choose $C_2 = 50\sim70$ which is calculated using the image data sets on internet.

3.1.4 Self Complexity Weight.

We consider that if an image is complex itself, it will be harder to find a similar one to it. In simple object while A_n' and B_n' are complex ones. We think the distance between A_n' other words, suppose two groups of images $\left\|A_n - B_n\right\|_F^2 = \left\|A_n' - B_n'\right\|_F^2$, A_n and B_n are two and B_n' is smaller. So

$$\omega_{complex} = \exp(-\frac{\sigma_{A_n}}{C_3}) \cdot \exp(-\frac{\sigma_{B_n}}{C_3}) \tag{13}$$

Where σ_{A_n} and σ_{B_n} can estimate their complexity, and C_3 is same as C_1, C_2. We choose $C_3 = 40\sim60$ for 8-bit grayscale or 24-bit true-color images.

3.1.5 Position Weight.

First, We think that the position of each block has relation to its contribution for the distance $d(A, B)$. The block close to the center will have greater weight. Second, the target is generally at the center of the image, while there may be some background objects around it. The block close to the edge may have more background information. To avoid these blocks' effect, they will be set smaller weight. So we hope ω_{pos} has Gauss form (in Figure 4, Equation 14), where the constant s is the value that is predefined for the block at the corner. r is the distance between the block and the center. r_0 is half of the diagonal length. We choose s=0.5 in our study.

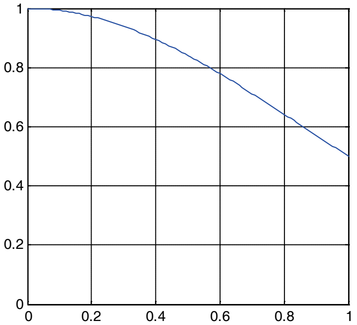


Fig. 4. The Relationship between Position Weight and its Position

$$\omega_{pos} = \exp(\log s \cdot (\frac{r}{r_0})^2) \tag{14}$$

Finally, all parts of ω_n are ready and we give

$$\omega_n = \left(m \cdot \omega_{lum} + (1-m) \cdot \omega_{con}\right) \cdot \omega_{complex} \cdot \omega_{pos} \tag{15}$$

where parameter m (0<m<1) is the weight of ω_{lum} and ω_{con}. Because we think ω_{lum} is from the whole view while ω_{con} is from the detail view. Parameter m is set to control the proportion how much they contribute to ω_n.

From all above discussions in Section 3.1, we know that

$$0 < \omega_n < 1 \tag{16}$$

It can be used as a weight. Then our $d(A, B)$ is accomplished finally.

3.2 Blocking Option

The size of each block and how the blocks are organized are also important to our RBF2D kernel function. But we will not discuss them in detail in this chapter. We only give our solution in this paper.

3.2.1 Size

We find that the size of each block can have effect on the performance of our image kernel. As shown in figure 5, if the blocks are too big, then the blocking operation is nonsense because the problem that we want to avoid when using the whole image will be met again. On the other side, if the blocks are too small, they can not contain the structural information that we hope they would have done.

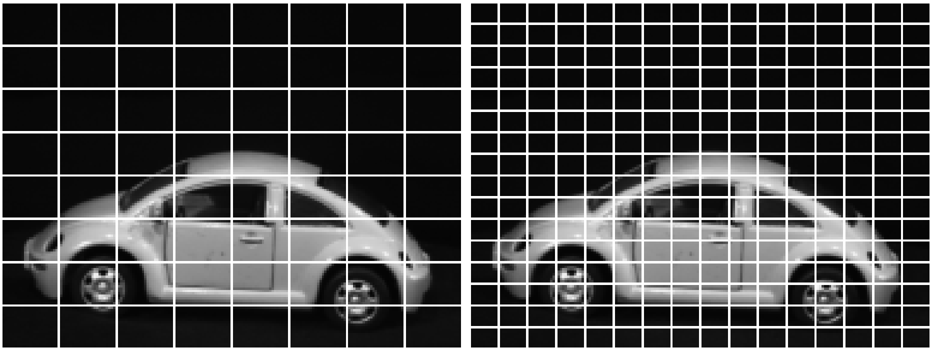


Fig. 5. Blocking mode which have different sizes

So, the size of block will be decided by the minimun shape in the image that can have some information.

3.2.2 Organization

We propose three image kernels based on their different organization forms of blocks.

(1) Normal Image Kernel. Blocks are organized as shown in figure 5. This is the simplest way. But its calculation efficiency is low and will be unstable when the edge of some block happens to overlap with that of the object.

(2) Redundant Image Kernel. We add redundancy part between two neighbors based on Normal Image Kernel. This image kernel is more stable but its calculation efficiency is even lower.

(3) Discrete Image Kernel. We use ROI (Regions of Interest) technology to decide some discrete blocks while other regions of the image will be ignored. This method has many advantages. First, the number of blocks which are calculated is small than the other two image kernel, its calculation efficiency is high. Second, the position of each block on one image is decided by its information distribution. The corresponding blocks on the other image is decided using image matching processing. So the positions of the object in the two images can have some difference, while the same thing will reduce the performance of the other two image kernels. The experiment result shows that the Discrete Image Kernel can work on the images in which the object can have different stances or be sheltered partly.

4. Experiment Results

4.1 UCI Optical Handwritten Digits Dataset

This dataset include about 5500 normalized bitmaps of handwritten digits gathered from 43 different people. Image size is 32×32.

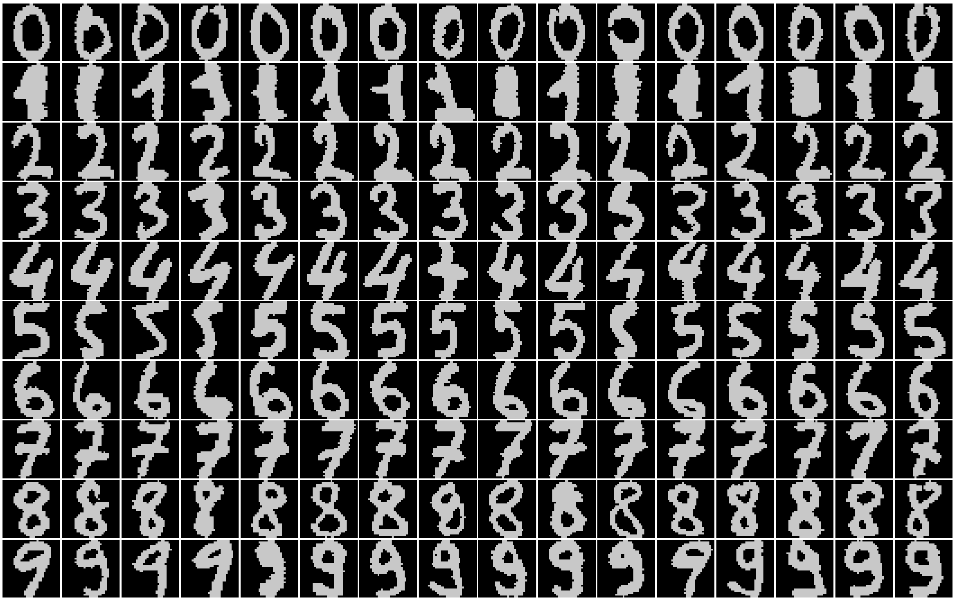


Fig. 6. UCI Optical Handwritten Digits Dataset

C. Kaynak divided these images into blocks of 4×4 and counted the pixels in each block. Then he got an input matrix of 8×8. His classifier was KNN based on Frobenius Norm and the final ratio of recognition is 97%~98% for each digits.

We choose digit '1' and digit '7' as our targets because they have similar appearance. 20% of total 400 samples are training set, others are testing set. The experiment will be repeated 10 times, and training set was selected randomly. And because the images are 1-bit bitmaps, we choose C_1=1, C_2=1, C_3=1, m=0.5, s=0.5. Blocks' size is 4×4. And finally our ratio of recognition is 99%~100%.

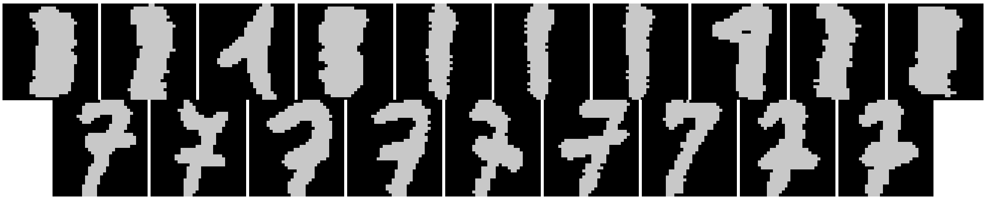


Fig. 7. SVs of RBF2D-based SVM

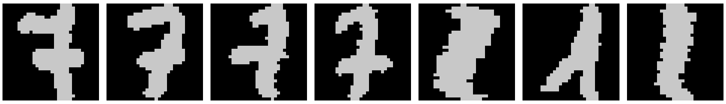


Fig. 8. Some samples which are difficult to recognized for Kaynak's method

From Figure 7 and Figure 8, we can see that the samples which are difficult to recognized for Kaynak's method are always the SVs of our SVM based on RBF2D. So our ratio can reach 100% when they are all in train set and treated as SVs.

4.2 COIL dataset

COIL data set includes 1000 objects and each object has 24 8-bit grayscale images under different luminance conditions. The size of each image is 144×192. We choose 3 objects. They are shown in Figure 9.

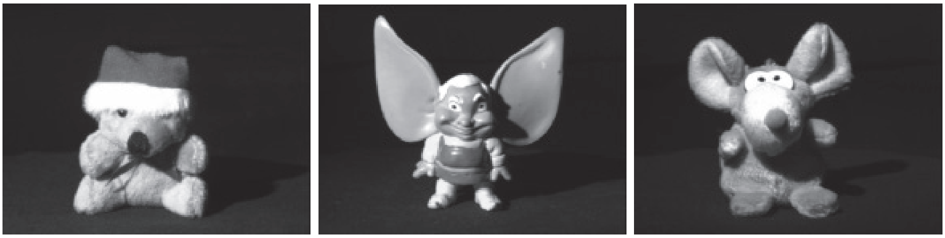


Fig. 9. Three Objects from COIL Data set

There are not enough samples. We only use our kernel function to calculate the gram matrix of the objects, because the learning mechanism of kernel-based classifiers is generally based on it. In our experiment, we set C_1 =20, C_2 =70, C_3 =60, m=0.5, s=0.5. Blocks' size is 24×24. The gram matrix is shown in Figure 10.

There are totally 72 samples, 24 for each object. From figure 6, it is obvious that only the two sample of the same object have the value close to 1, while others are close to 0. A kernel-based classifier (like SVM) can easily find the SVs of each class using linear programming algorithm or quadratic programming algorithm.

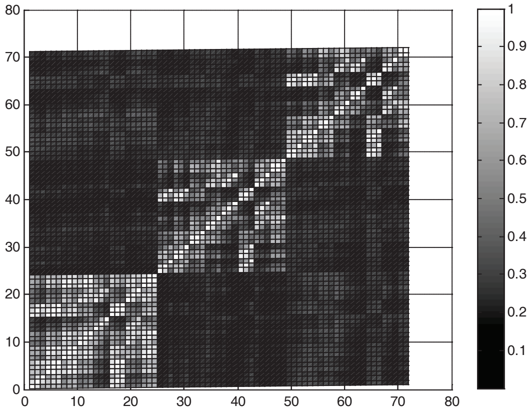
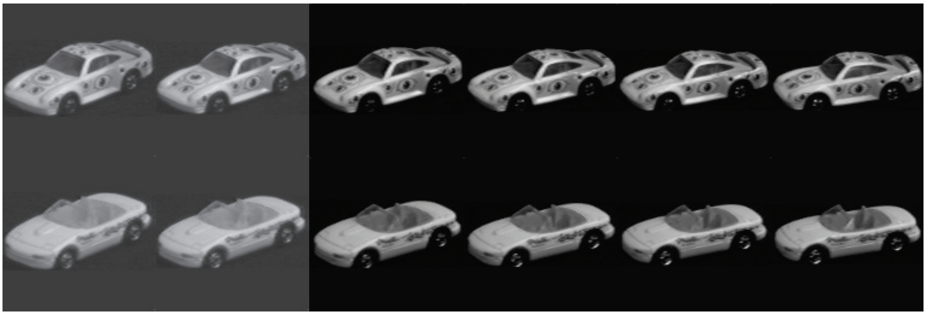


Fig. 10. The Gram Matrix of three Objects

4.2.1 COIL Objects with different angle of view

In Coil-100 dataset, the objects have different angles of view. We choose two as our targets.

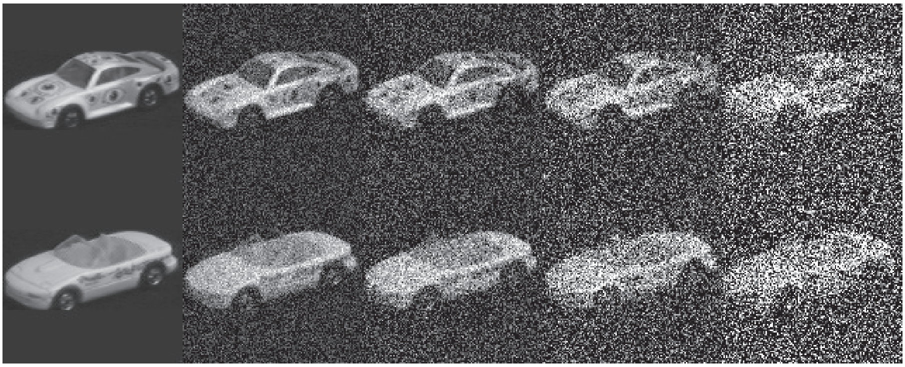


(a) 315° (b) 320° (c) 325° (d) 330° (e) 335° (f) 340°

Fig. 11. The Two Objects With Different Angel of View

There are only 12 images of the two objects. To expand the sample size, we add +2,+1 displacement to each object at four directions (up, down, left, right). And we get totally 12× 25=300 samples. We select 50% as train set and the others as test set and use Redundant Image Kernel and Discrete Image Kernel introduced in section 3. Classifier is the standard SVM.

For testing the performance, we also add noise to the images.



(a) no noise (b) 10db (c) 0db (d) -10db (e) -20db

Fig. 12. Images with noise

The experiment results are shown in table 1.

	Redundant Image Kernel		Discrete Image Kernel	
no noise	100%	34%	100%	3.3%
10 db	100%	100%	100%	30%
0 db	90%	100%	96%	66%
-10 db	85%	100%	90%	100%
-20 db	47%	100%	58%	100%

Table 1. Result of experiment, left is ratio of recognition, right is SVs/Samples

First, let's see the ratio of recognition. Discrete image kernel is the same as redundant image kernel. They can work until the noise level is reduced below -20 db. As introduced in sections before, discrete image kernel only uses part of the image, so its speed is higher.
Second, let's see the numbers of SVs. For redundant image kernel, all samples become SVs when the noise level is 10db. In this condition, every sample is SV and have to be saved so as to be used in testing processing. The classifier will have bad performance. While the discrete image kernel can work until noise is reduced below -10db.
To explain this problem, Figure 13 shows gram metrices of two kernels when there is no noise.

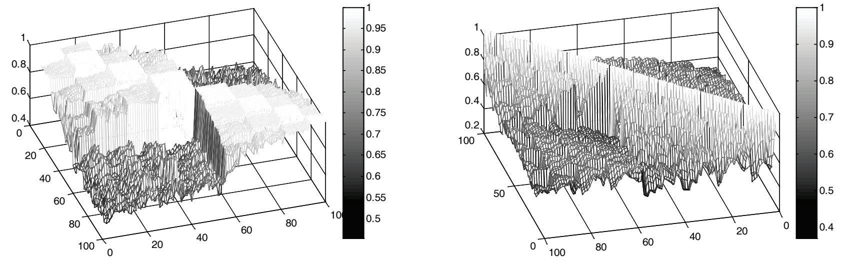


Fig. 13. Gram Metrix of two image kernel, left is Discrete Image Kernel, right is Redundant Image Kernel

From the gram metrix of two kernels, we can find that the difference of the two objects in discrete image kernel is more distinct and easier to use.

4.2.2 COIL Objects with sheltering

In this section, we will add some shelter to object images which are shown in figure 14.

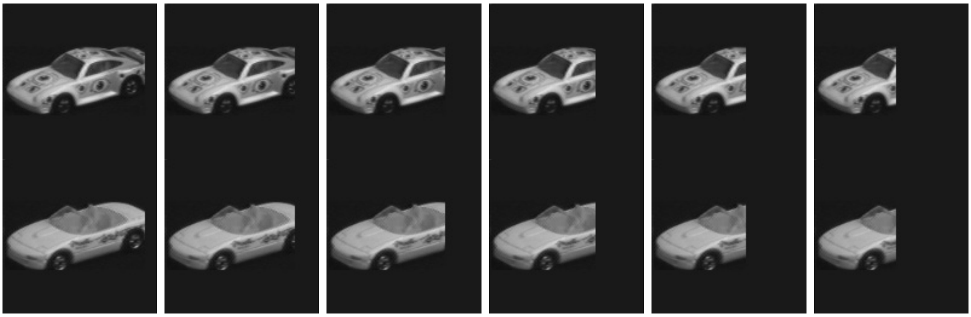


Fig. 14. Images with sheltering, shelter ratio is 7.8%,15.6%,23.4%,31.3%,39.1%,46.9%(from left to right)

The methods and parameters are the same as section 4.2.1. We use Redundant Image Kernel and Discrete Image Kernel introduced in section 3. Classifier is the standard SVM.
Train set is the full object image as in 4.2.1, while we add sheltering on test set, and use the new test set to test the two image kernel.

For each shelter ratio, the experimint is repeated once.
The result is shown in table 2.

	7.8%	15.6%	23.4%	31.3%	39.1%	46.9%
Discrete Image Kernel	Yes	Yes	Yes	Yes	Yes	No
Redundant Image Kernel	Yes	Yes	No	No	No	No

Table 2. Result of experiment, yes = can work, no = can't work

We can find that the discrete image kernel can work unless the shelter ratio reaches 40%, while the redundant image kernel can only work under 20%.
For Redundant Image Kernel, the object is sheltered means that the image has changed. While for Discrete Image Kernel, the algorithm only use part of the blocks (in figure 15).

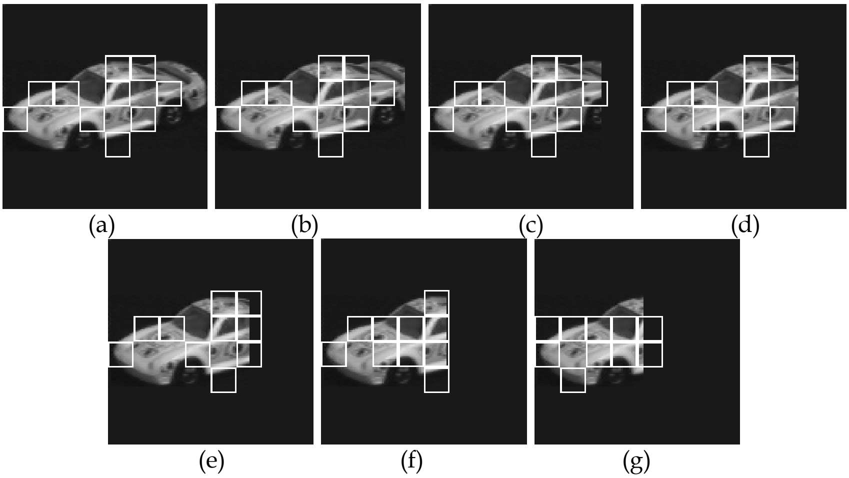

(a) (b) (c) (d)
(e) (f) (g)

Fig. 15. The Selected Blocks in Discrete Image Kernel. shelter ratio is (a) 0%, (b) 7.8%, (c) 15.6%, (d) 23.4%, (e) 31.3%, (f) 39.1%, (g) 46.9%

The seven image in figure 15 can be grouped into four categories.

(1) Figure 15(a). There is no sheltering in this condition. So (a) can be treat as a reference.
(2) Figure 15(b)~(c). Although the object has been sheltered, but all the selected blocks are the same as (a), so Discrete Image Kernel can work without any performance decrease.
(3) Figure 15(d)~(f). Here the object is sheltered partly and the selected are the same as (a) mostly. Only 1~2 blocks are changed, and they can not affect the total result.
(4) Figure 15(g). Mostly a large number of selected blocks are changed, and the result is unstable.

5. Conclusion

In this chapter, we have summarized the deficiency of traditional kernel functions on image recognition and proposed the distance measure of images and RBF2D kernel function which can accept two-dimensional image data as input data without abstracting the features that we often do nowadays.

We compare them to the old method using the UCI Optical Handwritten Digits dataset. The result indicates that RBF2D have good performance on image target.

Also we do some experiment to test the new image kernel when object are viewed from different angle, with noise, and even sheltered. The results show that our new image kernel can work in all these conditions.

6. Reference

John, S.T. & Nello, C. (2004). *Kernel Methods for Pattern Analysis.* Cambridge University Press, 7-111-17853-X, Cambridge

Horn, R.A. & Johnson, C.R. (1985). *Matrix Analysis*. Cambridge University Press, Cambridge

Sergios, T. & Konstantinos, K. (2006). *Pattern Recognition Third Edition*. Elsevier Pte Ltd, 981-259-707-7, 978-981-259-707-6, Singapore

Zhou, W. ; Alan, C.B. ; Hamid, R.S. & Eero, P.S. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, Vol.13, No.4, pp.600-612

C. Kaynak. (1995). Methods of Combining Multiple Classifiers and Their Applications to Handwritten Digit Recognition, MSc Thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University.

Statistical Inference on Markov Random Fields: Parameter Estimation, Asymptotic Evaluation and Contextual Classification of NMR Multispectral Images

Alexandre L. M. Levada[1], Nelson D. A. Mascarenhas[2] and Alberto Tannús[1]
[1]*University of São Paulo, Physics Institute of São Carlos*
[2]*Federal University of São Carlos, Computer Department*
Brazil

1. Introduction

Undoubtedly, Markov Random Fields (*MRF*) define a powerful mathematical tool for contextual modelling of spatial data. With advances in probability and statistics (Hammersley & Clifford, 1971), as the development of Markov Chain Monte Carlo (*MCMC*) simulation techniques (Metropolis et al., 1953; Geman & Geman, 1984; Swendsen & Wang, 1987; Wolff, 1989) and relaxation algorithms for combinatorial optimization (Besag, 1986; Marroquin et al., 1987; Yu & Berthod, 1995), *MRF's* became a central topic in fields including image processing, computer vision and pattern recognition. In this chapter, we are concerned with the multispectral image contextual classification problem. A Bayesian approach is used to combine two MRF models: a Gaussian Markov Random Field (*GMRF*) for the observations (likelihood) and a *Potts* model for the *a priori* knowledge. Hence, the problem is stated according to a Maximum *a Posteriori* (*MAP*) framework.

One of the main difficulties in contextual classification using a *MAP-MRF* approach relies on the *MRF* parameter estimation stage. Traditional methods, as Maximum Likelihood (*ML*), cannot be applied due to the existence of a partition function in the joint Gibbs distribution, which is computationally intractable. A solution proposed by Besag to surmount this problem is to use the local conditional density functions (*LCDF*) to perform maximum pseudo-likelihood (*MPL*) estimation (Besag, 1974). The main motivation for employing this approach is that *MPL* estimation is a computationally feasible method. Besides, from a statistical perspective, *MPL* estimators have a series of desirable and interesting properties, such as consistency and asymptotic normality (Jensen & Künsh, 1994). However, a serious limitation of contextual classification has been the use of extremely restricted neighbourhood systems. Actually, traditional methods often consider only first-order neighbourhood systems.

The main motivation for this chapter is to discuss the incorporation of higher-order neighbourhood systems in *MRF* models, since among several drawbacks existing in classification problems, the lack of an accurate contextual modelling is definitely a major one, especially when we are dealing with real image data, often degraded by noise. And, with the introduction of higher-order systems, novel, robust and suitable parameter estimation methods are also required. This chapter presents a novel framework for Bayesian image contextual classification through the definition of statistical inference and parameter estimation techniques in higher-order systems. Pseudo-likelihood equations for both *Potts* and *GMRF* models are presented and analysed using asymptotic evaluations and *MCMC* simulation algorithms. Two combinatorial optimization algorithms for *MAP-MRF* contextual classification are described and compared: *Iterated Conditional Modes* (*ICM*) and *Maximizer of the Posterior Marginals* (*MPM*). Experiments on real Nuclear Magnetic Resonance (*NMR*) images illustrate the proposed methodology.

The remaining of this chapter is organized as follows: Section 2 introduces the reader to the combined *GMRF* + *Potts MRF* model for contextual classification and discusses how to perform parameter estimation using the maximum pseudo-likelihood approach. In Section 3 we present asymptotic evaluations to assess the accuracy of *MPL* estimation by means of approximations for the *MPL* estimators' variances. Section 4 introduces the combinatorial optimization algorithms for contextual classification. Section 5 discusses metrics for quantitative performance evaluation, more precisely Cohen's Kappa coefficient. Section 6 shows results of the proposed classification methodology on real multispectral magnetic resonance images. Finally, Section 7 presents the conclusions and final remarks.

2. *MAP-MRF* Contextual Classification

Let $\mathbf{x}_w^{(p)}$ be the label field representing the classification map at the *p*-th iteration, $\boldsymbol{y}$ the observed multispectral image, $\boldsymbol{\theta}$ the vector of *GMRF* hyperparameters, $\boldsymbol{\Phi}$ the vector of *GMRF* spectral parameters for each class ($\boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_m$) and β the *Potts MRF* model hyperparameter. Considering a multispectral *GMRF* model for the observations and a *Potts* model for the *a priori* knowledge, according to the Bayes' rule, the current label of pixel (i,j) can be iteratively updated by choosing the label that maximizes the functional (Yamazaki & Gingras, 1996):

$$Q\left(x_{ij}=m \mid \boldsymbol{x}_w^{(p)},\boldsymbol{y},\boldsymbol{\theta},\boldsymbol{\Phi},\beta\right)=-\frac{1}{2}\ln\left|\hat{\boldsymbol{\Sigma}}_m\right|-\frac{1}{2}\left\{\boldsymbol{y}_{ij}-\hat{\boldsymbol{\mu}}_m\left[\hat{\boldsymbol{\theta}}^T\vec{y}_{\eta_{ij}}-2\left(\sum_{ct}\hat{\boldsymbol{\theta}}^{ct}\right)\hat{\boldsymbol{\mu}}_m\right]\right\}^T\hat{\boldsymbol{\Sigma}}_m^{-1} \\ \times\left\{\boldsymbol{y}_{ij}-\hat{\boldsymbol{\mu}}_m\left[\hat{\boldsymbol{\theta}}^T\boldsymbol{y}_{\eta_{ij}}-2\left(\sum_{ct}\hat{\boldsymbol{\theta}}^{ct}\right)\hat{\boldsymbol{\mu}}_m\right]\right\}+\hat{\beta}U_{ij}(m) \tag{1}$$

where $\hat{\boldsymbol{\theta}}^{ct}$ is a diagonal matrix whose elements are the horizontal, vertical and diagonals hyperparameters (4 x 4), $ct=1,\ldots,K$, where K is the number of bands, $\hat{\boldsymbol{\theta}}^T$ is a matrix build by stacking the $\hat{\boldsymbol{\theta}}^{ct}$ diagonal matrices from each image band (4 x 4*K*), that is, $\hat{\boldsymbol{\theta}}^T=\left[\hat{\theta}^{ct1},\hat{\theta}^{ct2},\ldots,\hat{\theta}^{ctK}\right]$ and $\boldsymbol{y}_{\eta_{ij}}$ is a vector whose elements are defined as the sum of the two neighbouring elements on each direction (horizontal, vertical and diagonals) for all the image bands (4*K* x 1).

2.1 *MPL* Estimation for *GMRF* model parameters

As the proposed model for contextual classification of multispectral images assumes independency between different image bands, it is quite reasonable to perform *MPL* estimation in each image band separately. Assuming this hypothesis and considering a second-order neighbourhood system, the pseudo-likelihood equation for the *GMRF* hyperparameters becomes (Won & Gray, 2004):

$$\log PL\left(\boldsymbol{\theta},\mu,\sigma^2\right)=\sum_{(i,j)\in W}\left\{-\frac{1}{2}\log\left(2\pi\sigma^2\right)-\frac{1}{2\sigma^2}\left[y_{ij}-\boldsymbol{\theta}^T\boldsymbol{\Psi}_{ij}-\mu\left(1-2\boldsymbol{\theta}\boldsymbol{I}\right)\right]^2\right\} \tag{2}$$

where W represents an image band, $\boldsymbol{\theta}^T=\left[\theta_1,\theta_2,\theta_3,\theta_4\right]$ is the hyperparameters vector and $\boldsymbol{\Psi}_{ij}=\left[\left(y_{i+1j}+y_{i-1j}\right),\left(y_{ij+1}+y_{ij-1}\right),\left(y_{i+1j-1}+y_{i-1j+1}\right),\left(y_{i-1j-1}+y_{i+1j+1}\right)\right]$. Fortunately, the *MPL* estimator of $\boldsymbol{\theta}$ admits a closed solution, given by (Won & Gray, 2004):

$$\hat{\boldsymbol{\theta}}=\left\{\left[\sum_{(i,j)\in W}\left(y_{ij}-\hat{\mu}\right)\tilde{\boldsymbol{\Psi}}_{ij}^T\right]\left[\sum_{(i,j)\in W}\tilde{\boldsymbol{\Psi}}_{ij}\tilde{\boldsymbol{\Psi}}_{ij}^T\right]^{-1}\right\} \tag{3}$$

where $\hat{\mu}$ is the sample mean of the image pixels, $\tilde{\boldsymbol{\Psi}}_{ij}=\boldsymbol{\Psi}_{ij}-\frac{1}{N}\sum_{(k,l)\in\Omega}\boldsymbol{\Psi}_{ij}$ and N is the number of image pixels.

2.2 *MPL* Estimation for *Potts MRF* model parameter

One of the most widely used prior models in Bayesian image modelling is the Potts *MRF* pair-wise interaction (*PWI*) model. Two fundamental characteristics of the Potts model considered here are: it is both isotropic and stationary. According to (Hammersley & Clifford, 1971), the Potts *MRF* model can be equivalently defined in two manners: by a joint Gibbs distribution (global model) or by a set of local conditional density functions (*LCDF's*). For a general *s-th* order neighbourhood system η^s, we define the former by the following expression:

$$p\left(x_{ij}=m|\eta^s,\beta\right)=\frac{\exp\left\{\beta U_{ij}(m)\right\}}{\sum_{l=1}^{M}\exp\left\{\beta U_{ij}(l)\right\}} \tag{4}$$

where $U_{ij}(l)$ is the number of neighbours of the *i-th* element having label equal to *l*, β is the spatial dependency parameter (known as inverse temperature), $m, l \in G = \{1, 2, \ldots, M\}$, with M denoting the number of classes. So, the pseudo-likelihood equation for the Potts model is given by:

$$PL(\beta) = \prod_{(i,j)\in W} \frac{\exp\left\{\beta U_{ij}(m)\right\}}{\sum\limits_{l=1}^{M} \exp\left\{\beta U_{ij}(l)\right\}} \tag{5}$$

Taking the logarithms, differentiating on the parameter and setting the result to zero, leads to the following expression, which is the basis for the derivation of the proposed equations:

$$\frac{\partial}{\partial\beta} \log PL(\beta) = \sum_{(i,j)\in W} U_{ij}(m) - \sum_{(i,j)\in W} \left[\frac{\sum\limits_{l=1}^{M} U_{ij}(l) \exp\left\{\beta U_{ij}(l)\right\}}{\sum\limits_{l=1}^{M} \exp\left\{\beta U_{ij}(l)\right\}} \right] = 0 \tag{6}$$

where m denotes the observed value for the *ij-th* element of the field.
Looking at equation (6) it is possible to see that its first term is independent of the parameter. Thus, it is possible to expand the second term of (6) in all possible spatial configuration patterns that provide different contributions to the pseudo-likelihood equation regarding a pre-defined neighbourhood system. For example, in first order systems, the enumeration of these configuration patterns is straightforward, since there are only five cases, from zero agreement (situation of four different labels) to total agreement (situation of four identical labels), as shown in Figure 1.

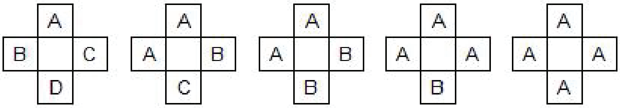


Fig. 1. Contextual configuration patterns for Potts *MRF* model in first-order neighbourhood systems

These configuration patterns can represented by vectors, as presented in relations (7), indicating the number of occurrences of each element around the central element. In the Potts model location information is irrelevant since it is an isotropic model:

$$v_0 = [1,1,1,1]; \quad v_1 = [2,1,1,0]; \quad v_2 = [2,2,0,0]; \quad v_3 = [3,1,0,0]; \quad v_4 = [4,0,0,0]; \tag{7}$$

Let N be the number of elements in the neighbourhood system η^s. For each $L = 1, \ldots, N$, let:

$$A_N(L) = \left\{ \left(a_1, \ldots, a_L\right) / \, a_i \in \left\{1, 2, \ldots, N\right\}, a_1 \le \cdots \le a_L, \sum_{i=1}^{L} a_i = N \right\} \tag{8}$$

and $n_N(L)$ the number of elements of the set $A_N(L)$. Then, the number of configuration patterns for the neighbourhood system η^s is $\lambda = n_N(1) + n_N(2) + \cdots + n_N(L)$.

Thus, the problem of finding the possible contextual configuration patterns can be solved automatically. The solution vectors can be found by exhaustive searching, by isolating one variable and searching on the subspace spanned by the remainder variables. Table 1 presents the number of configuration patterns for several neighbourhood systems.

Neighbourhood System	*Number of configuration patterns (λ)*
First order	5
Second order	22
Third order	77
Fourth order	637
Fifth order	1575

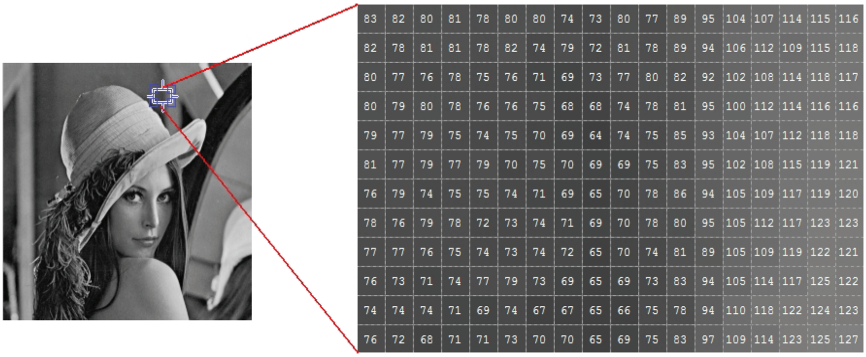
Table 1. Number of possible contextual configuration patterns for five neighbourhood systems

Now, given the complete set of contextual configuration patterns for a neighbourhood system, it is possible to expand the second term of equation (6). We can regard its numerator as an inner product of two vectors $\boldsymbol{U}_{ij}$ and $\boldsymbol{\omega}_{ij}$, where $\boldsymbol{U}_{ij}$ represents the contextual configuration vector for the current pixel (i.e., $\boldsymbol{U}_{ij} = [5,2,1,0,0,0,0,0]$ in case of a second-order system) and $\boldsymbol{\omega}_{ij}$ is a vector such that $\omega_{ij}[n] = \exp\left\{\beta U_{ij}[n]\right\}$. Similarly, the denominator is the inner product of $\boldsymbol{\omega}_{ij}$ with the identity column vector $\boldsymbol{r} = [1,1,\ldots,1]$. So, the second term of equation (6) can be expanded as a summation of λ terms, each one associated with a possible configuration pattern. However, as it involves a summation for all elements of the MRF, we define constants $K_i, i = 1, \ldots, \lambda$, representing the number of number of occurrences of each possible configuration patters along the entire image.

Basically, the idea is that the set of all K_i coefficients defines a contextual histogram, that is, instead of indicating the distribution of individual pixel gray levels, this set shows the distribution of spatial patterns defined in terms of the adopted neighbourhood system. For instance, in image analysis applications, smooth images, with many homogeneous regions, tend to present more concentration of configuration patterns with similar labels. On the other hand, heterogeneous regions tend to present concentration of configuration patterns with higher variation in the labels. Figures 2, 3 and 4 show an example with the Lena image.

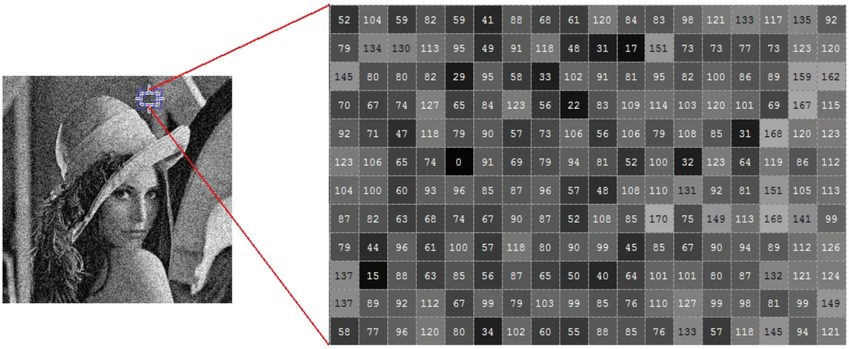
It is worthwhile noting that in a physical interpretation we are using the proposed equations to estimate a quantity called *inverse temperature* in a system of particles arranged on a 2-D lattice using only pair-wise interactions. The proposed pseudo-likelihood equation for second-order neighbourhood systems is given by equation (9) (Levada et al., 2008a).

Similarly, for third-order systems, the equation is obtained by expanding (6) on the 77 configuration patterns derived by solving (8) for *N=12*. The proposed pseudo-likelihood equation for third-order systems is given by equation (10) (Levada et al., 2008b).



83	82	80	81	78	80	80	74	73	80	77	89	95	104	107	114	115	116
82	78	81	81	78	82	74	79	72	81	78	89	94	106	112	109	115	118
80	77	76	78	75	76	71	69	73	77	80	82	92	102	108	114	118	117
80	79	80	78	76	76	75	68	68	74	78	81	95	100	112	114	116	116
79	77	79	75	74	75	70	69	64	74	75	85	93	104	107	112	118	118
81	77	79	77	79	70	75	70	69	69	75	83	95	102	108	115	119	121
76	79	74	75	75	74	71	69	65	70	78	86	94	105	109	117	119	120
78	76	79	78	72	73	74	71	69	70	78	80	95	105	112	117	123	123
77	77	76	75	74	73	74	72	65	70	74	81	89	105	109	119	122	121
76	73	71	74	77	79	73	69	65	69	73	83	94	105	114	117	125	122
74	74	74	71	69	74	67	67	65	66	75	78	94	110	118	122	124	123
76	72	68	71	71	73	70	70	65	69	75	83	97	109	114	123	125	127

Fig. 2. Smooth regions present more homogeneous contextual configuration patterns



52	104	59	82	59	41	88	68	61	120	84	83	98	121	133	117	135	92
79	134	130	113	95	49	91	118	48	31	17	151	73	73	77	73	123	120
145	80	80	82	29	95	58	33	102	91	81	95	82	100	86	89	159	162
70	67	74	127	65	84	123	56	22	83	109	114	103	120	101	69	167	115
92	71	47	118	79	90	57	73	106	56	106	79	108	85	31	168	120	123
123	106	65	74	0	91	69	79	94	81	52	100	32	123	64	119	86	112
104	100	60	93	96	85	87	96	57	48	108	110	131	92	81	151	105	113
87	82	63	68	74	67	90	87	52	108	85	170	75	149	113	168	141	99
79	44	96	61	100	57	118	80	90	99	45	85	67	90	94	89	112	126
137	15	88	63	85	56	87	65	50	40	64	101	101	80	87	132	121	124
137	89	92	112	67	99	79	103	99	85	76	110	127	99	98	81	99	149
58	77	96	120	80	34	102	60	55	88	85	76	133	57	118	145	94	121

Fig. 3. Noisy regions present more heterogeneous contextual configuration patterns

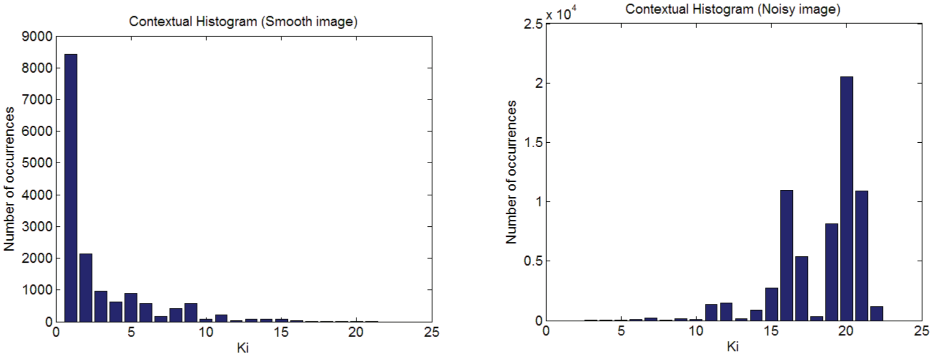


Fig. 4. Comparison between the distribution of contextual configuration patterns for both smooth and noisy Lena images (k_0 stands for total agreement and k_{22} for zero agreement).

$$\begin{aligned}
\frac{\partial}{\partial\beta}\log PL\left(\beta\right) = \sum_{(i,j)\in W} U_{ij}(m) - \frac{8e^{8\hat{\beta}}}{e^{8\hat{\beta}}+M-1}K_1 - \frac{7e^{7\hat{\beta}}+e^{\hat{\beta}}}{e^{7\hat{\beta}}+e^{\hat{\beta}}+M-2}K_2 - \frac{6e^{6\hat{\beta}}+2e^{2\hat{\beta}}}{e^{6\hat{\beta}}+e^{2\hat{\beta}}+M-2}K_3 \\
-\frac{6e^{6\hat{\beta}}+2e^{\hat{\beta}}}{e^{6\hat{\beta}}+2e^{\hat{\beta}}+M-3}K_4 - \frac{5e^{5\hat{\beta}}+3e^{3\hat{\beta}}}{e^{5\hat{\beta}}+e^{3\hat{\beta}}+M-2}K_5 - \frac{5e^{5\hat{\beta}}+2e^{2\hat{\beta}}+e^{\hat{\beta}}}{e^{5\hat{\beta}}+e^{2\hat{\beta}}+e^{\hat{\beta}}+M-3}K_6 - \frac{5e^{5\hat{\beta}}+3e^{\hat{\beta}}}{e^{5\hat{\beta}}+e^{3\hat{\beta}}+M-4}K_7 \\
-\frac{8e^{4\hat{\beta}}}{2e^{4\hat{\beta}}+M-2}K_8 - \frac{4e^{4\hat{\beta}}+3e^{3\hat{\beta}}+e^{\hat{\beta}}}{e^{4\hat{\beta}}+e^{3\hat{\beta}}+e^{\hat{\beta}}+M-3}K_9 - \frac{4e^{4\hat{\beta}}+4e^{2\hat{\beta}}}{e^{4\hat{\beta}}+2e^{2\hat{\beta}}+M-3}K_{10} - \frac{4e^{4\hat{\beta}}+2e^{2\hat{\beta}}+2e^{\hat{\beta}}}{e^{4\hat{\beta}}+e^{2\hat{\beta}}+2e^{\hat{\beta}}+M-4}K_{11} \\
-\frac{4e^{4\hat{\beta}}+4e^{\hat{\beta}}}{e^{4\hat{\beta}}+4e^{\hat{\beta}}+M-5}K_{12} - \frac{6e^{3\hat{\beta}}+2e^{2\hat{\beta}}}{2e^{3\hat{\beta}}+e^{2\hat{\beta}}+M-3}K_{13} - \frac{6e^{3\hat{\beta}}+2e^{\hat{\beta}}}{2e^{3\hat{\beta}}+2e^{\hat{\beta}}+M-4}K_{14} - \frac{3e^{3\hat{\beta}}+4e^{2\hat{\beta}}+e^{\hat{\beta}}}{e^{3\hat{\beta}}+2e^{2\hat{\beta}}+e^{\hat{\beta}}+M-4}K_{15} \\
-\frac{3e^{3\hat{\beta}}+2e^{2\hat{\beta}}+3e^{\hat{\beta}}}{e^{3\hat{\beta}}+e^{2\hat{\beta}}+3e^{\hat{\beta}}+M-5}K_{16} - \frac{3e^{3\hat{\beta}}+5e^{\hat{\beta}}}{e^{5\hat{\beta}}+5e^{\hat{\beta}}+M-6}K_{17} - \frac{8e^{2\hat{\beta}}}{4e^{2\hat{\beta}}+M-4}K_{18} - \frac{6e^{2\hat{\beta}}+2e^{\hat{\beta}}}{3e^{2\hat{\beta}}+2e^{\hat{\beta}}+M-5}K_{19} \\
-\frac{4e^{2\hat{\beta}}+4e^{\hat{\beta}}}{2e^{2\hat{\beta}}+4e^{\hat{\beta}}+M-6}K_{20} - \frac{2e^{2\hat{\beta}}+6e^{\hat{\beta}}}{e^{2\hat{\beta}}+6e^{\hat{\beta}}+M-7}K_{21} - \frac{8e^{\hat{\beta}}}{8e^{\hat{\beta}}+M-8}K_{22} = 0
\end{aligned} \tag{9}$$

The derived transcendental equations do not have closed solution, so in order to solve them a root-finding algorithm is required. In all experiments along this chapter, the *MPL* estimator is obtained by Brent's method (Brent, 1973), a numerical method that does not require the computation (not even the existence) of derivatives or analytical gradients. In this case, the computation of derivatives of the objective function would be prohibitive, given the large extension of the expressions. Basically, the advantages of this method can be summarized by: it uses a combination of bisection, secant and inverse quadratic interpolation methods, leading to a very robust approach and also it has super-linear convergence rate.

3. Statistical Inference and Asymptotic Evaluation on Markov Random Fields

Unbiasedness is not granted by either maximum likelihood (*ML*) or maximum pseudo-likelihood (*MPL*) estimation. Actually, according to statistical inference theory, there is no method that guarantees the existence of unbiased estimators for a fixed *N*-size sample. Often, in the exponential family, *ML* estimators coincide with *UMVU* (*Uniform Minimum Variance Unbiased*) estimators because they are functions of complete sufficient statistics (if a *ML* estimator is unique then it is a function of sufficient statistics). Besides, there are several characteristics that make *ML* estimation a reference method (Lehman, 1983; Bickel, 1991; Casella, 2002). Making the sample size grow infinitely ($N \to \infty$), ML estimator becomes asymptotically unbiased and efficient. Unfortunately, there is no result showing that the same occurs in *MPL* estimation. In this section, we show how to approximate the asymptotic variances of Potts and *GMRF* model parameters in terms of expressions for the observed Fisher information using both first and second derivatives.

$$
\begin{aligned}
&\frac{\partial}{\partial\beta}\log PL\left(\beta\right) = \sum_{(i,j)\in W} U_{ij}(m) - \frac{12e^{12\hat{\beta}}}{e^{12\hat{\beta}}+M-1}K_{1} - \frac{11e^{11\hat{\beta}}+e^{\hat{\beta}}}{e^{11\hat{\beta}}+e^{\hat{\beta}}+M-2}K_{2} - \frac{10e^{10\hat{\beta}}+2e^{2\hat{\beta}}}{e^{10\hat{\beta}}+e^{2\hat{\beta}}+M-2}K_{3} - \frac{9e^{9\hat{\beta}}+3e^{3\hat{\beta}}}{e^{9\hat{\beta}}+e^{3\hat{\beta}}+M-2}K_{4} \\
&- \frac{8e^{8\hat{\beta}}+4e^{4\hat{\beta}}}{e^{8\hat{\beta}}+e^{4\hat{\beta}}+M-2}K_{5} - \frac{7e^{7\hat{\beta}}+5e^{5\hat{\beta}}}{e^{7\hat{\beta}}+e^{5\hat{\beta}}+M-2}K_{6} - \frac{12e^{6\hat{\beta}}}{2e^{6\hat{\beta}}+M-2}K_{7} - \frac{12e^{4\hat{\beta}}}{3e^{4\hat{\beta}}+M-3}K_{8} - \frac{5e^{5\hat{\beta}}+4e^{4\hat{\beta}}+3e^{3\hat{\beta}}}{e^{5\hat{\beta}}+e^{4\hat{\beta}}+e^{3\hat{\beta}}+M-3}K_{9} \\
&- \frac{10e^{5\hat{\beta}}+2e^{2\hat{\beta}}}{2e^{5\hat{\beta}}+e^{2\hat{\beta}}+M-3}K_{10} - \frac{6e^{6\hat{\beta}}+6e^{3\hat{\beta}}}{e^{6\hat{\beta}}+2e^{3\hat{\beta}}+M-3}K_{11} - \frac{6e^{6\hat{\beta}}+4e^{4\hat{\beta}}+2e^{2\hat{\beta}}}{e^{6\hat{\beta}}+e^{4\hat{\beta}}+e^{2\hat{\beta}}+M-3}K_{12} - \frac{6e^{6\hat{\beta}}+5e^{5\hat{\beta}}+e^{\hat{\beta}}}{e^{6\hat{\beta}}+e^{5\hat{\beta}}+e^{\hat{\beta}}+M-3}K_{13} - \\
&\frac{7e^{7\hat{\beta}}+3e^{3\hat{\beta}}+2e^{2\hat{\beta}}}{e^{7\hat{\beta}}+e^{3\hat{\beta}}+e^{2\hat{\beta}}+M-3}K_{14} - \frac{7e^{7\hat{\beta}}+4e^{4\hat{\beta}}+e^{\hat{\beta}}}{e^{7\hat{\beta}}+e^{4\hat{\beta}}+e^{\hat{\beta}}+M-3}K_{15} - \frac{8e^{8\hat{\beta}}+4e^{2\hat{\beta}}}{e^{8\hat{\beta}}+2e^{2\hat{\beta}}+M-3}K_{16} - \frac{8e^{8\hat{\beta}}+3e^{3\hat{\beta}}+e^{\hat{\beta}}}{e^{8\hat{\beta}}+e^{3\hat{\beta}}+e^{\hat{\beta}}+M-3}K_{17} - \\
&\frac{9e^{9\hat{\beta}}+2e^{2\hat{\beta}}+e^{\hat{\beta}}}{e^{9\hat{\beta}}+e^{2\hat{\beta}}+e^{\hat{\beta}}+M-3}K_{18} - \frac{10e^{10\hat{\beta}}+2e^{\hat{\beta}}}{e^{10\hat{\beta}}+2e^{\hat{\beta}}+M-3}K_{19} - \frac{12e^{3\hat{\beta}}}{4e^{3\hat{\beta}}+M-4}K_{20} - \frac{4e^{4\hat{\beta}}+6e^{3\hat{\beta}}+2e^{2\hat{\beta}}}{e^{4\hat{\beta}}+2e^{3\hat{\beta}}+e^{2\hat{\beta}}+M-4}K_{21} - \frac{8e^{4\hat{\beta}}+4e^{2\hat{\beta}}}{2e^{4\hat{\beta}}+2e^{2\hat{\beta}}+M-4}K_{22} \\
&- \frac{8e^{4\hat{\beta}}+3e^{3\hat{\beta}}+e^{\hat{\beta}}}{2e^{4\hat{\beta}}+e^{3\hat{\beta}}+e^{\hat{\beta}}+M-4}K_{23} - \frac{5e^{5\hat{\beta}}+3e^{3\hat{\beta}}+4e^{2\hat{\beta}}}{e^{5\hat{\beta}}+e^{3\hat{\beta}}+2e^{2\hat{\beta}}+M-4}K_{24} - \frac{5e^{5\hat{\beta}}+6e^{3\hat{\beta}}+e^{\hat{\beta}}}{e^{5\hat{\beta}}+2e^{3\hat{\beta}}+e^{\hat{\beta}}+M-4}K_{25} - \frac{5e^{5\hat{\beta}}+4e^{4\hat{\beta}}+2e^{2\hat{\beta}}+e^{\hat{\beta}}}{e^{5\hat{\beta}}+e^{4\hat{\beta}}+e^{2\hat{\beta}}+e^{\hat{\beta}}+M-4}K_{26} \\
&- \frac{10e^{5\hat{\beta}}+2e^{\hat{\beta}}}{2e^{5\hat{\beta}}+2e^{\hat{\beta}}+M-4}K_{27} - \frac{6e^{6\hat{\beta}}+6e^{2\hat{\beta}}}{e^{6\hat{\beta}}+3e^{2\hat{\beta}}+M-4}K_{28} - \frac{6e^{6\hat{\beta}}+3e^{3\hat{\beta}}+2e^{2\hat{\beta}}+e^{\hat{\beta}}}{e^{6\hat{\beta}}+e^{3\hat{\beta}}+e^{2\hat{\beta}}+e^{\hat{\beta}}+M-4}K_{29} - \frac{6e^{6\hat{\beta}}+4e^{4\hat{\beta}}+2e^{\hat{\beta}}}{e^{6\hat{\beta}}+e^{4\hat{\beta}}+2e^{\hat{\beta}}+M-4}K_{30} \\
&- \frac{7e^{7\hat{\beta}}+4e^{2\hat{\beta}}+e^{\hat{\beta}}}{e^{7\hat{\beta}}+2e^{2\hat{\beta}}+e^{\hat{\beta}}+M-4}K_{31} - \frac{7e^{7\hat{\beta}}+3e^{3\hat{\beta}}+2e^{\hat{\beta}}}{e^{7\hat{\beta}}+e^{3\hat{\beta}}+2e^{\hat{\beta}}+M-4}K_{32} - \frac{8e^{8\hat{\beta}}+2e^{2\hat{\beta}}+2e^{\hat{\beta}}}{e^{8\hat{\beta}}+e^{2\hat{\beta}}+2e^{\hat{\beta}}+M-4}K_{33} - \frac{9e^{9\hat{\beta}}+3e^{\hat{\beta}}}{e^{9\hat{\beta}}+3e^{\hat{\beta}}+M-4}K_{34} \\
&- \frac{6e^{3\hat{\beta}}+6e^{2\hat{\beta}}}{2e^{3\hat{\beta}}+3e^{2\hat{\beta}}+M-5}K_{35} - \frac{9e^{3\hat{\beta}}+2e^{2\hat{\beta}}+e^{\hat{\beta}}}{3e^{3\hat{\beta}}+e^{2\hat{\beta}}+e^{\hat{\beta}}+M-5}K_{36} - \frac{4e^{4\hat{\beta}}+8e^{2\hat{\beta}}}{e^{4\hat{\beta}}+4e^{2\hat{\beta}}+M-5}K_{37} - \frac{4e^{4\hat{\beta}}+3e^{3\hat{\beta}}+4e^{2\hat{\beta}}+e^{\hat{\beta}}}{e^{4\hat{\beta}}+e^{3\hat{\beta}}+2e^{2\hat{\beta}}+e^{\hat{\beta}}+M-5}K_{38} \\
&- \frac{4e^{4\hat{\beta}}+6e^{3\hat{\beta}}+2e^{\hat{\beta}}}{e^{4\hat{\beta}}+2e^{3\hat{\beta}}+2e^{\hat{\beta}}+M-5}K_{39} - \frac{8e^{4\hat{\beta}}+2e^{2\hat{\beta}}+2e^{\hat{\beta}}}{2e^{4\hat{\beta}}+e^{2\hat{\beta}}+2e^{\hat{\beta}}+M-5}K_{40} - \frac{5e^{5\hat{\beta}}+6e^{2\hat{\beta}}+e^{\hat{\beta}}}{e^{5\hat{\beta}}+3e^{2\hat{\beta}}+e^{\hat{\beta}}+M-5}K_{41} - \frac{5e^{5\hat{\beta}}+3e^{3\hat{\beta}}+2e^{2\hat{\beta}}+2e^{\hat{\beta}}}{e^{5\hat{\beta}}+e^{3\hat{\beta}}+e^{2\hat{\beta}}+2e^{\hat{\beta}}+M-5}K_{42} \\
&- \frac{5e^{5\hat{\beta}}+4e^{4\hat{\beta}}+3e^{\hat{\beta}}}{e^{5\hat{\beta}}+e^{4\hat{\beta}}+3e^{\hat{\beta}}+M-5}K_{43} - \frac{6e^{6\hat{\beta}}+4e^{4\hat{\beta}}+2e^{\hat{\beta}}}{e^{6\hat{\beta}}+2e^{2\hat{\beta}}+2e^{\hat{\beta}}+M-5}K_{44} - \frac{6e^{6\hat{\beta}}+3e^{3\hat{\beta}}+3e^{\hat{\beta}}}{e^{6\hat{\beta}}+e^{3\hat{\beta}}+3e^{\hat{\beta}}+M-5}K_{45} - \frac{7e^{7\hat{\beta}}+2e^{2\hat{\beta}}+3e^{\hat{\beta}}}{e^{7\hat{\beta}}+e^{2\hat{\beta}}+3e^{\hat{\beta}}+M-5}K_{46} \\
&- \frac{8e^{8\hat{\beta}}+4e^{\hat{\beta}}}{e^{8\hat{\beta}}+4e^{\hat{\beta}}+M-5}K_{47} - \frac{12e^{2\hat{\beta}}}{6e^{2\hat{\beta}}+M-6}K_{48} - \frac{3e^{3\hat{\beta}}+8e^{2\hat{\beta}}+e^{\hat{\beta}}}{e^{3\hat{\beta}}+4e^{2\hat{\beta}}+e^{\hat{\beta}}+M-6}K_{49} - \frac{6e^{3\hat{\beta}}+4e^{2\hat{\beta}}+2e^{\hat{\beta}}}{2e^{3\hat{\beta}}+2e^{2\hat{\beta}}+2e^{\hat{\beta}}+M-6}K_{50} - \frac{9e^{3\hat{\beta}}+3e^{\hat{\beta}}}{3e^{3\hat{\beta}}+3e^{\hat{\beta}}+M-6}K_{51} \\
&- \frac{4e^{4\hat{\beta}}+6e^{2\hat{\beta}}+2e^{\hat{\beta}}}{e^{4\hat{\beta}}+3e^{2\hat{\beta}}+2e^{\hat{\beta}}+M-6}K_{52} - \frac{4e^{4\hat{\beta}}+3e^{3\hat{\beta}}+2e^{2\hat{\beta}}-3e^{\hat{\beta}}}{e^{4\hat{\beta}}+e^{3\hat{\beta}}+e^{2\hat{\beta}}+3e^{\hat{\beta}}+M-6}K_{53} - \frac{8e^{4\hat{\beta}}+4e^{\hat{\beta}}}{2e^{4\hat{\beta}}+4e^{\hat{\beta}}+M-6}K_{54} - \frac{5e^{5\hat{\beta}}+4e^{2\hat{\beta}}+3e^{\hat{\beta}}}{e^{5\hat{\beta}}+2e^{2\hat{\beta}}+3e^{\hat{\beta}}+M-6}K_{55} \\
&- \frac{5e^{5\hat{\beta}}+3e^{3\hat{\beta}}+4e^{\hat{\beta}}}{e^{5\hat{\beta}}+e^{3\hat{\beta}}+4e^{\hat{\beta}}+M-6}K_{56} - \frac{6e^{6\hat{\beta}}+2e^{2\hat{\beta}}+4e^{\hat{\beta}}}{e^{6\hat{\beta}}+e^{2\hat{\beta}}+4e^{\hat{\beta}}+M-6}K_{57} - \frac{7e^{7\hat{\beta}}+5e^{\hat{\beta}}}{e^{7\hat{\beta}}+5e^{\hat{\beta}}+M-6}K_{58} - \frac{10e^{2\hat{\beta}}+2e^{\hat{\beta}}}{5e^{2\hat{\beta}}+2e^{\hat{\beta}}+M-7}K_{59} \\
&- \frac{3e^{3\hat{\beta}}+6e^{2\hat{\beta}}+3e^{\hat{\beta}}}{e^{3\hat{\beta}}+3e^{2\hat{\beta}}+3e^{\hat{\beta}}+M-7}K_{60} - \frac{6e^{3\hat{\beta}}+2e^{2\hat{\beta}}+4e^{\hat{\beta}}}{2e^{3\hat{\beta}}+e^{2\hat{\beta}}+4e^{\hat{\beta}}+M-7}K_{61} - \frac{4e^{4\hat{\beta}}+4e^{2\hat{\beta}}+4e^{\hat{\beta}}}{e^{4\hat{\beta}}+2e^{2\hat{\beta}}+4e^{\hat{\beta}}+M-7}K_{62} - \frac{4e^{4\hat{\beta}}+3e^{3\hat{\beta}}+5e^{\hat{\beta}}}{e^{4\hat{\beta}}+e^{3\hat{\beta}}+5e^{\hat{\beta}}+M-7}K_{63} \\
&- \frac{5e^{5\hat{\beta}}+2e^{2\hat{\beta}}+5e^{\hat{\beta}}}{e^{5\hat{\beta}}+e^{2\hat{\beta}}+5e^{\hat{\beta}}+M-7}K_{64} - \frac{6e^{6\hat{\beta}}+6e^{\hat{\beta}}}{e^{6\hat{\beta}}+6e^{\hat{\beta}}+M-7}K_{65} - \frac{8e^{2\hat{\beta}}+4e^{\hat{\beta}}}{4e^{2\hat{\beta}}+4e^{\hat{\beta}}+M-8}K_{66} - \frac{3e^{3\hat{\beta}}+4e^{2\hat{\beta}}+5e^{\hat{\beta}}}{e^{3\hat{\beta}}+2e^{2\hat{\beta}}+5e^{\hat{\beta}}+M-8}K_{67} - \frac{6e^{3\hat{\beta}}+6e^{\hat{\beta}}}{2e^{3\hat{\beta}}+6e^{\hat{\beta}}+M-8}K_{68} \\
&- \frac{4e^{4\hat{\beta}}+2e^{2\hat{\beta}}+6e^{\hat{\beta}}}{e^{4\hat{\beta}}+e^{2\hat{\beta}}+6e^{\hat{\beta}}+M-8}K_{69} - \frac{5e^{5\hat{\beta}}+7e^{\hat{\beta}}}{e^{5\hat{\beta}}+7e^{\hat{\beta}}+M-8}K_{70} - \frac{6e^{2\hat{\beta}}+6e^{\hat{\beta}}}{3e^{2\hat{\beta}}+6e^{\hat{\beta}}+M-9}K_{71} - \frac{3e^{3\hat{\beta}}+2e^{2\hat{\beta}}+7e^{\hat{\beta}}}{e^{3\hat{\beta}}+e^{2\hat{\beta}}+7e^{\hat{\beta}}+M-9}K_{72} - \frac{4e^{4\hat{\beta}}+8e^{\hat{\beta}}}{e^{4\hat{\beta}}+8e^{\hat{\beta}}+M-9}K_{73} \\
&- \frac{4e^{2\hat{\beta}}+8e^{\hat{\beta}}}{2e^{2\hat{\beta}}+8e^{\hat{\beta}}+M-10}K_{74} - \frac{3e^{3\hat{\beta}}+9e^{\hat{\beta}}}{e^{3\hat{\beta}}+9e^{\hat{\beta}}+M-10}K_{75} - \frac{2e^{2\hat{\beta}}+10e^{\hat{\beta}}}{e^{2\hat{\beta}}+10e^{\hat{\beta}}+M-11}K_{76} - \frac{12e^{\hat{\beta}}}{12e^{\hat{\beta}}+M-12}K_{77}=0
\end{aligned}
\tag{10}
$$

3.1 Observed Fisher Information

Often, in practice, it is not possible to calculate the expected Fisher information $I\left(\theta\right)$. In such cases, we can adopt the observed Fisher information, $I_{obs}\left(\theta\right)$ instead. Furthermore, it has been shown (Efron & Hinkley, 1978) that the use of the observed information number is superior to the expected information number, as it appears in the Cramér–Rao lower bound. The observed Fisher information, in terms of the pseudo-likelihood function, is defined by

$$I_{obs}(\theta) = \left[\frac{\partial}{\partial\theta}\log PL(x;\theta)\right]^2 \tag{11}$$

and can be estimated by the following, justified by the Law of Large Numbers:

$$\hat{I}^1_{obs}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial\theta}\log p(x_i;\theta)\right]^2\Bigg|_{\theta=\hat{\theta}} \tag{12}$$

since $I(\theta) = E\left[\hat{I}^1_{obs}(\theta)\right]$, making $\hat{I}^1_{obs}(\theta) \approx I(\theta)$. Similarly, $I_{obs}(\theta)$ can be estimated using the second derivative of the pseudo-likelihood function:

$$\hat{I}^2_{obs}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial^2}{\partial\theta^2}\log p(x_i;\theta)\right]\Bigg|_{\theta=\hat{\theta}} \tag{13}$$

3.2 On the Asymptotic Variances of *GMRF* model *MPL* estimators

The asymptotic covariance matrix for *MPL* estimators is given by (Liang and Yu, 2003):

$$C(\boldsymbol{\theta}) = H^{-1}(\boldsymbol{\theta})\,J(\boldsymbol{\theta})\,H^{-1}(\boldsymbol{\theta}) \tag{14}$$

where $J(\boldsymbol{\theta})$ and $H(\boldsymbol{\theta})$ are functions of the Jacobian (first order partial derivatives) and Hessian (second order partial derivatives) matrices respectively:

$$\begin{aligned} H(\boldsymbol{\theta}) &= E_\theta\left[\nabla^2 F(X;\boldsymbol{\theta})\right] \\ J(\boldsymbol{\theta}) &= Var_\theta\left[\nabla F(X;\boldsymbol{\theta})\right] \end{aligned} \tag{15}$$

with $F(X;\boldsymbol{\theta})$ denoting the logarithm of the pseudo-likelihood function.

Considering that the *GMRF* hyperparameters $\theta_1, \theta_2, \theta_3, \theta_4$ are uncorrelated, we have a diagonal covariance matrix, given by:

$$C(\boldsymbol{\theta}) = \begin{bmatrix} c_{11}(\boldsymbol{\theta}) & 0 & 0 & 0 \\ 0 & c_{22}(\boldsymbol{\theta}) & 0 & 0 \\ 0 & 0 & c_{33}(\boldsymbol{\theta}) & 0 \\ 0 & 0 & 0 & c_{44}(\boldsymbol{\theta}) \end{bmatrix} \tag{16}$$

with asymptotic variances given by:

$$c_{kk}\left(\boldsymbol{\theta}\right)=\frac{Var_{\theta}\left[\frac{\partial}{\partial\theta_k}\log PL\left(\boldsymbol{\theta}\right)\right]}{E_{\theta}^{2}\left[\frac{\partial^{2}}{\partial\theta_k^{2}}\log PL\left(\boldsymbol{\theta}\right)\right]} \tag{17}$$

for $k=1,\ldots,4$. Rewriting the above equation using the definition of variance leads to:

$$c_{kk}\left(\boldsymbol{\theta}\right)=\frac{E_{\theta}\left[\left(\frac{\partial}{\partial\theta_k}\log PL\left(\boldsymbol{\theta}\right)\right)^{2}\right]}{E_{\theta}^{2}\left[\frac{\partial^{2}}{\partial\theta_k^{2}}\log PL\left(\boldsymbol{\theta}\right)\right]}\approx\frac{\hat{I}_{obs}^{1}\left(\boldsymbol{\theta}\right)}{\left[\hat{I}_{obs}^{2}\left(\boldsymbol{\theta}\right)\right]^{2}} \tag{18}$$

since the expected value of the log *PL* equation is zero:

$$\begin{aligned}E_{\theta}\left[\frac{\partial}{\partial\theta_k}\log PL\left(\boldsymbol{\theta}\right)\right]&\approx\frac{1}{N}\sum_{i=1}^{N}\left[\frac{\partial}{\partial\theta_k}\log p\left(x_i|\eta^{s},\boldsymbol{\theta}\right)\right]\Bigg|_{\theta=\hat{\theta}_{MPL}}\\&=\frac{1}{N}\frac{\partial}{\partial\theta_k}\log\prod_{i=1}^{N}p\left(x_i|\eta^{s},\boldsymbol{\theta}\right)\Bigg|_{\theta=\hat{\theta}_{MPL}}\\&=\frac{1}{N}\frac{\partial}{\partial\theta_k}\log PL\left(\boldsymbol{\theta}\right)\Big|_{\theta=\hat{\theta}_{MPL}}=0\end{aligned} \tag{19}$$

Thus, from the *LCDF* of the *GMRF* model and after some simple algebraic manipulations, we obtain the following expression for $\hat{I}_{obs}^{1}\left(\boldsymbol{\theta}\right)$ (Levada et al., 2008c):

$$\hat{I}_{obs}^{1}\left(\boldsymbol{\theta}\right)=\frac{1}{N\sigma^{4}}\sum_{i=1}^{N}\left\{\left[\boldsymbol{y}_{ij}-\boldsymbol{\theta}^{T}\boldsymbol{\Psi}_{ij}-\mu\left(1-2\boldsymbol{\theta}^{T}\boldsymbol{I}\right)\right]\left[\Psi_{ij}^{k}-2\mu\right]\right\}^{2} \tag{20}$$

where Ψ_{ij}^{k} denotes the *k-th* element of $\boldsymbol{\Psi}_{ij}$, $k=1,\ldots,4$. Similarly, for $\hat{I}_{obs}^{2}\left(\boldsymbol{\theta}\right)$ we have:

$$\hat{I}_{obs}^{2}\left(\boldsymbol{\theta}\right)=-\frac{1}{N\sigma^{2}}\sum_{i=1}^{N}\left\{\Psi_{ij}^{k}-2\mu\right\}^{2} \tag{21}$$

The proposed approximation allows the calculation of the asymptotic variance of maximum pseudo-likelihood estimators of the *GMRF* model in computationally feasible way. From previous works on statistical inference it can be shown that *MPL* estimators are asymptotically normal distributed. Therefore, with the proposed method, it is possible to completely characterize the asymptotic behavior of the *MPL* estimators of the *GMRF* model, allowing interval estimation, hypothesis testing and quantitative analysis on the model parameters in a variety of research areas, including image processing and pattern recognition (Levada et al., 2008d).

In order to demonstrate the application of the asymptotic variance estimation in stochastic image modeling, we present the results obtained in experiments using Markov Chain Monte Carlo simulation methods (Dubes & Jain, 1989; Winkler, 2006) by comparing the values of $\hat{\boldsymbol{\theta}}_{MPL}$ and asymptotic variances regarding second order neighbourhood systems using synthetic images, representing several *GMRF* model outcomes. For the experiments below, we adopted the Metropolis algorithm (Metropolis et al., 1953), a single spin flip *MCMC* method, to simulate occurrences of *GRMF* model using different known parameter vectors. Simulated images are shown in Figure 5. The *MPL* estimators, obtained by (3) were compared with the real parameter vectors. In all cases, the parameters μ and σ^2 were defined as zero and five, respectively. The vector parameters used in the simulated images were $\boldsymbol{\theta} = [0.25, 0.3, -0.1, 0.2]$ and $\boldsymbol{\theta} = [0.2, 0.15, 0.07, 0.05]$.

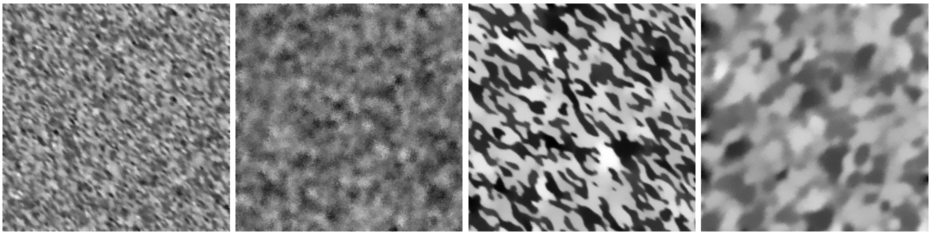

a.) b.) c.) d.)

Fig. 5. Synthetic images generated by *MCMC* simulation for *GMRF* model using a second-order neighbourhood system.

A major difficulty in *GMRF* models is the selection of parameters $\boldsymbol{\theta}$ for which the correlation matrix is positive definite, introducing one more problem in parameter estimation. With discrete *MRF's* almost any parameter in the parametric space lead to a mathematically valid model. However, only a portion of the parametric space generates valid *GMRF* models. The region of validity is known only for first-order systems, but not for higher-order *GMRF's* (Dubes & Jain, 1989). In fact, even parameters estimated by standard procedures may not be in the region of validity and simulations may not work properly.

Table 2 shows the *MPL* estimators, estimated asymptotic variances, 90% confidence intervals regarding the *GMRF* model parameters for the synthetic image indicated in Figure 1a. Similarly, Table 3 shows the same obtained results for the images shown in Figure 1b. The results on *MCMC* simulation images show that, in all cases, the true parameter value is contained in the obtained intervals, assessing the accuracy of the proposed methodology for asymptotic variance estimation.

K	θ_k	$\hat{\theta}_k$	$\hat{\sigma}_k$	90% CI
1	0.25	0.2217	0.0390	[0.1799 0.3077]
2	0.3	0.2758	0.0387	[0.2398 0.3667]
3	-0.1	-0.1145	0.0394	[-0.1711 -0.0479]
4	0.2	0.1743	0.0386	[0.1150 0.2416]

Table 2. *MPL* estimators, asymptotic variances and 90% confidence intervals for *GMRF* hyperparameters on simulated images (1a).

K	θ_k	$\hat{\theta}_k$	$\hat{\sigma}_k$	90% CI
1	0.2	0.1908	0.0506	[0.1079 0.2738]
2	0.15	0.1605	0.0524	[0.0746 0.2464]
3	0.07	0.0716	0.0482	[-0.0074 0.1506]
4	0.05	0.0523	0.0418	[-0.0146 0.1192]

Table 3. *MPL* estimators, asymptotic variances and 90% confidence intervals for *GMRF* hyperparameters on simulated images (1b).

3.3 On the Asymptotic Variance of *Potts MRF* model *MPL* estimator

Similarly to the *GMRF* model, we define an approximation for the asymptotic variance of Potts model *MPL* estimators through expressions for $\hat{I}^1_{obs}(\boldsymbol{\theta})$ and $\hat{I}^2_{obs}(\boldsymbol{\theta})$. From the *LCDF* of the Potts model (4) we have:

$$\hat{I}^1_{obs}(\beta) = \frac{1}{N}\sum_{i=1}^{N}\left\{\left[U_i(m) - \frac{\sum_{l=1}^{M} U_i(l)\exp\{\beta U_i(l)\}}{\sum_{l=1}^{M}\exp\{\beta U_i(l)\}}\right]^2\right\} \tag{22}$$

which, after some few algebraic manipulations, becomes (Levada et al, 2008a):

$$\hat{I}^1_{obs}(\beta) = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{\sum_{l=1}^{M}\left[\sum_{k=1}^{M}\left(U_i(m)-U_i(l)\right)\left(U_i(m)-U_i(k)\right)\exp\{\beta\left(U_i(l)+U_i(k)\right)\}\right]}{\left[\sum_{l=1}^{M}\exp\{\beta U_i(l)\}\right]^2}\right\} \tag{23}$$

Calculating the observed Fisher information using the second derivative of the pseudo-likelihood function leads to the following:

$$\hat{I}^2_{obs}(\beta) = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{\left[\sum_{l=1}^{M} U_i(l)^2\exp\{\beta U_i(l)\}\right]\left[\sum_{l=1}^{M}\exp\{\beta U_i(l)\}\right] - \left[\sum_{l=1}^{M} U_i(l)\exp\{\beta U_i(l)\}\right]^2}{\left[\sum_{l=1}^{M}\exp\{\beta U_i(l)\}\right]^2}\right\} \tag{24}$$

Simplifying equation (24), we have the final expression for $\hat{I}^2_{obs}(\boldsymbol{\theta})$ (Levada et al, 2008b):

$$\hat{I}^2_{obs}(\beta) = \frac{1}{N}\sum_{i=1}^{N}\left\{\frac{\sum_{l=1}^{M-1}\left[\sum_{k=l}^{M}\left(U_i(l)-U_i(k)\right)^2 \exp\left\{\beta\left(U_i(l)+U_i(k)\right)\right\}\right]}{\left[\sum_{l=1}^{M}\exp\left\{\beta U_i(l)\right\}\right]^2}\right\} \tag{25}$$

This approximation allows the calculation of the asymptotic variance of the maximum pseudo-likelihood estimator of the Potts *MRF* model. In order to demonstrate the application of the asymptotic variance in testing and evaluating the proposed pseudo-likelihood equation, some results obtained in experiments with Markov Chain Monte Carlo simulation methods are presented. The results show the values of $\hat{\beta}_{MPL}$, asymptotic variances, test statistics and *p-values* for several synthetic images generated using *MCMC* algorithms on second and third order neighbourhood systems. The objective is to validate the following hypothesis:

> *H*: the proposed pseudo-likelihood equations provide results that are statistically equivalent to the real parameter values, that is:

$$H : \beta = \hat{\beta}_{MPL} \tag{26}$$

Using the consistency property of *MPL* estimators and adopting the derived approximation for the asymptotic variance it is possible to completely characterize the asymptotic distribution of the Potts model parameter estimator and define the following test statistic:

$$Z = \frac{\beta-\hat{\beta}_{MPL}}{\hat{\sigma}^2\left(\hat{\beta}_{MPL}\right)} \approx N(0,1) \tag{27}$$

creating the decision rule: Reject *H* if $|Z| > c$. Considering a test size α (in all experiments along this chapter we set $\alpha = 0.1$), that is, the maximum probability of incorrectly rejecting *H* is α, we have $c = 1.64$. However, in order to quantify the evidence against or in favor of the hypothesis a complete analysis in terms the test size, test statistic and p-values, calculated by $P\left(|Z|>z_{obs}\right)$, is required. In case of a small p-value, we should doubt of the hypothesis being tested. In other words, to reject *H* we should have a test size α significantly higher than the p-value. This approach provides a statistically meaningful way to report the results of a hypothesis testing procedure.

For the experiments, to illustrate the example of statistical analysis in *MRF*, we adopted both single spin-flip *MCMC* methods, *Gibbs Sampler* and *Metropolis*, and a cluster-flipping *MCMC* method, the *Swendsen-Wang* algorithm, to generate several Potts model outcomes using different known parameter values. Figures 6, 7, 8 and 9 show the simulated images.

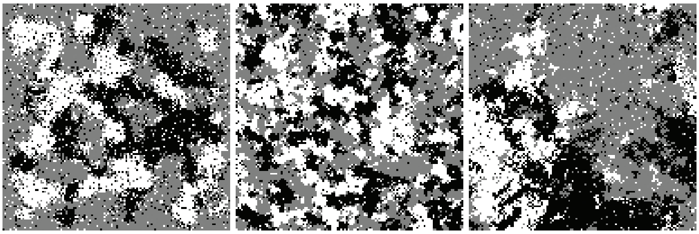


Fig. 6. Synthetic images generated by *MCMC* simulation algorithms using second-order neighbourhood systems for *M*=3: *Gibbs Sampler* ($\beta = 0.45$), *Metropolis* ($\beta = 0.5$) and *Swendsen-Wang* ($\beta = 0.4$), respectively.

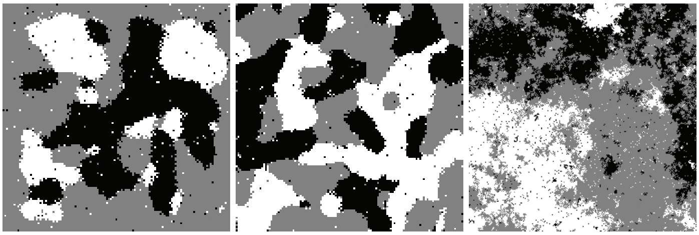


Fig. 7. Synthetic images generated by *MCMC* simulation algorithms using third-order neighbourhood systems for *M*=3: *Gibbs Sampler* ($\beta = 0.45$), *Metropolis* ($\beta = 0.5$) and *Swendsen-Wang* ($\beta = 0.4$), respectively.

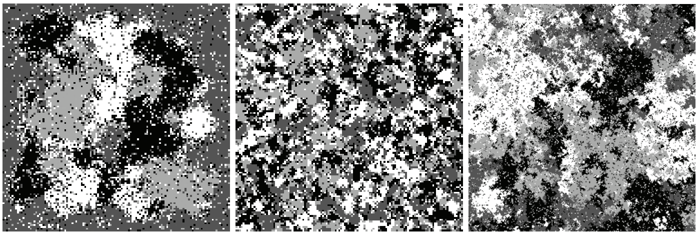


Fig. 8. Synthetic images generated by *MCMC* simulation algorithms using second-order neighbourhood systems for *M*=4: *Gibbs Sampler* ($\beta = 0.45$), *Metropolis* ($\beta = 0.5$) and *Swendsen-Wang* ($\beta = 0.4$), respectively.

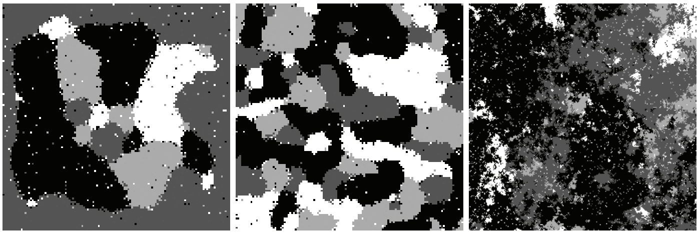


Fig. 9. Synthetic images generated by *MCMC* simulation algorithms using second-order neighbourhood systems for *M*=4: *Gibbs Sampler* ($\beta = 0.45$), *Metropolis* ($\beta = 0.5$) and *Swendsen-Wang* ($\beta = 0.4$), respectively.

The *MPL* estimators, obtained by the derived pseudo-likelihood equations were compared with the real parameter values. This information, together with the test statistics and the *p-values*, obtained from the approximation to the asymptotic variance provide a mathematical procedure to validate and assess the accuracy of the pseudo-likelihood equations. Tables 4 and 5 show the obtained results.

	Swendsen-Wang		*Gibbs Sampler*		*Metropolis*	
M	3	4	3	4	3	4
β	0.4	0.4	0.45	0.45	0.5	0.5
$\hat{\beta}_{MPL}$	0.4460	0.4878	0.3849	0.4064	0.4814	0.4884
$\lvert\beta-\hat{\beta}_{MPL}\rvert$	0.0460	0.0878	0.0651	0.0436	0.0186	0.0111
$\hat{I}_{obs}^{1}(\boldsymbol{\theta})$	0.4694	0.6825	0.8450	1.3106	0.3908	0.8258
$\hat{I}_{obs}^{2}(\boldsymbol{\theta})$	3.0080	3.3181	3.8248	4.5387	2.2935	2.6436
$\hat{\sigma}_n^2(\hat{\beta}_{MPL})$	0.0519	0.0620	0.0578	0.0636	0.0743	0.1182
Z_n	0.2458	0.3571	0.2707	0.1729	0.0682	0.0322
p-values	0.8104	0.7264	0.7872	0.8650	0.9520	0.9760

Table 4. *MPL* estimators, observed Fisher information, asymptotic variances, test statistics and p-values for synthetic *MCMC* images using second-order systems.

	Swendsen-Wang		*Gibbs Sampler*		*Metropolis*	
M	3	4	3	4	3	4
β	0.4	0.4	0.45	0.45	0.5	0.5
$\hat{\beta}_{MPL}$	0.3602	0.3772	0.4185	0.4309	0.4896	0.4988
$\lvert\beta-\hat{\beta}_{MPL}\rvert$	0.0398	0.0228	0.0315	0.0191	0.0104	0.0012
$\hat{I}_{obs}^{1}(\boldsymbol{\theta})$	0.2738	0.5372	0.1104	0.1433	0.0981	0.1269
$\hat{I}_{obs}^{2}(\boldsymbol{\theta})$	3.5691	4.6800	1.8703	2.3416	1.4165	1.4547
$\hat{\sigma}_n^2(\hat{\beta}_{MPL})$	0.0215	0.0245	0.0316	0.0261	0.0489	0.0600
Z_n	0.2510	0.1456	0.1772	0.1182	0.0470	0.0049
p-values	0.8036	0.8886	0.8572	0.9044	0.9602	0.9940

Table 5. *MPL* estimators, observed Fisher information, asymptotic variances, test statistics and p-values for synthetic *MCMC* images using third-order systems.

The obtained results show that the asymptotic variance is reduced in third-order systems, increasing the *p-values*, suggesting that the use of higher-order systems improves Potts *MRF* model *MPL* estimation. Considering the observed data, we conclude that there are no significant differences between β and $\hat{\beta}_{MPL}$. Therefore, based on statistical evidences, we should accept the hypothesis *H*, assessing the accuracy of the *MPL* estimation method.

4. Contextual Classification and Bayesian Inference

As presented in the previous sections of this chapter, multispectral image contextual classification through *MRF* models is stated as a Bayesian inference problem, since we are interested in the solution that maximizes the *a posteriori* distribution. Several combinatorial optimization algorithms can be used to approximate the *MAP* estimator, although is has been shown that the only optimal method is *Simulated Annealing* (*SA*). However, due to its high computational cost, *SA* may not be the best choice for real applications. In this chapter we discuss two suboptimal methods used in contextual classification: *ICM* and *MPM*. The main difference between these methods is that while the first one is the optimum Bayesian estimator in case of a uniform cost function, the later is optimum regarding a Hamming distance cost function (Won & Gray, 2004).

4.1 *Iterated Conditional Modes*

The *ICM* algorithm was originately proposed by Besag as a computationally feasible iterative and determinístic algorithm for approximating the *MAP* estimator in complex problems. The basic idea consists in, for each pixel, update its current value with the label that maximizes the *a posteriori* probability. By noting that $P(\boldsymbol{x}|\boldsymbol{y}) = P\left(x_{ij}|\boldsymbol{x}_{\overline{ij}},\boldsymbol{y}\right)P\left(\boldsymbol{x}_{\overline{ij}}|\boldsymbol{y}\right)$, where $\boldsymbol{x}_{\overline{ij}}$ denotes the entire random field without the current element x_{ij}, the subsequent maximization of $P\left(x_{ij}|\boldsymbol{x}_{\overline{ij}},\boldsymbol{y}\right)$ is always moving towards the a maximum of the *a posteriori* probability. Thus, *ICM* rapidly converges to a local maximum since its results are strongly dependent on the initialization. The *ICM* algorithm, as described in (Dubes & Jain, 1989) is given in the following.

Algoritm 1. *Iterated Conditinal Modes (ICM)*

```
1. Chose a MRF model for the label field $\boldsymbol{X}$ .
2. Initialize $\hat{x}$ by choosing the label $\hat{x}_{ij}$ that maximizes $p\left(y_{ij}|x_{ij}\right)$, that is, the result of
   maximum likelihood classification.
3. For $i = 1,\dots,M$ and $j = 1,\dots,N$
      a. Update the label $\hat{x}_{ij}$ with the value that maximizes
            $p\left(x_{ij}|\boldsymbol{x}_{\overline{ij}},\boldsymbol{y}\right) \propto p\left(y_{ij}|x_{ij}\right)p\left(x_{ij}|\eta_{ij}^{s}\right)$
4. Repeat (3) $N_{iter}$ times
```

So, in our case, we always update the current label with the new value that maximizes the product between the *LCDF's* of *GMRF* and Potts models. Note, however, that step (2) is being generalized, since instead of maximum likelihood classification, we can initialize *ICM* using several pattern classifiers.

4.2 Maximizer of the Posterior Marginals

As the name says, the *MPM* estimator is obtained by maximizing the posterior marginal probabilities $P\left(x_{ij}|\boldsymbol{y}\right)$. Thus, the fundamental point here is the calculation of these distributions. The *MPM* algorithm, as proposed in (Marroquin et al., 1987), uses *MCMC* methods to approximate these distributions. Basically, the *MPM* algorithm simulate a Markov chain over the states that represent all possible configurations of the random field. The idea is that as each pixel is repeatedly visited, the resulting Markov chain generates a sample of the posterior distribution $P\left(\boldsymbol{x}|\boldsymbol{y}\right)$, regardless of the initial conditions. As a result of that, a sequence of configurations $x(0) \to x(1) \to \cdots x(n) \to \cdots$, corresponding to a Makov chain that reaches its equilibrium state, is generated. Once this state is reached, we can regard all configurations from this point as a sample of $P\left(\boldsymbol{x}|\boldsymbol{y}\right)$. Besides, in the equilibrium state the expected value of a function of a random variable can be estimated through the ergodic principle. Using this result, the posterior marginal distribution for the *MPM* algorihtm can be approximated by (Dubes & Jain, 1989):

$$P\left(x_{ij}=g|\boldsymbol{y}\right) \approx \frac{1}{(n-k)} \sum_{p=k+1}^{n} \delta\left(x_{ij}^{p}-g\right) \tag{28}$$

where k is the number of steps needed to the sequence to stabilize and n is a sufficiently large number of iterations so that the estimation is accurate at a certain reasonable computational cost. One problem with this approach is exactly how to choose these values (also known as *magic numbers*). Usually, they are both chosen empiracally. The pseudo-code for *MPM*, as described in (Dubes & Jain, 1989), is shown in Algorithm 2.

5. Metrics for Performance Evaluation of Image Classification

In order to evaluate the performance of the contextual classification in an objective way, the use of quantitative measures is required. Often, the most widely used criteria for evaluation of classification tasks are the correct classification rate and/or the estimated classification error (*holdout, resubstitution*). However, these measures do not allow robust statistical analysis, neither inference about the obtained results. To surmount this problem, statisticians usually adopt Cohen's *Kappa* coefficient, a measure to assess the accuracy in classification problems.

5.1 Cohen's *Kappa* Coefficient

This coefficient was originally proposed by Cohen (Cohen, 1960), as a measure of agreement between rankings and opinions of different specialists. In pattern recognition, this coefficient determines a degree of agreement between the "*ground truth*" and the output of a given classifier. The better the classification performance, the higher is the *Kappa* value. In case of perfect agreement, *Kappa* is equal to one. The main motivation for the use of *Kappa* is that it has good statistical properties, such as asymptotic normality, and also the fact that it is easily computed from the confusion matrix.

```
Algorithm 2. Maximizer of the Posterior Marginals (MPM)
1. Chose a MRF model for the label field $\boldsymbol{X}$ .
2. Initialize $\hat{x}$ by choosing the label $\hat{x}_{ij}$ that maximizes $p\left(y_{ij}|x_{ij}\right)$, that is, the result of maximum likelihood classification.
3. For $i = 1,\dots,M$ and $j = 1,\dots,N$
    a. Choose a random label $g$ and define $z_{ij} = g$. Let $z_{kl} = x_{kl}$ for all $(k,l) \neq (i,j)$
    b. Let $p = \min\left\{1, \frac{P(X=z|Y=y)}{P(X=x|Y=y)}\right\}$
    c. Replace $x$ by $z$ with probability $p$ .
4. Repeat (3) N times, saving the realizations from $x^{(l+1)}$ to $x^{(n)}$
5. Calculate $P\left(x_{ij}=g|\boldsymbol{y}\right)$ for all pixels according equation (28).
6. For $i = 1,\dots,M$ and $j = 1,\dots,N$
    a. Choose the label $\hat{x}_{ij}$ that maximizes the posterior marginal among all possible labels
       $P\left(x_{ij}=\hat{x}_{ij}|\boldsymbol{y}\right) > P\left(x_{ij}=g|\boldsymbol{y}\right)$
```

The expression for the *Kappa* coefficient from the confusion matrix is given by (Congalton, 1991):

$$\hat{K} = \frac{N\sum_{i=1}^{c} c_{ii} - \sum_{i=1}^{c} x_{i+}x_{+i}}{N^2 - \sum_{i=1}^{c} x_{i+}x_{+i}} \tag{29}$$

where c_{ii} is an element of the confusion matrix, x_{+i} is the sum of the elements of column *i*, x_{i+} is the sum of the elements of the row *i*, c is the number of classes and *N* is the number of training samples. The asymptotic variance of this estimator is given by:

$$\hat{\sigma}_K^2 = \frac{1}{N}\left[\frac{\theta_1 \cdot \left(1-\theta_1\right)}{\left(1-\theta_2\right)^2} + \frac{2 \cdot \left(1-\theta_1\right) \cdot \left(2 \cdot \theta_1 \cdot \theta_2 - \theta_3\right)}{\left(1-\theta_2\right)^3} + \frac{\left(1-\theta_1\right)^2 \cdot \left(\theta_4 - 4 \cdot \theta_2^2\right)}{\left(1-\theta_2\right)^4}\right] \tag{30}$$

where

$$\theta_1 = \frac{1}{N}\sum_{i=1}^{c} x_{ii} \qquad \theta_2 = \frac{1}{N^2}\sum_{i=1}^{c} x_{i+}x_{+i} \tag{31}$$

$$\theta_3 = \frac{1}{N^2}\sum_{i=1}^{c} x_{ii} \cdot \left(x_{i+} + x_{+i}\right) \qquad \theta_4 = \frac{1}{N^3}\sum_{i=1}^{r}\sum_{j=1}^{r} x_{ij} \cdot \left(x_{j+} + x_{+i}\right)^2 \tag{32}$$

6. Experiments and Results in Nuclear Magnetic Resonance Images

In order to test and evaluate the contextual classification methods previously described in this chapter, we show some experiments in *NMR* images of *marmocets* brains, a monkey species often used in medical experiments. These images were acquired by the *CInAPCe* project, an abbreviation for the Portuguese expression "*Inter-Institutional Cooperation to Support Brain Research*", a Brazilian research project that has as main purpose the establishment of a scientific network seeking the development of neuroscience research through multidisciplinary approaches. In this sense, pattern recognition can contribute to the development of new methods and tools for processing and analyzing magnetic resonance imaging and its integration with other methodologies in the investigation of epilepsies. Figure 9 shows the bands *PD*, *T1* and *T2* of a *marmocet NMR* multispectral brain image.

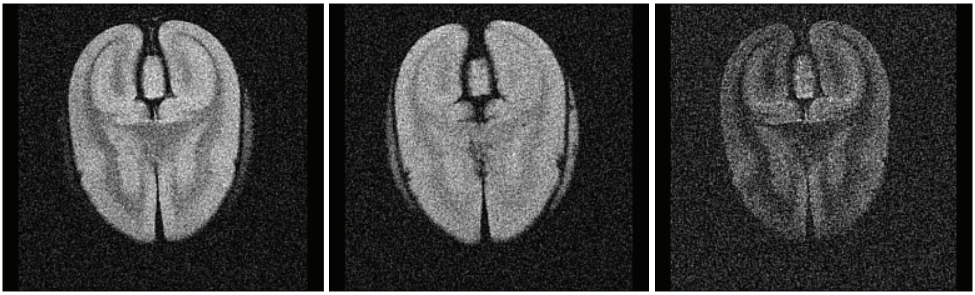


Fig. 9. *PD*, *T1* and *T2 NMR* image bands of the multispectral *marmocet* brain image.

The contextual classification of the multispectral image was performed by applying both *ICM* and *MPM* using second and third order neighbourhood systems on several initializations provided by seven different pattern classifiers: Quadratic Bayesian Classifier (*QDC*) and Linear Bayesian Classifier (*LDC*) under Gaussian hypothesis, Parzen-Windows Classifier (*PARZENC*), K-Nearest Neighbour Classifier (*KNNC*), Logistic Classifier (*LOGLC*), Nearest Mean Classifier (*NMC*) and Decision Tree Classifier (*TREEC*). All the experiments were implemented in *MATLAB*, using the pattern recognition toolbox *PRTOOLS v.4.1*[1], developed at Delft University, to provide the initializations through each one of the above classifiers.

The experiments were conducted to clarify some hypotheses and conjectures regarding contextual classification. We want to verify the following hypothesis:

A. Contextual classification is capable of significantly improving the performance of ordinary classification techniques (punctual methods).

B. Different initializations can lead to statistically different contextual classification results (for the same iterative algorithm).

C. The use of higher-order systems is capable of significantly improving the performance of contextual classification.

D. Different contextual classification algorithms are capable of producing statistically different results (for the same initialization).

We used 100 training samples for each class: white matter, gray matter and background. The classification errors and confusion matrix are estimated by the *10-Fold cross-validation* method. Convergence was considered by achieving one of two conditions: less than 1% of the pixels are updated in the current iteration, or the maximum of 5 iterations is reached. Tables 6, 7, 8, 9 and 10 show the obtained results.

Classifiers	$\hat{k}$	$\hat{\sigma}^2_{kappa}$
PARZENC	0.7816	0.00031061
KNNC	0.7550	0.00034100
LOGLC	0.7583	0.00033502
LDC	0.7716	0.00032177
QDC	0.7866	0.00030515
NMC	0.7850	0.00030629
TREEC	0.6500	0.00044737

Table 6. *Kappa* coefficients and variances for punctual classification results.

Classifiers	$\hat{k}$	$\hat{\sigma}^2_{kappa}$
PARZENC	0.9783	3.5584e-005
KNNC	0.9733	4.3614e-005
LOGLC	0.9900	1.6553e-005
LDC	0.9916	1.3811e-005
QDC	0.9700	4.8952e-005
NMC	0.9966	5.5431e-006
TREEC	1.0000	0.0000

Table 7. *Kappa* coefficients and variances for *MPM* classification on second order systems.

Classifiers	$\hat{k}$	$\hat{\sigma}^2_{kappa}$
PARZENC	0.9966	5.5431e-006

[1] *Available online at http://www.prtools.org*

KNNC	1.0000	0.0000
LOGLC	0.9950	8.3055e-006
LDC	0.9966	5.5431e-006
QDC	0.9966	5.5431e-006
NMC	0.9933	1.1062e-005
TREEC	0.9966	5.5431e-006

Table 8. *Kappa* coefficients and variances for *MPM* classification on third order systems.

Classifiers	$\hat{k}$	$\hat{\sigma}^2_{kappa}$
PARZENC	0.9700	4.8975e-005
KNNC	0.9550	7.2614e-005
LOGLC	0.9600	6.4864e-005
LDC	0.9616	6.2206e-005
QDC	0.9516	7.7788e-005
NMC	0.9583	6.7446e-005
TREEC	1.0000	0.0000

Table 9. *Kappa* coefficients and variances for *ICM* classification on second order systems.

Classifiers	$\hat{k}$	$\hat{\sigma}^2_{kappa}$
PARZENC	0.9983	2.7747e-006
KNNC	0.9866	2.202e-005
LOGLC	0.9983	2.7747e-006
LDC	0.9950	8.3053e-006
QDC	0.9850	2.4743e-005
NMC	0.9950	8.3053e-006
TREEC	1.0000	0.0000

Table 10. *Kappa* coefficients and variances for *ICM* classification on third order systems.

6.1 Statistical Analysis

To test the hypothesis and validate the proposed methodology for contextual classification, both local (confidence intervals) and global (*T* test) analysis are performed. Let $\bar{k}_1$ and $\bar{k}_2$ be the mean *Kappa* coefficients before and after the application of a given technique. Defining $\bar{k} = \bar{k}_1 - \bar{k}_2$, it is desirable to test the following hypothesis:

$$\begin{aligned} H_0 &: \bar{k} = 0 \\ H_1 &: \bar{k} \neq 0 \end{aligned} \tag{33}$$

The test statistic *T*, defined as follows, has a *t-student* distribution with *n-1* degrees of freedom (*n=7* in this case):

$$T = \frac{\bar{k}}{\sigma_{\bar{k}}/\sqrt{n}} \tag{34}$$

where $\sigma_{\bar{k}}$ denotes the standard deviation of the punctual differences. Thus, considering a α test size (i.e., $\alpha = 0.05$ or $\alpha = 0.01$), H_0 must be rejected if T is less than a critical value t_c. This information, together with the *p-values*, allows a robust statistical analysis, as well as inferences about the problem in study. Note that the decision based on the *T* statistic is quite intuitive, since the greater the difference between the two means, more chance that we are dealing with distinct groups (captured by the numerator of *T*). On the other hand, the greater the variability of the results, more difficult is to detect differences on the means (captured by the denominator of *T*).

To verify the hypothesis A, a *T* test was performed using the data presented in Tables 6 and 9 (punctual classification x *ICM* on second order systems). Considering a test size $\alpha = 0.05$, we have a critical value of $t_c = -1.943$. The obtained results indicate strong evidences against H_0, since $\bar{k} = -0.2098$, $T = -8.7741$, leading to a *p-value* smaller than 0.0005. Therefore, we should reject H_0, assessing that combinatorial optimization algorithms can significantly improve the classification performance. Figure 10 shows a comparison of the visual results obtained by the *LOGLC* classifier and the *LOGLC+ICM* classification.

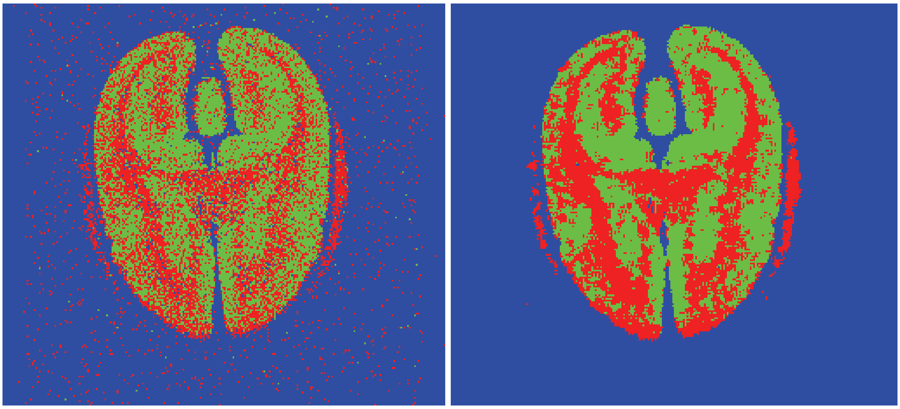


Fig. 10. Comparison between classification maps for the *marmocet* brain multispectral *NMR* image obtained using *LOGLC* and *LOGLC+ICM*

The hypothesis B was tested by simply constructing 95% confidence intervals (*CI*) for the respective *Kappa* coefficients. To illustrate the scenario, we compared the results of *KNNC+ICM* and *TREEC+ICM* classification form Table 9. The confidence intervals show that the results are statistically different, since for the *KNNC+ICM* we have [0.9387, 0.9713] and the *TREEC+ICM* provides a *Kappa* coefficient equal to one and with zero variance. Figure 11 compares the visual results. Actually, these results were expected since both combinatorial optimization algorithms *ICM* and *MPM* are sub-optimal, that is, they converge to different local maxima depending on the initialization.

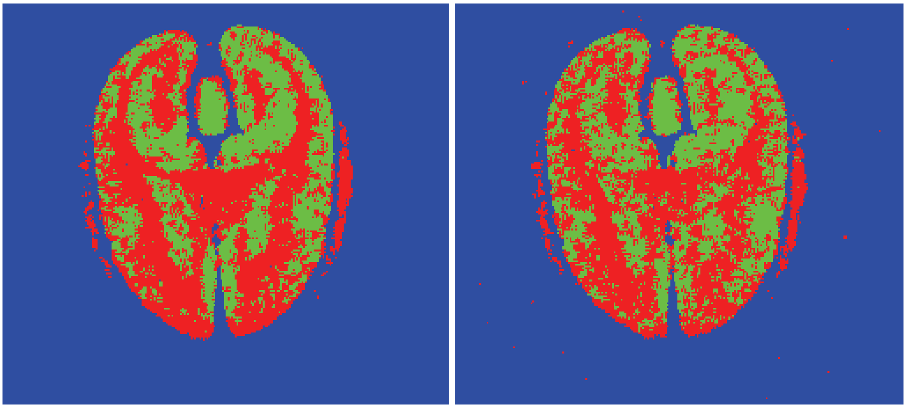


Fig. 11. Comparison between classification maps for the *marmocet* brain multispectral NMR image obtained using *KNNC+ICM* and *TREEC+ICM*.

To test the third hypothesis (C), a T test was performed in data from Tables 9 and 10 to compare the mean performances. The results show that the mean performances are significantly different, since $\bar{k} = -0.0288$, $T = -5.8115$, leading to a *p-value* smaller than 0.005 and once again, strong evidences against H_0. Figures 12 and 13 shows a comparison between *LOGLC+ICM* and *NMC+ICM* on second and third order systems.

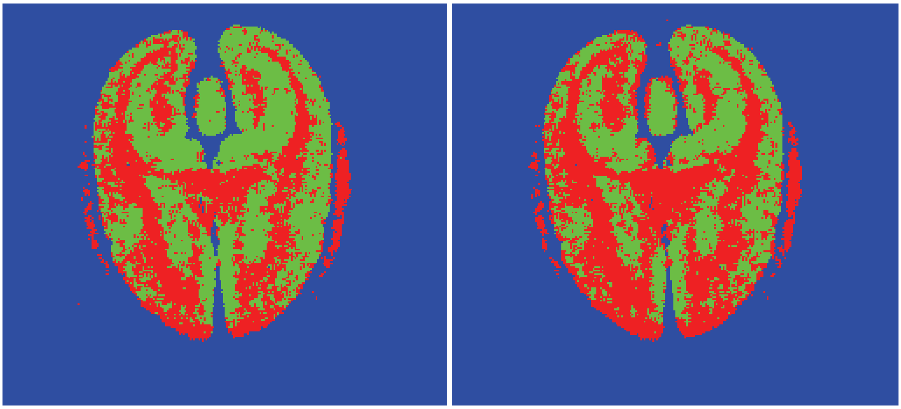


Fig. 12. Classification maps for *LOGLC+ICM* and *NMC+ICM* on second order systems.

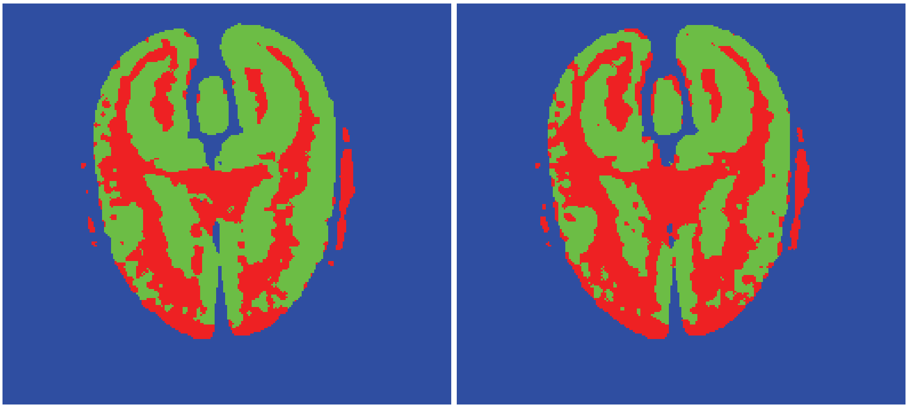


Fig. 13. Classification maps for *LOGLC+ICM* and *NMC+ICM* on third order systems.

Finally, to test the last hypothesis, 95% confidence intervals were constructed using the *Kappa* coefficient values together with the asymptotic variances. From Tables 7 and 9 it is possible to observe that *NMC+MPM* produces the interval [0.9920, 1], while *NMC+ICM* gives [0.9423, 0.9743], assessing that the performances are significantly different. This results and the result obtained on testing the hypothesis B suggest that the use of classifier combination rules can be explored in contextual classification problems, since there is significant differences in the results, providing complementary information that can be used to improve the performance even more. Figure 14 shows a comparison between the results of *MPM* and *ICM* for the same initialization (*NMC*).

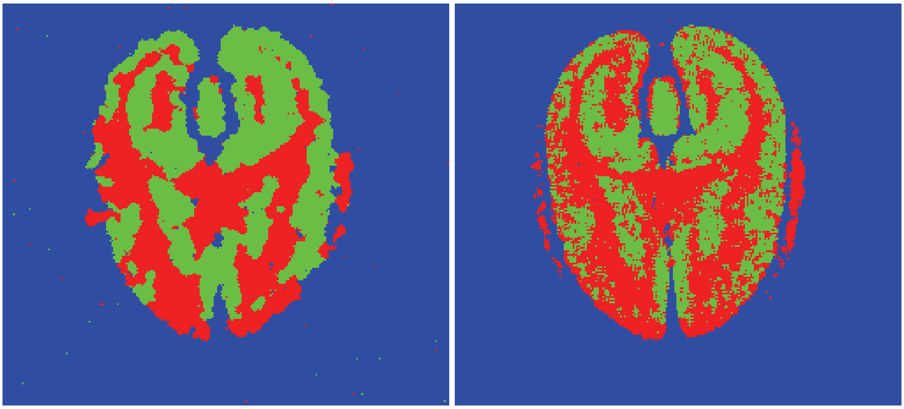


Fig. 14. Classification maps for *NMC+MPM* and *NMC+ICM* on second order systems.

7. Conclusions

In this chapter we discussed three important problems in pattern recognition. First, the derivation of novel pseudo-likelihood equations for Potts MRF model parameter estimation on higher-order neighbourhood systems. Then, the accuracy of MPL estimation was assessed through approximations for the asymptotic variance of these estimators. Finally, multispectral image contextual classification was stated as a Bayesian inference problem. The obtained results show that the approach discussed here is valid, and more, is capable of significantly improving classification performance. Future works in this research area include the study about the efficiency of the MPL estimation through the analysis of necessary/sufficient conditions of information equality in MRF models, as well as the combination of contextual classifiers aiming for a further improvement in classification performance.

8. Acknowledgments

We would like to thank FAPESP for the financial support through Alexandre L. M. Levada student scholarship (grant n. 06/01711-4). We also would like to thank Hilde Buzzá for the NMR images acquisition and for all the support and assistance throughout this process.

9. References

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society – Series B,* vol.36, pp. 192-236.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society – Series B,* vol.48, n. 3, pp. 259-302.

Bickel, P. J. (1991). *Mathematical Statistics,* Holden Day, New York.

Brent, R. P. (1973). *Algorithms for minimization without derivatives,* Prentice Hall, New York.

Casella, G. & Berger, R. L. (2002). *Statistical Inference,* 2nd Edition, Duxbury, New York.

Cohen, J. A. (1960). A coefficient of agreement for nominal scales, *Educational and Psycological Measurement*, v. 20, n. 1, pp. 37-46.

Congalton, R. G. (1991). A review of assessing the accuracy of classification of remotely sensed data, *Remote Sensing of Enviroment*, v. 37, pp. 35-46.

Dubes, R. & Jain, A. (1989). Random field models in image analysis, *Journal of Applied Statistics*, v. 16, n. 2, pp. 131-164.

Efron, B. F. & Hinkley, D. V. (1978). Assessing the accuracy of the ML estimator : observed versus expected Fisher information, *Biometrika*, vol. 65, pp. 457-487.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Trans. On Pattern Analysis and Machine Intelligence,* vol.6, n. 6, pp. 721-742.

Hammersley, J. M. & Clifford, P. (1971). Markov field on finite graphs and lattices. *Unpublished.*

Jensen, J. L. & Künsh, H. R. (1994). On asymptotic normality of pseudo likelihood estimates for pairwise interaction processes. *Annals of the Institute of Statistical Mathematics,* vol.46, n. 3, pp. 475-486.

Lehmann, E. L. (1983). *Theory of Point Estimation,* Wiley, New York.

Levada, A. L. M. ; Mascarenhas, N. D. A. & Tannús, A. (2008a). A novel pseudo-likelihood equation for Potts MRF model parameter estimation in image analysis, *Proceedings of the 15th International Symposium on Image Processing (ICIP)*, pp. 1828-1831, October 2008, IEEE, San Diego.

Levada, A. L. M. ; Mascarenhas, N. D. A. & Tannús, A. (2008b). Pseudo-likelihood equations for Potts MRF model parameter estimation on higher-order neighbourhood systems, *IEEE Trans. On Geoscience and Remote Sensing Letters*, vol. 5, pp. 522-526.

Levada, A. L. M. ; Mascarenhas, N. D. A. & Tannús, A. (2008c). On the asymptotic variances of Gaussian Markov Random Field model hyperparameters in stochastic image modeling, *Proceedings of the 19th International Symposium on Pattern Recognition (ICPR)*, pp. 1-4, December 2008, IEEE, Tampa.

Levada, A. L. M. ; Mascarenhas, N. D. A. ; Tannús, A. & Salvadeo, D. H. P. (2008d). Spatially non-homogeneous Potts model parameter estimation on higher-order neighbourhood systems by maximum pseudo-likelihood, *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (ACM SAC)*, pp. 1733-1737, March 2008, ACM, Fortaleza, Brazil.

Liang, G. & Yu, B. (2003). Maximum pseudo likelihood estimation in network tomography, *IEEE Trans. On Signal Processing*, vol. 51, n. 8, pp. 2043-2053.

Marroquin, J. L.; Mitter, S. K. & Poggio, T. A. (1987). A probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Society,* vol.82, n. 397, pp. 76-89.

Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A. & Teller, E. (1953). Equation of state calculations by fast computer machines. *Journal of Physical Chemistry,* vol. 21, n. 6, pp. 1087-1092.

Swendsen, R. & Wang, J. (1987). Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters,* vol.58, pp. 86-88.

Winkler, G. (2006). *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction,* 2nd Edition, *Springer-Verlag*, New York.

Wolff, U. (1989). Collective Monte Carlo updating for spin systems. *Physical Review Letters,* vol.62, pp. 361-364.

Won, C. S. & Gray, R. M. (2004). *Stochastic Image Processing,* Kluwer Academic/Plenum Publishers, New York.

Yamazaki, T. & Gingras, D. (1996). A contextual classification system for remote sensing using a multivariate Gaussian MRF model, *Proceedings of the 9th International Symposium on Circuits and Systems (ISCAS)*, pp. 648-651, May 1996, IEEE, Atlanta.

Yu, S. & Berthod, M. (1995). A game strategy approach for image labeling. *Computer Vision and Image Understanding,* vol.61, n. 1, pp. 32-37.

BIVSEE – A Biologically Inspired Vision System for Enclosed Environments

Fernando López-García[1], Xosé Ramón Fdez-Vidal[2],
Xosé Manuel Pardo[2] and Raquel Dosil[2]
[1] *Universidad Politécnica de Valencia*
[2] *Universidade de Santiago de Compostela*
Spain

1. Introduction

Many people working in Computer Vision and related areas, have, at some stage, thought about the possibility of developing a machine able to imitate the Human Visual System, i.e. to develop a computational model of human vision. However, to date, the goal of creating a general purpose vision system close, or even slightly close, to the robust and resilient capabilities of the human visual system remains unreachable (Vernon, 2006).

In the history of Computer Vision many works related to this issue have been released. A survey about this subject is out of the scope of the present work, an introduction can be found in (Vernon, 2006). Nevertheless, we would like to draw our attention to two remarkable books dealing with human and computer vision appeared in mid 1980s. The authors were M. D. Levine and S. Watanabe (Levine, 1985; Watanabe, 1985). In addition to appearing the same year, they also turned out to be complementary to each other. In his book, Levine defined low-level and high-level tasks for computer and human vision. Related to computer vision, he defined the levels of analysis presented in Table 1.

Level	Description
M + 3	3D Scene interpretation
M + 2	3D Scene Description
M + 1	2D Image description
6 to M	Higher level aggregation and model matching
5	Discovery of structural relationships
4	Feature classification
3	Image segmentation and feature detection
2	Preprocessing and restoration
1	Sensor representation
0	Scene

Table 1. Levels of Analysis for Computer Vision (by Levine).

In his book, Levine dealt only with the levels from 0th to 3th, which correspond to the transformation from the raw sensed scene to a coded version of it. Watanabe, working independently, devoted his book to the 4th level, which corresponds to the task of pattern recognition. In their books, both authors presented and compared the human process of vision with the state-of-the-art of mathematical and computational developments at that time. However, they did not provide a computational model of human vision.

The higher levels of analysis given in Table 1 would correspond to what is called perceptual organization. A review on this subject can be found in (Sarkar & Boyer, 1993).

Recently, efforts to compile and group scattered research on this subject have led to the definition of a new Computer Vision field; the Cognitive Vision Systems. Although this new area is not yet well-defined (Christensen and Nagel, 2006; Vernon, 2008), Cognitive Vision Systems are defined by highlighting generic functionalities and non-functional attributes (Vernon, 2006). Thus, it is said that "a cognitive vision system can achieve four levels of generic functionality: *Detection* of an object or event in the visual field; *Localization* of the position and extent of a detected entity; *Recognition* of a localized entity by a labeling process; and *Understanding* or comprehending the role, context, and purpose of a recognized entity". It is easy to find that these functionalities match the computer vision levels of analysis provided by Levine in Table 1. But, in addition, the definition of Cognitive Vision Systems is extended underlining the fact that "they can engage in purposive goal-directed behavior, adapting to unforeseen changes of the visual environment, and anticipating the occurrence of objects or events. These capabilities (non-functional attributes) are achieved through; a faculty for learning semantic knowledge, and for the development of perceptual strategies and behaviors; the retention of knowledge about the environment, the cognitive system itself, and the relationship between the system and its environment; and the deliberation about objects and events in the environment, including cognitive system itself... The three non-functional attributes of *purposive behavior*, *adaptability*, and *anticipation*, taken together, allow a cognitive vision system to achieve certain goals, even in circumstances not expected when the system was designed".

Under the term of Cognitive Vision many works have been developed recently (Christensen and Nagel, 2006; Vernon, 2008). Nevertheless, these works are devoted to specific aspects related to the Cognitive Vision processes, such as object recognition, adaptive knowledge, predictive element of cognition, and others, rather to define complete systems, which was the natural goal in early works, to develop a complete computer vision model imitating the human visual system. Nevertheless, this goal was, and continues to be, an unreachable target since the way that human vision actually works is mostly unknown, despite the advances on this subject achieved in the neurophysiology and psychology sciences (Levine, 1985). At the moment, we can approach the human vision mainly from its functionality, and also try to provide it with higher level capabilities by adding some kind of artificial intelligence (non-functional attributes of Cognitive Vision Systems). Current knowledge about how the human vision system works is limited to the first levels of analysis (0th to 4th levels of analysis given in Table 1). Thus, it is only in these levels where we can propose approaches that strictly can be called *biologically inspired*, such as Visual Attention (García-Díaz et al., 2008) for scene recognition and Visual Context Models (Ehtiati & Clark, 2004; Oliva & Torralba, 2007; Perko & Leonardis, 2008) to improve recognition performance. However, we think there are approaches inspired in the way humans carry out their cognitive vision at higher levels that can be considered *biologically inspired*. We refer to the way that humans organize visual information, object building from parts and localization in a given environment, and use it to achieve certain goals.

In this context, we carry out a theoretical exercise and present here the outlines of one of the first complete designs[1] for an artificial vision system which can be included within the definition of Cognitive Vision Systems; the BIVSEE system. This is biologically inspired (in the sense exposed in the previous paragraph) and also it is intended to imitate the early functionalities of the human visual system in enclosed environments[2]. The goal is to define a system able to perform basic recognition of objects, determine the spatial interrelations among the objects, and interact with the environment with a purposive goal, e.g. to survey a specific area, track a moving object, etc. We present a simple but valuable design which is a first approach on the path to develop wide-purpose humanlike vision systems, and it is intended to serve as the basis for future more complex developments. The system is defined through a cyclic and modular architecture that includes the following levels of analysis; preprocessing, scene location, tree description, analytic projection, and decision making.

We also present experimental work related to the scene recognition task, which is used in the scene location module to pre-localize sub-areas in the enclosed environment (the application scenario) and speed-up the computation of the tree description. For this purpose we use the saliency maps of a biologically inspired Visual Attention approach in combination with image features, SIFT (Lowe, 2004) and SURF (Bay et al., 2008) and find out the superiority of SIFT approach over SURF for the studied task. This is an interesting result as it is one of the few comparison works of these competitive methods in the area of image features[3] (Bauer et al., 2007; Mikolajczyk & Schmid, 2005).

2. BIVSEE Architecture

The system architecture is simple. It is based on a set of cyclically interconnected modules. Each module deals with a specific type of input data that is elaborated to provide appropriate data to the next module. The architecture (see Figure 1) starts with the camera, placed in a fixed location or mounted on a mobile device (robotic applications) to inspect the environment. The camera provides the first module with a stream of raw data composed of scene frames at a given rate (typically 5 fps correspond to robot navigation). The *Preprocessing* module improves the image data by applying image preprocessing techniques, e.g. noise removal. This module feeds the *Scene Location* module which localizes the current scene into one of the several sub-areas that set the complete environment. This will help, in next module, to reduce the search area within the reference tree. In the *Tree Description* module, segmentation techniques are used to divide the scene into homogeneous regions (regions with a homogeneous visual feature which can be a colour texture feature) which are used to build a tree data structure that describes the scene by means of the recognized objects present in it, and also compiles geometric and localization data for these objects. To carry out this recognition task it is necessary to use prior information which in this case is a reference tree which describes the complete scenario (the enclosed environment). This reference tree is the innate knowledge of the system and it must be previously created in a

[1] As far as we know in literature.

[2] Areas with a limited number of patterns and controlled illumination.

[3] Also called interest point detectors.

supervised manner with the aid of human operators. The *Tree Description* module provides the next module, the *Analytic Projection* module, with a tree structure that includes the recognized objects and geometric and localization data of them in the current frame. Here the tree description is analyzed and projected into a semantic description of the scene. This semantic description includes the objects present in the frame, their location, geometrical properties and spatial interrelations. Finally, the semantic description is used in the *Decision Making* module to elaborate and decide adequate actions coherent with the expected purposive behaviour of the system. For example, the system can be used to monitor the environment or track moving objects. The semantic description in the *Analytic Projection* module is performed using Semantic Networks, specifically the ERNEST formalism (Niemann et al., 1990) which contains extensions oriented to pattern recognition. *Decision Making* is carried out through the use of Decision Networks also called Influence Diagrams (Russell & Norvig, 2003). All the architecture cycle is intended to work at the frame rate provided by the camera.

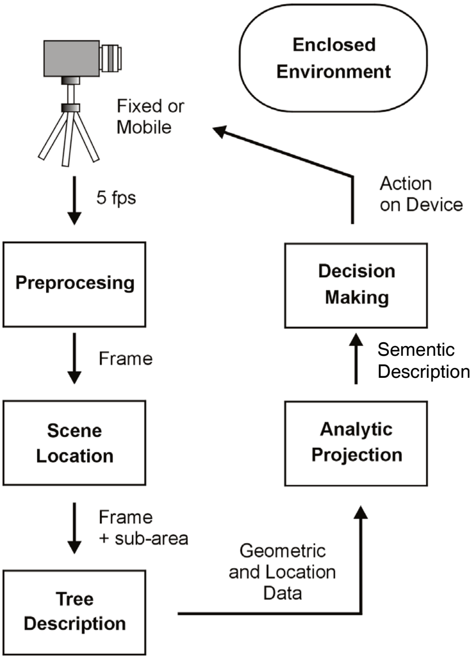


Fig. 1. BIVSEE Architecture.

2.1 Preprocessing

This module is used to enhance the original image data, which is commonly affected by noise and illumination variance (Petrou & Bosdogianni, 1999). The most usual kind of noise is Gaussian, which can be removed with an average filter. A median filter should be used instead in case of impulsive noise. With regards to the illumination variance, although enclosed environments are usually provided with controlled illumination, some kind of variability can still be present mainly due to the flicker effects introduced by lamps which operate directly from main frequency AC, e.g. fluorescents. In this case, it is better to manage it within the objects extraction stage, as suggested in the literature (Zhou et al., 2006). In the literature, object extraction with variable illumination is dealt with using different approaches, such as colour constancy models, invariant features, or learning from many samples taken under different illumination conditions. We propose to use this last approach as we will see in Section 2.3.

2.2 Scene Location

This module receives the enhanced image data of the current frame and performs a pre-localization of it into one of the several sub-areas that form the complete scenario (the complete enclosed environment). In a general case, a complex scenario can be composed by several sub-areas, e.g. an application scenario could be a University facility composed by several halls and rooms. This is the case of the data used in Section 3. In this case, the complete scenario is divided into seven sub-areas; hall-1, hall-2, hall-3, room-1, room-2, room-3 and room-4. If we pre-localize the current scene into a specific sub-area (through a scene recognition application) we can save computing time by reducing the search area within the reference tree, which is the prior information used to build the tree description of the current frame or scene.

The scene recognition task is developed in Section 3, where also experimental work is presented. The scene recognition is carried out using a combination of saliency maps coming from a novel approach of Visual Attention and the SIFT and SURF image features.

2.3 Tree Description

This module receives the enhanced image data of current frame plus its localization into one of the several sub-areas that could form the complete scenario. These data is transformed into a tree data structure which describes the captured scene. This module implements the first levels in Table 1 from 0th to M+1 providing a 2D scene description.

In this module, the image is first segmented into its different homogeneous components (regions with a homogeneous visual feature) using a state-of-the-art segmentation method. One generic and fast method that provides very good results is the Efficient Graph-Based Image Segmentation method, by (Felzenszwalb & Huttenlocher, 2004).

Once we have segmented the image, it is divided into several regions. At this point we introduce the biologically inspired approach mentioned in Section 1 for higher levels of human vision. We, the humans, manage the visual data of a scene dividing it into objects and, at the same time, we compose these objects by grouping the several parts that form them (see Figure 2). In our approach we consider that each part of an object (an image region with a homogenous visual feature) has been correctly segmented by the segmentation algorithm.

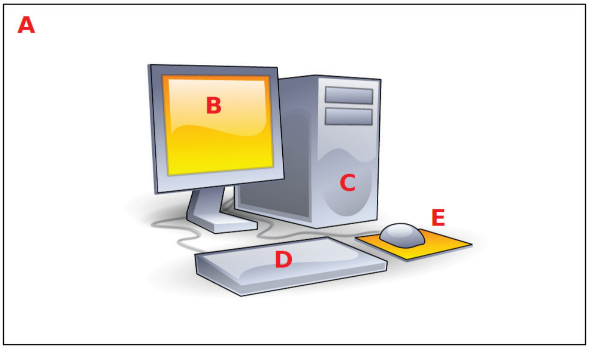


Fig. 2. An artificial scene showing a computer and the different parts of it.

Taking into account the decomposition of objects into their homogeneous parts, the enclosed environment, can be described through a tree structure that hierarchically compiles data about the segmented regions forming the different objects. Thus, the scene presented in Figure 2 would correspond to a tree data structure as it is shown in Figure 3.

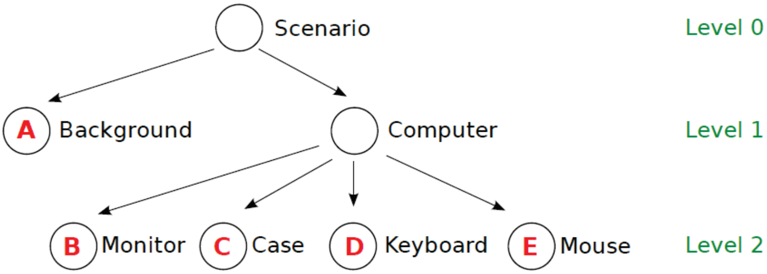


Fig. 3. Tree description of the scene presented in Figure 1.

As we can see in Figure 3, the scene is decomposed from a primary node (Level 0) to several sub-nodes that refer to the objects present in Level 1. Also these objects are formed by the union of several leave-nodes (or final nodes) that correspond to the segmented areas achieved by the segmentation algorithm. Objects can be present in deeper levels of the tree, e.g. the mouse object could be decomposed into mouse and mousepad. These two last regions would be then leave-nodes in Level 3.

If we go to a more general case, a complex scenario composed by several sub-areas, we could use the first level under the primary node to divide the complete scenario into different sub-areas. For example, a scenario could be a University facility composed by several halls and rooms. This is the case of the data used in Section 3 for experimental work. In this case, the complete scenario is divided into seven sub-areas; hall-1, hall-2, hall-3, room-1, room-2, room-3 and room-4. Then, from each one of these sub-areas, the different objects present in them would hold in the way shown in Figure 3. Going further, it would be possible to divide each sub-area into sub-sub-areas using another level at the beginning of the tree structure.

RECOGNITION

If we want the system to work with a specific scenario we should first create a prior information of this scenario. This prior information will be contained in a reference tree description of the scenario built in a supervised manner with the aid of human operators. This will give the innate knowledge to the system. The operators will study the segmented regions from the segmentation algorithm and from them compose the different objects building the reference tree data structure. In order to be able to perform object recognition from this prior information, we have to compile discriminative information into the leave-nodes, classical pattern recognition data (e.g. colour, texture, shape, etc) for each segmented region.

Once we have the reference tree that contains the prior information, the application will work using one or several cameras acquiring scenes in the enclosed environment. The idea is to provide the system with recognition information about what is being "seen" by the cameras. To do this, a segmentation method is applied and it divides the image into different homogeneous regions. After this, pattern recognition data compatible with that compiled for the leave-nodes in the reference tree is computed. This data is used to perform the recognition of these regions using one or more well-known pattern recognition methods (Duda et al., 2002).

Once the segmented regions in the scene are recognized and classified using the pattern recognition data of the reference tree structure, a new tree corresponding to the current scene is built to describe the objects present in it. That way, the result of recognition is also a tree structure that describes the captured scene in the enclosed environment.

In order to help to build the new tree for the current frame, we can define in the reference tree, for each object, what the object "is" using the regions that belong to it and the regions that do not belong to it. For example, the computer object in Figure 2 can be defined as:

$$\text{Computer} = B+C+D+E \quad \text{(using the regions that belong to it)} \tag{1}$$

$$\text{Computer} = S-A \quad \text{(using the regions that do not belong to it)} \tag{2}$$

Being S the complete scene.

When we build the different objects in the current scene, we shall use different sets of regions that should maximize the first formula and minimize the second one. Thus, we can use a juxtaposition of formulas to find the correct set of segmented regions that correspond to a specific object in the reference tree (the prior information of the environment).

Apart of compiling pattern recognition data in the leave-nodes, other important kind of information stored in the tree structure is geometric and localization information. Localization data, 2D position in a first approach or 3D in advanced developments, can be stored in leave-nodes and also in object-nodes. Geometric data can be also introduced in the nodes, e.g. the Fourier signatures (Loncaric, 1998). This will provide the next modules of the system with helpful high level information of the scene. Localization and geometric information in a specific node of the tree is always referred to the nodes that hold from it.

If we want to introduce into the reference tree some kind of invariance to changes in illumination, scale, rotation and viewpoint, we can introduce in an object-node several representations of it, different sets of segmented regions (leave-nodes) corresponding to different illumination conditions, scales, viewpoints and rotations.

New objects can be introduced a posteriori in the prior information (the reference tree) with the aid of human operators.

Computing time can be saved if the complete scenario is sub-divided into different sub-areas in the reference tree. In this case, the search area within the reference tree used to carry out the recognition of leave-nodes and object building will be significatively reduced.

2.4 Analytic Projection

This module receives the tree data structure which describes the scene in the current frame. It is analyzed and then projected into a semantic description of the scene which includes the objects present in the frame, their location, geometry and spatial interrelations. The semantic description is performed using Semantic Networks. Specifically, we propose to use ERNEST semantic networks.

Semantic networks were introduced in late sixties to model the semantics of English words (Quillian, 1969). These networks corresponded to directed, labelled graphs, where nodes contained information about objects, events or facts. Lately, semantic networks were improved to achieve problem-independent control algorithms giving rise to several semantic networks formalisms such as KRIPTON, NIKL, SB-ONE and ERNEST. We propose the ERNEST formalism (Niemann et al., 1990) because it contains useful extensions oriented to pattern recognition.

2.5 Decision Making

Finally, to implement the *Decision Making* module, which has to evaluate and decide adequate actions coherent with the expected purposive behaviour of the system, we turn to the areas of Decision Analysis (Machine Learning) and Artificial Intelligence. Among the variety of methods developed in these areas, we propose the use of decision networks (Russell & Norvig, 2003), also called influence diagrams. Decision networks are an extension of the Bayesian networks and they can be used to solve probabilistic inference problems (Bayesian networks) and also decision making problems (by using a maximum expected utility criterion). Decision networks are now widely used and are becoming an alternative to decision trees which typically suffer from exponential growth in the number of branches when new variables are added to the model. Although semantic networks can include control algorithms, that is, they can provide a semantic description and also implement the purposive behaviour of the system, we propose to use decision networks instead because they are more flexible and allow more complex decision schemes to be implemented, which is desirable if we want to extend system capabilities.

3. Scene Recognition

Scene recognition is used within the BIVSEE system to recognize local areas (sub-areas) in the global scenario, reducing the searching area in the reference tree and thus accelerating the recognition process.

Here we present the experimental work that we have carried out related to this issue. We study how the use of a model of bottom-up saliency (visual attention), based on local energy and colour, can significantly accelerate scene recognition and, at the same time, preserve the recognition performance. We do this in the context of a mobile robot-like application where scene recognition is performed through the use of image features (SURF and SIFT alternatives are compared) to characterize the different scenarios, and the Nearest Neighbour rule to carry out the classification. Experimental work shows that SIFT features are the best alternative achieving important reductions in the size of the database of prototypes without significant losses in recognition performance and thus accelerating the scene recognition task.

3.1 Introduction

Visual attention is related with the process by which the human visual system is able to select [from a scene] regions of interest that contain salient information, reducing the amount of information to be processed and therefore the complexity of viewing (Treisman & Gelade, 1980; Koch & Ullman, 1985). In the last decade, several computational models biologically inspired have been released to implement visual attention in image and video processing (García-Díaz et al., 2008; Itti and Koch, 2000; Milanese et al., 1995). Visual attention has also been used to improve object recognition and scene analysis (Bonaiuto & Itti, 2005; Walther et al., 2005).

We study the utility of using a recently presented novel model of bottom-up saliency (García-Díaz et al., 2008) to improve a scene recognition application by reducing the amount of prototypes needed to carry out the classification. The application is based on mobile robot-like video sequences taken in an indoor university area formed by several rooms and halls. The aim is to recognize the different scenarios in order to provide a mobile robot system with general location data.

The visual attention approach that we use (García-Díaz et al., 2008) is a novel model for the implementation of the Koch & Ullman (Koch & Ullman, 1985) architecture of bottom-up saliency for static images. Two features are used to measure the saliency: local energy and colour. From them, we extract local maxima of variability through the decorrelation of responses and the measurement of statistical distance, followed by a non-linear local maxima excitation process to deliver a final map of saliency. With this method we obtain saliency areas in images that point out to relevant regions from the point of view of visual attention. In addition, saliency is not measured in a binary manner (salient or not) but scaled from 0 to 1, which permits to determine different levels of relevance by simply thresholding the saliency map.

Scene recognition is performed using SIFT (Lowe, 2004) and SURF (Bay et al., 2008) image features (two different approaches which are compared) and the Nearest Neighbour rule. SIFT features are distinctive image features that are invariant to image scale and rotation, and partially invariant to change in illumination and 3D viewpoint. They are fast [to compute] and robust to disruptions due to occlusion, clutter or noise. SIFT features have proven to be useful in many object recognition applications and currently they are considered the state-of-the-art for general purpose real-world object learning and recognition, together with SURF features. SURF is a robust image descriptor, first presented by Herbert Bay et al. in 2006 (Bay et al., 2006), that can be used in computer vision tasks like object recognition or 3D reconstruction. It is partly inspired by the SIFT descriptor. The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT.

Results of experimental work have shown that the use of saliency maps in combination with SIFT features permit to drastically reduce the size of the database of prototypes, used in the

1-NN recognition process, achieving very good recognition performance. Thus, the computing costs of classification are reduced proportionally to the database size and the scene recognition application is accelerated. The database was reduced to 10.6% of its original size achieving a recognition performance of 91.9%, only a drop of 3.4% from the original performance 95.3% achieved without using saliency maps.

3.2 Visual Attention

In this model, following the standard model of V1, we use a decomposition of the image by means of a Gabor-like bank of filters. We employ two feature dimensions: colour and local energy. By decorrelating responses and extracting local maxima of variability we obtain a unique, and efficient, measure of saliency.

Local Energy and Colour Maps. Local energy is extracted through the convolution of the intensity, the average of the three channels r, g and b, with a bank of log Gabor filters (Field, 1987), which presents a number of advantages against Gabor filters, have complex valued responses. Hence, they provide in each scale and orientation a pair of filters in phase quadrature (Kovesi, 1996), an even filter and its Hilbert transform, an odd filter; allowing us to extract local energy as the modulus (Morrone and Burr, 1988) of the response to this filter vector. A more detailed description of our approach to local energy extraction can be found in (García-Díaz et al., 2008). With regards to Colour Maps, we extract first two colour opponent components: r/g and b/y. From them we obtain a multi-scale centre-surround representation obtained from the responses of the two double opponent components to high-pass logarithmic Gaussian filters. By subtracting large scales from small scales (1-3, 1-4, 2-4, 2-5), we obtain a pyramid of four centre-surround maps for each colour component, r/g and b/y.

Measurement of Variability. Difference and richness of structural content have been proven as driving attention in psychophysical experiments (Zetzsche, 2005). Observations from neurobiology show decorrelation of neural responses, as well as an increased population sparseness in comparison to what can be expected from a standard Gabor-like representation (Vinje & Gallant, 2000)(Weliky et al., 2003). Hence, we use decorrelation of the responses to further measure the statistical distance of local structure from the average structure. To decorrelate the multi-scale information of each sub-feature (orientations and colour components) we perform a PCA on the corresponding sets of scales. From the decorrelated responses, we extract the statistical distance at each point as the T2 of Hotelling.

Excitation of Local Maxima. Once the structural distance within each sub-feature has been measured, we force a spatial competition exciting local maxima in a non-linear approach already described in (García-Díaz et al., 2008). Next, we fuse the resultant sub-feature maps simply gathering the surviving maxima, with a max() operation, in a local energy saliency map, and in a colour saliency map. Finally, we repeat the process, with these two maps to extract a final measure of salience. All the process is illustrated in Figure 4.

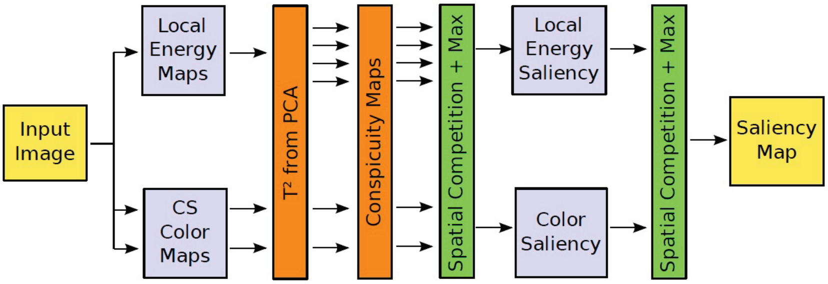


Fig. 4. Saliency computation using the bottom-up model of visual attention.

3.3 Scene Recognition Application

Scene recognition or classification is related with the recognition of general scenarios rather than local objects. This approach is useful in many applications such as mobile robot navigation, image retrieval, extraction of contextual information for object recognition, and even to provide access to tourist information using camera phones, apart of its use within the BIVSEE system to pre-localize sub-areas in the global scenario. In our case, we are interested in recognize a set of different areas which are part of the facilities of the Electronics and Computer Science Department of the University of Santiago de Compostela. These facilities are formed by four class rooms and three halls that connect them. The final aim is to provide general location data useful for the navigation of a mobile robot system.

Scene recognition is commonly performed using generic image features that try to collect enough information to be able to distinguish the different scenarios. In our case, to achieve this aim, we used image features comparing the SIFT and SURF alternatives.

With regards to SIFT features, we used Lowe's algorithm (Lowe, 2004) which is applied to each image [or frame] and works as follows. To identify candidate keypoint locations, scale space extrema are found in a difference-of-Gaussian (DoG) function convolved with the image. The extremas are found by comparing each point with its neighbours in the current image and adjacent scales. Points are selected as candidate keypoint locations if they are the maximum or minimum value in their neighbourhood. Then image gradients and orientations, at each pixel of the Gaussian convolved image at each scale, are computed. For each key location an orientation, determined by the peak of a histogram of previously computed neighbourhood orientations, is assigned. Once the orientation, scale, and location of the keypoints have been computed, invariance to these values is achieved by computing the keypoint local feature descriptors relative to them. Local feature descriptors are 128-dimensional vectors obtained from the pre-computed image orientations and gradients around the keypoints. With regards to SURF features (Bay et al., 2008), they are based on sums of 2D Haar wavelet responses and make an efficient use of integral images. As basic image descriptors they use a Haar wavelet approximation of the determinant of Hessian blob detector. There are two SURF versions: the standard version which uses a descriptor vector of 64 components (SURF-64), and the extended version (SUFR-128) which uses 128 components. SURF are robust image features partly inspired by the SIFT features, and the standard version of SURF is several times faster than SIFT.

To compute the SIFT features we used Lowe's original implementation[4]. We also used the original implementation of SURF features[5] by Bay et al (see Figure 5).

To carry out the classification task we used the 1-NN rule, which is a simple classification approach but fast [to compute] and robust. For this approach, we need to previously build a database of prototypes that will collect the recognition knowledge of the classifier. These prototypes are in fact a set of labelled SIFT/SURF keypoints obtained from training frames. The class (or label) of the keypoints computed for a specific training frame will be that previously assigned to this frame in an off-line supervised labelling process. This database is then incorporated into the 1-NN classifier, which uses the Euclidean distance to select the closest prototype to the test SIFT/SURF keypoint being classified. The class of the test keypoint will be assigned to the class of the closest prototype in the database, and finally, the class of the test frame will be that of the majority of its test keypoints.

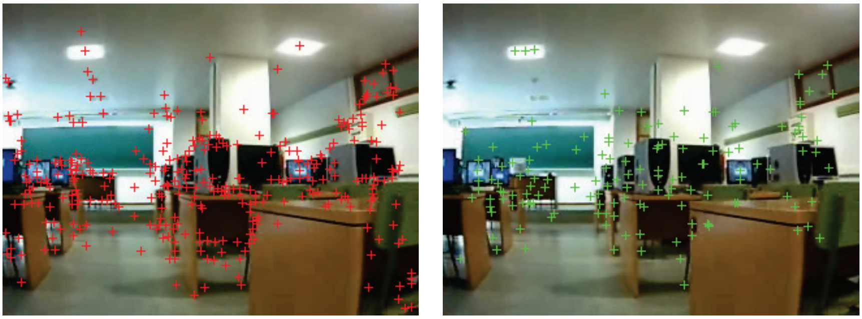


Fig. 5. SIFT (left) and SURF (right) keypoints computed on the same frame.

3.4 Experiments and Results

Experimental work consisted in a set of experiments carried out using four video sequences taken in a robot-navigation manner. These video sequences were grabbed in an university area covering several rooms and halls. Sequences were taken at 5 fps collecting a total number of 2,174 frames (7:15 minutes) for the first sequence, 1,986 frames for the second (6:37 minutes), 1,816 frames for the third (6:03 minutes) and 1,753 frames for the fourth (5:50 minutes). The first and third sequences were taken following and specific order of halls and rooms: hall-1, room-1, hall-1, room-2, hall-1, room-3, hall-1, hall-2, hall-3, room-4, hall-3, hall-2, hall-1. The second and fourth sequences were grabbed following the opposite order. This was done to collect all possible viewpoints of the robot-navigation through these University facilities. In all the experiments, we used the first and second sequences for training and the third and fourth ones for testing.

In the first experiment we computed the SIFT keypoints for all the frames of the training video sequences. Then, we labelled these keypoints with the corresponding frame class. The labels we used were: room-1, room-2, room-3, room-4, hall-1, hall-2 and hall-3. The whole set of labelled keypoints formed itself the database of prototypes to be used by the 1-NN classifier to carry out the classification on the testing sequences. For each frame of the testing

[4] http://www.cs.ubc.ca/~lowe/keypoints/

[5] http://www.vision.ee.ethz.ch/~surf/index.html

sequences their corresponding SIFT keypoints were computed and classified. The final frame class was set to the majority class within its keypoints. This experiment achieved very good recognition performance, 95.25% of correct frame classification, although, an important drawback was the high computational costs of classification, despite the fact that the 1-NN is a simple classifier. This was due to the very large size of the knowledge database of prototypes formed by 1,170,215 samples.

In the next experiment, we followed the previous steps but using SURF features instead of SIFT features. In this case, the recognition results were very bad achieving only 35.09% of recognition performance with SURF-128 (version that uses 128 descriptors per keypoint), and 25.05% of recognition performance using SURF-64 (faster version which uses only 64 descriptors). In both cases the size of the database of prototypes was 415,845.

Although there are well known techniques for NN classifiers to optimize the database of prototypes (e.g. feature selection, feature extraction, condensing, editing) and also for the acceleration of the classification computation (e.g. kd-trees), at this point we are interested in the utility of using the saliency maps derived from the Visual Attention approach shown in Section 3.2. The idea is to achieve a significant reduction of the original database by selecting in each training frame only those keypoints that are included within the saliency map computed for this frame. Also, in recognition, only those keypoints lying within the saliency maps, computed for the testing frames, will be considered for classification. Once the database is reduced that way, optimizing techniques could be used to achieve even further improvements.

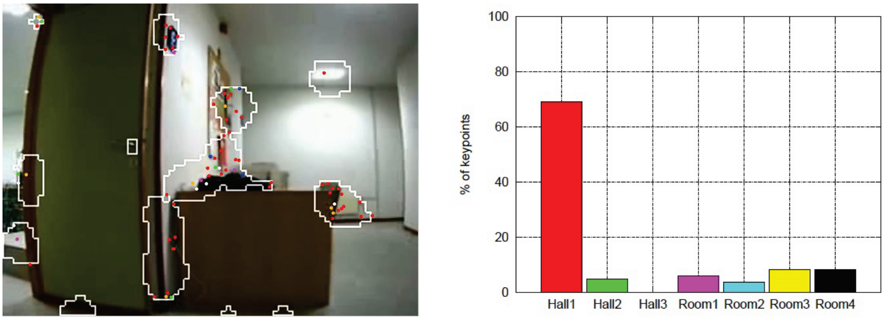


Fig. 6. A frame processed using its saliency map at threshold 0.250.

In the next experiment we carried out the idea exposed in the previous paragraph. Nevertheless, we wanted to explore more in-depth the possibilities of saliency maps. As it was commented, saliency measures are set in a range between 0 and 1, thus, we can choose different levels of saliency by simply using thresholds. We will be the least restrictive if we choose a saliency > 0.000, and more restrictive if we choose higher levels (e.g. 0.125, 0.250, etc). We planed to use 8 different saliency levels or thresholds: 0.000, 0.125, 0.250, 0.375, 0.500, 0.625, 0.750 and 0.875. For each of these levels we carried out the recognition experiment (see Figure 6), and two were the results obtained: the percentage of recognition performance for each saliency level, and the size reduction of the original database. Results using SIFT and SURF features are shown in Tables 2 and 3 and Figures 7 and 8.

	Recognition %	Database Size	Database Size %
Original	95.3	1,170,215	100.0
Saliency > 0.000	94.9	870,455	74.4
Saliency > 0.125	94.2	340,517	29.2
Saliency > 0.250	93.2	200,097	17.1
Saliency > 0.375	91.9	123,463	10.6
Saliency > 0.500	89.7	76,543	6.6
Saliency > 0.650	84.6	45,982	4.9
Saliency > 0.750	64.8	24,525	2.1
Saliency > 0.875	29.3	9,814	0.8

Table 2. Results achieved using original frames and saliency maps with SIFT features.

	Recognition %	Database Size	Database Size %
Original	35.1	415,845	100.0
Saliency > 0.000	33.0	334,159	80.4
Saliency > 0.125	72.2	141,524	34.0
Saliency > 0.250	73.9	84,599	20.3
Saliency > 0.375	69.2	52,682	12.7
Saliency > 0.500	59.3	32,715	7.9
Saliency > 0.650	40.2	19,794	4.8
Saliency > 0.750	41.4	10,583	2.6
Saliency > 0.875	20.7	4,373	1.1

Table 3. Results achieved using original frames and saliency maps with SURF-128 features.

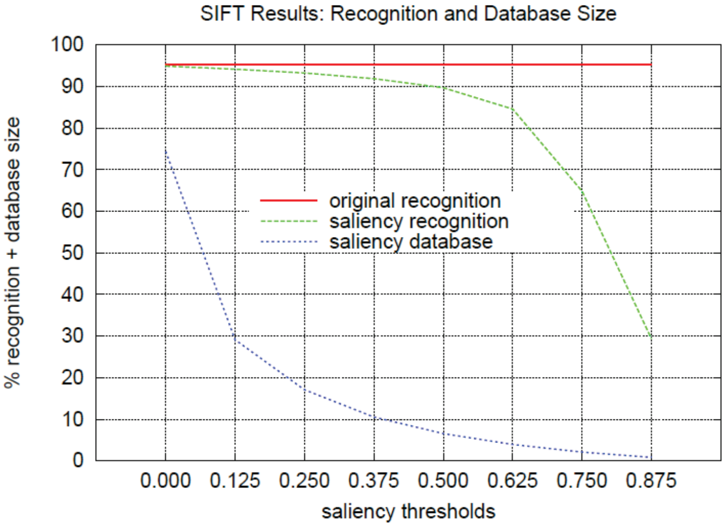


Fig. 7. Graphical results of recognition and database size using SIFT features.

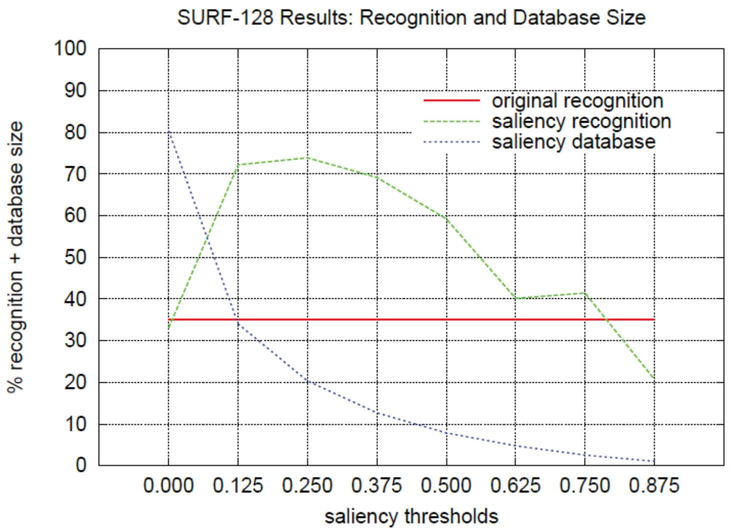


Fig. 8. Graphical results of recognition and database size using SURF-128 features.

Experimental results clearly show that although SURF features have the advantage that they collect significantly less keypoints than SIFT (approximately half of them) their performance results are not adequate for the application of scene recognition that we are studying. Nevertheless, they have proven to be adequate, and faster than SIFT features, in other applications (Bay et al., 2008).

Another interesting result with regards to the SIFT/SURF comparison is the following. In Figure 8, we can see that SURF-128 performance recognition improves as we use more restrictive saliency maps, with a 73.9% of maximum performance at 0.250 saliency threshold, then it drops from threshold 0.375 on. This result shows that SURF descriptors loose distinctiveness as we use more keypoints in each frame (less restrictive saliency maps). This fact does not occur when we use SIFT features, thus, SIFT descriptors show more distinctiveness than SURF descriptors when using very large keypoint databases.

Finally, we have to point out that the best results are achieved using SIFT features, and also that saliency maps can reduce the amount of prototypes in the knowledge database and testing images up to one order of magnitude, while the recognition performance is held, saliency threshold 0.375 at Figure 7 and Table 2. In this case, the recognition performance drops to 91.9% (only 3.4 points from the maximum 95.3%) while the database size drastically falls from 1,170,215 to 123,463 prototypes.

4. Summary

In this chapter we have presented the outlines of a complete design for a Cognitive Vision System which includes in its development methods of biological inspiration, that is, methods inspired in the way humans perform our vision task. The BIVSEE system, is a system able to perform basic recognition of objects, determine the spatial interrelations among the objects, and interact with the environment with a purposive goal. We have presented a valuable simple design which is intended to serve as the basis for future more complex developments.

The system is defined through an architecture composed of fifth cyclically interconnected modules; *Preprocessing*, *Scene Location*, *Tree Description*, *Analytic Projection* and *Decision Making*. Each of these modules deals with a specific type of input data which is elaborated to provide the next module with adequate data. The *Preprocessing* module enhances the raw image (frame) acquired by the camera sensor. Then, the enhanced frame is passed to the *Scene Location* module which pre-localizes the scene into one of the several sub-areas of the complete scenario or environment. Then, the *Tree Description* module using a reference tree of the complete enclosed environment generates a tree data structure that describes the scene; the objects present in the scene and also geometric and localization data on these objects. This data is passed to the *Analytic Projection* module which elaborates this data to produce a semantic description of the scene. We propose to use the ENERST formalism of semantic networks to carry out this task. ENERST networks provide useful extension for pattern recognition, which is coherent with the kind of information that the system has to manage. This semantic description includes the objects present in the scene, their geometry, location and spatial interrelations. Finally, the semantic description is the input data used in the *Decision Making* module to decide the adequate actions coherent with the purposive behaviour that we want to implement into the system. For this module we propose to use Decision Networks. They come from the areas of Decision Analysis and Artificial Intelligence and allow implementing complex decision schemes.

All the system cycle is intended to work at the frame ratio provided by the camera, which usually is of 5 frames per second in robot-navigation applications.

In the second part of the chapter we present an application and experimental work related to the scene recognition task, which is used in the *Scene Location* module of the BIVSEE system to recognize specific sub-areas of the enclosed environment and thus reduce the search area in the reference tree. This will be useful to accelerate the computation of the tree description of the current frame.

The scene recognition is performed using a biologically inspired Visual Attention approach in combination with image features or interest point detectors. We compare the SIFT and SURF approaches to extract image features. These two competitive approaches belong to the current state-of-the-art in this area and their comparison is a current issue in literature. Experimental results show that although SURF features imply less interest points, the best performance corresponds to SIFT features. The SIFT method achieves a 95.3% of correct scene recognition in the best case, while SURF method only reach to 73.9%. Another important result is achieved when we use the saliency maps from the Visual Attention approach. Using these saliency maps we can drastically reduce the database of prototypes used in the scene recognition application (up to one order of magnitude) without a significant loss in recognition performance, and thus it can accelerate the scene recognition process. In addition, experiments show that SURF features are less distinctive than SIFT features when the number of prototypes in the database grows.

5. Acknowledgements

Work supported by the Ministry of Education and Science of the Spanish Government (AVISTA-TIN2006-08447) and the Government of Galicia (PGIDIT07PXIB206028PR).

6. References

Bauer, J.; Sünderhauf, N. & Protzel, P. (2007). Comparing Several Implementations of Two Recently Published Feature Detectors. *Proceedings of The International Conference on Intelligent and Autonomous Systems*, (IAV), Toulouse, France.

Bay, H.; Ess, A.; Tuytelaars, T. & Gool, L. V. (2006). SURF: Speeded Up Robust Features, *Proceedings of the 9th European Conference on Computer Vision*, pp. 404-417, ISBN 978-3-540-3971-7, May 2006, Graz (Austria), Springer LNCS volume 3951.

Bay, H.; Ess, A.; Tuytelaars, T. & Gool, L. V. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346-359, ISSN 1077-3142.

Bonaiuto, J. J. & Itti, L. (2005). Combining Attention and Recognition for Rapid Scene Analysis, *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05)*, pp. 90-90, ISBN 978-88-89884-09-6, San Diego (USA), June 2005, IEEE Computer Society.

Christensen, H. & Nagel, H. (Eds.)(2006). *Cognitive Vision Systems: Sampling the Spectrum of Approaches*, LNCS, Springer, ISBN 978-3-540-3971-7, Heidelberg.

Duda, R. O.; Hart, P. E. & Stork D. G. (2002). *Pattern Classification (2nd Edition)*. John Wiley & Sons, New York, ISBN: 0-471-05669-3.

Ehtiati, T. & Clark, J. J. (2004). A Strongly Coupled Architecture for Contextual Object and Scene identification. In: *Proceedings of International Conference on Pattern Recognitionn (ICPR 2004)*, Vol. 3, pp. 69-72, ISBN 0-7695-2128-2.

Felzenszwalb, P. F. & Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, Vol. 59, No. 2, pp. 167-181, ISSN 0920-5691 (print version), ISSN 1573-1405 (electronic version).

Field, D. J. (1987). Relations Between the Statistics of Natural Images and the Response Properties of Cortical Cells. *Journal of the Optical Society of America A*, Vol. 4, No. 12, pp. 2379-2394, ISSN 1084-7529 (print version), ISSN 1520-8532 (electronic version).

García-Díaz, A.; Fdez-Vidal, X. R.; Dosil, R. and Pardo, X. M. (2008). Local Energy Variability as a Generic Measure of Bottom-Up Salience, In: *Pattern Recognition Techniques, Technology and Applications*, Peng-Yeng Yin (Ed.), pp. 1–24 (Chapter 1), In-Teh, ISBN 978-953-7619-24-4,Vienna.

Itti, L. & Koch, C. (2000). A Saliency-based Search Mechanism for Overt and Covert Shifts of Visual Attention. *Vision Research*, Vol. 40, pp. 1489–1506, ISSN 0042-6989.

Koch, C. & Ullman, S. (1985). Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. *Human Neurobiology*, Vol. 4, No. 4, pp. 219-227, ISSN 0721-9075.

Kovesi, P. (1996). Invariant Measures of Image Features from Phase Information. Ph.D. Thesis, The University or Western Australia.

Levine, M. D. (1985). *Vision in Man and Machine*, McGraw-Hill, New York.

Loncaric, S. (1998). A Survey of Shape Analysis Techniques. *Pattern Recognition*. Vol. 31, No. 8, pp. 983-1001, ISSN 0031-3203.

Lowe, D. G. (2004). Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, ISSN 0920-5691 (print version), ISSN 1573-1405 (electronic version).

Mikolajczyk, K. & Schmid, C. (2005). A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp. 1615-1630, ISSN 0162-8828.

Milanese, R.; Gil, S. and Pun, T. (1995). Attentive Mechanisms for Dynamic and Static Scene Analysis. *Optical Engineering*, Vol. 34, No. 8, pp. 2428–2434, ISSN 0091-3286.

Morrone, M. C. & Burr, D. C. (1988). Feature Detection in Human Vision: A Phase-Dependent Energy Model. In: *Proceedings of the Royal Society of London B*, Vol. 235, pp. 221-245, ISSN 0080–4649.

Niemann, h.; Sagerer, G.; Schröder, S. and Kummert, F. (1990). ERNEST: A Semantic Network System for Pattern Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 12, No. 9, pp. 883-905, ISSN 0162-8828.

Oliva, A. & Torralba, A. (2007). The Role of Context in Object Recognition. *Trends in Cognitive Sciences*, Vol. 11, No. 12, pp. 520-527, ISSN 1364-6613.

Perko, R. and Leonardis, A. (2008). Context Driven Focus of Attention for Object Detection, In: *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint*, pp. 216-233, ISBN 978-3-540-77342-9, Springer, Berlin.

Petrou, M. & Bosdogianni, P. (1999). *Image Processing: The Fundamentals*. John Wiley & Sons, New York, ISBN: 978-0-471-99883-9.

Quillian, M. R. (1969). *Semantic Memory in Semantic Information Processing*. MIT Press, ISBN 978-0262130448.

Russell, S. and Norvig, P. (2003). *Artifical Intelligence: A Modern Approach (2nd Edition)*, Prentice Hall, ISBN 978-0-13-790395-5.

Sarkar, S. & Boyer, K. L. (1993). Perceptual Organization in Computer Vision: A Review and a Proposal for a Classificatory Structure. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 2, March/April 1993, pp. 382-399, ISSN 0018-9472.

Treisman, A. & Gelade, G. (1980). A Feature Integration Theory of Attention. *Cognitive Psychologhy*, Vol. 12, pp. 97–136, ISSN 0010-0285.

Vernon, D. (2006). The Space of Cognitive Vision, In: *Cognitive Vision Systems: Sampling the Spectrum of Approaches*, Christensen H. and Nagel H. (Eds.), LNCS, pp. 7-26, Springer, ISBN 978-3-540-3971-7, Heidelberg.

Vernon, D. (2008). Editorial of Image and Vision Computing Special Issue on Cognitive Vision. *Image and Vision Computing*, Vol. 26, No. 1, January 2008, pp. 1-4, ISSN 0262-8856.

Vinje, W. E. & Gallant, J. L. (2000). Sparse Coding and Decorrelation in Primary Visual Cortex During Natural Vision. *Science*, Vol. 287, pp. 1273–1276, ISSN 0036-8075 (print version), ISSN 1095-9203 (electronic version).

Walther, D.; Rutishauser, U.; Koch, C. & Perona, P. (2005). Selective Visual Attention Enables Learning and Recognition of Multiple Objects in Cluttered Scenes. *Computer Vision and Image Understanding*, Vol. 100, pp. 1-63, ISSN 1077-3142.

Watanabe, S. (1985). *Pattern Recognition: Human and Mechanical*, John Wiley & Sons, New York.

Weliky, M.; Fiser, J.; Hunt R. H. & Wagner D. N. (2003). Coding of Natural Scenes in Primary Visual Cortex. *Neuron*, Vol. 37, pp. 703-718, ISSN 0896-6273.

Zetzsche, C. (2005). Natural Scene Statistics and Salient Visual Features. In: *Neurobiology of Attention*, Itti, L.; Rees, G. & Tsotsos, J. K. (Eds), pp. 226-232 (Chapter 37), Elsevier Academia Press, ISBN 0-12-375731-2, London.

Zhou, Q.; Ma, L.; and Chelberg, D. (2006). Adaptive Object Detection and Recognition Based on a Feedback Strategy. *Image and Vision Computing*, Vol. 24, No. 1, pp. 80-93, ISSN 0262-8856.

Multidirectional Binary Pattern for Face Recognition

Sanqiang Zhao and Yongsheng Gao
Griffith University
Australia

1. Introduction

During the past three decades, extensive research has been conducted on automatically recognising the identity of individuals from their facial images. In spite of the existence of alternative technologies such as fingerprint and iris recognition, human face remains one of the most popular cues for identity recognition in biometrics. Face recognition possesses the non-intrusive nature and are often effective without the participant's cooperation or knowledge. It makes a good compromise between performance reliability and social acceptance and well balances security and privacy. Other biometric methods do not possess these advantages. For instance, fingerprint recognition methods require the subjects to cooperate in making explicit physical contact with the sensor surface (Maltoni et al., 2003). Similarly, iris recognition methods require the subjects to cooperate in placing their eyes carefully relative to the camera. Nowadays, automatic face recognition has become one of the most active research topics in computer vision and pattern recognition, and received much attention from both scientific and engineering communities.

The immediate motivation for this growing interest stems from various commercial applications relating to security and surveillance, such as bankcard identification, access control, airport monitoring and law enforcement. The availability of public large-scale datasets of face images, e.g. (Bailly-Bailliére et al., 2003; Martínez & Benavente, 1998; Messer et al., 1999; Phillips et al., 2005; Phillips et al., 1998; Sim et al., 2003), and evaluation protocols for assessing the performance of different techniques, e.g. (Bailly-Bailliére et al., 2003; Beveridge et al., 2005; Gao et al., 2008; Messer et al., 1999; Phillips et al., 2005; Phillips et al., 2000), further advances the development of face recognition algorithms. Possibly, understanding of our human selves also forms a motivating factor of face recognition (Martínez et al., 2003). In fact, researchers have investigated various issues related to face recognition by humans and machines. Many studies in psychophysics and neuroscience have direct relevance to engineers working on designing algorithms or systems for face recognition (Zhao et al., 2003).

The purpose of face recognition is to visually identify or verify one or more persons from input still or video images. This task is performed by matching the input images (also known as the *probe*) against the model images (also known as the *gallery*), which are the faces of known people in a database. A typical face recognition system contains four major steps: 1) face detection, in which the presence of one or more faces in an input image is detected and the rough positions of these faces are located; 2) face localisation, in which the accurate positions and sizes of faces are decided; 3) feature extraction, in which discriminative features are extracted from each face region to represent identity information. A prior face normalisation procedure may be involved in this step; and 4) feature classification, in which discriminative features are fed into the classification algorithm for identification or verification. After more than 30 years of research, the state-of-the-art face recognition techniques have demonstrated a certain level of maturity on large databases in well-controlled environments (Li & Jain, 2005; Matas et al., 2000; Messer et al., 2004a; Messer et al., 2004b; Messer et al., 2003; Phillips et al., 2003; Phillips et al., 2000; Zhao et al., 2003). Nevertheless, face recognition in uncontrolled conditions is still challenging and far from adequate to deal with most general purpose tasks (Li & Jain, 2005; Phillips et al., 2006; Phillips et al., 2003; Phillips et al., 2007; Zhao et al., 2003). A wide range of variations are inevitable when face images are acquired in an uncontrolled and uncooperative scenario. These variations, such as pose variation, illumination variation and facial expression variation, can cause serious performance degradation, and thus form important challenges to be solved in the research community.

Existing technologies for face recognition can be roughly classified into holistic approaches and analytic approaches. Using information derived from the whole face image, holistic approaches, such as Eigenface (Turk & Pentland, 1991) and Fisherface (Belhumeur et al., 1997), are conceptually simple and easy to implement, but their performance is affected by facial expression, pose and illumination changes in practice. On the other hand, analytic approaches, such as Elastic Bunch Graph Matching (EBGM) (Lades et al., 1993; Wiskott et al., 1997), Line Edge Map (LEM) (Gao & Leung, 2002) and Directional Corner Point (DCP) (Gao & Qi, 2005), extract local information from salient facial features to distinguish faces. Represented by a set of low dimensional local feature vectors, these methods have the advantage of robustness to environmental variations. Recently, the Local Binary Pattern (LBP) approach (Ahonen et al., 2004; Ahonen et al., 2006) has proven to be a quite successful achievement for face recognition, providing a new way of investigation into face analysis. As a non-parametric local descriptor, LBP was originally designed for texture description (Ojala et al., 1996; Ojala et al., 2002; Ojala et al., 2001), but later extended to face recognition and outperformed existing methods such as PCA, Bayesian and EBGM methods (Ahonen et al., 2006). Two most important properties of the LBP operator in real-world applications are its computational efficiency and robustness against monotonic gray-level changes. The first property makes it possible to analyse images in challenging real-time settings. LBP has also been applied to facial expression analysis (Zhao & Pietikäinen, 2007) and background modelling (Heikkilä & Pietikäinen, 2006).

The basic principle of LBP is that a face can be seen as a composition of micropatterns generated by the concatenation of the circular binary gradients. The statistical distribution (histogram) of these illumination invariant micropatterns is used as a discriminative feature for identification. The LBP operator is, by design, suitable for modelling repetitive texture patches, and is sensitive to random and quantisation noise in uniform image areas. Due to the fact that a holistic LBP histogramming retains only the frequencies of micropatterns and discards all information about their spatial layout, Ahonen et al. (2006) employed a spatially enhanced histogram for face recognition, which is extracted from evenly divided subregions of a face, followed by a histogram concatenation. This arbitrary spatial partition is not in accordance with local facial morphology, and thus inevitably leads to loss of discriminative power.

In this chapter, we propose to extract micropatterns from the neighbourhoods of a sparse set of shape-driven points which are detected from edge map with rich information content on a face image. Both the number and the locations of the points vary with different individuals such that diverse facial characteristics of these individuals can be represented. To enhance the discriminative power of micropatterns, we also propose a Multidirectional Binary Pattern (MBP) to reflect binary patterns spanning multiple directions. The new representation is capable of describing both global structure and local texture, and also significantly reduces the high dimensionality of LBP histogram description. It inherits most of the other advantages of LBP such as computational efficiency and exemption from training. Besides, the proposed method can effectively alleviate the problem of sensitivity to random noise in uniform image areas, because MBP features are only extracted from the neighbourhoods of the sparse points, which are generally non-uniform areas. Using a new MBP measurement, we performed an investigation and evaluation of the proposed method for establishing point correspondence on the publicly available AR face database (Martínez & Benavente, 1998). A higher recognition accuracy than that of the Directional Corner Point (DCP) method (Gao & Qi, 2005) was obtained in our experiments, demonstrating the validity of this method on face recognition.

The remainder of this chapter is organised as follows. Section 2 presents the details of the proposed MBP representation, which is derived from a detection algorithm of sparse points and an illumination-insensitive pattern descriptor attached on each point. Section 3 describes using the specially designed MBP measurement to establish the correspondence among sparse points. In Section 4, the proposed method is experimentally evaluated through comparative experiments on the AR database. The last section summarises this chapter.

2. Representation

In this section, we first present a brief introduction of Local Binary Pattern (LBP), and then propose a new Multidirectional Binary Pattern (MBP). MBP is extracted from a sparse set of shape-driven points. This is different from most LBP approaches that cluster LBP occurrences from local image patches and thus can better represent both global structure and local texture for coding a face.

2.1 Local Binary Pattern

Initially derived from texture analysis community, the LBP operator was created as a gray-level invariant texture measure to model texture images (Ojala et al., 1996; Ojala et al., 2002; Ojala et al., 2001). Later, it demonstrated excellent performance in many other research fields in terms of both speed and discrimination capability (Ahonen et al., 2006; Heikkilä & Pietikäinen, 2006; Zhao & Pietikäinen, 2007).

Specifically, the LBP operator marks each pixel I_c of an image as a decimal number $LBP_{P,R}(I_c)$, which is formed by thresholding the P equally spaced neighbour pixels $I_{p,R}$ $(p = 0,\cdots,P-1)$ on a circle of radius R with the centre pixel I_c and concatenating the results binomially with factor 2^p:

$$LBP_{P,R} = \sum_{p=0}^{P-1} T\left(I_{p,R} - I_c\right) 2^p \tag{1}$$

where the thresholding function $T(x)$ is defined as

$$T(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

If the coordinate of I_c is $(0,\ 0)$, the coordinates of $I_{p,R}$ are given by $(-R\sin(2\pi p/P),\ R\cos(2\pi p/P))$. The gray-level values of neighbours $I_{p,R}$ not falling exactly in the centre of pixels are estimated by interpolation (Ojala et al., 2002). Fig. 1 illustrates an example of obtaining a LBP micropattern $LBP_{8,1}$ with the parameters $P = 8$ and $R = 1$.

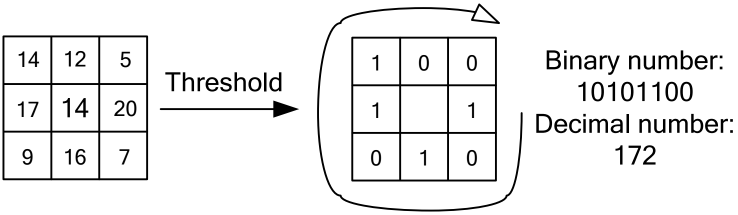


Fig. 1. The LBP operator.

2.2 Sparse points detection

The edges in an image reflect large local intensity changes that are caused by the geometric structure of the object, the characteristics of the surface reflectance of the object and the viewing direction (Gao & Leung, 2002). Containing spatial information, a sparse set of points is detected at positions which have rich edge information in a face image. In contrast to traditional methods where feature points are often predefined as the locations of eyes, nose, mouth, etc., we do not fix either the number or the locations of the sparse points. The number of the sparse points and their locations can vary in order to better represent diverse facial characteristics of different persons, such as dimples, moles, etc. These diverse features are also important cues that humans might use for recognising faces.

In order to ensure less demand on storage space and less sensitivity to illumination changes, the sparse points should be placed on the significant edge curves with high curvatures. While any general edge detection method can be used to detect the sparse points, we use an edge detector from (Nevatia & Babu, 1980), followed by the Dynamic Two-Strip algorithm (Dyn2S) (Leung & Yang, 1990) to obtain these points. After the edge map of a face image is detected, a strip is fitted to the left and right of each point on an edge curve, and the points inside each strip are approximated as a straight line. The orientation and width of the strip are adjusted automatically. Longer and narrower strips are favoured. In addition, the curvature and a measure of merit of each point can be calculated. Sparse points are selected in a three-step procedure:

1) Points with a small merit compared to their neighbours are eliminated.
2) A number of points, chosen from any points that are not covered by one of the strips selected in the first step, are reinstated to avoid over-elimination.

3) Points that align approximately on a straight line are deleted except for the two endpoints on the curve.

The remaining points after these steps are the detected sparse points. Fig. 2 illustrates two examples of sparse points superimposed on the original face image from the AR database.

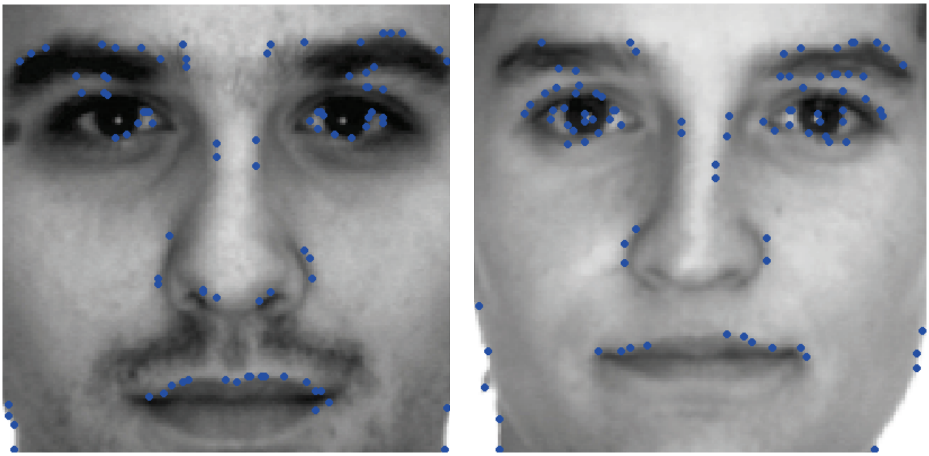


Fig. 2. Detected sparse points.

2.3 Multidirectional Binary Pattern

After the sparse points are detected, MBPs are extracted from these point positions. A MBP is defined as a pattern set which consists of four bunches of directional binary patterns: Horizontal Binary Patterns (HBPs), Vertical Binary Patterns (VBPs), Ascending Binary Patterns (ABPs) and Descending Binary Patterns (DBPs). In other words, MBP is composed of binary pattern bunches collected from four different directions. Fig. 3 visually illustrates the positions covered by these four pattern bunches. Similar to LBP, the pixels in the neighbourhoods are thresholded with the value of the centre pixel, and then linearly concatenated into four directional binary patterns as a local descriptor. One difference between MBP and LBP is that MBP is kept as original *binary patterns*, without being transformed into decimal figures for histogramming as in LBP. It should be noted that although the four bunches of directional binary patterns may be derived from the same pixels, the pattern-level features they represent are different. This is demonstrated from the example in Fig. 4. Mathematically, a MBP set takes the form

$$MBP = \{HBP_{L,N}, VBP_{L,N}, ABP_{L,N}, DBP_{L,N}\} \tag{3}$$

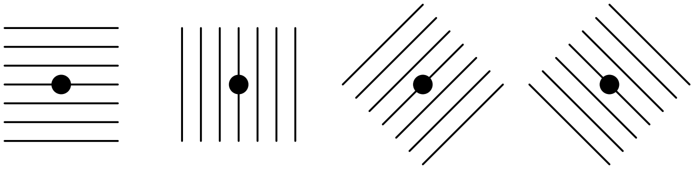
where *HBP*, *VBP*, *ABP*, and *DBP* refer to the four bunches of directional binary patterns respectively, with each bunch containing *N* binary patterns of the length *L*. For instance, the bunch of HBPs can be represented as

$$HBP_{L,N} = \{HBP_{L,1}, HBP_{L,2}, \cdots, HBP_{L,N}\} \tag{4}$$

where each HBP is composed of concatenated *L* binary values:

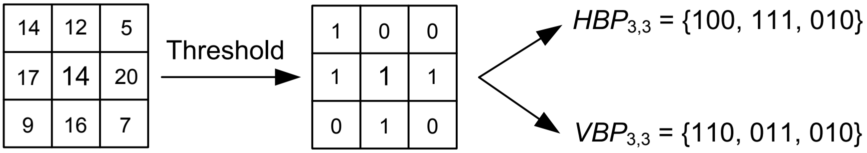
$$HBP_{L,n} = [T(I_{H(1,n)} - I_c), T(I_{H(2,n)} - I_c), \cdots, T(I_{H(L,n)} - I_c)]. \quad 1 \le n \le N \tag{5}$$

Here $I_{H(l,n)}$ $(1 \le l \le L,\ 1 \le n \le N)$ represent the horizontally spaced pixels located at $L \times N$ positions in the neighbourhood of the centre pixel I_c (see Fig. 3a). Similar representations are applied to the remaining three bunches of directional binary patterns. Fig. 4 provides an example of obtaining two bunches of directional binary patterns $HBP_{3,3}$ and $VBP_{3,3}$ with the parameters $L = 3$ and $N = 3$.



(a) (b) (c) (d)

Fig. 3. Multidirectional Binary Pattern (MBP). (a) Horizontal Binary Patterns (HBPs). (b) Vertical Binary Patterns (VBPs). (c) Ascending Binary Patterns (ABPs). (d) Descending Binary Patterns (DBPs). The black dot stands for the centre pixel.



14	12	5
17	14	20
9	16	7

1	0	0
1	1	1
0	1	0

$HBP_{3,3}$ = {100, 111, 010}

$VBP_{3,3}$ = {110, 011, 010}

Fig. 4. An illustration of obtaining $HBP_{3,3}$ and $VBP_{3,3}$.

Based on this description, a face is represented by a sparse set of shape-driven points with MBP attached on each point as local texture. The MBP representation is extracted from sparse points rather than from histogramming all the pixels, and thus reduces the storage demand of an image. It also inherits LBP's advantage of insensitivity to illumination changes. Because the sparse points are derived from low-level edge map with rich feature information, they circumvent uniform areas where LBP suffers from random and quantisation noise. Meanwhile, the four-bunch MBP provides enhanced discriminative power for representation in order to improve the recognition accuracy.

3. Measurement

In practical applications, face images of a same individual generally suffer from intra-class variations such as illumination, expression and ageing. Finding correspondence of MBP pairs between two images is therefore very important to reveal the substantial similarity/difference of two faces. In this section, we first propose a new Binary Pattern Distance (BPD) to measure binary patterns, and then integrate it into a compound cost function to establish MBP correspondence for face recognition. The cost function is motivated by the Hausdorff distance concept (Dubuisson & Jain, 1994). Hausdorff distance has been widely utilised as shape comparison metrics on binary images.

3.1 Binary Pattern Distance

As a preliminary step, two distances are proposed to take measurement of two binary patterns (a model binary pattern BP^M and a test binary pattern BP^T): *pattern distance* and *shifting distance*. Representing the pattern-level disparity between two binary patterns, the pattern distance d_p is measured by examining the Hamming distances (the accumulated sum of the disagreeing bits in between) of the model pattern and the test pattern with several bit-wise shifts. The minimal value of these distances is selected as d_p. The shifting distance d_s is defined as the number of shifting bits at which the pattern distance reaches the minimum. Fig. 5 provides examples of the proposed two distances. It is possible to assume that the pattern-level disparity originates from inter-class variation and the bit-wise shifting comes from intra-class variation. Therefore, d_p and d_s have the ability to reveal the local feature's inter-class and intra-class variations respectively.

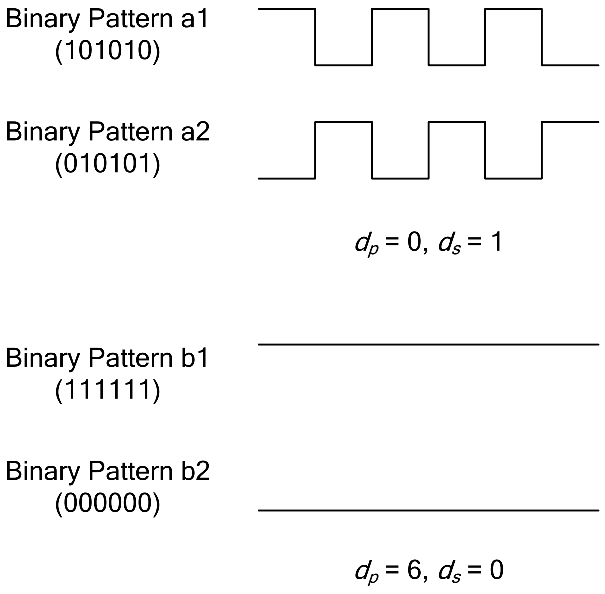


Fig. 5. The pattern distance (d_p) and the shifting distance (d_s).

Following the definitions of d_p and d_s, the BPD of binary patterns BP^M and BP^T is represented as:

$$BPD(BP^M, BP^T) = \sqrt{d_p^2 + \rho d_s^2} \tag{6}$$

where

$$\begin{cases} d_p = \min\limits_{-K \le k \le K} \{HD(BP^M, SH(BP^T, k))\} \\ d_s = |\arg\min\limits_{-K \le k \le K} \{HD(BP^M, SH(BP^T, k))\}| \end{cases} \tag{7}$$

Here ρ is used to balance the contributions of d_p and d_s. HD stands for the Hamming distance. The operation $SH(BP^T, k)$ performs a bit-wise directional shifting on BP^T for k $(k = -K, \cdots, 0, \cdots, K)$ times. A positive k stands for a forward-shifting; a negative k stands for a backward-shifting; and when k equals 0, no shifting operation is performed. K $(K \ge 0)$ is the bit-wise shifting limit.

3.2 MBP distance

For two MBPs (MBP^M and MBP^T) composed of four bunches of binary patterns respectively, the average BPD in each directional bunch is calculated, and then the minimal mean of four bunches is selected and defined as the MBP distance:

$$\begin{aligned} d(MBP^M, MBP^T) = \min\Big\{ & {}^1\!/_N \textstyle\sum_{n=1}^{N} BPD(HBP^M_{L,n}, HBP^T_{L,n}), \\ & {}^1\!/_N \textstyle\sum_{n=1}^{N} BPD(VBP^M_{L,n}, VBP^T_{L,n}), \\ & {}^1\!/_N \textstyle\sum_{n=1}^{N} BPD(ABP^M_{L,n}, ABP^T_{L,n}), \\ & {}^1\!/_N \textstyle\sum_{n=1}^{N} BPD(DBP^M_{L,n}, DBP^T_{L,n}) \Big\}. \end{aligned} \tag{8}$$

This measurement involves bit-wise shifting of local patterns in four different directions (see Fig. 6). By using a small balancing factor ρ, it can provide robustness to small local feature distortion caused by intra-class variation.

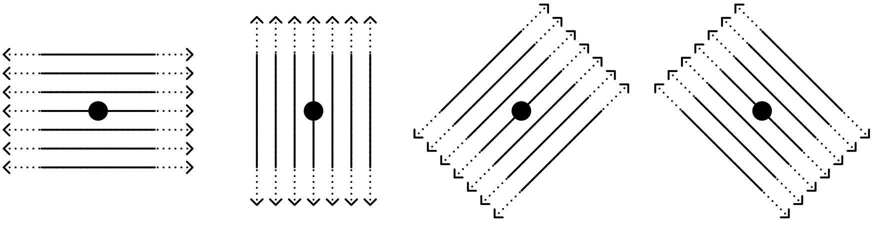


Fig. 6. The bit-wise directional shifting of MBP.

3.3 Compound cost function

A cost function is defined to find correspondence of MBP pairs between two face images. Given two finite MBP sets $M = \{MBP_1^M, MBP_2^M, \cdots, MBP_P^M\}$ representing a model face in the database and $T = \{MBP_1^T, MBP_2^T, \cdots, MBP_Q^T\}$ representing a test face from input, where P and Q are the numbers of MBPs in M and T respectively. The cost function takes the form:

$$D(M,T) = \max\{dirD(M,T), dirD(T,M)\} \tag{9}$$

where the function $dirD(M,T)$ is the directed cost function from set M to T. Since the point position (x,y) where each MBP is extracted has been recorded, the directed MBP cost function can be defined as

$$dirD(M,T)=\frac{1}{P}\sum_{p=1}^{P}\min_{1\le q\le Q}\sqrt{(x_p^M-x_q^T)^2+(y_p^M-y_q^T)^2+\lambda d^2(MBP_p^M,MBP_q^T)} \tag{10}$$

This is a compound measurement composed of both spatial information and MBP features. The weight λ is used to balance the contributions of Euclidean distance and MBP distance. The cost function $D(M,T)$ evaluates the degree of mismatch between two MBP sets by measuring the distance of the MBP of M that has the largest distance from any MBP of T, and vice versa.

4. Experimental results

The proposed method was assessed on the public available AR face database (Martínez & Benavente, 1998), which contains over 4000 colour images from 126 peoples (70 men and 56 women). The database covers frontal view faces under controlled condition, different facial expressions and different illumination conditions. There are 26 different images per person, recorded in two different sessions with a two-week time interval, and each session consists of 13 images. Because images in some sessions were missing, we eventually obtained 120 complete set of images (65 men and 55 women). All the images were normalised (in scale and rotation) and cropped to 160 × 160 pixels based on the manually labelled positions of two eyes. We fixed the MBP size as $L=8$, $N=8$, the bit-wise shifting limit as $K=4$ and the balancing factor as $\rho=0.1$ in our experiments.

4.1 Determination of parameters

The weight λ In Equation (10) balances the contributions of spatial and MBP measurements. In this subsection, a set of experiments was performed to determine λ using all the neutral expression faces in the AR database. The model set is the neutral faces in the first session, and the test set is those in the second session. The top-one recognition accuracy against the weight λ is displayed in Fig. 7. The recognition accuracy reached and remained maximum when λ ranged from 120 to 300. The weight $\lambda=160$ was selected and used in the rest of the experiments.

In the following, the MBP method was compared with the Directional Corner Point (DCP) method (Gao & Qi, 2005) under various situations, using the neutral faces in normal condition taken in the first session as the model set.

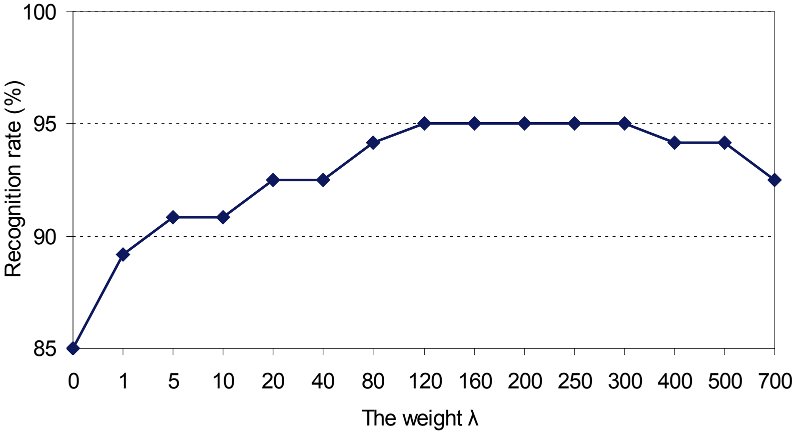


Fig. 7. Recognition accuracy against the weight.

4.2 Face recognition results

The face images under controlled condition in the second session were first used to evaluate the proposed method. The comparative recognition accuracy is illustrated in Fig. 8. Although the number of subjects used in this study (120) was more than that in DCP (112), the proposed MBP method still outperformed the DCP method.

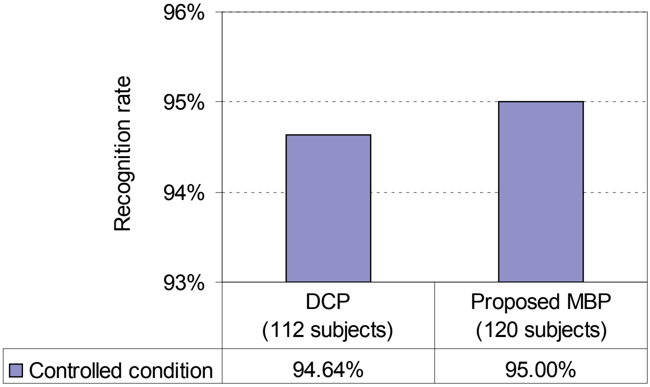
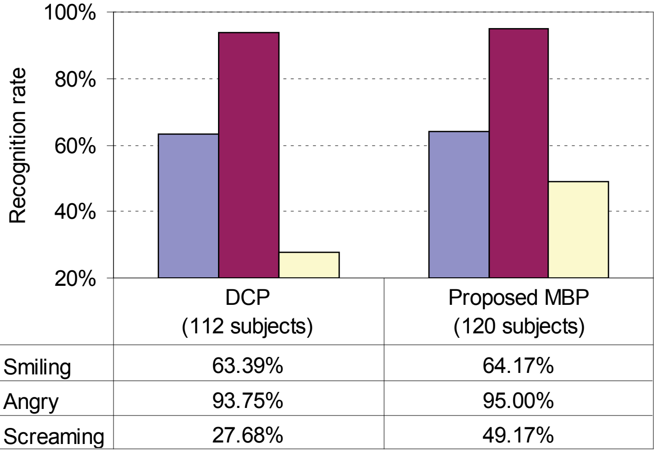


Fig. 8. Comparative recognition accuracy under controlled condition.

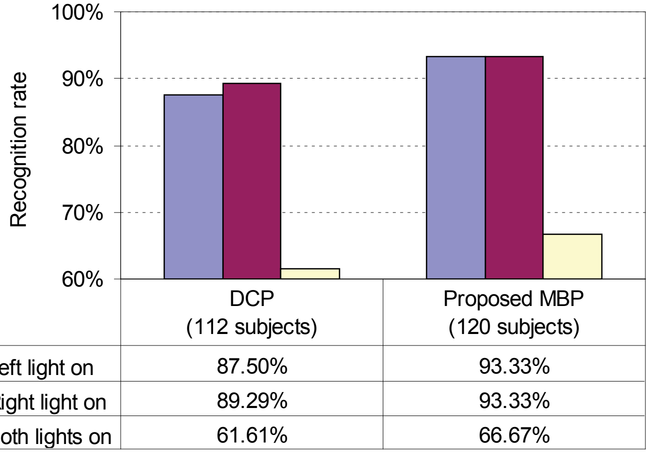
To compare the recognition accuracy with expression variations, the experiment was also performed on three different sets of images with smiling, angry and screaming expressions in the first session. The results are listed in Fig. 9. It can be seen from the figure that the performance of the proposed method is much better than the DCP method under all three expression variations, especially under the screaming condition, where the improvement is over 20%. This can be explained by the robustness of MBP against local feature distortion. It indicates that the locations of feature points might be subject to significant change from screaming, but the pattern-level disparity of their neighbourhoods is comparably stable.



	DCP (112 subjects)	Proposed MBP (120 subjects)
🟦 Smiling	63.39%	64.17%
🟪 Angry	93.75%	95.00%
🟨 Screaming	27.68%	49.17%

Fig. 9. Comparative recognition accuracy under different expressions.

We finally performed the experiment under the condition of illumination changes. The AR database contains three different lighting conditions: left light, right light and both lights on. Fig. 10 displays these experimental results. The recognition accuracy of the proposed method is noticeably above 90% when either left or right light on. This demonstrates that MBP is very tolerant to lighting changes. However, it is still sensitive to extreme lighting, which causes strong specular reflectance on the face skin and thus could erase some sparse points.



	DCP (112 subjects)	Proposed MBP (120 subjects)
🟦 Left light on	87.50%	93.33%
🟪 Right light on	89.29%	93.33%
🟨 Both lights on	61.61%	66.67%

Fig. 10. Comparative recognition accuracy under different illuminations.

5. Conclusions

Local Binary Pattern (LBP) has proved to be a powerful descriptor for both texture and facial images, demonstrating excellent performance in computer vision community. This chapter proposed a more discriminative Multidirectional Binary Pattern (MBP) for face representation. Faces are modelled as a sparse set of shape-driven points with MBP attached on each point. The main contributions of the proposed method are: 1) Binary pattern bunches from multiple directions are collected to enhance the discriminative power of local features. 2) In stead of histogramming all the pixels of an image, local features are extracted from sparse points to reduce the storage demand. 3) A specially designed MBP measurement is proposed to evaluate binary patterns and establish point correspondence. The experiments on face recognition demonstrated the effectiveness of the proposed method against different environmental variations. This study reveals that the proposed MBP method provides a new solution towards robust face recognition.

6. References

Ahonen, T.; Hadid, A. & Pietikäinen, M. (2004). Face Recognition with Local Binary Patterns, *Proceedings of the European Conference on Computer Vision*, Vol. 3021, pp. 469-481

Ahonen, T.; Hadid, A. & Pietikäinen, M. (2006). Face Description with Local Binary Patterns: Application to Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 12, pp. 2037-2041

Bailly-Bailliére, E.; Bengio, S.; Bimbot, F.; Hamouz, M.; Kittler, J.; Mariéthoz, J.; Matas, J.; Messer, K.; Popovici, V.; Porée, F.; Ruiz, B. & Thiran, J.-P. (2003). The BANCA Database and Evaluation Protocol, *Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication*, Vol. 2688, pp. 625-638

Belhumeur, P. N.; Hespanha, J. P. & Kriegman, D. J. (1997). Eigenfaces Vs. Fisherfaces: Recognition Using Class Specific Linear Projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 711-720

Beveridge, J. R.; Bolme, D.; Draper, B. A. & Teixeira, M. (2005). The CSU Face Identification Evaluation System: Its Purpose, Features, and Structure, *Machine Vision and Applications*, Vol. 16, No. 2, pp. 128-138

Dubuisson, M. P. & Jain, A. K. (1994). A Modified Hausdorff Distance for Object Matching, *Proceedings of the International Conference on Pattern Recognition*, Vol. 1, pp. 566-568

Gao, W.; Cao, B.; Shan, S.; Chen, X.; Zhou, D.; Zhang, X. & Zhao, D. (2008). The CAS-PEAL Large-Scale Chinese Face Database and Baseline Evaluations, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 38, No. 1, pp. 149-161

Gao, Y. & Leung, M. K. H. (2002). Face Recognition Using Line Edge Map, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 6, pp. 764-779

Gao, Y. & Qi, Y. (2005). Robust Visual Similarity Retrieval in Single Model Face Databases, *Pattern Recognition*, Vol. 38, No. 7, pp. 1009-1020

Heikkilä, M. & Pietikäinen, M. (2006). A Texture-Based Method for Modeling the Background and Detecting Moving Objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 4, pp. 657-662

Lades, M.; Vorbrüggen, J. C.; Buhmann, J.; Lange, J.; von der Malsburg, C.; Würtz, R. P. & Konen, W. (1993). Distortion Invariant Object Recognition in the Dynamic Link Architecture, *IEEE Transactions on Computers*, Vol. 42, No. 3, pp. 300-311

Leung, M. K. & Yang, Y.-H. (1990). Dynamic Two-Strip Algorithm in Curve Fitting, *Pattern Recognition*, Vol. 23, No. 1-2, pp. 69-79

Li, S. Z. & Jain, A. K. (2005). Handbook of Face Recognition, Springer

Maltoni, D.; Maio, D.; Jain, A. K. & Prabhakar, S. (2003). Handbook of Fingerprint Recognition, Springer

Martínez, A. M. & Benavente, R. (1998). The AR Face Database, *CVC Technical Report #24*

Martínez, A. M.; Yang, M. H. & Kriegman, D. J. (2003). Special Issue on Face Recognition, *Computer Vision and Image Understanding*, Vol. 91, No. 1-2, pp. 1-5

Matas, J.; Hamouz, M.; Jonsson, K.; Kittler, J.; Li, Y.; Kotropoulos, C.; Tefas, A.; Pitas, I.; Teewoon, T.; Hong, Y.; Smeraldi, F.; Bigun, J.; Capdevielle, N.; Gerstner, W.; Ben-Yacoub, S.; Abeljaoued, Y. & Mayoraz, E. (2000). Comparison of Face Verification Results on the XM2VTFS Database, *Proceedings of the International Conference on Pattern Recognition*, Vol. 4, pp. 858-863

Messer, K.; Kittler, J.; Sadeghi, M.; Hamouz, M.; Kostin, A.; Cardinaux, F.; Marcel, S.; Bengio, S.; Sanderson, C.; Poh, N.; Rodriguez, Y.; Czyz, J.; Vandendorpe, L.; McCool, C.; Lowther, S.; Sridharan, S.; Chandran, V.; Palacios, R. P.; Vidal, E.; Bai, L.; Shen, L.; Wang, Y.; Chiang, Y.-H.; Liu, H.-C.; Hung, Y.-P.; Heinrichs, A.; Müller, M.; Tewes, A.; von der Malsburg, C.; Würtz, R.; Wang, Z.; Xue, F.; Ma, Y.; Yang, Q.; Fang, C.; Ding, X.; Lucey, S.; Goss, R. & Schneiderman, H. (2004a). Face Authentication Test on the BANCA Database, *Proceedings of the International Conference on Pattern Recognition*, Vol. 4, pp. 523-532

Messer, K.; Kittler, J.; Sadeghi, M.; Hamouz, M.; Kostyn, A.; Marcel, S.; Bengio, S.; Cardinaux, F.; Sanderson, C.; Poh, N.; Rodriguez, Y.; Kryszczuk, K.; Czyz, J.; Vandendorpe, L.; Ng, J.; Cheung, H. & Tang, B. (2004b). Face Authentication Competition on the BANCA Database, *Proceedings of the International Conference on Biometric Authentication*, Vol. 3072, pp. 8-15

Messer, K.; Kittler, J.; Sadeghi, M.; Marcel, S.; Marcel, C.; Bengio, S.; Cardinaux, F.; Sanderson, C.; Czyz, J.; Vandendorpe, L.; Srisuk, S.; Petrou, M.; Kurutach, W.; Kadyrov, A.; Paredes, R.; Kepenekci, B.; Tek, F.; Akar, G.; Deravi, F. & Mavity, N. (2003). Face Verification Competition on the XM2VTS Database, *Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication*, Vol. 2688, pp. 964-974

Messer, K.; Matas, J.; Kittler, J.; Luettin, J. & Maitre, G. (1999). XM2VTSDB: The Extended M2VTS Database, *Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication*, pp. 72-77

Nevatia, R. & Babu, K. R. (1980). Linear Feature Extraction and Description, *Computer Graphics and Image Processing*, Vol. 13, No. 3, pp. 257-269

Ojala, T.; Pietikäinen, M. & Harwood, D. (1996). A Comparative Study of Texture Measures with Classification Based on Feature Distributions, *Pattern Recognition*, Vol. 29, No. 1, pp. 51-59

Ojala, T.; Pietikäinen, M. & Mäenpää, T. (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 971-987

Ojala, T.; Valkealahti, K.; Oja, E. & Pietikäinen, M. (2001). Texture Discrimination with Multidimensional Distributions of Signed Gray-Level Differences, *Pattern Recognition*, Vol. 34, No. 3, pp. 727-739

Phillips, P. J.; Flynn, P. J.; Scruggs, T.; Bowyer, K. W.; Jin, C.; Hoffman, K.; Marques, J.; Jaesik, M. & Worek, W. (2005). Overview of the Face Recognition Grand Challenge, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 947-954

Phillips, P. J.; Flynn, P. J.; Scruggs, T.; Bowyer, K. W. & Worek, W. (2006). Preliminary Face Recognition Grand Challenge Results, *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 15-24

Phillips, P. J.; Grother, P.; Micheals, R. J.; Blackburn, D. M.; Tabassi, E. & Bone, M. (2003). Face Recognition Vendor Test 2002: Evaluation Report, *NISTIR 6965*

Phillips, P. J.; Moon, H.; Rizvi, S. A. & Rauss, P. J. (2000). The FERET Evaluation Methodology for Face-Recognition Algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, pp. 1090-1104

Phillips, P. J.; Scruggs, W. T.; O'Toole, A. J.; Flynn, P. J.; Bowyer, K. W.; Schott, C. L. & Sharpe, M. (2007). FRVT 2006 and ICE 2006 Large-Scale Results, *NISTIR 7408*

Phillips, P. J.; Wechsler, H.; Huang, J. & Rauss, P. J. (1998). The FERET Database and Evaluation Procedure for Face-Recognition Algorithms, *Image and Vision Computing*, Vol. 16, No. 5, pp. 295-306

Sim, T.; Baker, S. & Bsat, M. (2003). The CMU Pose, Illumination, and Expression Database, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, pp. 1615-1618

Turk, M. & Pentland, A. (1991). Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71-86

Wiskott, L.; Fellous, J. M.; Krüger, N. & von der Malsburg, C. (1997). Face Recognition by Elastic Bunch Graph Matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 775-779

Zhao, G. & Pietikäinen, M. (2007). Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, pp. 915-928

Zhao, W.; Chellappa, R.; Phillips, P. J. & Rosenfeld, A. (2003). Face Recognition: A Literature Survey, *ACM Computing Surveys*, Vol. 35, No. 4, pp. 399-459

Bayesian Video Face Detection with Applications in Broadcasting

Atsushi Matsui [1], Simon Clippingdale [1], Norifumi Okabe [2],
Takashi Matsumoto [3] and Nobuyuki Yagi [1]
[1] *Science & Technology Research Laboratories, Japan Broadcasting Corporation, Tokyo, Japan*
[2] *Broadcast Engineering Department, Japan Broadcasting Corporation, Tokyo, Japan*
[3] *Faculty of Advanced Science and Engineering, Waseda University, Tokyo, Japan*

1. Introduction

Face detection is the most fundamental and critical process in any automated system that deals with face images. Errors in this first step affect subsequent processes such as face localization, face modeling, facial expression analysis, face recognition, user classification, and so on. A face detector should have sufficient robustness to possible variations of face position, scale, orientation, aging, make-up, and illumination.

Several approaches based on different features are available for face detection: model-based, color-based, appearance-based, or a combination of these. Model based approaches can deal with variations in face pose and illumination, but require the initial position of the target a priori. Color based approaches can reduce the search space of the detection system. However, skin color models are not effective where the spectrum of the light source varies significantly, i.e. color appearance is often unstable due to changes in both background and foreground lighting. To the best of the authors' knowledge, the most successful face detection algorithms are based on appearance without using other cues. Although there has been much reported research in this field, it is probably fair to say that the framework of Viola and Jones (Viola & Jones, 2001) has attracted the most attention for its combination of detection accuracy and speed. They introduced a cascaded structure of weak classifiers using a boosted learning algorithm on a pool of simple Haar-like features (Papageorgiou, et al., 1998), and an extremely fast method of evaluating the features at any image location and scale. Lienhart and Maydt (Lienhart & Maydt, 2002) demonstrated the efficacy of extending the feature set; the modified version of the Haar-cascade face detector is available in the open-source computer vision library OpenCV. The computational processes for the face detection system can be divided into two stages: the learning stage and the detection stage. The learning stage involves the selection of the feature pool, variation of training samples, and training algorithm. The detection stage includes the search algorithm, structure, and merging post-processing of multiple outputs.

In this paper, we introduce an online learning scheme based on a Sequential Monte Carlo method (Doucet, 1998); (Freund & Schapire, 1997) to boost computational efficiency on the detection stage.

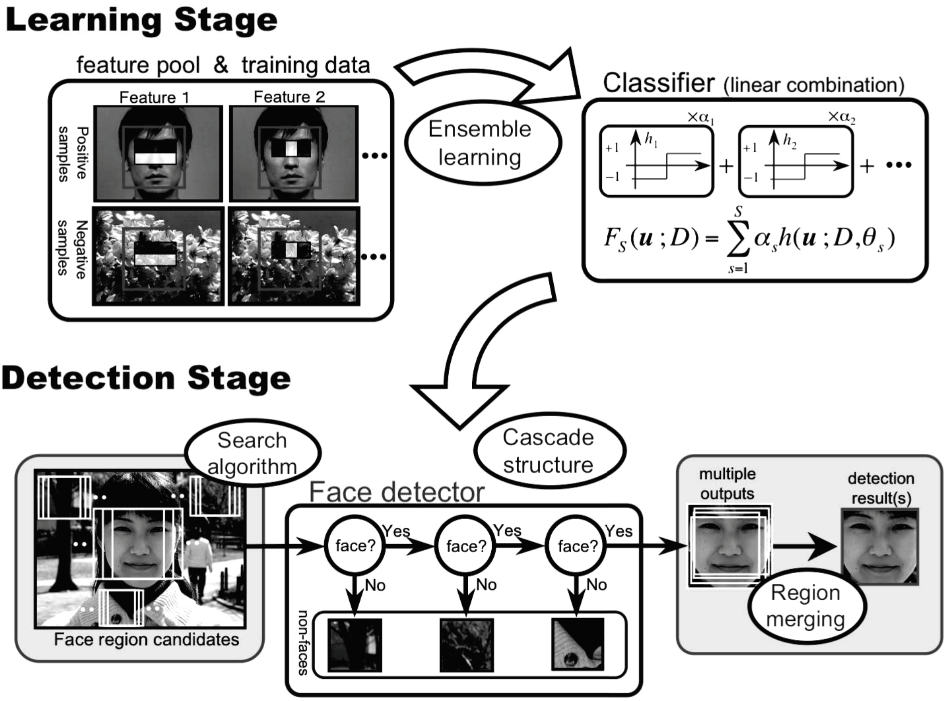


Fig. 1. An overview of a Haar-cascade face detector and associated processes.

2. Bayesian approaches for detection stage

As in numerous other pattern classification tasks, accuracy and speed are twin requirements of face detection systems, with speed especially vital for some video-based applications. More accuracy, however, often requires more computation and thus entails less speed. A combination of a face detector and a tracker (to reduce the search volume in the current frame based on results up to the previous frame) may be used to increase processing speed in such cases. However, the underlying assumption of temporal continuity is violated at scene changes, which must be detected and the system re-initialized in order to avoid generating erroneous detections. In the following sections, we introduce a sequential face detection scheme for video sequences containing scene changes. The algorithm estimates probability distributions of a sequence of face region parameters using a Sequential Monte Carlo (SMC) method (Doucet, 1998). We show that this SMC approach successfully predicts possible regions in face parameter space for the current frame using temporal continuity from past frames, while resetting the distributions automatically at temporal discontinuities corresponding to scene changes.

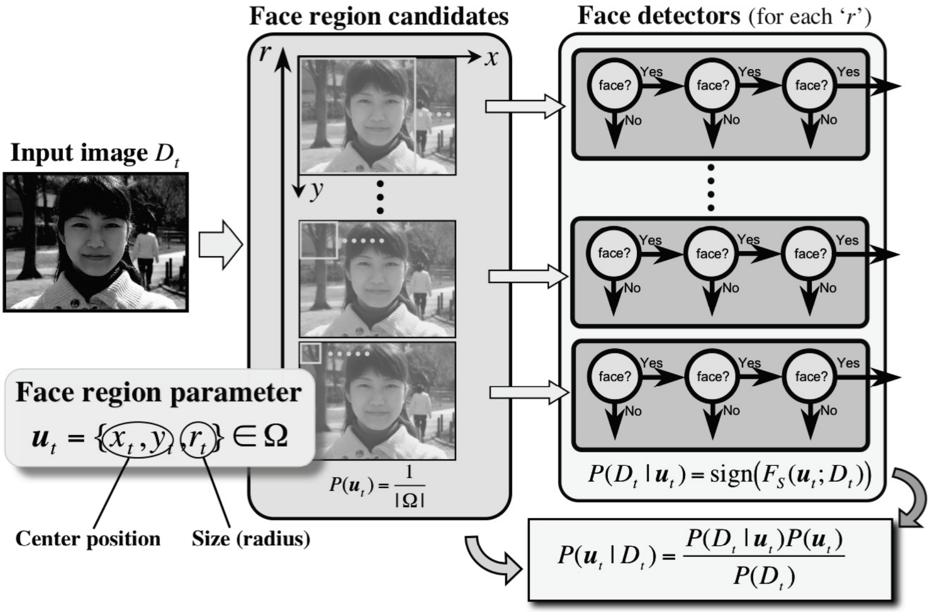

Face region candidates
Face detectors (for each 'r')
Input image D_t
r
x
y
face?
Yes
No
Face region parameter
$u_t = \{x_t, y_t, r_t\} \in \Omega$
Center position
Size (radius)
$P(u_t) = \frac{1}{|\Omega|}$
$P(D_t | u_t) = \mathrm{sign}(F_S(u_t; D_t))$
$P(u_t | D_t) = \frac{P(D_t | u_t)P(u_t)}{P(D_t)}$

Fig. 2. Overview of an exhaustive search algorithm.

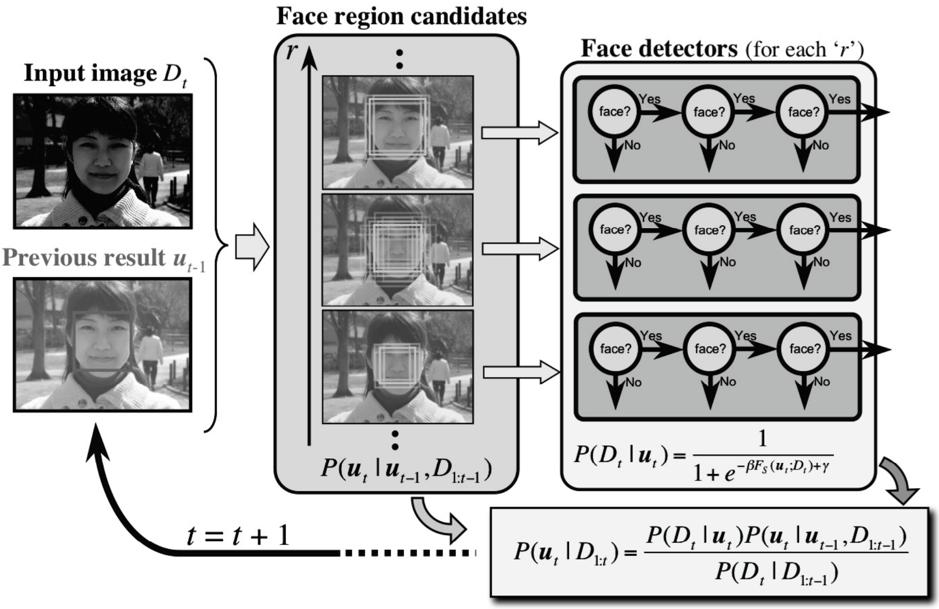

Face region candidates
Face detectors (for each 'r')
Input image D_t
Previous result u_{t-1}
r
face?
Yes
No
$P(u_t | u_{t-1}, D_{1:t-1})$
$P(D_t | u_t) = \frac{1}{1+e^{-\beta F_S(u_t;D_t)+\gamma}}$
$t = t + 1$
$P(u_t | D_{1:t}) = \frac{P(D_t | u_t)P(u_t | u_{t-1}, D_{1:t-1})}{P(D_t | D_{1:t-1})}$

Fig. 3. Overview of the Bayesian sequential importance search algorithm.

2.1 Exhaustive search and frequency-based thresholding

Given an input image, the face detector scans possible combinations of the center positions and sizes of face rectangles. The computational cost reflects the total number of face candidates. Such face detectors are often trained using a training data set with some variations in position and scale, and the prevailing features for each weak learner are relatively simple and have a degree of uncertainty such that a single face in an input image can yield multiple face candidates with a range of positions and scales. One possible approach to handling these multiple outputs is to classify neighboring candidates into groups G_j , and to take an average over all elements of each group(see figure 1).

$$\hat{\boldsymbol{u}}^{[j]} = \frac{1}{\left|G_j\right|} \sum_{\boldsymbol{u}_i \in G_j} \boldsymbol{u}_i \ , \ \ \boldsymbol{u}_i ; \ H_S(\boldsymbol{u}_i ; D_t) = 1 \tag{1}$$

Since true positives usually occur with greater consistency than false positives, if the number of elements belonging to the group G exceeds some threshold η the group may be considered likely to represent a true positive, while groups with fewer members may be rejected on the grounds that they are more likely to represent false positives.

2.2 Bayesian importance search and marginalization

In the general case, a face detection system is designed to scan the entire possible space of face region parameters uniformly unless we have prior knowledge of the statistics of those face parameters in the input image. It can be regarded as a maximization of the posterior distribution of the parameters using the uniform prior distribution (see figure 2). However, in the case of video face detection, we can assume temporal continuity of face regions as well as that of input image frames in general (see figure 3). Avidan (Avidan, 2001) introduced the first combined approach which uses the output of an SVM object detector to perform a tracking task. Okuma (Okuma, et al., 2004) proposed an SMC based algorithm which combined the AdaBoost detector and a color-based object tracker. The system can successfully track multiple targets from a given video sequence, but does not use the classifier directly as a likelihood function.

Consider the situation where data is given as a video sequence. Let D be image data at the current (t^{th}) frame and let $D_{1:n} = \{D_1, D_2, ..., D_n\}$ be the image data set up to the current frame. Several approaches regard the output of the face classifier $F_S(\boldsymbol{u}_t ; D_t)$ as a confidence score for a face candidate $\boldsymbol{u}_t$. The likelihood function in this work is defined by using the calibration technique of Platt's scaling (Platt, 1999)

$$P(D_t | \boldsymbol{u}_t) = \frac{1}{1 + \exp\left(-\beta F_S(\boldsymbol{u}_t ; D_t) + \gamma\right)} \tag{2}$$

Using Bayes' theorem we obtain straightforwardly a recursive formula for $P(\boldsymbol{u}_{0:t}|D_{1:t})$, the joint posterior distribution of a series of face region parameters $\boldsymbol{u}_{0:n} = \{ \boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_n \}$ given the data:

$$P(\boldsymbol{u}_{0:t} \mid D_{1:t}) = \frac{P(D_t \mid \boldsymbol{u}_t) P(\boldsymbol{u}_t \mid \boldsymbol{u}_{t-1}, D_{1:t-1})}{P(D_t \mid D_{1:t-1})} P(\boldsymbol{u}_{0:t-1} \mid D_{1:t-1}) \tag{3}$$

with $P(D_t \,|D_{t\text{-}1}) = P(D_1)$ at $t = 1$. Consider a *proposal distribution* π such that draws

$$\boldsymbol{u}_{0:t}^{(i)} \sim \pi(\boldsymbol{u}_{0:t} \,|\, D_{1:t-1}), \quad i = 1,\ldots,N \tag{4}$$

can be obtained by standard methods, and define the *importance weights*

$$w_{0:t}^{(i)} = \frac{P(D_{1:t} \,|\, \boldsymbol{u}_{0:t}^{(i)}) P(\boldsymbol{u}_{0:t}^{(i)} \,|\, D_{1:t-1})}{\pi(\boldsymbol{u}_{0:t}^{(i)} \,|\, D_{1:t-1})}, \quad i = 1,\ldots,N \tag{5}$$

Then Sequential Monte Carlo approximates the joint posterior distribution by

$$\hat{P}(\boldsymbol{u}_{0:t} \,|\, D_{1:t-1}) = \sum_{i=1}^{N} \tilde{w}_{0:t}^{(i)} \delta\left(\left\| \boldsymbol{u}_{0:t} - \boldsymbol{u}_{0:t}^{(i)} \right\|\right) \tag{6}$$

$$\tilde{w}_{0:t}^{(i)} := \frac{w_{0:t}^{(i)}}{\sum_{j=1}^{N} w_{0:t}^{(j)}} \tag{7}$$

In order to perform Monte Carlo evaluation of the one-step marginal likelihood, assume that the draws at the previous step

$$\boldsymbol{u}_{0:t}^{(i)} \sim P(\boldsymbol{u}_{0:t-1} \,|\, D_{1:t-1}), \quad i = 1,\ldots,N \tag{8}$$

are available. Use a stochastic dynamics for the parameters $P(\boldsymbol{u}_t \,|\boldsymbol{u}_{t-1}, D_{1:t-1})$ to generate $\{\tilde{u}_t^{(i)}\}_{i=1}^{N}$ and evaluate

$$\hat{P}(D_t \,|\, D_{1:t-1}) = \sum_{i=1}^{N} P(D_t \,|\, \tilde{\boldsymbol{u}}_t^{(i)}) \tilde{w}_{0:t-1}^{(i)} \tag{9}$$

If we design a sequential proposal distribution of the form

$$\pi(\boldsymbol{u}_{0:t} \,|\, D_{1:t-1}) = \prod_{s=1}^{t} \pi(\boldsymbol{u}_s \,|\, \boldsymbol{u}_{0:s-1}) \pi(\boldsymbol{u}_0) \tag{10}$$

then, instead of drawing entire sample trajectories at each time by (8), we only need to draw one-step samples

$$\boldsymbol{u}_t^{(i)} \sim \pi(\boldsymbol{u}_{0:t} \,|\, \boldsymbol{u}_{0:t-1}^{(i)}, D_{1:t-1}) \tag{11}$$

which significantly reduces computational costs. A sequential proposal distribution also leads to sequential importance weights:

$$w_{0:t}^{(i)} = \frac{P(D_t \,|\, \boldsymbol{u}_t^{(i)}) P(\boldsymbol{u}_t^{(i)} \,|\, \boldsymbol{u}_{0:t-1}^{(i)}, D_{1:t-1})}{\pi(\boldsymbol{u}_t \,|\, \boldsymbol{u}_{0:t-1}^{(i)}, D_{1:t-1})} w_{0:t-1}^{(i)}, \quad i = 1,\ldots,N \tag{12}$$

We predict the region for the jth face candidate group G_j by the marginal posterior mean of the parameters at time t :

$$\hat{\boldsymbol{u}}_t^{[j]} = \frac{\int_{\boldsymbol{u}_t \in G_j} \boldsymbol{u}_t P(\boldsymbol{u}_t \,|\, D_{1:t}) d\boldsymbol{u}_t}{P(D_t \,|\, D_{1:t-1})_{G_j}} \cong \frac{\sum_{\boldsymbol{u}_t \in G_j} \boldsymbol{u}_t^{(i)} \tilde{w}_{0:t}^{(i)}}{\sum_{\boldsymbol{u}_t \in G_j} \tilde{w}_{0:t}^{(i)}} \tag{13}$$

It should be noted that the Monte Carlo marginalization can be implemented simply by discarding those components that are not of interest. This is one of the advantages of Monte Carlo methods in general, and Sequential Monte Carlo in particular.

2.3 Change detection and re-initialization

In many applications including broadcasting, input video sequences can contain discontinuities corresponding to scene changes. Since the approach described above assumes continuity between the current video frame D_t and the past series of frames $D_{1:t-1}$, we need a change detection framework to re-initialize the probability distributions at scene changes.

We propose the *sequential marginal likelihood* (Matsumoto & Yosui, 2007) $P(D_t| D_{1:t-1})$ as representing the degree of continuity of the given sequence $D_{1:t}$. If the marginal likelihood suddenly drops, it is natural to suppose that an unexpected phenomenon on the input sequence, like a scene change, may have occurred. We therefore reset the probability distribution of $\boldsymbol{u}_t$ to beflat if the sequential marginal likelihood $P(D_t \mid D_{1:t-1})$ for all face candidate groups G falls below a threshold ε:

$$\pi(\boldsymbol{u}_t \mid \boldsymbol{u}_{0:t-1}^{(i)}, D_{1:t-1}) = \begin{cases} \mathcal{N}(\boldsymbol{u}_{t-1}^{(i)}, \Sigma) & \text{if } {}^{\exists}G_j; P(D_t \mid D_{1:t-1})_{G_j} > \varepsilon, \\ 1/|\Omega| & \text{otherwise,} \end{cases} \tag{14}$$

where Σ denotes the 3x3 covariance matrix of the Gaussian distribution $\mathcal{N}$ and Ω denotes the size of the entire parameter space of $\boldsymbol{u}_t$.

2.4 Experiments

We evaluated the performance of the proposed algorithms, compared with a face detection algorithm provided by the OpenCV libraries[11]. As a test image sequence, we selected 500 frames from a video sequence from the TRECVID 2007 development data, a collection of broadcast videos. The development data provides a wide variety of broadcast programs to train content-based retrieval systems for the TRECVID contest, at QVGA resolution (320x240) with the frame rate of the PAL video format (25Hz). We selected the "BG_15190" video sequence, which contains many frontal faces at relatively large sizes. We constructed the test sequence from the first 10 frames of each of the first 50 shots, giving 500 frames containing 450 faces. We entered ground truth data by hand for all frontal and semi-frontal faces. Evaluation of the detection results was performed against the ground truth data, allowing position and size errors of up to 10% of the true face size. To simply compare each search strategy in face region parameter space, we used the same core object detector function from OpenCV as the face classifier $F_S(\boldsymbol{u}_t ; D_t)$, and used the same runtime data file for frontal faces. For the baseline algorithm, we tried 10 settings for the threshold η (= {1, 2, . . . , 10}). For the proposed algorithm, we tried 10 settings for the threshold ε (= {0.2, 0.4, . . . , 2.0}), and fixed the number of Monte Carlo samples at N = 500.

Figure 4 and figure 5 show performance curves (precision vs recall, and F-measure vs detection speed) for the baseline algorithm and the proposed algorithm. Figure 4 shows that the average speed of the proposed approach was roughly double that of the baseline scheme for all settings, without sacrificing detection performance.

Table 1 shows details of the best results in terms of F-measure for each approach. Figure 5 shows temporal variations in detection time per frame for the proposed method with ε = 1.2. Spikes in the detection time are due to enlargement of the parameter search space corresponding to re-initialization when the lower alternative in equation (14) is selected. The frequency of such spikes on the trajectory corresponds to the 10-frame interval at which discontinuities occurred in the test sequence (constructed from 50 different shots of length 10 frames).

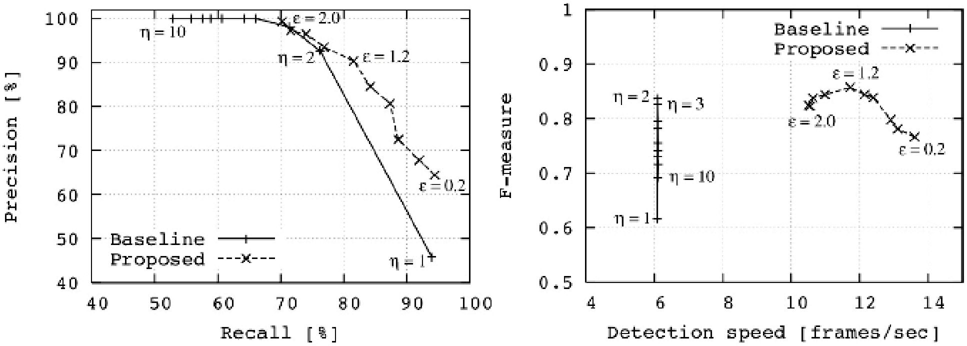


Fig. 4. Face detection performance of both methods with various thresholds (Left: precision vs recall, Right: F-measure vs detection speed).

Search scheme	Baseline ($\eta = 2$)	Proposed ($\varepsilon = 1.2$)
Recall	76.2 % (343/450)	81.6 % (367/450)
Precision	92.7 % (343/370)	90.4 % (367/406)
F-measure	0.84	0.86
frame rate	6.1 fps	11.7 fps

Table 1. Parameters at best (largest F-measure) result for each algorithm.

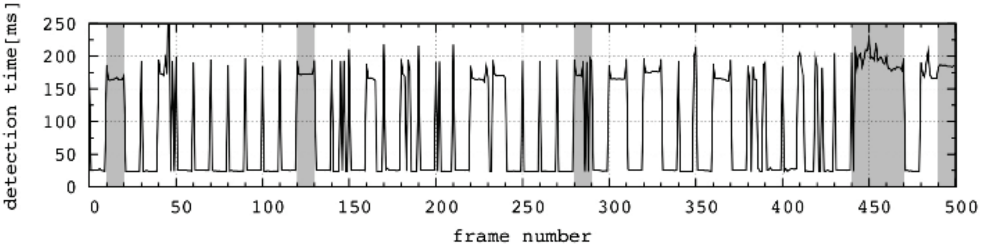


Fig. 5. Trajectory of detection time for proposed method (ε = 1.2), performed on a 3.8 GHz Xeon CPU. The grey regions indicate the input frame with no face.

3. Applications in Broadcasting

Many applications of face detection technologies have appeared in the last few years, and digital still camera featuring the technologies have achieved commercial success. However, there are still few applications for video sequences. In the following sections, we introduce two particular applications of video face detection in broadcasting.

3.1 Video Correction Support System

In this section, we describe a video correction system for broadcast video materials. The system automatically detects human faces in the input video sequence as target regions, then estimates an appropriate set of correction parameters for the white/black/gamma levels of the detected regions. Figure 6 shows an overview of the system, constructed from 5 modules: color corrector, 4-split multi viewer, face detection PC, correction control PC, and GUI terminal.

The 4-split multi viewer encodes the input video signal as a sequence of JPEG images and broadcasts them to the two PCs, locally connected with Ethernet cables. Then the face detection PC detects all possible face candidate regions and passes them to the correction control PC. For each detected face region, the correction control PC evaluates the average luminance, and estimates the optimum luminance transformation so as to achieve a natural face tone.

The correction control PC also evaluates a histogram of luminance over the whole input image, and sets black/white/gamma levels so as to achieve appropriate contrast and dynamic range for broadcasting. The set of video correction parameters are smoothed using a moving average filter, so as to eliminate over-sensitive responses to transient changes like flashing, and to eliminate possible errors in the face detection stage. Then the color corrector transforms the input video signal according to the set of smoothed correction parameters, as determined by the correction control PC. Using the GUI terminal (see figure 7), operator(s) can check a set of sample images from the input/output video signals. The GUI also provides adjustment/preset functions for each system parameter, including weighting factors for the whole image and detected face regions as the targets of video correction processes.

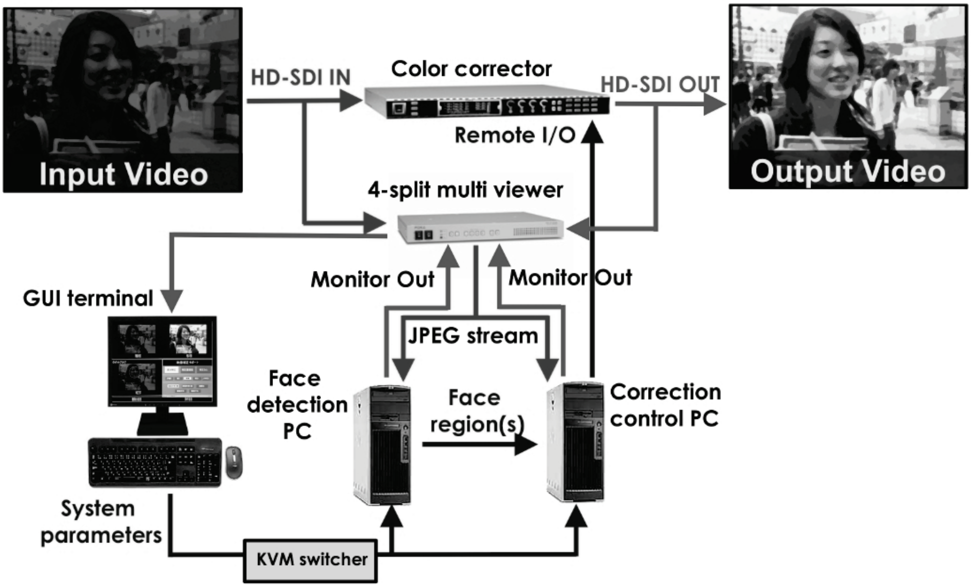
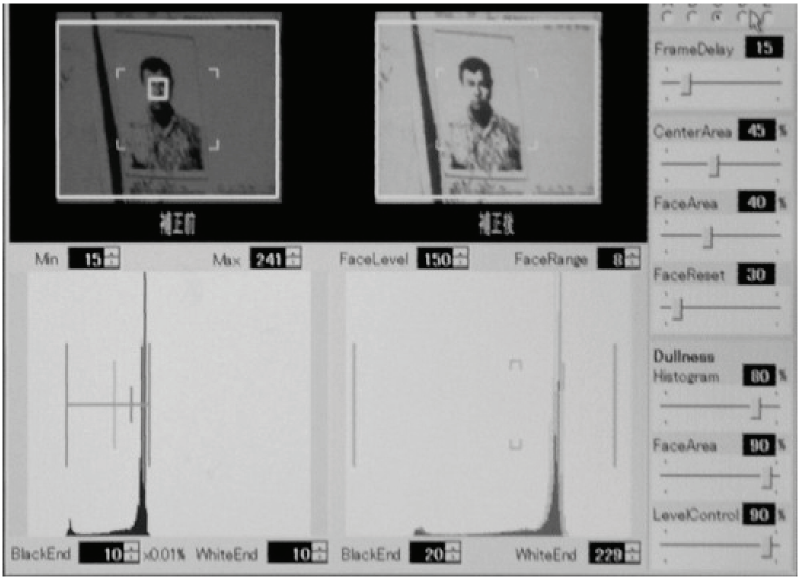


Fig. 6. Overview of the video correction system.



FrameDelay 15
CenterArea 45 %
FaceArea 40 %
FaceReset 30
Dullness
Histogram 80 %
FaceArea 90 %
LevelControl 90 %

Min 15 Max 241 FaceLevel 150 FaceRange 8

BlackEnd 10 ×0.01% WhiteEnd 10 BlackEnd 20 WhiteEnd 229

Fig. 7. Example of the GUI screen in parameter adjustment/presetting mode.

3.2 Facial Occlusion Spotting

In many cases, a broadcast image sequence is assembled from several video materials, with various visual effects. In this work, we assume a standard video effect, in which some video materials are superimposed on a base image. The main purpose of this section is to spot occlusions, in which the main broadcast subjects are overlapped completely or partially by the superimposed video materials.

Figure 8 shows an overview of our occlusion spotting system. This system is constructed from two parts: a face detection system and an occlusion detector.

Although there are various different situations where occlusions can arise in broadcast scenes, we focus on human faces as a major object of interest in broadcasting. We adopt the following three criteria for a target scene to be detected:

i. Size of the face region is significant.
ii. Spatial ratio of the overlapped area is significant.
iii. The occlusion spans several contiguous frames.

The first criterion reflects a general trend in broadcasting that main image objects are bigger than other incidental objects. The second criterion relates to the degree of occlusion. It reflects a natural assumption that an occlusion is troublesome if it hides more than a certain proportion of the whole face region. The third criterion is for temporal consistency. It aims to exclude accidental occlusions and noise due to camera work and/or movement of the face itself. We show a particular application for a broadcast news program, detecting face occlusions using our face detection technologies and an occlusion detector based on the three criteria described above.

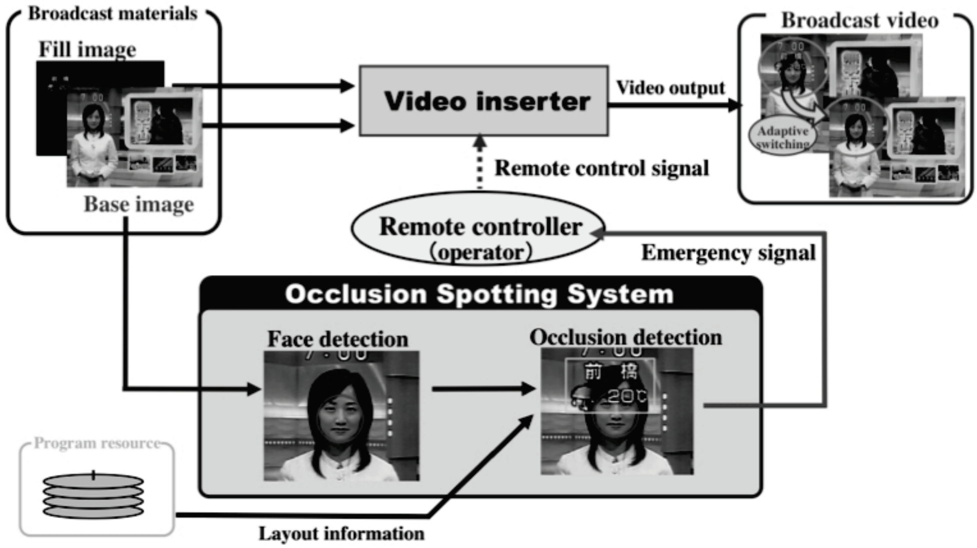


Fig. 8. Architecture of the facial occlusion spotting system.

Occlusion Spotting

Assume a set of face regions $\{ \Omega_i \, ; i \in \mathrm{I} \}$ in an input image D_t at frame t, i.e. regions where the output of a face detector is +1.

According to criteria (i) and (ii), we define the degree of occlusion (occlusion score) for input image D_t as follows:

$$V_t := \max_{i \in I} \left[\lambda_a \phi\left(\frac{r_i}{r_0} \, ; \rho_a \right) + \lambda_b \phi\left(O_i \, ; \rho_b \right) \right] \tag{15}$$

$$\phi(x\,;\rho) = \begin{cases} 0 & \text{if } x \le 0 \\ x/\rho & \text{if } 0 < x < \rho \\ 1 & \text{otherwise} \end{cases} \tag{16}$$

where $\phi(x; \rho)$ is a score function parameterized by ρ, and the weighting parameters λ_a and λ_b satisfy the following equation:

$$\lambda_a + \lambda_b = 1 \tag{17}$$

Here r_0 is a standard radius for face regions in the given input images. O_i stands for the degree of spatial overlap between the i^{th} face region Ω_i and the target area $\Omega_0 = \{(x, y) : x_R < x < x_L, y_T < y < y_B\}$ to be superimposed. The first term of equation (14) embodies criterion (i), and the second term criterion (ii). As shown in equation (15), we assume that the occlusion score would saturate if the size of face region were large enough, and the overlapping factor were large enough.

We define an importance factor for a pixel (x, y) in the i^{th} face region as a Gaussian distribution, $G(x,y;\, \Omega_i)$. Then we define an overlapping factor O_i as the integral of the distribution $G(x,y;\, \Omega_i)$ inside the target area $0 < O_i < 1$ (see figure 9).

Smoothing of Occlusion Score

From criterion (iii), we apply a smoothing filtering process to the trajectories of occlusion scores, V_t . As a criterion for occlusion, we define the alarm level at time t, U_t . Observing the general trend of temporal occlusions due to pan/tilt movement of the camera and isolated false positive errors in the face detector, we define using a Median filter with buffer length τ.

$$U_t := \text{MED}\ [V_{t-\tau+1},\ldots,V_t] \tag{18}$$

where $0 < U_t < 1$ since V_t is bounded by [0, 1]. We set $V_t = 0$ ($t < 0$), $t = 2n+1$ ($n \in Z+$). The system assumes there is a continuous occlusion when U_t exceeds a threshold , and outputs warning signals. Therefore, the system response U_t is delayed relative to V_t by (τ-1)/2. A delay of several frames is acceptable because human operators can not respond in less than several hundreds of milliseconds in an occlusion spotting task.

Figure 10 shows trajectories of V_t and U_t for the proposed scheme. We show examples of true positive and true negative scenes in figures 11 and 12, respectively. Some unstable responses at around $570 < t < 590$ on the trajectories of figure 10 were caused by continuous missing of face regions in some critical scenes, such as scenes in which the target person is walking and shifting his/her head pose. These errors are mainly caused by face detection errors due to pose variations in the broadcast video and insufficient robustness of the face detector we used. We expect that the robustness can be improved if we use a variety of face pose data at the training stage of the face detector, and detectors tuned to multiple poses.

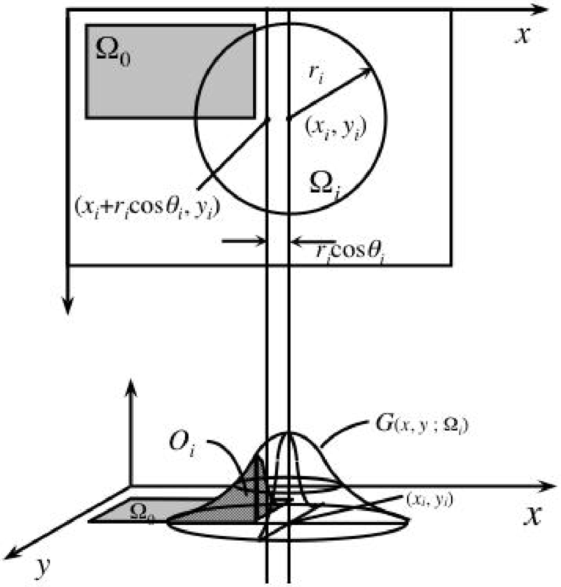


Fig. 9. Importance factor for pixel (x, y) in i^{th} face region.

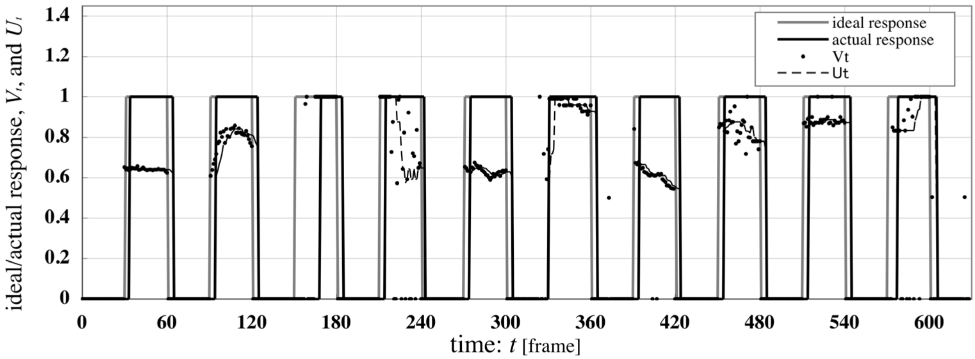


Fig. 10. Trajectories of ideal response, actual response, V_t, and U_t of the system.

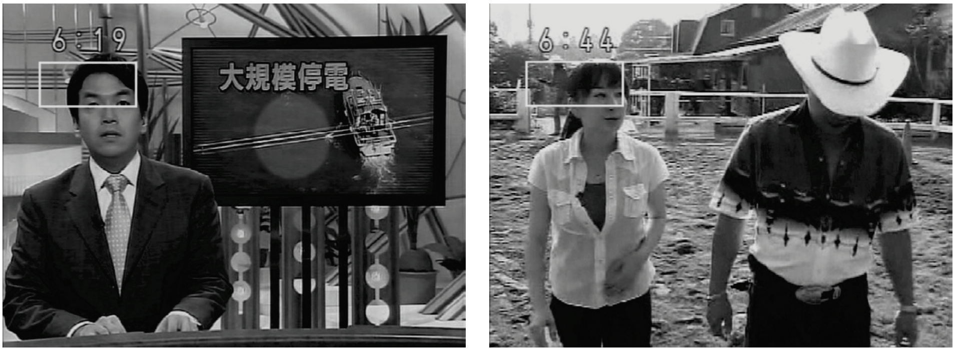


Fig. 11. Examples of true positives.

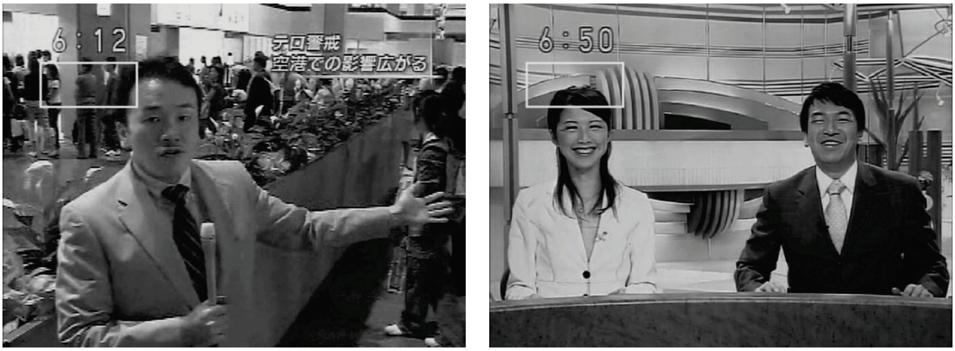


Fig. 12. Examples of true negatives.

4. Conclusion

In this paper we introduced a Bayesian online learning approach, Sequential Importance Search, for face detection in video which dramatically boosts detection speed for sequential input images. The online learning approach successfully prunes a considerable amount of the search space of face region parameters, and can automatically reset its prediction process in response to discontinuities (cuts) in the input video stream. Experimental results showed the efficiency of the proposed method in comparison to the conventional exhaustive search algorithm, and demonstrated its automatic re-initialization process, Bayesian change detection, at each scene boundary in the input sequence of a simulated broadcasting program compiled from the TRECVID video data. The re-initialization process rests on the sequential model marginal likelihood from the online learning process, which can be derived naturally from the methodology of hierarchical Bayesian inference.

We also introduced two possible applications of our new face detection algorithm, in a video correcting system, and an occlusion spotting system. Although there still remain open issues, these technologies offer the prospect of improved performance in various practical engineering tasks in broadcasting.

5. References

Avidan, S. (2001). Support Vector Tracking, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR2001), Vol.1, pp.184-191, 2001.

Doucet, A. (1998). On Sequential Simulation-Based Methods for Bayesian Filtering, CUED/F-INFENG/TR-310, University of Cambridge, 1998.

Freund Y. & Schapire, R. E. (1997). A Decision-theoretic Generalization of On-line Learning and an Application to Boosting, Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119-139, 1997.

Lienhart R. & Maydt J. (2002). An Extened Set of Haar-like Features for Rapid Object Detection", Proceedings of ICIP2002, vol. 1, pp. 900-903, 2002.

Matsumoto T. & Yosui, K. (2007). Adaptation and Change Detection with a Sequential Monte Carlo Scheme, IEEE Transaction on Systems, Man, and Cybernetics, vol. 37, no. 3, pp. 592-606, 2007.

Okuma, K.; Taleghani, A.; Freitas, N. de.; Little, J. J. & Lowe, D. G. (2004) A Boosted Particle Filter: Multitarget Detection and Tracking, Proceedings of ECCV2004, LNCS 3021, pp. 28-39, 2004.

Papageorgiou, C.; Oren, M. & Poggio, T. (1998). A General Framework for Object Detection, Proceedings of ICCV '98, pp. 555-562, 1998.

Platt J. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, Advances in Large Margin Classifiers, MIT Press, 1999.

Viola, P. & Jones, M. (2001). Robust Object Detection using a Boosted Cascade of Simple Features, Proceedings of CVPR2001, vol. 1, pp. 511-518, 2001.

16

3D Human Posture Estimation Using HOG Features of Monocular Images

Katsunori Onishi, Tetsuya Takiguchi & Yasuo Ariki
Kobe University
Japan

1. Introduction

The accurate estimation of the 3D configurations of complex articulated objects from monocular images [5] has been widely studied. Once the technology is perfected, there will be potential applications in many fields related to human posture and kinematic information, such as computer interfaces with gesture input, interaction with robots, video surveillance, and entertainment. However, this problem is extremely challenging due to the complicated nature of human motion and information limitations associated with 2D images.

Various methods focus on human posture estimation. There are methods to extract features from images, based on the structure of the human body, for example, using skin color or facial position [3]. However, they impose restrictions on features, such as clothes and orientation. There are other methods to extract silhouettes and edges from images as features [1, 6, 7]. Many methods represent human images using body silhouettes. This representation has the advantage of containing strong cues for posture estimation while being unaffected by changes in appearance and lighting. However, they rely on the stable extraction of the silhouettes and edges, and they are weak in regard to self-occlusion. To solve these problems, it is necessary to extract features inside the silhouettes, being independent of skin color or orientation. HOG [2] was originally proposed as features to express the shape of an object, but it is also effective for human posture estimation from the above viewpoint.

In this chapter, we propose an appearance-based approach to estimate human posture using HOG features, which can describe the shape of the object. The method does not depend on clothes and orientation under noisy conditions, so 3D human posture can be estimated stably. However, the dimension of the extracted HOG features vector is usually high in the background region because the HOG features are computed over the entire image. To solve this problem, we also propose a method to reduce feature dimension in the background regions using principal component analysis (PCA) on every HOG block. Using the proposed methods, 3D human posture can be estimated by linear regression of HOG features.

It was confirmed that our method worked effectively for real images, and the experimental results show that our method reduces the RMS estimation error compared to the conventional method (shape contexts).

2. Features

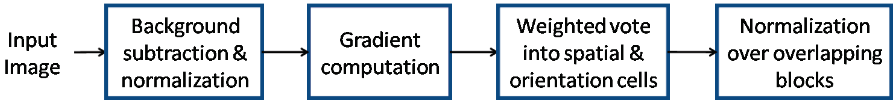

Input Image
Background subtraction & normalization
Gradient computation
Weighted vote into spatial & orientation cells
Normalization over overlapping blocks

Fig. 1. The flow of feature extraction

This section describes the HOG features extracted from an image and the structure for representing the 3D human model. Moreover, this section describes the method to reduce the dimension of the HOG features vector in the background region using PCA on every block. Fig. 1 shows the flow of HOG features extraction.

2.1 Histograms of Oriented Gradients

HOG [2] and SIFT [4] were proposed for gradient-based features for general object recognition. HOG and SIFT describe similar features. The difference is that SIFT describes the features at the candidate location (keypoint), while HOG describes the features over the given region. This means that HOG can represent the rough shape of the object as shown in Fig. 2.

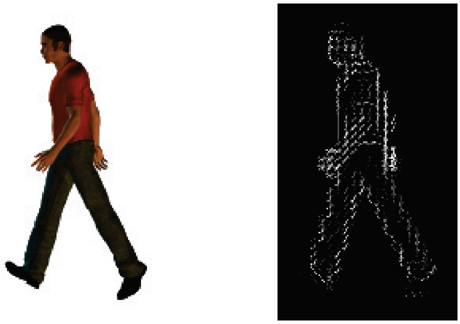


Fig. 2. Input image (left) and image represented by HOG features (right)

2.1.1 Gradient computation

Before extracting the HOG features, the human region has to be detected using the background subtraction method on the input image. The image size is normalized at this time, and the human region is located in the central position on the image. Then the image gradient is computed as follows.

$$\begin{cases} f_x(x,y) = I(x+1,y) - I(x-1,y) & \forall x,y \\ f_y(x,y) = I(x,y+1) - I(x,y-1) & \forall x,y \end{cases} \tag{1}$$

where f_x and f_y denote x and y components of the image gradient, respectively. $I(x,y)$ denotes the pixel intensity at position (x,y) . The magnitude $m(x,y)$ and orientation $\theta(x,y)$ are computed by

$$m(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2} \tag{2}$$

$$\theta(x,y) = \tan^{-1}(f_y(x,y) / f_x(x,y)) \tag{3}$$

In order to make the HOG features insensitive to the clothes and facial expressions, we use the unsigned orientation of the image gradient computed as follows.

$$\tilde{\theta}(x,y) = \begin{cases} \theta(x,y) + \pi & \text{if} \quad \theta < (x,y) < 0 \\ \theta(x,y) & \text{otherwise} \end{cases} \tag{4}$$

2.1.2 Orientation histograms

The gradient image is divided into cells ($c_w \times c_h$ pixels) as shown in Fig. 3. For each cell, the orientation $\tilde{\theta}(x,y)$ is quantized into c_b orientation bins, weighted by its magnitude $m(x,y)$ to make histogram. That is, the histogram with the c_b orientations is computed for each cell.

2.1.3 Block normalization

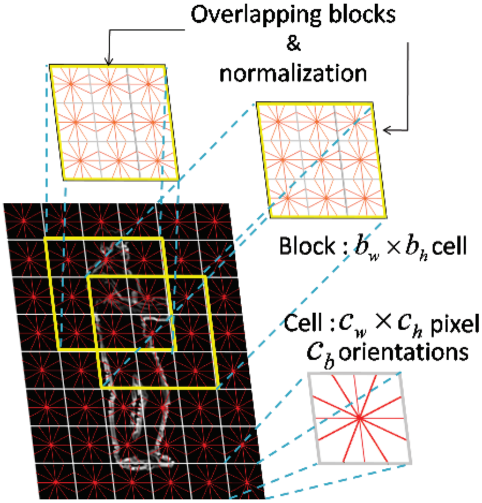


Fig. 3. Block normalization

Fig. 3 shows the orientation histogram extracted for every cell and the larger spatial blocks with $b_w \times b_h$ cells. Since a cell has c_b orientations, the feature dimension of each block is $d_b = b_w \times b_h \times c_b$ for each block. Let v denote a feature vector in a block, and h_{ij} denote the unnormalized histogram of the cell in the position (i,j), $\{1 \le i \le b_w, 1 \le j \le b_h\}$ in a block. The feature vector of a certain block is normalized as follows.

$$h'_{ij} = \frac{h_{ij}}{\sqrt{\|v\|^2 + \varepsilon}} \qquad (\varepsilon = 1) \tag{5}$$

Since the normalization is done by overlapping the block, the histograms h$_{ij}$ are repeatedly normalized by different blocks.

2.2 Dimension reduction using block-based PCA

A HOG features vector usually has high dimension even in the background region because the gradients are computed over the entire image. Since the features are required inside the human region, the features in the background region should be removed for 3D human posture estimation.

For this purpose, PCA is carried out for every block using training data. The gray value in the background region is almost constant, although it includes noises, because background subtraction is already performed as preprocessing. Therefore, a lot of feature dimensions in the background region can be reduced by PCA. Conversely, number of features in the human region cannot be reduced too much because their values change in various ways. Therefore, the human region has a lot of feature dimensions, and in the background region is reduced as shown in Fig. 4.

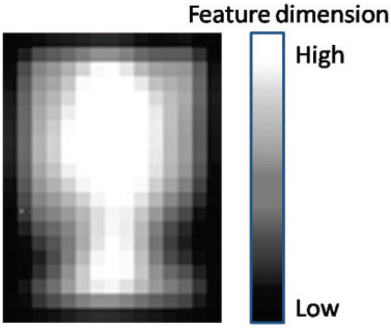


Fig. 4. Dimension reduction from block-based PCA

2.3 Structure of 3D human model

Humans are regarded as multi-joint objects that transform into various shapes. However, the segment part which connects two joints can be regarded as rigid. Therefore, it is possible to express a 3D human model by joint angles. That is, in order to express the posture of a 3D human model, the values of joint angles are important.

Let y denote the vector composed of the angles at joints (elbow, waist, knee, etc.) of the 3D human model. Various postures can be expressed by changing these joint angles. The y has 24 (3×6+1×6) dimensions for the joint angles (except for joints like a finger), as shown in Fig. 5.

The various postures are expressed by estimating these joint angles from the input image.

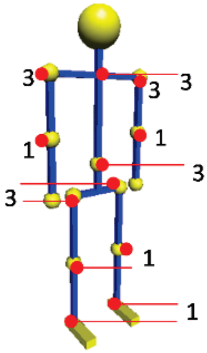


Fig. 5. Structure of 3D human model

3. Regression-based approach

This section describes the method for estimating 3D human posture from image features. Regression analysis is employed to estimate the posture as used in [1]. The relation between the HOG features vector $x \in \mathbb{R}^d$ and 3D human model vector $y \in \mathbb{R}^m$ is approximated by the following formula.

$$y = Ax + \varepsilon \tag{6}$$

A is the $m \times d$ matrix, and ε is the residual error vector. The 3D human posture is estimated by converting the input image feature x to the 3D human model vector y. In training the model (estimate A), a set of n training pairs $\{(y_i, x_i) \mid i = 1, \cdots, n\}$ is given (in our case, 3D poses and the corresponding image HOG features). The conversion matrix A is estimated by minimizing the least mean square error. Packing the training data into an $m \times n$ 3D posture matrix $Y \equiv (y_1, y_2, \cdots, y_n)$ and $d \times n$ image feature matrix $X \equiv (x_1, x_2, \cdots, x_n)$, the training is performed as follows.

$$A := \arg\min_A \|AX - Y\|^2 \tag{7}$$

In testing, the 3D human pose vector y is estimated by converting HOG features vector x using the computed conversion matrix A. Fig. 6 shows the regression-based estimation method.

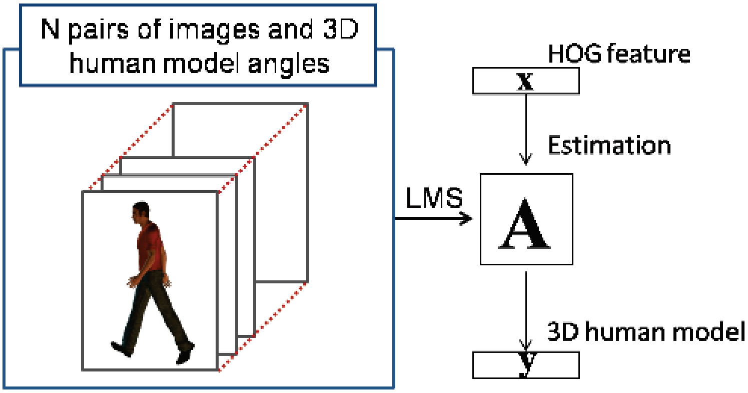


Fig. 6. Regression-based estimation method

4. Experiment

In this section, we show the results of our proposed method in comparison to the conventional method, which utilizes shape context descriptors [6] extracted from silhouettes.

4.1 Data and ground truth

Images were taken with a monocular camera with a resolution of 640×480 pixels, as shown in Fig. 7. A standing human body was rotated horizontally. Images were taken from 8 directions at intervals of 45 degrees using a fixed camera. Five actions (standing, hands raised, arms open, walking, running) were taken continuously in each direction. We manually assigned joint angles to each posture beforehand, and the estimation result was evaluated by RMS error.

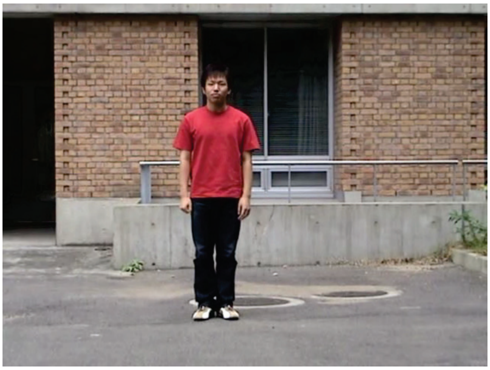


Fig. 7. A sample image that was taken for the experiment

For training data, 30 images were used in each direction, for 5 postures in total. For test data, 123 images were used; image (a) under the same condition as the training data, image (b) under various conditions, and image (c) downloaded from http://www.nada.kth.se/

~hedvig/data.html. The images used are summarized in Table 1.

Posture	The number of images			
	Training data	Test data		
		(a)	(b)	(c)
Standing	16	8	8	0
Hands raised	40	8	8	0
Arms open	24	8	8	0
Walking	80	16	16	11
Running	80	16	16	0

Table 1. The number of images

The image size was normalized to 150×200 using the background subtraction method. The values of HOG parameters were c_w=10, c_h=10, c_b=9, b_w=3, b_h=3. In computing the HOG features vector, the block was moved cell by cell. Because 234 blocks were made from an image, the dimension of the HOG features was 18,954. PCA was carried out for every block to reduce the 81 dimensions until the 98% cumulative proportion of the HOG features was achieved. The dimension of the HOG features was reduced to 8,998 as a result of computing block-based PCA .

4.2 Experimental result

It was confirmed that our method worked effectively for a real image. The results of the comparison experiment are shown in Fig. 8.

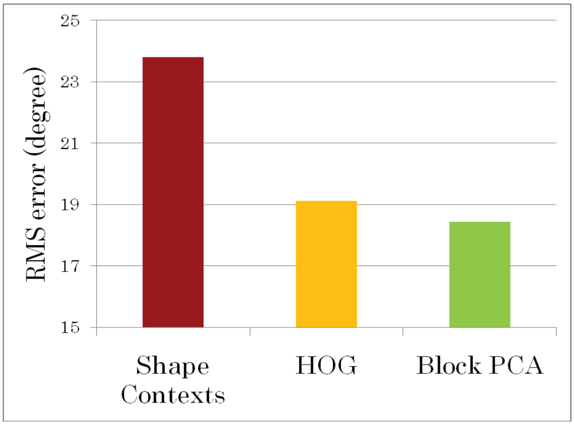


Fig. 8. Comparison experiment results

Our method reduces the RMS estimating error by 5.35 degrees compared to the conventional method (shape contexts). Concerning the silhouette images, the limbs were sometimes ambiguous due to self-occlusion. However, in the HOG features, since it takes the internal edge into consideration, the posture differences can be distinguished so that the error decreased, as shown in Fig. 9. In addition, HOG after PCA at each block can improve the RMS error by 0.68 degrees compared to the original HOG.

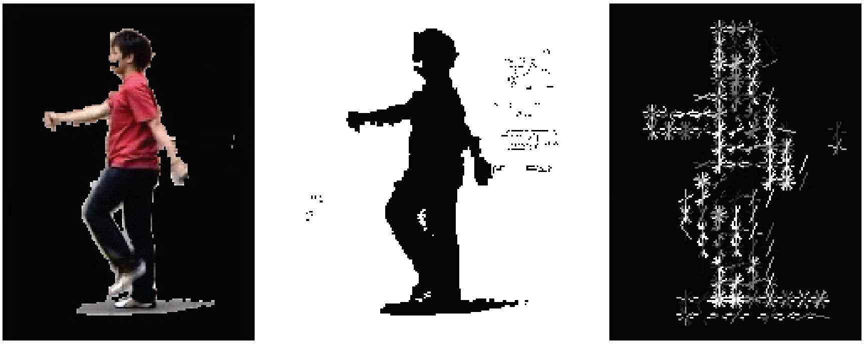


Fig. 9. Images walking leftward. The left image is the input image, the middle image is the silhouette image, and the right image is the HOG features image.

The dimension reduction of PCA was decided according to the cumulative proportion rate. In Fig. 10, the RMS error vs. the cumulative proportion rate (the number of feature dimensions) is plotted. As shown in Fig. 10, the optimum cumulative rate (98%) was selected in our experiments.

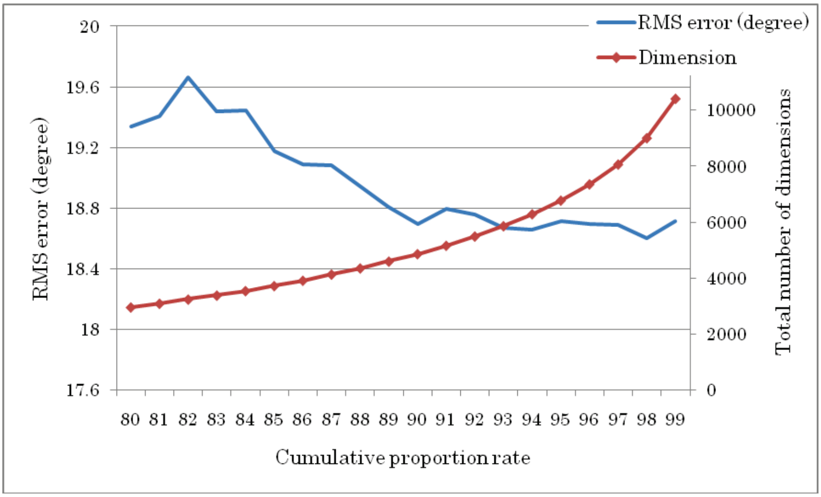


Fig. 10. Results of dimension reduction using block-based PCA. The RMS error decreased most when the cumulative proportion rate was 98%.

Next, the evaluation results of postures are shown in Fig. 11. The conventional method (shape contexts) showed a small error in the standing posture. This is because noises occurred when the human moved quickly. In a case of stationary posture, such as standing, was little noise in an image, so it was stabilized, and the human silhouette was extracted accurately.

However, the purpose is to estimate not only standing but various other postures as well. With this in mind, our method can be said to be effective, as shown in Fig. 8 when considered all postures.

Fig. 12 shows examples of the results of 3D posture estimation.

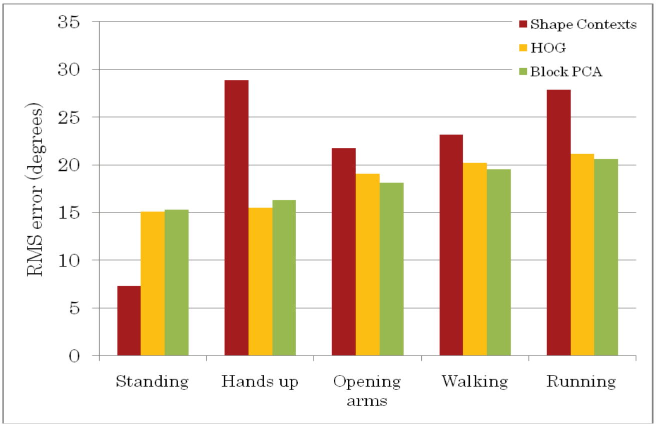


Fig. 11. Estimation result of postures

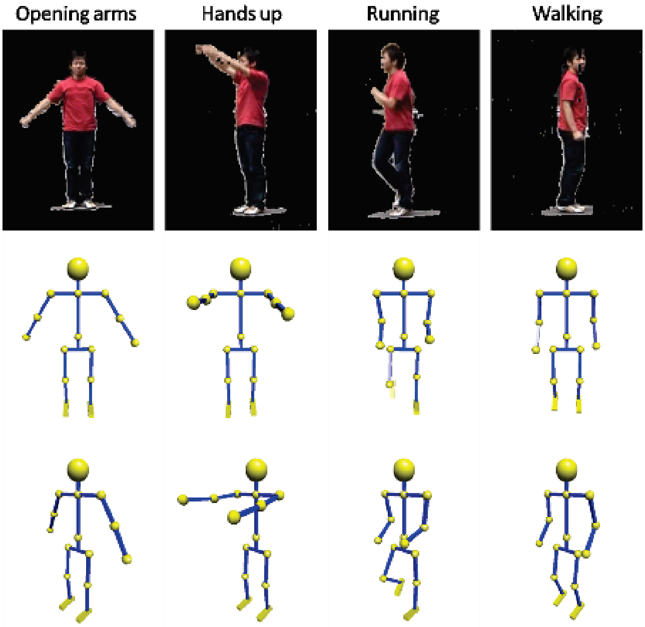


Fig. 12. Reconstructed sample postures

7. Conclusion

We described a method for estimating 3D human posture from a monocular image. In this chapter, we proposed the use of HOG features (which can be extracted without dependence upon clothes and orientation) and reducing the feature dimension in the background region by PCA for every block. In future research, human detection with HOG features will be integrated with our method without using background subtraction.

8. References

[1]A. Agarwal and B. Triggs. 3D Human Pose from Silhouettes by Relevance Vector Regression, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR 2004, pp. 882-888, Washington, DC, USA, July 2004

[2]N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection, *IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, CVPR 2005, pp. 886-893, San Diego, CA, USA, June 2005

[3]M. Lee, I. Cohen. A Model-Based Approach for Estimating Human 3D Poses in Static Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, TPAMI, Vol.28, No.6, pp.905-916, June 2006

[4]David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision,* CVIU, Vol.60, No.2, pp.91-110, 2004

[5]Thomas B. Moeslund and Erik Granum. A Survey of Computer Vision-Based Human Motion Capture. *International Journal of Computer Vision and Image Understanding,* IJCV, Vol.81, pp.231-268, 2001

[6]G. Mori and J. Malik. Recovering 3D Human Body Configurations using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* TPAMI, Vol.28, No.7, pp. 1052-1062, June 2006

[7]C. Sminchisescu, A. Kanaujia, D.N.Metaxas. BM^{3}E : Discriminative Density Propagation for Visual Tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, TPAMI, Vol.29, No.11, pp.2030-2044, June 2007

17

Frequency Shifting for Emotional Speaker Recognition

Yingchun Yang, Zhenyu Shan and Zhaohui Wu
College of Computer Science and Technology, Zhejiang University
P.R.China

1. Introduction

The task of automatic speaker recognition is to determine the identity of a speaker by machine(Herbert & Michael, 1994 and Bimbot et al., 2004), which is the one of the main applications in pattern recognition. According to different application fields, it encompasses speaker verification and speaker identification. Speaker verification is to verify a person's claimed identity from his/her voice, and gives a result "Yes" or "No" (Joseph & Campbell, 1997). In speaker identification, there is no a priori identity claim, and the system decides who the person is, what group the person is a member of, or that the person is unknown.

In general, the process of speaker recognition has two steps: feature extraction and classification. The process of feature extraction is to extract features from the utterance for training or testing. Many effective features are found in researches, such as MFCC(Rivaral et al., 1996 & 1999), LPC(Philippe &, 1995), PLP(Hermansky, 1990), Pitch(e.g. Atal, 1969) and Energy. The classification includes training and testing. Before testing, the recognition system must get familiar with the voices. Thus, in training, the system needs to collect speech of the registered person. Then in testing, the system compares an unidentified utterance to the trained model and makes identification or verification. The popular classification methods(or models) include GMM(Reynolds & Richard, 1995), GMM-UBM(Reynolds et al., 1999), SVM(e.g., Wan & Campbell, 2000), HMM(Tishby, 1991), and DBN(Sang et al., 2003) .

Although great progress has been made in last 20 years, there are still many problems for speaker recognition system in real applications. The recognition performance is reduced by many factors(Sadaoki 1997), such as background noise(e.g. Ming et al., 2008), channel effects(e.g. Reynolds, 2004) and emotion variability. In the paper(Shan et al., 2006), a speaker check-in system is evaluated in an office with 38 users(25 males, 13 female, aged from 20 to 38) for 13 months. In the system, MFCC and GMM are applied as feature extraction method and modelling method. About 34% of the recognition errors are caused by noise and 35% by un-described reasons, in which emotion variability is the main factor.

Speaker emotion variability means the emotion states mismatch between the training and testing speech, and such kind of recognition is called emotional speaker recognition or affective speaker recognition. Different emotion states will affect speech production mechanism of a speaker in different ways, thus lead to acoustical changes in his/her speech

(Scherer, 1998 and 2000). As concluded in (Louis, 2003 and Scherer, 2003), the emotion factor will change more than 20 types of speech features, such as pitch, Intensity, formants, speech rate, energy and duration. These changes can induce intra-speaker vocal variability, which will aggravate the recognition performance.

The remainder of this chapter is organized as follows. In Section 2, the problem of emotional speaker recognition is analyzed and some methods are introduced. In Section 3, we proposed the frequency shifting method for emotional speaker recognition, including modelling and feature extraction method. In Section 4, an emotional speech corpus MASC is described briefly. In Section 5, the performance evaluation for the shifting method is given. Discussions and future works are drawn in Section 6.

2. Emotional Speaker Recognition

This section will give a review of some methods and related work in emotional speaker recognition. According to the difference of emotion states of the training and testing utterances, the emotional speaker recognition system often works in two situations. The first one is that both the training and testing data include the utterances with the same emotion states, though its performance is still worse than the situation when the emotion state is neutral(Scherer et al., 2000 and Wu W. et al.,2006). The other is that the emotion state of testing utterance is excluded in the training utterance. It usually occurs in the real application, where users often provide neutral speech in training yet various emotional utterances in testing. The system in this situation performs even worse than in the first one(Wu et al.,2006; Shan et al.,2007 and Huang et al.,2008), and it attracts our focal attention.

The reason for poor performance is the mismatching between training and testing conditions. It also exists in other recognition applications, such as noise and channel effect. Some methods has been proposed to overcome this limitation. For example, for solving noise or channel effect, the compensation method, based on the relationship between different channels(Wu et al., 2006) or noise types, is applied to eliminate the negative effect. Under noisy surroundings, some rules are presented to see whether one frame is contaminated by noise and the clean part is used for testing. The idea of these two methods is helpful to emotional speaker recognition. However, emotion has its particularities. For instance, the deformation of voice by the same channel is fixed thus has its own rule, and the effects of emotion on voice deformation are more complex, for different person and text will affect the degree of deformation.

The mismatch in emotional speaker recognition contains three conditions. In the first condition, the pattern of testing data is not contained in training speech. For example, the testing utterances contains sad speech, and the training utterances not, and the recognition system can't match the pattern of the utterance in testing. In the second, the training data contains all the pattern in the testing utterance, but the pattern is not matched extract. Owning to the difference of emotional intensity and depth, the pattern may not be similar even if the emotion state is the same. In some cases, the mismatch happens, where the testing data is mixture of happy and neutral utterance. In the third condition, only neutral speech can be gathered in training and various emotion utterances exist in testing. In the ordinary, the emotion state of testing data is also unknown. It may need to identity the emotion state in testing and the recognition performance will be affected by the emotion identification rate. It is the most common condition in real application. And in the following, we primarily review the methods for solving this type problem.

In our conclusion, the method of solving the mismatch problem has three ideas. The first idea is to extract the feature which is not varied by the emotion variation, which is the most effective way. Nevertheless it is the hardest, because recent experiment result shows that all types of features for speaker recognition (as we known) will be affected by emotion variation. For a brief explanation, the emotion varies the frequency spectrum in a complex way and the common features for recognition are extracted from the frequency spectrum. The second method tries to find a relation between different emotion states, including its feature, model and scores. The aim is to build a relationship between neutral utterance and other emotional utterance. Then, this relationship is adopted to transfer neutral speech to emotion speech. We named this method emotion enriching. The third method is to collect the neutral part from the emotion utterance for testing. It often needs to compute the emotional factor to define the emotion part, which is named emotion regulation. In the following of this section, we will introduce the recent presented methods from the second or third idea.

Zheng(Bao et al., 2007 & Wu et al., 2006) considered that emotion effect is somewhat similar to that of channel effect on speaker recognition. They realized that when training and testing speech are in different emotions, the discrepancy between the speaker models and the test utterances will induce biases in verification scores. Thus, the ideas of handling the channel effect was borrowed to alleviate the negative effect of emotion(Bao et al., 2006). In their former work, an emotion-dependent score normalization method(E-norm) (Wu et al., 2006)derived from H-norm(Reynolds et al., 2000) was proposed, which was originally designed to alleviate channel effect in cross-channel speaker verification. It is designed to estimate these emotion-dependent biases from development data. Then, the biases are removed from verification scores. Soon afterwards, they furthered their approach by proposing emotion attribute projection(EAP) method(Bao et al., 2007), which removes emotion variability from SVM expansion dimensions. The idea is borrowed from the nuisance attribute projection (NAP), which has been proven to be a successful channel compensation method(Campbell et al., 2006 and Solomonoff et al., 2005). It can achieve a better result because it removes the subspace that may cause the emotion variability in the kernel of an SVM system.

Huang (Huang et al., 2008) studied the difference between various emotions and propose a method based on the Pitch-dependent difference detection and modification (PDDM). The basic idea is to choose the neutral part of testing utterance for final scoring. The pitch is used as the emotion factor to distinguish "neutral" part from emotion utterance. First, it classifies the test utterances into HD (High Different from neutral, and with high identification error rate) and LD (Low Different from neutral, and with low identification error rate) group. Then, it modifies the segments with intense emotional information in the HD utterance to reduce the impact of unmatched emotion states in the training and testing.

In most circumstances, the emotional element, if there is any, always lasts ephemeral when users provide testing utterance, since they tend to provide neutral speech for testing in real application(as discussed in (Shan et al., 2006). The testing utterance is a mixture of neutral and emotion speech, and it is common in emotional speaker recognition. The paper(Shan et al., 2009) experimentally analyzed the performance of the GMM-based verification system with the utterances in this situation and two results are concluded. One is applying the test utterances with low emotion ratio, which suggests the user to provide pure neutral speech for testing or eliminating the non-neutral part from the testing speech. The other is increasing the testing utterances length, yet it can't be satisfied in most applications. Thus, a simple is proposed method to distinguish the non-neutral features from the neutral ones in the scoring processing of testing with the purpose to reduce the emotion ratio of testing utterances. First, all features' scores and the average of these scores are calculated. Second, the scores higher than the average are selected for the final score computation. In other words, not all features are effective in the score computation thus some of them are treated as non-neutral features and eliminated to decrease the emotion ratio.

In about ten years ago, Scherer(Scherer, 2000) presented a structured training approach which aims at making the system get familiar with the emotion variation of the user's voice in training. In this method, the registered speakers are asked to provide speech on various emotional states. It doesn't fit the common situation as we described above. However, it gives a direction for eliminating the emotion effect: the question is that how to synthesis emotion speech, or how to obtain the emotion feature, or how to train the emotion model, or how to calculate the scores of the feature vectors against his/her emotion model, when only the natural utterance is obtained in training.

In the paper(Wu Z.H. et al., 2006), it was found the main reason that causes the performance degradation is the absence of emotion speech. Analysis of emotional speech and its synthesis rules have been researched for many years. It is a way to synthesize emotion speech based on these rules. They investigate the rules based feature modification for robust speaker recognition with emotional speech. A feature modification rule is developed to convert neutral speech to emotion speech. First, the rules of prosodic features modification are learned from a small amount of the content matched source-target pairs. Then, features with emotion information are adapted from the prevalent neutral features by applying the modification rules. Finally, the converted features are trained together with the neutral features to build the speaker models. The speaker models are trained with both the neutral speech provided by the users and generated output speech perceived as conveying emotions.

In order to train the emotion model from his/her neutral model, Shan(Shan et al., 2007) presented a neutral-emotion GMM transformation algorithm, which is based on the assumption: if two speakers' natural speech space satisfies the similar distribution, so does their emotion speech space, especially when they share the same culture. First, the KL-distance between neutral GMMs (order n) is calculated to find out k speakers from the emotion database, who have the similar feature space with the registered speaker. Then, these speakers' emotion GMM is transformed to the registered speaker's emotion GMM. However, the order of the transformed GMM is n^k times of the original one and leads to an exponential increase in computation cost. To overcome this limitation, a neutral-emotion GMM transformation algorithm was presented based on the same assumption as above. In this method, the transformation function is defined by a polynomial function(Shan et al., 2008). It establishes a relationship between neutral and emotion GMM. Applying this relationship, the emotion GMM is obtained from the neutral GMM with the same order. The model is adopted in the emotional speaker recognition with increasing less cost of training and testing. However, these methods demand that the system is aware of all the emotion states of the testing utterances in training. And only the emotion states in testing dataset are considered. If the utterance with new emotion state is added into the testing dataset, the model needs to be retrained for this emotion state.

In following sections, we present the frequency shifting method for emotional speaker recognition. In this new method, only neutral speech is required in training, and the testing speech is in multi-emotion conditions and the emotion state is not available. To model the speech of multi-emotion conditions, we introduce the multi-condition model, which has been successfully used in speech recognition to account for varying noise sources or speaking styles. The researches (e.g., Lippmann et al., 1987 and Scherer, 2000) indicate that it can improve the system robustness when the training data sets are in multi-conditions, thus we believe it is suitable for the multi-emotion conditions. In further, the maximum scoring method is proposed which takes the emotion occurrences of testing utterances into consideration. Thus the adversely effect of the mismatch of emotion state in training and testing can be weakened. In order to generate training speech with multi-emotions, we proposed a method to synthesize emotion utterances by shifting the spectrum of neutral speech. The energy distribution of neutral utterance is changed to create utterances with different energy distribution, which are used as emotion utterances for training. Comparative experiments carried on the MASC database(Wu, T., 2006) show that the new method is superior to the baseline system. The frequency shifting method combining with the maximum scoring method improve the system robustness.

3. Frequency Shifting Method

In this section, the frequency shifting method is introduced in two parts: model and emotion speech synthesis.

3.1 Model

In general speaker recognition, the training data set (Φ_0) only contains neutral utterance of speaker S , which is represented by the likelihood function $p(X|S,\Phi_0)$ of feature vector X associated with speaker S trained on data set Φ_0 . The Gaussian Mixture Model (GMM) is a preferable likelihood function(Reynolds et al., 1995), which is a weighted sum of Gaussian components. Each speaker S is indicated by a GMM λ . Generally, Expectation-Maximization (EM) algorithm is applied in the training process to find out the parameters of λ to maximize $p(X|S,\Phi_0)$ with respect to Φ_0 .

In emotional speaker recognition, training speech with various emotion states should be created from Φ_0 at first. This leads to augment training sets $\Phi_1,\Phi_2,...\Phi_l...,\Phi_L$, where Φ_l denotes the l^{th} emotion speech and L is the number of emotion states. The direct modeling method is to apply the structured training method to employ all the emotional data sets to train a likelihood function $p(X|S,\Phi_0,\Phi_1,...,\Phi_L)$. It demands that the length of the training utterance of each emotion state is almost the same; otherwise its performance will be affected, when the specific emotion state of the testing utterance takes small proportion in training. According to overcome this limitation, we introduce the multi-condition model(Atal, 1974), which is formed by combining the likelihood functions trained on the individual training set:

$$p(X \mid S) = \sum_{l=0}^{L} p(X \mid S,\Phi_l) P(\Phi_l \mid S) \tag{1}$$

where $p(X \mid S,\Phi_l)$ is the likelihood function of frame vector X trained on set Φ_l , and $P(\Phi_l \mid S)$ is the prior probability for the occurrence of l^{th} emotion speech of speaker S . As $P(\Phi_l \mid S)$ is fixed, maximizing $p(X \mid S)$ in training is to maximize $p(X \mid S,\Phi_l)$ individually.

In testing, the first is to calculate the posterior probability of speech given feature vector X with respect to speaker S :

$$p(S \mid X) = \frac{p(X \mid S) P(S)}{P(X)} \tag{2}$$

where the equation is based on Bayes' rule. By replacing $p(X \mid S)$ in Equation (2) with Equation (1) and assuming an equal prior $P(S)$ for all the speakers, we obtain an operational version of Equation (2) for recognition.

$$p(S \mid X) = \sum_{l=0}^{L} p(X \mid S,\Phi_l) P(\Phi_l \mid S) \tag{3}$$

And the posterior likelihood of a speaker given an utterance $X^T = \{X_1, X_2, ..., X_T\}$ with T frames is defined as:

$$p(S \mid X^T) = \frac{1}{T} \sum_{t=1}^{T} \log p(S \mid X_t^T) \tag{4}$$

where $p(S \mid X_t)$ is defined by Equation (3).

In our assumption, the emotion state of feature vector X is not available, so the value of $P(\Phi_l \mid S)$ can't be decided. If each emotion state occurrences in testing is the same, $P(\Phi_l \mid S)$ has the equal value for each emotion state. Equation (3) can be simplified:

$$p(S \mid X) = \underset{l=0}{\overset{L}{mean}}\, p(X \mid S,\Phi_l) \tag{5}$$

In this equation, the average of $p(X \mid S,\Phi_l)$ is used for scoring, so it is named the mean scoring method. Since all of $P(\Phi_l \mid S)$ is the same, it neglects the effect of the emotion state of feature vector X . In our opinion, the emotion information of testing speech may overcome the degradation of the performance caused by different emotion states of testing and training speech. In testing, it needs to find the most likely emotion state for a given feature vector X . One feature vector only belongs to one emotion condition. Thus, the value of $P(\Phi_l \mid S)$ for a feature vector X is defined as:

$$P(\Phi_l \mid S) = \{ \begin{matrix} 1, \textit{ if the emotion state of } X \textit{ is } l \\ 0, \textit{otherwise} \end{matrix} \tag{6}$$

As $p(X\,|\,S,\Phi_l)$ in Equation (1) describes the distribution of both the speaker and emotion feature space, a simple method based on $p(X\,|\,S,\Phi_l)$ is proposed. The object is to find which emotion state has the maximum posteriori probability to feature X . Formally,

$$l = \arg\max p(S,\Phi_l\,|\,X) = \arg\max \frac{p(X\,|\,\Phi_l,S)p(X)}{p(\Phi_l,S)} \approx \arg\max p(X\,|\,\Phi_l,S) \tag{7}$$

where the second equation is due to Bayes' rule. Combining Equation (6) and Equation (7), Equation (3) can be reduced to:

$$p(S\,|\,X) = \max p(X\,|\,S,\Phi_l) \tag{8}$$

According to the deduction, the final result relates to the maximum of $p(X\,|\,S,\Phi_l)$, so it is named the maximum scoring method.

3.2 Emotion Speech Synthesis

In the last subsection, we assume only neutral speech Φ_0 can be obtained in training. To adopt the multi-condition model in multi-emotion conditions, the training sets $\Phi_1,\Phi_2,...,\Phi_L$ should be generated from neutral speech Φ_0 . The method of converting neutral speech to emotion speech will be described in this section.

Much effort has been devoted to analyze the relationship between neutral and emotion speech, and many qualitative relations have been published. But it is troublesome to quantitatively analyze the relationship, because it is not only speaker dependent but also text dependent. Thus, it is hard to convert neutral speech to emotion speech exactly in the text independent speaker recognition. However, the research on acoustic changes benefits us to transform neutral speech to emotion speech.

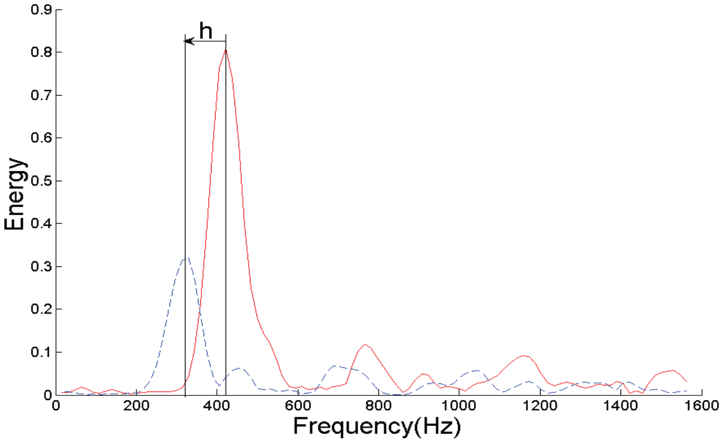


Fig. 1. The spectrum of certain frame vector of anger and neutral speech (voicing [e]). The dashed and solid line represents the anger and neutral spectrum individually.

Nowadays, the research on the acoustic changes is mainly in the frequency domain. The variation of formats represents the difference between neutral and emotion speech. Take anger speech for example, the average of F0, F1 and F2 of anger speech is smaller than neutral speech. Figure 1 shows the spectrum of certain feature vector extracted from anger and neutral utterance (voicing [e]). It indicates the anger speech can be shifted from neutral spectrum to change the formats. Hence, the emotion spectrum can be converted from the neutral spectrum by shifting its spectrum, though it is not precise.

In the multi-conditions model, the aim is to synthesize all kinds of emotion speeches while the emotion state of synthesized speech is not concerned. Thus, we are even unaware of that in which direction and how long to shift the neutral spectrum. The training sets $\Phi_1, \Phi_2, ..., \Phi_L$ can be synthesized by changing the spectrum of neutral speech Φ_0 . The spectrum shifting method is proposed:

$$E_E(f) = E_N(f + h) \tag{9}$$

where E_E and E_N is the energy in frequency f of emotion and neutral speech individually, and h is the length of shifting. According to this, the "emotion" speech can be obtained from the neutral speech.

The synthesized speech can't be categorized as a specific emotion state, only indicating the formats variation of neutral speech. In fact, it also changes the pitch and the energy distribution of the frame. In the structured training method, the aim is to synthesize all kinds of emotion utterances while the emotion state of synthesized speech is not concerned. Thus, the shifting method is suitable for obtaining the training dataset with different emotion states.

The MFCC is most widely used frequency domain feature in speaker recognition. Adding the spectrum shifting method to MFCC extraction, it can extract emotion features from neutral speech with the different value of h . As shown in Figure 2, the process in the dashed rectangle is added and other steps are the same with the original.

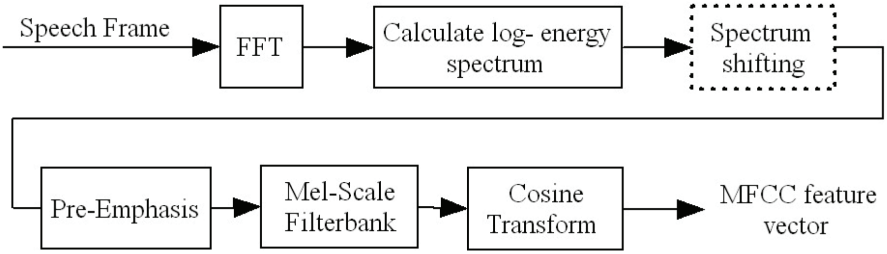


Fig. 2. The process of MFCC and the crewel frame is added to extract emotion features from neutral speech.

4. Emotional Database

An emotional speech database MASC (Mandarin Affective Speech Corpus) is used in our experiments(Wu, T. et al., 2006), which is available publicly. It is constructed with two major objectives. On one hand, it is used for prosodic and linguistic investigation of emotion expression in Mandarin. On the other hand, it supplies a training set as well as a test data set for speaker recognition system affected by emotional factors. Compared with other emotional speech database, MASC concentrates on showing both the characteristics of different emotional states and the intra-speaker variability caused by state changes of speakers. In particular, these records are spoken in Mandarin which is a fairly fresh area of emotional speech studies.

The selection of emotional states is expected to put the speech on the emotion wheel which has been derived from the Plutchik's work (Plutchik, 1980). It is a model to describe the activation-evaluation-power space of emotion. According to this emotion model, emotional states distribute on a circle which is named Emotion Wheel. The centre of this circle stands for the natural origin, a state which gathers all kinds of emotional factors. However, the effects from these emotional factors are so weak that they cannot emerge at the origin. Each emotional state is defined with a unique planar vector $\vec{E}$ that has two parameters, emotional intensity and emotional orientation. Emotional intensity indicates the range of $\vec{E}$ and emotional orientation renders the angle of $\vec{E}$. In terms of the emotion wheel, four emotion types are selected in the corpus: anger, elation, panic and sadness, whose descriptions are consulted by Banse & Scherer (1996).

- Neutral - Simple statements without any emotion.
- Anger - A strong feeling of displeasure or hostility.
- Elation - Be glad or happy because of praise.
- Panic - A sudden, overpowering terror, often affecting many people at once.
- Sadness - Affected or characterized by sorrow or unhappiness.

The corpus contains recordings of 68 native speakers (23 female and 45 male) and five kinds of emotion states: neutral, anger, elation, panic and sadness. Each speaker reads 5 phrases, 20 sentences three times for each emotion state and 2 paragraphs only for neutral. These materials cover all the phonemes in Chinese. The sentences include all the phonemes and most common consonant clusters in Mandarin. The types of sentences are: simple statements, a declarative sentence with an enumeration, general questions (yes/no question), alternative questions, imperative sentences, exclamatory sentences, special questions (wh-questions). The sampling rate of the utterances used in the experiments is 8000Hz. More details can be found in (Wu, T. et al., 2006).

5. Experiments

5.1 Evaluation of the Shifting Method

In this subsection, an experiment is designed to compare the speakers’ emotion speech with the ones synthesized by the spectrum shifting method. The object is to see their similarity. The experiment employs five phrases read for three times of all speakers from the MASC. The neutral and emotion phrase with the same text and same speaker is compared, so the comparison is text dependent and speaker dependent.

First, the speech is segmented into frames by a 20-ms window progressing at a 10-ms frame rate. Secondly, FFT of the size 512 is used to transform the speech frames into frequency domain. Then, spectrogram shifting method is applied to obtain synthesized spectrums. The length unit of shifting is 8000/512=15.625Hz. And in the experiment, the shifting length h is ±1, ±2, ±3 ±4, ±5 unit. Thus, 10 phrases are created from one neutral phrase. Finally, reverse FFT transforms them into time domain to get emotion utterance.

Here, the similarity of two phrases is indicated by the DTW(Sakoe, 1978 and Joseph 1997) distance. The distances from speaker's emotion phrase to the neutral one and to the synthesized phrases are calculated. For each emotion state to neutral, there is 5*3*68 = 1020 times of distance calculation. Because the number of synthesized phrases of one neutral phrase is 10, so its distance is based on DTW:

$$D = \min_{h}(DTW(X_E \,||\, X_N^h)) \tag{10}$$

where X_E and X_N^h represent emotion speech and synthesized speech by shifting h HZ individually.

The result is shown in Figure 3. The distance from synthesized phrase to emotion phrase is about half the distance to neutral one. The synthesized is more similar to speaker's emotion speech than neutral. It indicates the synthesized speech can make the system familiar with the emotion speech in a certain way.

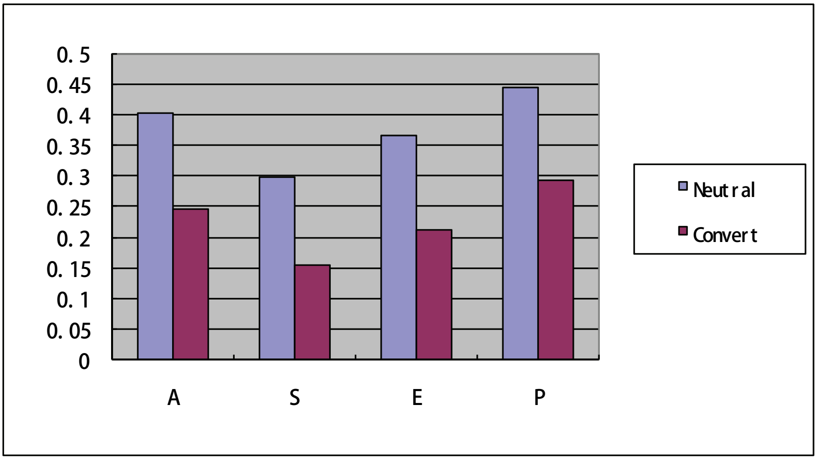


Fig. 3. The comparison of the distances from speaker's emotion phrase to the neutral one and to the synthesized phrases. Note: A : Anger, S : Sadness, E : Elation, P : Panic.

5.2 Experiment Strategies

In the following experiments, the speech is segmented into frames by a 20-ms window progressing at a 10-ms frame rate. An energy-based voice activity detector is used to remove silence. Then, the 13-dimension MFCCs are extracted from the speech frames. The final performance is evaluated by identification rate (IR), which is computed as the percent of correctly identified sentence over all testing sentences.

Only the sentences(2-10s) of the corpus are used in the experiments. All sentences in MASC are divided into two parts. The first 5 sentences read for three times for each speaker are used for training, named training data set. And the left 5*3*15*68=15300 are for testing.

Six speaker identification experiments are designed to evaluate the performance. In the first experiment, only the neutral sentences are used to train a GMM with 32 order and the scores is calculated as Equation(2). Because the emotion mismatch between the training and testing speech, the result is the lower limit of the speaker identification performance. In the second, all the sentences in training data set are used to train a GMM with 160 orders and the scoring method is the same as the first experiment. It aims to evaluate the structured training approach. The training and testing speech in the third and fourth is the same with the second experiment. The difference is that they apply the multi-condition model with 32 order GMM for each. The mean scoring method and maximum scoring method are adopted individually. These three methods are design to evaluate the performance of multi-condition model and maximum scoring method. The training and testing sentences in the fifth and sixth experiment are the same as the first one. The multi-emotion training data is generated by the frequency shifting method to evaluate its performance. Their training and testing methods are the same as the third and forth one individually. The experiment strategies are summarized in Table 1.

5.3 Results

	Experiment Strategies				Results
	Train dataset	Test dataset	Scoring Method	Frequency Shifting	IR (%)
1	Neutral	ALL	Eq.2	No	45.17
2	ALL	ALL	Eq.2	No	63.54
3	ALL	ALL	Eq.4	No	58.71
4	ALL	ALL	Eq.7	No	77.76
5	Neutral	ALL	Eq.4	Yes	47.49
6	Neutral	ALL	Eq.7	Yes	54.43

Table 1. The strategies and results of the experiments.

The details of identification rate are shown in Table 1. Compared with the first experiment, the IR of the second, third and forth one is increased 12% at least. It indicates the system performance can be improved if the training data contains all kinds of emotion states. And it is important to make the system get familiar with the emotion speech in training.

The multi-condition model combining with the maximum scoring method (77.76%) exceeds the structured training method (63.54%), with an increasing of 14.22%. The multi-condition model serves to model the speech of multi-emotions. And the maximum scoring method can alleviate the negative effect of unavailable of the emotion state of testing speech. However, it is surprising that the structured training method outperforms the multi-condition model combining with mean scoring method (58.71%). The possible reason is that the length of speech with each emotion state is almost same in the experiment which is suitable for the structured training method.

The use of shifting method improves the IR increased from 45.17% to 54.43%. It shows the shifting method indeed improves the performance when there is only neutral speech in training. The synthesized speech is helpful to make the system get familiar with the emotion speech. However, it can't achieve the result of forth experiment, in which all kinds of emotion states can be obtained. In summary, our emotional speaker recognition method based on the multi-condition model is more robust, when there is only neutral speech in training and various emotion utterances in testing.

6. Discussions & Conclusions

This chapter introduces the emotional speaker recognition, and presents a frequency shifting method. The shifting method combined with MFCC is adopted for feature extraction. It converts neutral speech to emotion speech for emotional speaker recognition. The extracted emotion feature is applied with maximum scoring method for emotional speaker recognition. In our experiments, the synthesized emotion utterance by shifting frequency spectrum proves more similar to emotion speech than the neutral is. The results demonstrate the shifting method combined with maximum scoring method is effective to improve the performance of emotional speaker recognition.

All the methods we introduced above indeed outperform the baseline speaker recognition system in emotional condition. However, it can't achieve the result where neutral speech is used for both training and testing. Even the performance is still worse than the situation where all kinds of emotion states can be obtained in training. It shows that we haven't building an exact relationship between neutral and emotion utterances. More works should be carried out to find the variation rules of different emotion speeches.

7. Acknowledgements

This work is supported by the foundings: NCET-04-0545, NSFC_60525202/ 60533040, 863 Program 2006AA01Z136, PCSIRT0652, ZPNSF Y106705, National Key Technology R&D Program (No. 2006BAH02A01).

8. References

Atal, B.S.(1969). Automatic Speaker Recognition Based on Pitch Contours, Acoust. Soc. Am, Vol 45, Issue 1, pp. 309-309, 1969.

Atal, B.S.(1974). Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, Acoust. Soc. Amer., vol 55, pp. 1304–1312, 1974.

Banse, R.; Scherer, K.R.(1996). Acoustic profiles in vocal emotion expression, Journal of Personality and Social Psychology, vol 70, pp. 614-636, 1996.

Bao, H.J.; Xu, M.X. & Zheng, T.F.(2007). Emotion Attribute Projection for Speaker Recognition on Emotional Speech, InterSpeech 2007, pp. 758-761, 2007.

Bimbot, F.; Bonastre, J.F. & Fredouille, C.(2004). A Tutorial on Text-independent Speaker Verification, Journal on Applied Signal Processing, Vol.4, pp.430-451, 2004.

Campbell, W.M.; Sturim, D.E. and Reynolds, D.A.(2006), SVM based speaker verification suing a GMM supervector kernel and NAP variability compensation, Signal Processing Letters, vol 13, no 5, pp.308-311, 2006.

Hermansky, H.(1990). Perceptual linear predictive (PLP) analysis of speech, The Journal of the Acoustical Society of America, 1990

Herbert, G.; Michael, S.(1994). Text-Independent Speaker Identification, IEEE SIGNAL PROCESSING MAGAZINE, pp.18-32, 1994.

Huang, T.; Yang, Y.C.(2008). Applying pitch-dependent difference detection and modification to emotional speaker recognition, InterSpeech 2008, pp. 2751-2754, 2008.

Joseph, P.; Campbell, J.R.(1997). Speaker Recognition: A Tutorial, PROCEEDINGS OF THE IEEE, VOL. 85, NO. 9, pp.1437-1462, 1997.

Lippmann, R. P.; Martin, E. A. and Paul, D. B.(1987). Multi-style training for robust isolated-word speech recognition," in Proc. ICASSP87, pp. 705–708, 1987.

Louis, T.B. (2003). Emotions, speech and the ASR framework, Speech Communication, vol 40, pp.213–225, 2003.

Ming, J.; Hazen, T. and et al.(2008) Robust Speaker Recognition in Noisy Conditions, IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, 2008.

Philippe, T.; Heinz, H.(1995). Usefulness of the LPC-residue in text-independent speaker verification, Speech Communication, Vol17, pp.145 – 157, 1995.

Plutchik, R.(1980). Emotion: A Psycho evolutionary Synthesis. New York: Harper and Row, 1980.

Reynolds, D.A.; Richard, C.(1995). Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models, IEEE Transactions on Speech and Audio Processing, Vol.1, No.3, pp.72-83, 1995.

Reynolds, D.A.; Quatieri T.F. & Dunn R.B. (2000). Speaker Verification Using Adapted Gaussian Mixture Models, Digital Signal Processing, Vol 10, pp.19–41, 2000.

Reynolds, D.A.(2004). Channel robust speaker verification via feature mapping, ICASSP2004, 2004.

Rivaral, V.; Douglas, S. & Vishwa, G.(1996). Compensated Mel Frequency Cepstrum Coefficients, Proceedings of the Acoustics, Speech, and Signal Processing, pp.323-326, 1996.

Rivaral, V.; Douglas, S. & Azarshid,F.(1999). Generalized Mel Frequency Cepstral Coefficients for Large-Vocabulary Speaker-Independent Continuous-Speech Recognition, IEEE Transactions on speech and audio processing,Vol.7, No.5, pp.525-532, Sep., 1999.

Sakoe, H.; Chiba, J.(1978). Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoustic, Speech, Signal Processing, vol. ASSP-26, no. 1, pp. 43–49, 1978.

Sang, L.F.; Wu Z.H.; Yang Y.C.; Zhang, W.F.(2003). Automatic speaker recognition using dynamic Bayesian network, ICASPP, 2003.

Sadaoki, F.(1997). Recent advances in speaker recognition. Pattern Recognition Letters, vol 18, pp.859-872, 1997.

Scherer, K.R.(2000). A cross-cultural investigation of 40 E-norm emotion inferences from voice and speech: implication for speech technology, Proceedings of ICSLP, 2000.

Scherer, K.R.; Johnstone, T. & Klasmeyer, G.(2000). Can automatic speaker verification be improved by training the algorithms on emotional speech, Proceedings of ICSLP, Vol.2, pp.807-810, Beijing, China, 2000.

Scherer, K.R.; Johnstone, T. & Bänziger, T.(1998). Automatic verification of emotionally stressed speakers: The problem of individual differences, Proc. of SPECOM, 1998.

Scherer, K.R.(2003). Vocal communication of emotion: A review of research paradigms, Speech Communication, vol 40,pp.227–256, 2003.

Shan, Z.Y.; Yang, Y.C. & Ye R.Z.(2007). Natural-Emotion GMM Transformation Algorithm for Emotional Speaker Recognition, InterSpeech, pp. 782-785, 2007.

Shan, Z.Y.; Yang, Y.C.(2008). Learning Polynomial Function Based Neutral-Emotion GMM Transformation for Speaker Recognition, ICPR 2008, pp.236-240, 2008.

Shan, Z.Y.; Yang, Y.C. & Wu Z.H.(2006). SCS: A Speech Check-in System, The 8th International Conference on Signal Processing, pp.752-756, 2006.

Shan, Z.Y.; Yang, Y.C. (2009). Scores Selection for Emotional Speaker Recognition, The 3rd edition of the International Conference on Biometrics, 2009.

Solomonoff, A.; Campbell, W. M.; and Boardman I.(2005). Advances in channel compensation for SVM speaker recognition, in Proceedings of ICASSP, 2005.

Tishby, N. Z.(1991). On the application of mixture AR hidden Markov models to text independent speaker recognition, IEEE Transaction on Acoustic, Speech, Signal Processing, vol. 39, no. 3, pp. 563–570, 1991.

Wan, V.; Campbell, W.M.(2000). Support vector machines for speaker verification and identification, Neural Networks for Signal Processing, 2000.

Wu, T.; Yang, Y.C., Wu, Z.H. & Li, D.D(2006). MASC: A Speech Corpus in Mandarin for Emotion Analysis and Affective Speaker Recognition, ODYSSEY 2006, pp.1-5, June 2006.

Wu, W.; Zheng, T.F., Xu, M.X. & Bao, H.J.(2006). Study on Speaker Verification on Emotional Speech, Proceedings of ICSLP 2006, pp.2102-2105, 2006.

Wu, W.; Zheng T.F. and Xu, M.(2006). Cohort-based speaker model synthesis for channel robust speaker recognition, ICASSP, 2006.

Wu, Z.H.; Li, D.D. & Yang, Y.C.(2006). Rules Based Feature Modification for Affective Speaker Recognition, ICASSP 2006, pp.661-664, May 2006.

Pattern Recognition in Medical Image Diagnosis

Noriyasu Homma
Cyberscience Center, Tohoku University
Japan

1. Introduction

Medical image diagnosis can be conducted through a highly intelligent cognitive process that requires special medical knowledge and experiences. It is not, of course, still completely clear what kind of information is needed and used for the highly intelligent diagnosis, but relatively low level features such as shapes, texture, and other pixel based statics extracted from the images can be used for the diagnosis. In this sense, medical images can be diagnosed, at least partially, by using pattern recognition algorithms. In this chapter, for lung cancer diagnosis by using X-ray computed tomography (CT) images, fundamentals and some advanced techniques of pattern recognition in medical image diagnosis will be studied extensively.

An early stage detection of the lung cancer is extremely important for survival rate and quality of life (QOL) of patients (Naruke et al., 1988). Although a periodical group medical examination is widely conducted by diagnosing chest X-ray images, such group examination is not often good enough to detect the lung cancer accurately and there is a high possibility that the cancer at an early stage cannot be detected by using only the chest X-ray images. To improve the detection rate for the cancer at early stages, X-ray CT has been used for a group medical examination as well (Iinuma et al., 1992; Yamamoto et al., 1993).

Using the X-ray CT, pulmonary nodules that are typical shadows of pathological changes of lung cancer (Prokop and Galanski, 2003) can be detected more clearly compared to the chest X-ray examination even if they are at early stages. This is an advantage of the X-ray CT diagnosis. In fact, it has been reported that the survival rate of the later ten years can reach 90% after the detection at early stages using X-ray CT images (I-ELCAP, 2006).

On the other hand, using the X-ray CT may exhaust radiologists because the CT generates a large number of images (at least over 30 images per patient) and they must diagnose all of them. The radiologists' exhaustion and physical tiredness might cause a wrong diagnosis especially for a group medical examination where most of CT images are healthy and only very few images involve the pathological changes. Therefore, some computer-aided diagnosis (CAD) systems have been developed to help their diagnosis work (Okumura et al., 1998; Lee et al., 1997; Yamamoto et al., 1994; Miwa et al., 1999).

Although these CAD systems can automatically detect pulmonary nodules with a high true positive rate (TP), the false positive rate (FP) is also high. To reduce the FP, several advanced methods such as neural network approaches have been proposed (Suzuki et al., 2003; Nakamura et al., 2005). However, there are still some fundamental problems such as a low discrimination rate for variations of size and positional shift of nodule images. This is because they are still based on so-called low level or simple image recognition mechanisms with pixel oriented features compared to the radiologist's intelligent diagnosis process with more complex features.

In this chapter, to further reduce the FP, we propose new methods to extract and combine novel features from the CT images of pulmonary nodules (Homma et al., 2008). The extraction and combination of new features are motivated by the radiologist's higher level cognitive process in which several features are combined and integrated to conduct precise diagnosis. Simulation results demonstrate the effectiveness of the new features and the combination method for discriminating nodule shadows from non-nodule ones.

The rest of this chapter consists of as follows. Some filtering and feature extraction techniques will be introduced in section 2. To diagnose medical images, the region of interest (ROI) that is the target region of the recognition must be detected first and then some features can be extracted from the ROI for next step of the diagnosis. In section 3, for the extracted feature sets, the principal component analysis can be used to reduce the informational redundancy of the feature sets and to avoid unnecessary computational expensiveness of the classification algorithms. Then the effect of the extracted features on the nodule pattern classification will be evaluated by using the receiver operating characteristic (ROC) analysis. This evaluation will reveal what have been achieved and what cannot be achieved by the classification. To overcome the disadvantage of the classification algorithm, some advanced techniques will be developed in section 4. In section 5, concluding remarks including future perspective of this field will be given.

2. Filtering and Feature Extraction: Fundamentals for Lung Nodule Detection

In general, a discrimination method mainly consists of the feature extraction and pattern recognition techniques. The conventional image features are such as average, variance, and entropy of pixel values (Takizawa et al., 2001). However, they are not very effective and don't directly reflect target shapes in CT images that are one of the most important pieces of information used to discriminate between nodules and non-nodules. Therefore, we first pay attention to extracting a new shape feature that is more effective than conventional ones (Homma et al., 2008).

2.1 Detection of ROI: Morphological filters

First, we use the variable N-quoit filter (Okumura et al., 1998), based on a mathematical morphological technique (Haralick et al., 1987), to detect ROI from the original CT images. Let us consider an original image $I(x, y)$ of the pixel values at position (x, y). To apply the N-quoit filter to the image I, we define two elemental functions, D with a disk domain K_D and R with a ring domain K_R, as follows (Nakayama et al., 1995).

$$D(x_1, y_1) = \begin{cases} 0, & (x_1, y_1) \in K_D \\ -\infty, & \text{otherwise} \end{cases} \tag{1}$$

$$R(x_1, y_1) = \begin{cases} 0, & (x_1, y_1) \in K_R \\ -\infty, & \text{otherwise} \end{cases} \tag{2}$$

where

$$K_D = \left\{(x_1, y_1) \mid x_1^2 + y_1^2 \le r_1^2\right\} \tag{3}$$

$$K_R = \left\{(x_1, y_1) \mid r_1^2 \le x_1^2 + y_1^2 \le r_3^2\right\} \tag{4}$$

r_1, r_2, and r_3 are radii of the disk, internal, and external rings, respectively. Usually, $r_1 = r_3$ and $r_2 < r_3$.

The output of the N-quoit filter, q, is calculated as

$$q(x, y) = h_D(x, y) - h_R(x, y) \tag{5}$$

where h_D and h_R can be defined by using the operator $\oplus$ of the Murkowski's set addition (Haralick et al., 1987)

$$\begin{aligned} h_D(x, y) &= I(x, y) \oplus D(x_1, y_1) \\ &= \max_{(x_1, y_1) \in K_D} \left\{ I(x - x_1, y - y_1) + D(x_1, y_1) \right\} \end{aligned} \tag{6}$$

$$\begin{aligned} h_R(x, y) &= I(x, y) \oplus R(x_1, y_1) \\ &= \max_{(x_1, y_1) \in K_R} \left\{ I(x - x_1, y - y_1) + R(x_1, y_1) \right\} \end{aligned} \tag{7}$$

Using the disk and ring functions, the output $q(x, y)$ results in large for island shadows in the image I, otherwise $q(x, y)$ becomes small. Since the pulmonary nodules often look like small islands in the CT slice images, the filter can effectively detect regions including nodule candidates with high q values.

2.2 Orientation features extraction

To extract features for nodules recognition, we binarize the original images I in the ROI as

$$I_\beta(x, y) = \begin{cases} 1, & I(x, y) \ge \text{mean}(I) + \beta \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

and calculate three conventional features (mean, variance, and entropy of pixels intensity) of the binarized image I_β (Kondo et al., 2000).

Then, we apply a Gabor filter to the binarized image I_β and extract M orientation outputs. The impulse response of the filter is defined as the harmonic function multiplied by the Gaussian function

$$g(x, y, \sigma, \lambda, \gamma, \theta) = \cos\left(2\pi \frac{x'}{\lambda}\right) \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \tag{9}$$

where θ is the angle of orientation, σ is the bandwidth, γ is the aspect ratio, and λ is the wave length, respectively. x' and y' are given by

$$\begin{cases} x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{cases} \tag{10}$$

Orientation features are obtained from the convolution of image $I_\beta(x, y)$ and $g(x, y, \sigma, \lambda, \gamma, \theta)$ as

$$O(x, y) = I_\beta(x, y) * g(x, y, \sigma, \lambda, \gamma, \theta) \tag{11}$$

Fig. 1 shows examples of filtered images of four orientations. Using the new orientation features, the circle-like shadows can be discriminated from the other shapes. This is a promising result because nodule shadows often look like circles. The orientation features involving such circle-shape information can thus be appropriate for the discrimination.

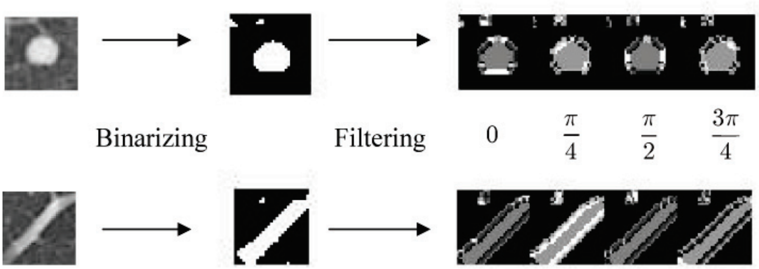


Fig. 1. Examples of four orientation filtered outputs.

For each orientation, we calculate the three features of mean, variance, and entropy of intensity. Consequently, we extract a total $3\times(M+1)$ features from the binarized image. Then we define a vector X of $3\times(M+1)$ features, $X=[x_1,x_2,...,x_{3(M+1)}]^T$, for the image in ROI. To eliminate the dimensional redundancy of the vector, we finally define a feature vector X' from the vector X by using the principal component analysis technique.

3. Nodule Pattern Classification

3.1 Pattern classification in principal component space

We make, respectively, P and Q clusters of nodules and non-nodules images of training data on the principal component feature space by K-means method. The numbers of nodule and non-nodule clusters, P and Q, can be determined automatically on the basis of variance equalization between clusters (Ngo et al., 2002). Then, we project test data X' into the feature space and calculate Euclidean distances between test data and all the cluster centers (Fig. 2). Here any other distances such as the inner product and Maharanobis distance can be used as the similarity measure, but if the variances are almost the same among clusters, then Maharanobis distance are equivalent to Euclidean distance.

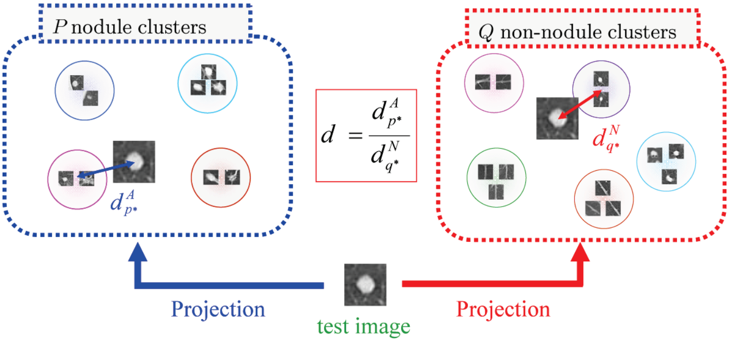


Fig. 2. Distances from the test image to centers of P nodule and Q non-nodule clusters.

Let us consider the $(P+Q)$ distances d_p^A, $p=1, 2, ..., P$, from the P nodule clusters and d_q^N, $q=1,2,...,Q$, from the Q non-nodule ones. The discrimination is conducted by comparing the minimum distances d_{p*}^A, $p* \in \{1, 2, ..., P\}$, from the nearest nodule cluster and d_{q*}^N, $q* \in \{1, 2, ..., Q\}$, from the non-nodule one. That is, if the ratio

$$d = \frac{d_{p*}^{\ A}}{d_{q*}^{\ N}} \tag{12}$$

is less than a threshold α, then the test image can be a nodule candidate; otherwise it is a non-nodule candidate.

3.2 Effect of orientation feature

To evaluate the effect of the orientation feature on the discrimination between nodule and non-nodule images, we have tested the proposed method by using a data set from the Web database of CT images (NICA, https://imaging.nci.nih.gov/ncia/faces/baseDef.tiles). We used a set of 297 nodule data images (208 training and 89 test images) and 1929 non-nodule data images (1351 training and 578 test images). The ROI size was 33 × 33 pixels and the binarizing threshold β was 40. The number of orientations M was 4 and the Gabor filter's parameters λ, σ, and γ were 1.5, 2.6, and 1, respectively. The number of principal components C was 5, defined as the minimum number that satisfies $\sum_{j=1}^{C} u_j > 0.95$ where u_j is the contribution ratio of principal component j. The number of clusters was 35 (25 nodules and 10 non-nodules).

3.2.1 Clustering results

Figs. 3 - 5 show sample images of feature vectors belonging to clusters made from training nodule images. The results demonstrate that each cluster consists of similar circle-like shapes of nodules. This suggests that the orientation features extracted from the nodule images can be useful for clustering them, and thus the proposed feature is effective for nodule discrimination.

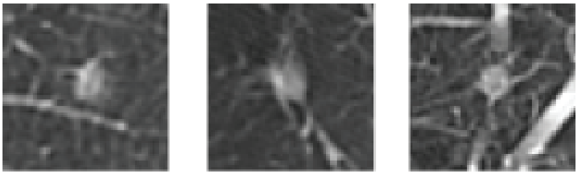


Fig. 3. Nodule images in cluster A. Images including relatively light and fuzzy boundary shadows are involved in this cluster.

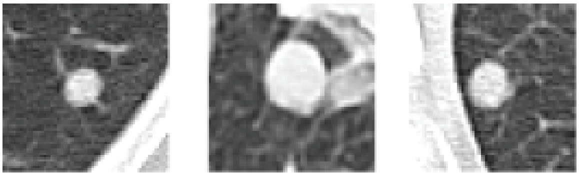


Fig. 4. Nodule images in cluster B. Images including relatively bright, smooth boundary and large circle shadows are involved in this cluster.

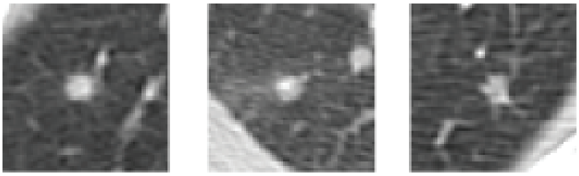


Fig. 5. Nodule images in cluster C. Images including small circle with spiculated boundary shadows are involved in this cluster.

On the other hand, Figs. 6 - 10 show sample images of feature vectors belonging to non-nodule clusters. The results demonstrate that some clusters are composed of similar shapes of non-nodules, but some are not. For example, in cluster nA (Fig. 6), most of images look like small circles, but there are a few images not involving such small circle shapes. Also, there are no similar shapes with each other in cluster nE (Fig. 10). This implies that non-nodule clusters are composed of various images with relatively large variance of feature vectors, compared to similar images with small variance of feature vectors in nodule clusters. Indeed, variances of feature vectors in non-nodule clusters are relatively large, while variances in nodule clusters are relatively small, although both variances were not large: The averages of the variances in non-nodule and nodule clusters were 0.003 and 0.001, respectively.

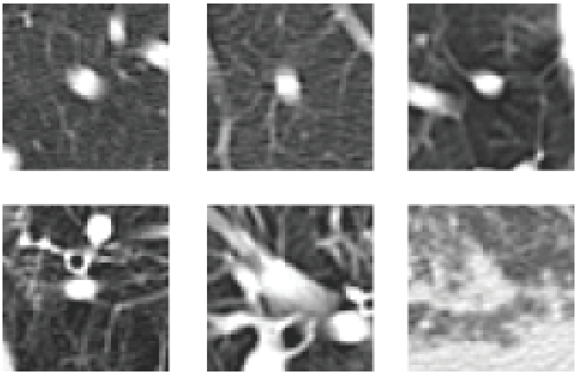


Fig. 6. Non-nodule images in cluster nA. Images including relatively small circle shadows are involved in this cluster.

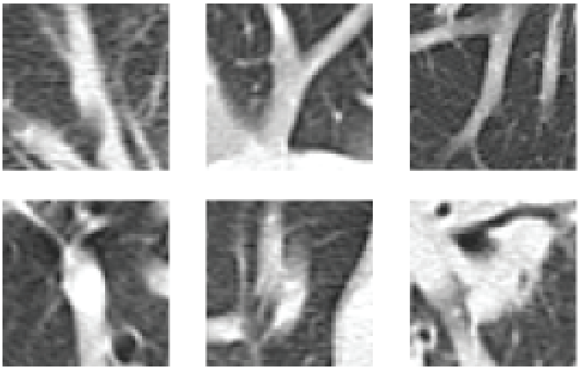


Fig. 7. Non-nodule images in cluster nB. Images including vertical line segments are involved in this cluster.

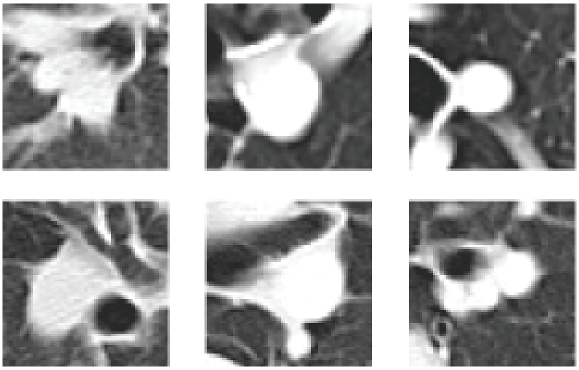


Fig. 8. Non-nodule images in cluster nC. Images including relatively large circle shadows are involved in this cluster.

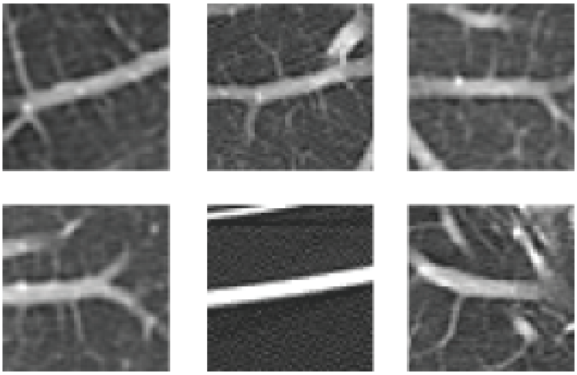


Fig. 9. Non-nodule images in cluster nD. Images including horizontal line segments are involved in this cluster.

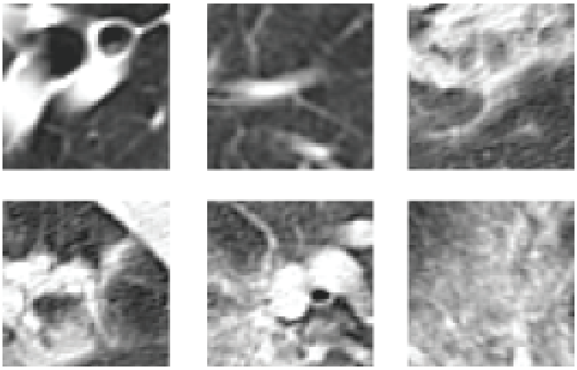


Fig. 10. Non-nodule images in cluster nE. Various shapes are involved in this cluster.

This suggests that further improvement for non-nodule clustering can be done by extracting more effective features from the original images. Such improvement will be discussed in section 4.2.

3.2.1 ROC analysis

Fig. 11 shows the 3 receiver operating characteristic (ROC) curves. Without 12 features of 4 orientations extracted by the Gabor filter, FP was about 80% when TP was 80%, while FP was about 35% by using the orientation features. The improvement of the discrimination rate (FP was improved from 80% to 35%) clearly demonstrates the effectiveness of the proposed feature on the diagnosis of pulmonary nodules.

On the other hand, FP was about 30% under the same condition by using a massive training artificial neural network (MTANN) (Suzuki et al., 2003). Although these rates can be improved if we could choose more suitable settings for both the proposed and MTANN methods, we may claim that the discrimination performances of both methods are almost equivalent.

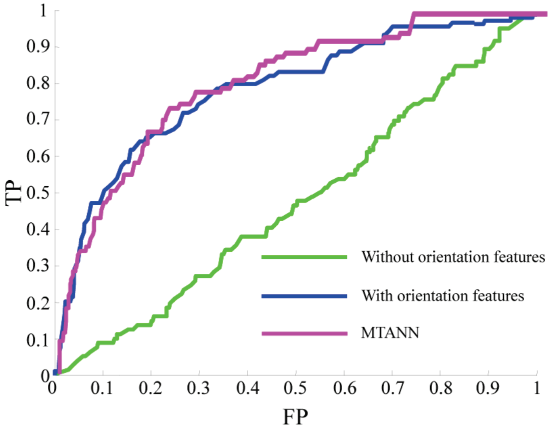


Fig. 11. ROC curves.

4. Advanced Methods for Lung Cancer Diagnosis

4.1 Variation feature along body axis

To further improve the discrimination rate for clinical use, we will now try to extract another effective feature (Homma et al., 2008). To begin with, let us consider why the discrimination performance using the orientation feature is not enough and what kind of images can be misjudged. For example, Fig. 12 shows a CT slice image of a patient. As mentioned in section 2.2, nodules often have circle-like shadows and thus we want to extract such shape information by using the Gabor filter. There are, however, some cases in which it is hard to discriminate between nodule and non-nodule images by using only such shape feature although the proposed one can be more effective than some conventional ones as demonstrated in the preceding section. Both images of nodules in cluster C (Fig. 5) and non-nodules in cluster nA (Fig. 6) have similar circle-like shapes, for example.

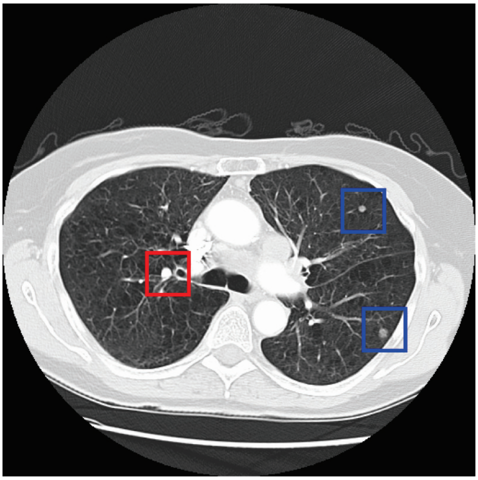


Fig. 12. ROI images detected by the variable N-quoit filter. Blue frames indicate images including nodules, whereas the red frame indicates a non-nodule image.

Different from the shape information within a CT slice, a novel feature can be extracted from shadow shapes across CT slices in the direction along the body axis (cranio-caudal direction). For example, Figs. 13 and 14 are CT slices above and below Fig. 12. Note that, according to a common opinion of several radiologists, circle-like shapes of non-nodules are almost shadows of blood vessels in the direction along the body axis. In this case, as seen in these figures, the blood vessels are cylinder-like shapes and thus the circle-like shadows remain at the same position if we look at slices above and below the target slice. On the other hand, nodules are often ball shapes. In this case, if we look at a slice above or below the target slice, the circle-like shadows often disappear. Thus, as a new feature, we employ the variation of CT values in the direction along body axis.

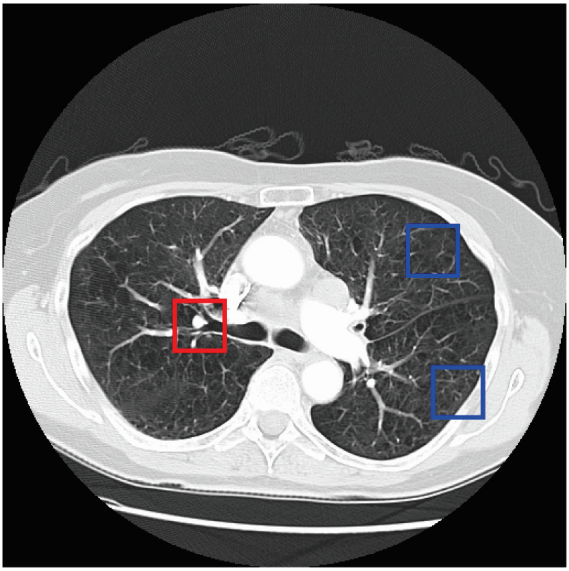


Fig. 13. CT slice image above the slice of Fig. 12. The red frame shows continuity between Figs. 12 and 13, in which a circle-like shadow remains at the same position in both figures. On the other hand, the blue frames show discontinuity that sizes and CT values of circle-like shadows in both figures are different from each other.

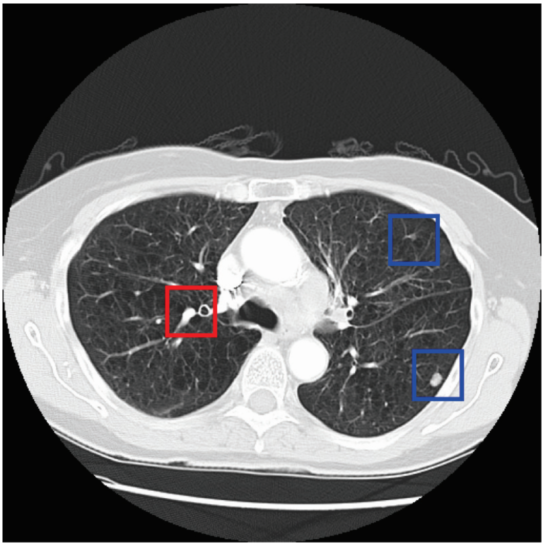


Fig. 14. CT slice image below the slice of Fig. 12. As same in Fig. 13, we can see continuity of a non-nodule shadow and discontinuity of nodule shadows.

To extract the variation feature, we first calculate the average pixel value of the shadow image in the ROI. If a shadow is of a non-nodule and a part of the cylinder-shape blood vessels along the body axis, continuity of the average values can be observed. On the other hand, if the shadow is of a nodule, then discontinuity of the average can be observed. In other words, for the non-nodule case, the average value is almost the same in above and below slices, while the average changes depended on the slices for the nodule case.

Let us denote the average values of the shadows V_m, V_u, and V_l for the target slice, and slices above and below the target, respectively. Using the averages, we define a new feature of shadow variation in the direction along the body axis T by

$$T = \max(T_u, T_l) \tag{13}$$

where

$$T_u = | 1 - V_m / V_u | \tag{14}$$

$$T_l = | 1 - V_m / V_l | \tag{15}$$

The concept of calculation of the feature extraction is illustrated in Fig. 15.

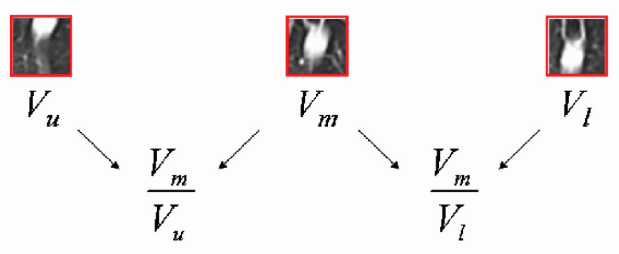


Fig. 15. Extraction of the shadow variation feature T.

The new feature T tends to be small for non-nodule shadows of the continuity case while it is large for nodule shadows of the discontinuity case. In fact, for the data used in section 3.2, the average value of the variation T for non-nodule images was 0.182, while the average of T was 0.479 for nodules.

4.1.1 Effect of variation feature

Here the shadow variation feature T was first applied to the ROI images and then more careful discrimination using the orientation features was conducted. That is, if the variation feature T of a candidate shadow in a ROI is less than a threshold T_h, the proposed method regards the shadow as a non-nodule. Otherwise, if $T \geq T_h$, the candidate shadow in the ROI is discriminated by using the orientation features as described in section 3.1. It might be worth mentioning an interesting fact that radiologists first detect ROIs of candidate shadows from the original CT slices by using such variation information along the body axis, and then diagnose the detected ROIs by using more detailed information such as shape, size, and CT values of shadows. This is the reason why we use the variation feature T before the orientation ones.

We have evaluated the effect of the new features on the discrimination rate by using the ROC analysis. Fig. 16 shows ROC curves by the conventional method and proposed methods without and with the variation feature T. By using the variation feature, FP was about 20% when TP was 90% in the case of the threshold T_h =0.206. On the other hand, FP was beyond 50% without the feature *T*. In other words, the discrimination rate FP was improved from about 50% to 20% under the condition TP=90%. Note that the condition TP=90% is good enough for clinical applications of pulmonary nodules diagnosis. Thus, the improvement clearly demonstrates usefulness of the variation feature.

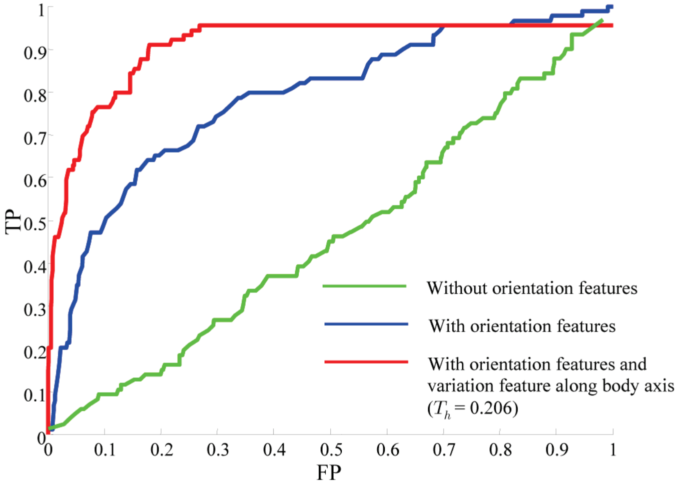


Fig. 16. ROC curves.

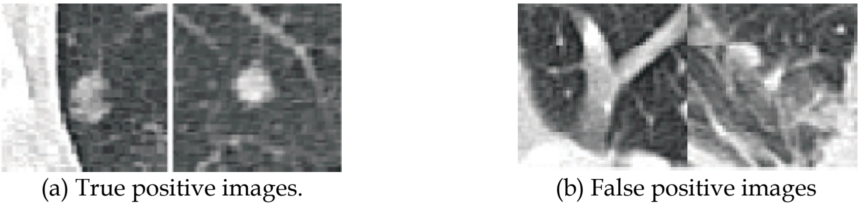
The fact that TP does not reach 100% in Fig. 16 might be a disadvantage of the proposed method with the variation *T*. This is because a few nodule shadows were regarded as non-nodule shadows by the variation threshold. As a second opinion for clinical use, however, robustness of the performance for various conditions is more important than TP=100% (TP ≥ 90% is often good enough). Indeed, the performance is robust for various threshold values and thus it can be another advantage for clinical use.

In addition to this, as shown in Fig. 11, performance of MTANN (Suzuki et al., 2003) was almost the same as that of the proposed method without the variation feature. Consequently, performance of the proposed method with the variation feature can be superior to that of the MTANN. Also, similar information to the variation *T* can be obtained by 3-dimensional images reconstructed from helical CT data (Nakayama et al., 1995). However, calculation of the variation *T* is very simple and thus less computationally expensive.

4.2 Toward further improvement

4.2.1 A new feature of circle-like shapes

Fig. 17 shows examples of the true positive and false positive images under the condition TP=90% and FP=20%. It seems that the TP and FP images can further be distinguished by their shapes: TP images are circle-like shapes while FP images are tree branch-like shapes of blood vessels or more complex shapes. The proposed orientation features do not work well for these images, although they are very effective for the greater part of images.


(a) True positive images.

(b) False positive images

Fig. 17. Examples of the true positive and false positive images. True positive means that the discrimination result of the CAD system is nodule and it is really nodule whereas the false positive means that the system's result is nodule, but it is non-nodule.

As discussed in section 3.2.1, nodule images are clustered well compared to non-nodule images clustering. A wide non-nodule cluster region with high variances can affect the FP results because the distance to the FP image may be overestimated even if an image involved in a non-nodule cluster is close to the FP image in the feature vector space.
Another reason for this may be that the features are calculated for each orientation independently, but their relation among the orientations is not considered at least explicitly. For example, as illustrated in Fig. 18, we can expect that average pixel values extracted by the Gabor filter for all orientations are almost the same for circle-like shapes, while the averages are different from each other for line segments or tree branch shapes.

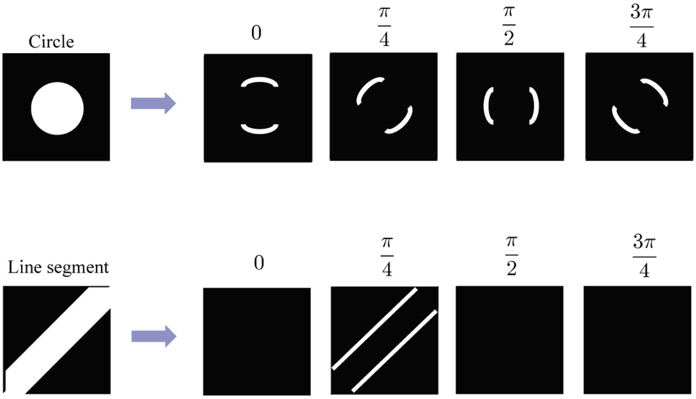


Fig. 18. Expected relation between different angles for circle-like and tree branch-like shapes.

To extract such differences between orientations, higher angle resolution may be necessary. However, as shown in Fig. 19, the discrete Gabor filter function is depended on the angle because of the small size of ROI. In such case, the sums of pixel values extracted the Gabor filter are different from each other even for the circle-like shapes.

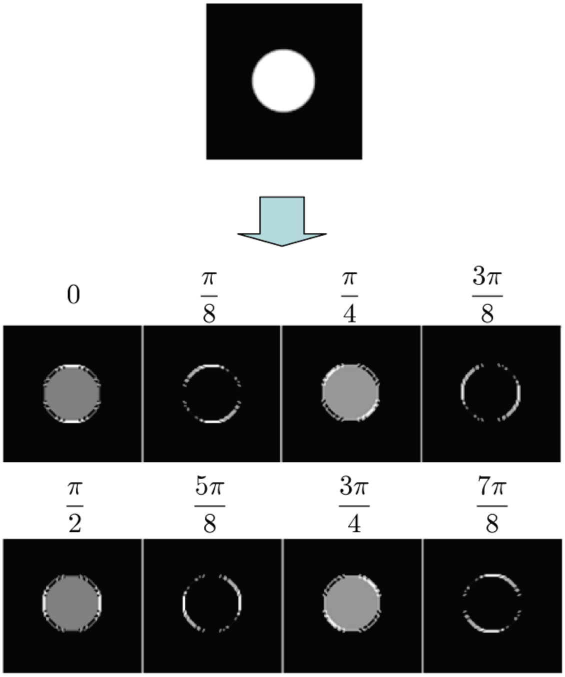
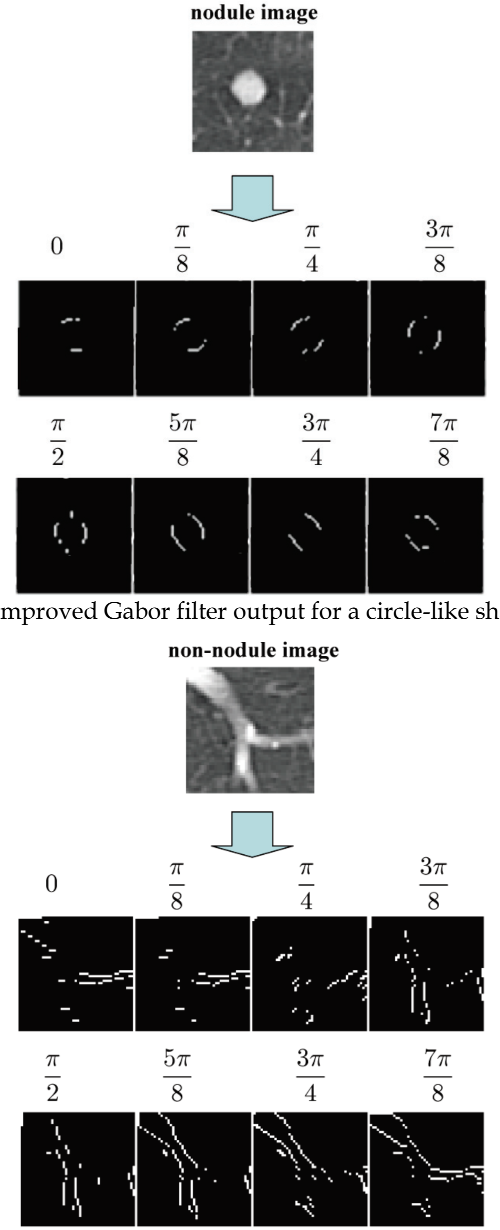


Fig. 19 Angle dependency of the Gabor filter outputs: Extracted values of inner and edge pixels are different from each other for various angles.

To overcome this problem, we conducted an edge detection technique as preprocessing the original images, and then the output images of the Gabor filter were binarized to eliminate the error caused by the spatial resolution (Homma et al., 2008). By this improvement, as shown in Fig. 20, average values of *M*=8 orientations can be almost the same for all orientations for the circle-like shape, while for the branch-like shapes, the 8 average values are different from each other depending on the orientation of the branches.



(a) Improved Gabor filter output for a circle-like shape.

(b) Improved Gabor filter output for a tree branch-like shape.

Fig. 20. Extracted values of inner and edge pixels are (a) almost the same for a circle-like shape, but (b) different from each other for a tree branch-like shape.

4.2.2 Improved results

Fig. 21 shows the standard deviation *s* of 8 average values for the TP and FP images. The numbers of TP and FP images were 72 and 76, respectively. Note that the standard deviations *s* for TP images are relatively small as expected for circle-like shapes, whereas the deviations for FP images are relatively large or widely distributed from large to small. Thus, after the discrimination by the variation along body axis and orientation features proposed in section 2, the TP and FP images can further be distinguished by the new feature *s*. In fact, FP=8% under the condition TP=90% when the algorithm classifies the images with $s > 0.01$ into non-nodules. In other words, FP decreased from 20% to 8% under the condition TP=90%.

Although the improvement achieved by the new feature is a good result, what we would like to stress here is that the combination of several effective features and classification techniques might be the most important for developing clinically useful CAD systems. The methods and the promising results presented in this chapter may support the importance of the combination.

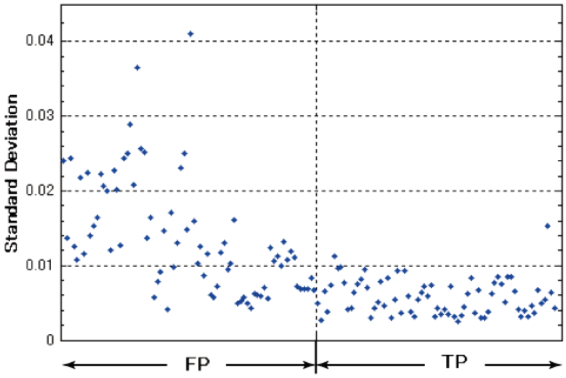


Fig. 21. The standard deviation *s* of *M* average values for TP and FP images. *s* for TP images are relatively small as expected for circle-like shapes.

5. Concluding Remarks

In this chapter, we have proposed a new method to detect pulmonary nodules in X-ray CT images. From results in this study, we may claim that the proposed orientation and variation features of nodules can be useful for the pulmonary nodule diagnosis. The proposed method is based on the radiologist's diagnosis process. That is, by using the variation feature of shadows in the direction along the body axis, the method first selects nodule candidates and then only for the candidates, instead of all the images, the method further discriminates nodules from non-nodules by using the orientation feature details of shadow shapes. The selection can thus contribute to less computational expense.

The methods introduced and developed in this chapter, however, aimed at only isolated circle-like shapes with the some morphological features, and thus non-isolated nodules (pathological changes) may not be detected by such methods. To improve the detection rate of such non-isolated nodules, Homma et al. have proposed a new technique that transforms the non-isolated nodules connected to the walls of the chest into isolated ones (Homma et al., 2009). Again, we can combine the technique for non-isolated nodules with the methods for isolated nodules for clinical usefulness. The other drawbacks of the conventional methods can further be improved by incorporating some specific methods to solve the drawbacks.

6. References

Haralick, RM. et al. (1987). Image Analysis Using Mathematical Morphology, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 9, No. 4, pp. 532-550

Homma, N., Takei, K., & Ishibashi, T. (2008). Combinatorial Effect of Various Features Extraction on Computer Aided Detection of Pulmonary Nodules in X-ray CT Images, *WSEAS Trans. Information Science and Applications*, Vol. 5, Issue 7, pp. 1127-1136

Homma, N., Simoyama, S., Ishibashi, T., & Yoshizawa, M. (2009). Lung Area Extraction from X-ray CT Images for Computer-aided Diagnosis of Pulmonary Nodules by using Active Contour Model, *WSEAS Trans. Information Science and Applications*, Vol. 6, Issue 5, pp. 746-755

Iinuma, T. ; Tateno, Y., Matsumoto, T. et al. (1992). Preliminary specification of X-ray CT for lung cancer screening (LSCT) and its evaluation on risk-cost-effectiveness. *Nippon Acta Radiologica*, Vol. 52, pp. 182-190 (in Japanese)

International Early Lung Cancer Action Program (I-ELCAP). (2006). Survival of Patients with Stage I Lung Cancer Detected on CT Screening, *New England J. Medicine*, Vol. 355, No. 17, pp. 1763-1771

Kondo, M., Hirano, Y., Hasegawa, J., Toriwaki, J., Ohmatsu, H. & Eguchi, K. (2000). Classification of tumors in chest X-ray CT images into the solid and air-containing type and application to discrimination of the benign and malignant tumors, *TECHNICAL REPORT OF IEICE*, Vol. 100, No. 46, MI2000-16, pp. 27-32 (in Japanese)

Lee, Y., Hara, T., Fujita, H., Itoh, S., & Ishigaki, T. (1997). Nodule detection on chest helical CT scans by using a genetic algorithm, *Proc. of IASTED International Conference on Intelligent Information Systems*, pp. 67-70

Miwa, T., Kako, J., Yamamoto, S., Matsumoto, M., Tateno, Y., Iinuma, T., & Matsumoto, T. (1999). Automatic Detection of Lung Cancers in Chest CT Images by the Variable N-Quoit Filter, *Trans. Institute of Electronics, Information and Communication Engineers*, Vol. 82-D-II, pp. 178-187 (in Japanese)

Nakamura, Y., Fukano, G., Takizawa, H., Mizuno, S., Yamamoto, S., Matsumoto, T., Sone, S., Takayama, F., Koyama, M., & Wada, S. (2005). Recognition of X-ray CT image using subspace method considering translation and rotation of pulmonary nodules, *TECHNICAL REPORT OF IEICE*, Vol. 104, No. 580, MI2004-102, pp. 119-124 (in Japanese)

Nakayama, M., Tomita, T., Yamamoto, S. et al. (1995). Study of 3D Morphological Filtering Applied for Automatic Detection of Lung Cancer X-ray CT, *Medical Imaging Technology*, Vol. 13, No. 2, pp. 155-164 (in Japanese)

Naruke, T., Goya, T., Tsuhiya, R. et al. (1988). Prognosis and survival in resected lung carcinoma based on the new international staging system. *J. Thorac Cardiovasc Surg*, Vol. 96, pp. 440-447

National Cancer Imaging Archive (NCIA), https://imaging.nci.nih.gov/ncia/faces/baseDef.tiles

Ngo, C. W., Pong, T. C., & Zhang, H. J. (2002). On clustering and retrieval of video shots through temporal slice analysis, *IEEE Trans. Mlt.*, Vol. 4, No. 4, pp. 446-458

Okumura, T., Miwa, T., Kako, J., Yamamoto, S., Matsumoto, M., Tateno, Y., Iinuma, T., & Matsumoto, T. (1998). Variable-N-Quoit filter applied for automatic detection of lung cancer by X-ray CT, *Proc. of Computer-Assisted Radiology*, pp. 242-247 (in Japanese)

Prokop, M. & Galanski, M. (2003). *Spiral and Multislice Computed Tomography of the Body*, Thieme Medical Publishers, Stuttgart

Suzuki, K., Armato, S. G., Le, F., Sone, S. & Doi, K. (2003). Massive training artificial neural network (MTANN) for reduction of false-positives in computerized detection of lung nodules in lowdose computed tomography, *Med. Phys*, Vol. 30, No. 7, pp. 1602-1617

Takizawa, H., Kamano, S., Yamamoto, S., et al. (2001). Quantitative analysis of cancer candidate regions in chest X-ray CT images, *J. Computer Aided Diagnosis of Medical Images*, Vol. 5, No. 2, pp. 4-11 (in Japanese)

Yamamoto, S., Tanaka, I., Senda, M., Tateno, Y., Iinuma, T., Matsumoto, T., & Matsumoto, M. (1993). Image Processing for Computer Aided Diagnosis in the Lung Cancer Screening System by CT (LSCT), *Trans. Institute of Electronics, Information and Communication Engineers*, Vol. 76-D-2, pp. 250-260 (in Japanese)

Yamamoto, S., Nakayama, M., Senda, M., Matsumoto, M., Tateno, Y., Iinuma, T., & Matsumoto, T. (1994). A Modified MIP Processing Method for Reducing the Lung Cancer X-ray CT Display Images, *Medical Imaging Technology*, Vol. 12, No. 6 (in Japanese)

Neural Network Based Classification of Myocardial Infarction: A Comparative Study of Wavelet and Fourier Transforms

Fawzi Al-Naima[1] & Ali Al-Timemy[2]
[1]*Computer Engineering Department, College of Engineering,*
Nahrain University, Baghdad,
Iraq
[2]*School of Computing, Communications and Electronics,*
University of Plymouth, Plymouth, Devon,
United Kingdom

1. Introduction

The development of automated systems for medical diagnosis is a significant challenge faced by physicians, engineers and computer scientists. Such system requires a data set sufficiently large in order to be considered a reliable statistical sample (Karuiannis & Venetsanepouious, 1997). Conventional methods of monitoring and diagnosing cardiac abnormalities rely on detecting the presence of particular signal features by human observer. Due to the large number of patients in intensive care units and the need for continuous observation of such conditions; several techniques for automated abnormality detection have been developed in recent years to attempt to solve this problem. Such techniques work by transforming the mostly qualitative diagnostic criteria into a more objective quantitative signal feature classification problem (Owis et al., 2002).

Computer technology has an important role in structuring biological systems. The explosive growth of high-performance computing techniques in recent years with regard to the development of good and accurate models of biological systems has contributed significantly to new approaches to fundamental problems of modelling transient behaviour of biological systems. Computer based analytical tools for the in-depth study and classification of data over day-long intervals can be very useful in diagnostics (Acharya et al., 2004).

The ECG signal represents the changes in electrical potential during the cardiac cycle as recorded between surface electrodes on the human body. The characteristic shape of this signal is the result of an action potential that propagates within the heart and causes the contraction of the various portions of cardiac muscle. This internal excitation starts at the sinus node which acts as a pacemaker, and then spreads to the atria, this generates the characteristic P wave in the ECG. The cardiac excitation then reaches the ventricles giving rise to the characteristic QRS complex. After that the ventricles repolarises corresponding to the T wave of the ECG. The automatic detection and timing of these waves is important for diagnostic purposes (Unser & Aldroubi, 1996; Prasad & Sahambi, 2003).

The classification of ECG signals into different disease categories is a complex pattern recognition task. In conventional systems, a typical heart beat is identified from the ECG. The P, QRS and T waves are characterized using measurements such as magnitude, duration and area. Classification is performed on these measurements, but the measurements of P, QRS and T sections vary significantly even among normal beats and can lead to misclassification (Kim et al., 2001).

Electrocardiograms are signals that originate from the action of the human heart. The ECG is the graphical representation of the potential difference between two points on body surface, versus time. Its historical development has resulted in a tool for clinical diagnosis, the 12-lead ECG.

ECG signals are largely employed as a diagnostic tool in clinical practice in order to assess the cardiac status of a subject. They are used to examine ambulatory patients who are at rest during a recording or performing an exercise program and also patients in intensive care units. ECG recordings are examined by a physician who visually checks features of the signal and estimates the most important parameters of the signal. Using his expertise the physician judges the status of a patient. The recognition and the analysis of the ECG signal is difficult, since their size and form may change eventually and there can be a considerable amount of noise in the signal. Since the processing of ECG signal is a very important step in the process of ECG examination by physicians many tools, methods and algorithms from signal processing theory have been proposed, described and implemented. The Wavelet Transform (WT) is a new and promising set of tools and techniques for doing this. Wavelets have generated a tremendous interest in both theoretical and applied areas, especially over the past few years. A number of papers, already large, continues to grow, thus progress is being made at a rapid pace (Anan et al. 1995; Novak, 2000; Afsar & Arif, 2008).

2. Wavelets Applications in Medicine

The results of the studies in the literature have demonstrated that the WT is the most promising method to extract features from the ECG signals. The WT can be thought of as an extension of the classic Fourier transform, except that, instead of working on a single scale (time or frequency), it works on a multi-scale basis. This multi-scale feature of the WT allows the decomposition of a signal into a number of scales, each scale representing a particular coarseness of the signal under study. The WT provides very general techniques, which can be applied to many tasks in signal processing. One very important application is the ability to compute and manipulate data in compressed parameters which are often called features. Thus, the ECG signal, consisting of many data points, can be compressed into a few parameters. These parameters characterize the behavior of the ECG signal. These features of using a smaller number of parameters to represent the ECG signal are particularly important for recognition and diagnostic purposes (Güler & Übeyli, 2004).

The WT of a signal consists of breaking up a signal into shifted and scaled versions of a reference (mother) wavelet and has good properties of time and frequency localization. It is robust to time varying signal analysis. The wavelet coefficients represent measures of similarity of the local shape of the signal to the mother wavelet under different shifts and scales (Unser & Aldroubi, 1996).

Early diagnosis of acute Myocardial Infarction (MI) is of vital importance for patients attending the emergency department with chest pain, as there are large benefits for immediate treatment of acute MI patients. With appropriate therapy, the size of an infarct can be reduced which helps in preserving long-term cardiac function. On the contrary, without proper treatment the result may be severe cardiac damage that significantly reduces the prognosis for the patient. Computer-based ECG interpretations for the detection of acute MI are, therefore, of importance as they can improve the early diagnosis of acute MI (Haraldsson et al., 2004).

3. ECG Data Collection and Sample Selection

The ECG data for the present work were collected from Al-Kadhimya Teaching Hospital and from Al-Iskandaryia General Hospital in Baghdad. The ECG trace papers were recorded on an A4 trace paper containing the 12 ECG leads. The recorded ECGs were taken from MAC 1200 device. The collected ECG traces contain many diseases like MI and ischaemia (anterior, lateral and inferior wall infarction), ventricular hypertrophy and heart block. Sixty seven ECG traces had been recorded from these two hospitals. Some of them are corrupted by base line wander noise. All the ECG records had been diagnosed with the help of expert cardiologist from Al-Iskandaryia General Hospital. The job of the cardiologist was to diagnose the sixty seven ECG traces and to assess whether they are normal or abnormal.

In the present work, since most of the collected cases contain normal and inferior infarction, the inferior MI and ischaemia were selected to be analyzed by our proposed classification system. Forty-three normal and abnormal subjects (patients with inferior MI) had been selected from the sixty seven cases. These records were displayed on ECG graph paper to be analyzed in the present work. The next step was the conversion of the ECG paper into image file in the computer by using high-resolution scanner. The resulting image file was saved as a bitmap image. The horizontal and vertical resolutions were both set to 150 dots per inch (dpi). The resulting image size is 1230*1630 pixels. This procedure was done for all the collected ECG papers.

4. Methodology

The block diagram of the proposed system is shown in Fig.1. The following sub-sections will describe each component of the block diagram in detail.

4.1 Lead Selection

A single lead ECG is selected from the total ECG image. Lead III has been selected to be processed for the next operations because the inferior infarction appears only in lead III and aVF lead. The selected lead has a length of 13 large squares of the ECG image. The extracted lead III picture is shown in Fig. 2.

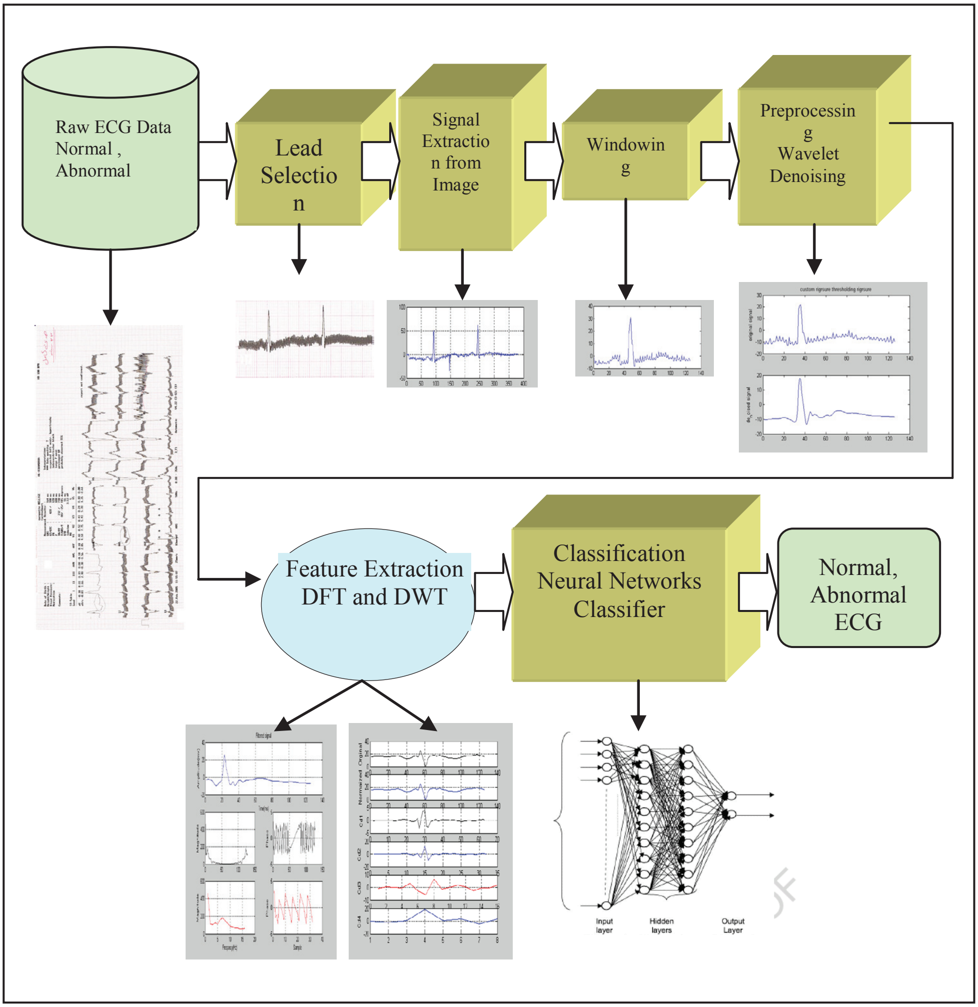


Fig. 1. Block diagram of the ECG classification system

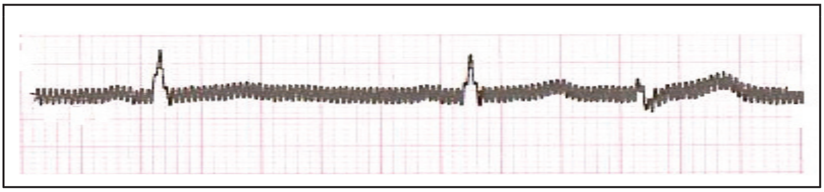


Fig. 2. Lead III picture selected from the total ECG image of sample No. 7

4.2 Signal Extraction from Image and Heart Rate Calculation

The multi-beat lead image in the previous block is converted into a multi- beat signal by the signal extraction block. The flowchart of signal extraction from the image is shown in Fig. 3. The multi-gray leveled ECG image is separated into a single gray level image. Then, scanning for the gray levels between 0 and 200 is done to capture the signal and exclude the background. The gray levels above 200 are discarded based on histogram technique to remove the background. The resulted lead III multi-beat signal is shown in Fig. 4. From this signal, the heart rate is calculated by measurement of R-R interval in large squares since it has a multi-beat by using the formula (Guyton & Hall, 2000).

$$HR=300/R\text{-}R\ (\text{distance in large squares}) \tag{1}$$

The heart rate is measured for all ECG data and it is expressed in bmp. After heart rate calculation, the algorithm can decide whether the subjects had a tachycardia, bradycardia or normal heart rate.

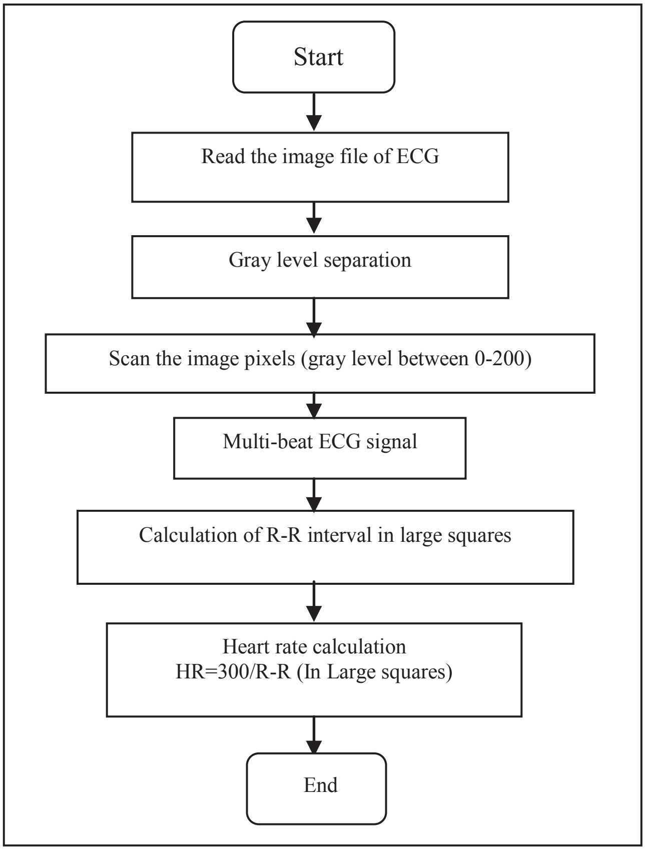

Start
Read the image file of ECG
Gray level separation
Scan the image pixels (gray level between 0-200)
Multi-beat ECG signal
Calculation of R-R interval in large squares
Heart rate calculation
HR=300/R-R (In Large squares)
End

Fig. 3. Flowchart of signal extraction from image and heart rate calculation

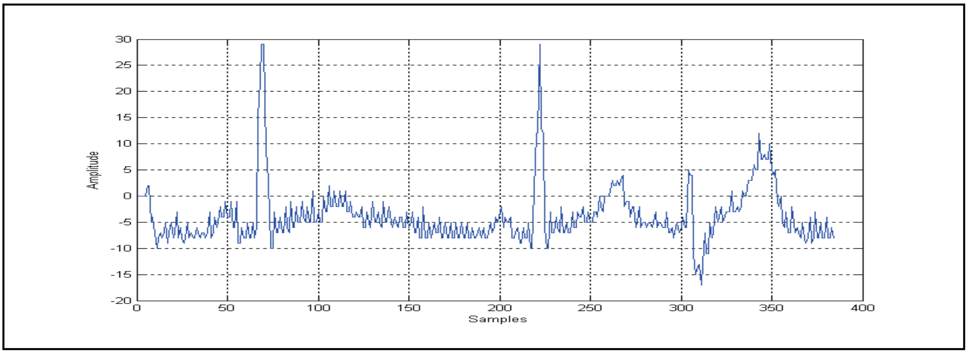


Fig. 4. Extracted lead III multi-beat signal from lead III image of sample No. 7

4.3 Windowing

As the shape of each beat in ECG waves is asymmetric, P-QRS-T complexes are selected by using windows with the range 80 samples before the R-wave maximum point to 48 samples after the R-wave maximum. This is to extract a single beat ECG signal from the multi-beat data. The resulting single beat ECG is shown in Fig. 5. The signal has been normalized to 1 as shown in Fig. 6.

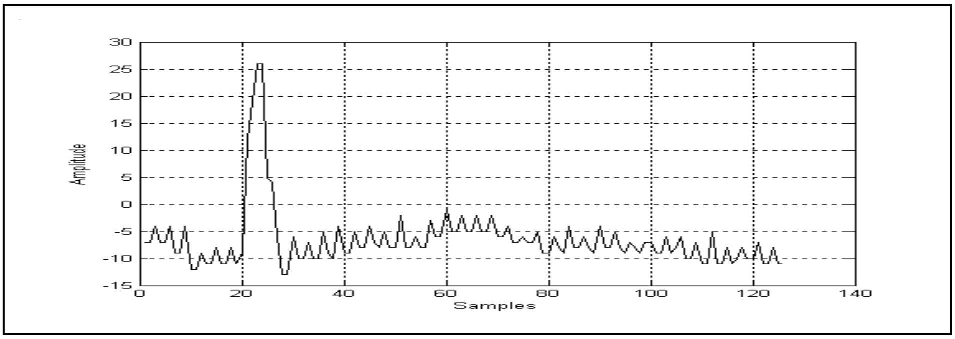


Fig. 5. Single beat lead III signal of sample No. 7

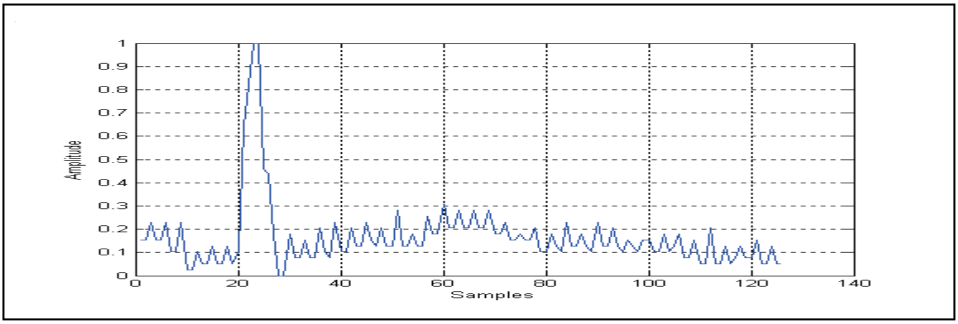


Fig. 6. Normalized single beat signal of sample No. 7

4.4 Wavelet Denoising

Wavelet denoising (WD) method or nonlinear wavelet filtering is based on taking Discrete Wavelet Transform (DWT) of a signal, passing the transform through a threshold, which removes the coefficients below a certain value, then taking the inverse DWT, as illustrated in Fig. 7. The method is able to remove noise and achieve high compression ratios because of the "concentrating" ability of the wavelet transform. The DWT localizes the most important spatial and frequency features of a regular signal in a limited number of wavelet coefficients (Novák, 2000).

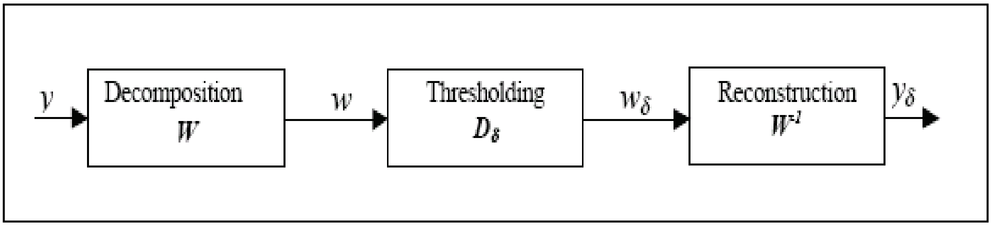


Fig. 7. Basic denoising concept

The general wavelet denoising procedure is as follows (Graps, 1995):

1. Apply wavelet transform to the noisy signal to produce the noisy wavelet coefficients to the desired level.
2. Select appropriate threshold limit at each level and threshold method to best remove the noise.
3. Take the inverse wavelet transform of the threshold wavelet coefficients to obtain a de-noised signal.

4.4.1 Threshold method

The nonnegative garrote shrinkage function which was first introduced by Breiman (Brieman, 1995); (Poornachandra & Kumaravel, 2008) is defined as:

$$\delta^{G}{}_{\lambda}(x) = x\left\{ - \left(\lambda/x\right)^{2} \right\}_{+} = \begin{cases} 0 & for \quad |x| \le \lambda \\ x - \lambda^{2}/x & for \quad |x| > \lambda \end{cases} \tag{2}$$

Where $\delta_{\lambda}{}^{G}(x)$ is the non-negative garrote shrinkage function. A garrote shrinkage function is plotted in Fig. 8. The shrinkage function $\delta_{\lambda}{}^{G}(x)$ is continuous (like the soft shrinkage, therefore more stable than hard), and approaches the identity line as $|x|$ gets large (close to the hard shrinkage, smaller bias than the soft shrinkage for large coefficient). The non-negative garrote shrinkage function provides a good compromise between the hard and the soft shrinkage functions.

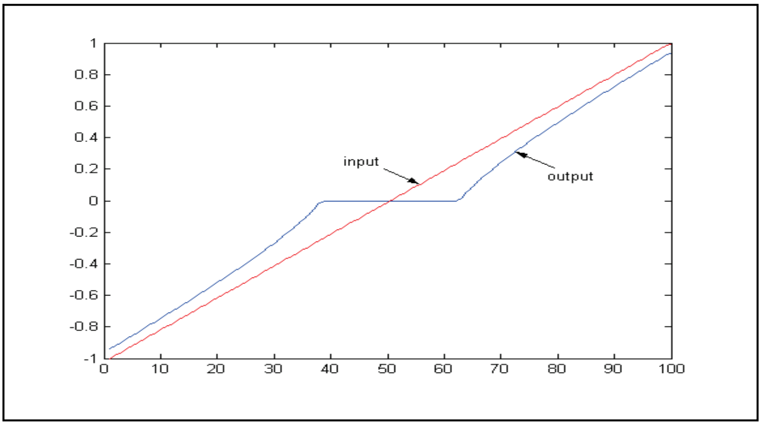


Fig. 8. Garrote thresholding function

4.4.2 Threshold Selection Rules

There are three main threshold selection rules (Graps, 1995).

a) Rigsure: threshold is selected using the principle of Stein's Unbiased Risk Estimate (SURE) quadrature loss function. An estimate of the risk can be obtained for a particular threshold value λ. Minimizing the risks in λ gives a selection of the threshold value.

b) Universal: Fixed form threshold yielding minimax performance multiplied by a small factor proportional to the length of the signal.

c) Heursure: Threshold is selected using a mixture of the first two methods. In the present work, Heursure is used as a threshold selection rule.

The original noisy ECG signal and the new denoised signal are shown in Fig. 9.

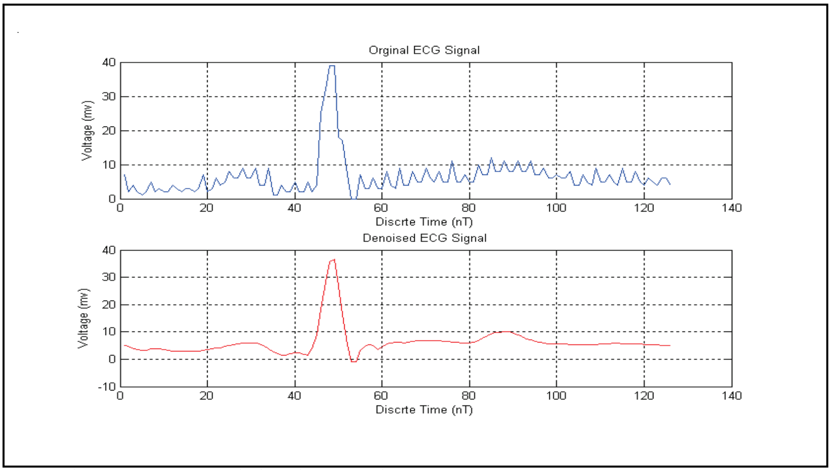


Fig. 9. Original noisy ECG and the denoised one

4.5 Feature Extraction

4.5.1 Feature Selection

It is basically impossible to apply any classification method directly to the ECG samples, because of the large amount and the high dimension of the examples necessary to describe such a big variety of clinical situations. A set of algorithms from signal conditioning to measurements of average wave amplitudes, durations, and areas, is usually adopted to perform a quantitative description of the signal and a parameter extraction. On this set of extracted ECG parameters, several techniques for medical diagnostic classification are then applied, such as probabilistic approaches, heuristic models, and knowledge-based systems. The aim of this work was to determine suitable input feature vectors which would discriminate between normal and abnormal MI beats (Al-Naima et al., 2008).

4.5.2 Reasons for the Use of Wavelet

This is the most interesting question for most of the users. The wavelet has one or two parameters. Because wavelets have so many constraints that are not associated with the signal, but more with mathematical and calculation limitations, it is virtually impossible to blindly select a wavelet. Wavelets are usually chosen on the basis of "If you see what you need to see, then that's that, if not, then try something else". The most general-purpose usable wavelet is Daubechies (Belgacem et al., 2003).

4.5.3 DWT Coefficients Extraction

In the present work Db4 and Haar wavelet have been used as the mother wavelets. MATLAB software package version 7 used to extract the DWT coefficients. For achieving good time-frequency localization, the preprocessed ECG signal is decomposed by using the DWT up to the fourth level. The smoothing feature of Haar wavelets and Db4 made them more suitable to ECG changes and the feature set is composed of level 1,2,3, 4 coefficients cd1,cd2 cd3,cd4 and ca4. Most of the energy of the ECG signal lies between 0.5 Hz and 40 Hz. This energy of the wavelet coefficients is concentrated in the lower sub-bands ca4, cd4, and cd3. The level 1, 2 coefficients cd1 and cd2 are the most detail information of the signal and they are discarded since the frequency band covered by these levels contains much noise and is less necessary for representing the approximate shape of ECG. The frequencies covered by these levels were higher than frequency content of the ECG. Coefficients cd3 and cd4 represent the highest frequency components and ca4 represent the lowest one.

For the Db4 wavelets, cd3 and cd4 having lengths of 21 and 14 coefficients are generated respectively. The DWT coefficients of Db4 wavelets of ECG segment of sample no. 7 are shown in Fig. 10. For the Haar wavelets, cd3 and cd4 having lengths of 16 and 8 coefficients are generated respectively. The obtained feature vectors form Db4 and Haar wavelets decomposition are used as an input to the NN classifier. The above procedure of decomposition is done for the 43 ECG segments for both normal and inferior MI patients.

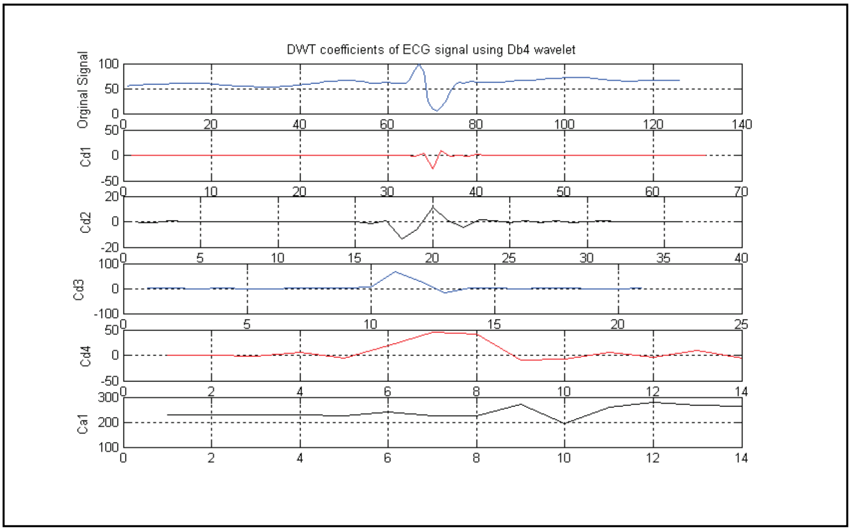


Fig. 10. The DWT coefficients of ECG Db4 wavelet of level-4 of sample No. 7

4.5.4 Discrete Fourier Transform

Discrete Fourier Transform of each data set was performed after getting the sampling period to observe both frequency and phase response properties of every ECG signals. Each ECG segment is analyzed by DFT of 128 points. Due to symmetry of the DFT, only 64 points are considered. Thus, the Fourier magnitude and phase of 64 points length have been obtained. To reduce the dimensionality, 16 points are selected from the 64 points sample of the Fourier magnitude. For the phase, a set of 32 points are selected from the 128 points phase. Both the 16 samples Fourier magnitude and 32 samples phase are used as feature vectors to be introduced to the NN classifier. The above procedure is done for each noisy and denoised ECG segments of the 43 cases of normal and patients with MI. The Fourier magnitude and phase for denoised ECG segment of sample no. 7 are shown in Fig. 11.

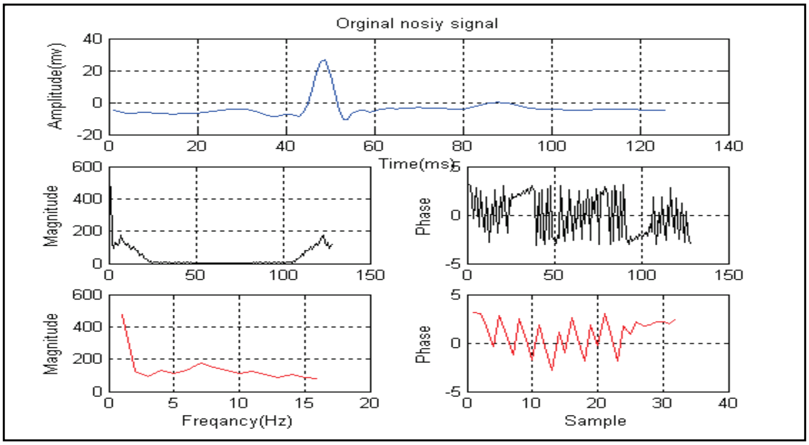


Fig. 11. The Fourier magnitude and phase for the denoised ECG signal

4.6 Neural Networks Classifier

In the present work, the neural networks are used for the classification purposes. The neural networks derive their power due to their massively parallel structure, and an ability to learn from experience. They can be used for fairly accurate classification of input data into categories, provided they are previously trained to do so. The accuracy of the classification depends on the efficiency of training. The knowledge gained by the learning experience is stored in the form of connection weights, which are used to make decisions on fresh input.
Three issues need to be settled in designing an ANN for a specific application:

1. Topology of the network.
2. Training algorithm.
3. Neuron activation functions.

In the topology we adopted, the number of neurons in the input layer was fixed by the number of elements in the input feature vector. Therefore the input layer had 16 neurons for both the first and the third ANN classifiers, 21 neurons for the second, and 32 neurons for the fourth one using DWT (Db4), DWT (Haar), Fourier magnitude, Fourier phase respectively. The output layer was determined by the number of classes desired. In our study, the unique neuron of the output layer corresponds to the normal and MI beats.
The proposed network was trained with all 45 cases (26 normal and 19 abnormal cases). These 45 cases are fed to the four feed forward neural network proposed in this study.
MATLAB software package version 7 is used to implement the software in the current work. When the training process is completed for the training data (45 cases), the last weights of the network were saved to be ready for the testing procedure. Learning rate is set to 0.5, the output of the network was (-1) for the class normal and (1) for the class abnormal. The training algorithm used for this network is the Back Propagation Algorithm (BPA). The performance goal was met at 2300 epochs after a training time of 45 sec. The summary of the back-propagation algorithm applied in the present work can be described as:

1. Initialization: Assuming no prior information is available, the synaptic weights and thresholds have picked to a random value.
2. Presentations of the training examples The network is presented with an epoch of training examples. For each example in the set, ordered in some fashion, the sequence of forward and backward computations described under points 3 and 4 is performed.
3. Forward computation
4. Backward computation
5. Iteration The forward and backward computations under points 3 and 4 are iterated by presenting new epochs of training examples to the network to reach the stopping criteria.

The testing process is done for 20 cases (12 normal and 8 abnormal). These 79 cases are fed to the proposed network and the output is recorded for calculation of the sensitivity, specificity and accuracy of prediction.

The accuracy of the classification depends on the efficiency of training. The knowledge gained by the learning experience is stored in the form of connection weights which are used to make decisions on fresh input.

Classification of MI is a complicated problem. To solve this two hidden layers are taken in a feed forward neural network. The single hidden layer is set for our four neural classifiers as follows: For the DWT (Db4) NN and Fourier magnitude NN, the hidden layer consists of four neurons. For the DWT (Haar) NN, the hidden layer consists of six neurons. And for the Fourier phase NN, the hidden layer consists of 8 neurons. The BPA is a supervised learning algorithm, which aims at reducing the overall system error to a minimum. The connection weights are randomly assigned at the beginning and progressively modified to reduce the overall mean square system error. The weight updating starts with the output layer, and progresses backwards. The weight update aims at maximizing the rate of error reduction, and hence, it is termed as 'gradient descent' algorithm. It is desirable that the training data set be large in size, and also uniformly spread throughout the class domains. In the absence of a large training data set, the available data may be used iteratively, until the error function is reduced to an optimum level. For quick and effective training, data are fed from all classes in a routine sequence, so that the right message about the class boundaries is communicated to the ANN.

Before the training process is started, all the weights are initialized to small random numbers. This ensured that the classifier network was not saturated by large values of the weights. In this experiment, the training set was formed by choosing 15 normal beats and 12 MI beats obtained from the selected cases.

The sigmoid function was used as the neural transfer function. The most important reason for choosing the sigmoid as an activation function for our networks is that the sigmoid function f(x) is differentiable for all values of x, which allows the use of the powerful BPA.

5. Results and Discussion

The performance of the algorithm was tested by computing the percentages of the three parameters; *Sensitivity (SE), Specificity (SP) and Accuracy (AC)* as follows (Al-Timemy, 2008; Al-Timemy & Al-Namia, 2009):

$$SE = \frac{TP}{(TP + FN)} \times 100 \tag{3}$$

$$SP = \frac{TN}{(TN + FP)} \times 100 \tag{4}$$

$$AC = \frac{(TP + TN)}{(TN + TP + FN + FP)} \times 100 \tag{5}$$

Where TP is the number of true positives, TN is the number of true negatives, FN is the number of false negatives, and FP is the number of false positives. The true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) are defined appropriately as shown below:

FP: Classifies normal as abnormal.
TP: Classifies abnormal as abnormal.
FN: Classifies abnormal as normal.
TN: Classifies normal as normal.

In our study, the unique neuron of the output layer corresponds to the normal and infarction beats. In practice, the number of neurons in the hidden layer varies according to the specific recognition task and is determined by the complexity and amount of training data available. If too many neurons are used in the hidden layer, the network will tend to memorize the data instead of discovering the features. This will result in failing to classify new input data. Using a trial-and-error method, we tested hidden layers varying between two and 20 neurons. The optimum number of neurons in the hidden layer was found to be five for the first ANN classifier, three for the second and two for the last one. The resulted accuracy, sensitivity and specificity for DFT-NN and DWT-NN are shown in Table 1.

Disease	**cases**	**Accuracy**	**Sensitivity**	**Specificity**
DFT-NN	20	85%	80%	90%
DWT-NN	20	95%	90%	90%

Table 1. The results after training of the proposed network

6. Conclusion

ECG signals of the human generated by the conduction system of the heart are usually non-stationary signals. A method based on image processing techniques was presented for data acquisition of the ECG cases. This method of obtaining ECG samples was shown to be efficient in ECG samples acquisition. In the present work, classification of ECG patterns was achieved by means of DWT and DFT combined with BP NN. In its current form, BPA uses the gradient descent to train the network.

The objective is to minimize the BP error to reach the desired response. Denoising process was adopted to remove different types of noise corrupting the ECG samples. The ECG signal can be used as a reliable indicator of heart diseases. In the present work, the DWT, DFT and the NN classifier are presented as diagnostic tools to aid the physician in the analysis of heart diseases. A wavelet based NN classifier has been proposed for MI classification. The feature set has been carefully chosen to have enough information for good accuracy. This feature set is a subset of DWT coefficients based on 'Db4' and 'Haar' wavelets.

7. References

Acharya, R., Kumar, A., Bhat, P. S., Lim, C. M., Iyengar, S. S., Kannathal, N. & Krishnan, A. S. M. (2004). Classification of Cardiac Abnormalities using Heart Rate Signal. *Medical & Biological Engineering & Computing*, 42

Afsar, F. A. & Arif, M. (2008). Robust Electrocardiogram Beat Classification using Discrete Wavelet Transform. Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on.

Al-Naima, F. M. M, Al-Timemy, A. H. A. & Mahdi, S. S. (2008). Data Acquisition for Myocardial Infarction Classification based on Wavelets and Neural Networks. *5th International Multi-Conference on Systems, Signals and Devices, 2008. IEEE SSD 2008***.** 20-22 July 2008, pp. 1-6, Amman, Jordan, ISBN:978-1-4244-2205-0

Al-Timemy, A. H. A. (2008). Self-Organization Maps for Prediction of Kidney Dysfunction. *Proc. 16th Telecommunications Forum, TELFOR*, Belgrade, Serbia, 2008, pp. 775-778

Al-Timemy, A. H. A. & Al-Naima, F. M. M. (2009). Comparison of Different Neural Network Approaches for the Prediction of Kidney Dysfunction. *International Journal of Biological and Medical Sciences* Vol. 5, No. 1, pp. 15-20, ISSN**:** 2070-3791

Anan, K., Dowla, F. & Rodrigue, G. (1995). Vector Quantization of ECG Wavelet Coefficients. *Signal Processing Letters*, IEEE, 2, pp.129-131

Belgacem, N., Chikh, M. A., & Bereksi-Rreguig, A. (2003). Detection of Cardiac Arrhythmias By Neural Networks, *The 5th International Conference on Enterprise Information Systems, ICEIS 2003*, Angers, France, April 22-26, 2003, Angers, France

Breiman, L. (1995). Better Subset Regression Using the Nonnegative Garrote. *Technometrics*, Vol. 37, No. 4, Nov., 1995, pp. 373-384

Graps, A. (1995). An Introduction to Wavelets. *IEEE Computational Science and Engineering* Vol. 2, No. 2 (1995), pp. 1-19

Güler, I. & Übeyli, E. D. (2004). Application of Adaptive Neuro-fuzzy Inference System for Detection of Electrocardiographic Changes in Patients with Partial Epilepsy using Feature Extraction. *Expert Systems with Applications,* Vol. 27, No. 3, (2004), pp. 323-330

Guyton, A. & Hall, A. J. E. (2000). *Text Book of Medical Physiology*, W.B. Saunders Company, U.S.A

Haraldsson, H.; Edenbrandt, L. & Ohlsson, M. (2004). Detecting Acute Myocardial Infarction in the 12-lead ECG using Hermite Expansions and Neural Networks. *Artificial Intelligence in Medicine,* Vol. 32, No. 2 (2004), pp. 127-136

Karuiannis, N. B. & Venetsanepouious, A. N. (1997). *Artificial Neural Network, Learning Algorithm, Performance, Evolution and Applications*, Kluwer Academic Publisher, London

Kim, M. J., Han, J. S.; Park, K. H., Bang, W. C. & Bien, A. Z. Z. (2001). Classification of Arrhythmia Based on Discrete Wavelet Transform and Rough Set Theory. *International Conference on Control Automation and Systems*, *(ICCAS 2001),* Oct 17-21, 2001, Jeju National University, Jeju, Korea

Novak, D. (2000). *ECG Processing using Wavelets*, Final Year Thesis, Valencia Technical University and Czech Technical University in Prague, Czech Republic

Owis, M. I.; Abou-Zied, A. H.; Youssef, A. B. M. & Kadah, Y. M. (2002). Study of Features based on Nonlinear Dynamical Modeling in ECG Arrhythmia Detection and

Classification. *IEEE Transactions on Biomedical Engineering*, Vol. 49, No.7, (2002), pp. 733-736

Poornachandra & S., Kumaravel, N. (2008). A Novel Method for the Elimination of Power Line Frequency in ECG Signal using Hyper Shrinkage Function. *Digital Signal Processing, Elsevier,*. Vol. 18,(2008), pp. 116–126

Prasad, G. K. & Sahambi, J. S. (2003). Classification of ECG Arrhythmias using Multi-resolution Analysis and Neural Networks. *Proc. IEEE Conf. on Convergent. Technologies (Tecon2003), Bangalore, India*, Vol. 1, pp. 227-231

Unser, M. & Aldroubi, A. (1996). A Review of Wavelets in Biomedical Applications. *Proceedings of the IEEE,* Vol. 84, No. 4, pp. 626-638

A Cellular Automaton Framework for Image Processing on GPU

Claude Kauffmann[1] and Nicolas Piché[2]
[1] *Department of medical imaging, Notre-Dame Hospital, CHUM, 1560 Sherbrooke East, Montreal, QC H2L 4M1 Canada*
[2]*Object Research System Inc. 760 St-Paul W, suite 101 Montreal, QC H3C 1M4 Canada*

1. Introduction

1.1 GPU Computing

Since 1999, which marks the introduction of the Graphics Processing Unit (GPU) for the PC industry, the use of commodity graphics hardware for non-graphic applications has become an important research topic. At the time, GPUs were specially designed for 3D computer graphics based on a fixed-function processor built around the graphic pipeline. These GPUs were difficult to program so the first programming efforts were regarded as academic projects. A major limitation of this generation of GPUs was the lack of floating-point precision in the fragmenting processors. This limitation has vanished with the introduction of floating-point pixel processing with the ATI Radeon 9700 GPU in the late 2002s, and the NVIDIA GeForce FX GPU in the early 2003s. After only a few years, we are now entering the third stage of GPU computing : GPUs are now general-purpose parallel processors (GPGPU) with support for accessible programming interfaces and industry-standard languages. Nowadays, the GPU's have drastically changed the face of computing and a large community of developers successfully applies their findings to GPUs, in various application domains, in order to achieve speedups of orders of magnitude superior to optimized CPU implementations. GPUs are playing an increasing role in non-graphic and scientific computing applications. For a good overview and the state of the art of GPUs architecture, computing and applications, the reader can refer to (Owens et al., 2008, Owens et al., 2007).

The high computational rates of the GPU have made graphics hardware an attractive target for demanding applications such as those in the field of signal and image processing. Among the most prominent applications in this area are those related to image segmentation, a driving application concerning the field of medical imaging.

1.2 Medical imaging context

Medical imaging must deal with increasingly large sized medical image data produced by modern imaging systems such as multidetector computed tomography (MDCT) or magnetic resonance imaging (MRI). The evaluation process of these images is not trivial and can result in missed abnormalities due to the inherent limitations of human perception. These evaluation difficulties increase with the use of 3D or 4D images. In clinical settings it is fundamental to extract quantitative and qualitative information from these images, in a reproducible and reliable way. The basic problem is the identification and extraction of anatomical or pathological features embedded in 2D or 3D images by a process called segmentation. Image processing on large medical volume data rely on computationally demanding applications that can benefit from the development of GPUs.

Medical image processing addresses a large range of problems and for each area of application, the segmentation strategies must be adapted to specific needs and clinical constraints. In several cases, fully automatic segmentation processes fail because the quality of images is directly affected by variability in imaging parameters, human anatomy and pathological changes over time. Therefore, the medical practitioner must take the time to correct the segmentation errors in 3D or 4D images in a tedious and time consuming task that is not compatible with clinical workflow. Medical images are complex and, in absence of higher level knowledge, this may result in several different interpretations. It is therefore imperative to incorporate some user intervention, which adds prior information in order to guide the segmentation process towards a reliable solution. This approach is suited in a clinical context because segmentation errors and computing time can be reduced within the same step. The principle is that high level image understanding is used to initialize low-level tasks operated by the computer. Specifically, the medical practitioner can draw rough scribbles labeling the regions of interest (each label related to a specific object) and then the image is automatically segmented by propagating these labels over the whole image. If needed, the user can add more seeds to achieve the ideal result.

Seeded segmentation approaches have been widely used in a very inspiring way in various domains, such as segmentation and matting of photo/video or in the treatment of natural images (Bai and Sapiro, 2007), cartoons or natural image colorization (Qu et al., 2006, Konushin et al., 2006), and interactive segmentation for image compositing (Vezhnevets et al., 2005, Yin et al., 2004, Rother et al., 2004, Mortensen and Barrett, 1998). These segmentation approaches rely on weighted-distance-based techniques. From each user-provided label, a weighted distance is computed in order to find out the probability for each unlabelled pixel to be assigned to a particular label. In these techniques, the image grid is seen as a graph with pixels as nodes and edges connecting neighboring pixels. Since the work initiated by Boykov and Jolly (Boykov and Jolly, 2000), the Graph Cuts method has become a very attractive way for interactive organ segmentation in medical imaging and different work based on weighted distance for image segmentation has been published (Xu et al., 2007, Protiere and Sapiro, 2007, Chefd'hotel and Sebbane, 2007). Additionally, the geodesic weighted distances can be seen as a specific case of the more general technique presented in (Falcão et al., 2004) as well as the $q = \infty$ case presented by Sinop and Grady (Sinop and Grady, 2007). In this last publication, the authors showed a general seeded image segmentation algorithm based on the minimization of ℓ_q norms. They showed that two popular seeded image segmentation algorithms, Graph Cuts (Boykov and Funka-Lea, 2006) and Random Walker (Grady and Funka-Lea, 2004), correspond to the parameter choices of q = 1 and q = 2 respectively.

Since graph-cuts algorithms are computationally heavy on large datasets, recent work has been done to find solutions to implement these algorithms on graphics hardware. Two different approaches are found.

The first looks at the new GPU implementation of the push-relabel algorithm (Vineet and Narayanan, 2008, Dixit et al., 2005) to solve the mincut/maxflow problem. Reported performances on 2D images show that the GPU approaches are very promising (Vineet and Narayanan, 2008). Nevertheless, it is important to notice that the implementation was performed using CUDA, a parallel computing architecture developed by NVIDIA, which works with all NVIDIA GPUs from the G8X series or newer. Compared to low level languages, CUDA needs only a short learning curve to build proof of concepts but the application is limited to NVIDIA cards.

Another approach is to focus on efficient GPU implementation for large sparse matrix solver (Volkov and Demmel) required by spectral methods for image segmentation such as Normalized-Cuts, or to find the solution to a sparse linear system as needed by the isoperimetric algorithm (Aharon et al., 2005). Implementation of these algorithms is not straightforward and for this reason, we focus our energy on developing simple iterative algorithms which can be formulated as a cellular automaton.

1.3 Cellular automaton: a model suited to GPU computing

Since the days when Von Neumann and Ulam (Von Neumann and Burks, 1966) first proposed the concept of cellular automata (CA) until the recent book of Wolfram `A New Kind of Science' (Wolfram, 2002), the simple structure of CA has attracted researchers from various disciplines. An exhaustive survey of literature, on cellular automata for modeling purposes, can be found in (Ganguly et al., 2003). CA became more practical and immensely popular in the late 1960's, when John Conway developed the cellular automata based *Game of Life* - an elementary computerized model of a colony of living cells (Gardner, 1970). The popularity of cellular automata can be explained by the enormous potential that they hold in modeling complex systems, in spite of their simplicity.

A cellular automaton (CA) is a collection of cells arranged in an N-D lattice, such that each cell's state changes as a function of time according to a defined set of rules that includes the states of neighbouring cells. Typically, the rule for updating the cells state is the same for each cell, it does not change over time and it is applied to the whole grid simultaneously. That is, the new state of each cell, at the next time step, depends only on the current state of the cell and the states of the cells in its neighbourhood. All cells on the lattice are updated synchronously. Thus, the state of the entire lattice advances in discrete time steps. It is clear that the concept of parallelism is implicit to cellular automata. In terms of computing, we need to process many elements (cells) with the same program. Each element is independent of the other elements and basically, elements cannot communicate with each other. These constraints match exactly with the GPU programming model, which is called SPMD and stands for Single Program, Multiple Data. Technically, each element can operate on a 32-bit integer or on floating-point data with a reasonably complete general-purpose instruction set. Elements can read data from a shared global memory (a "gather" operation) and with the newest GPUs, they can also write back onto arbitrary locations in shared global memory ("scatter") (Owens et al., 2008). This programming model is well-suited for programs formulated as CA, as many elements can be processed in lockstep running the exact same code. Code written in this manner is called "SIMD", for Single Instruction, Multiple Data.

The most recent published papers using CA and GPU cover simulation of physical (Zhao, 2008), biological (Gobron et al., 2007) or physiological (Alonso Atienza et al., 2005) processes. It would not be surprising to see an increase in the popularity of CA algorithms now that powerful GPUs are widely available, at low cost, on home computers.

2. Our contribution

Our work seeks to develop simple CA algorithms to perform efficient multi-label segmentation tasks on, but not restricted to N-Dimensional medical images, and implement them on low cost graphical hardware (GPUs).

In this paper we propose a massively parallel implementation of the Ford-Bellman's shortest paths algorithm (FBA) to perform graph-based segmentation.

Two applications based on FBA are presented. The first focuses on ultra-fast computation of the watershed transform on N-D images, while the second presents an alternative and an optimized framework to perform graph-based seeded segmentation on N-D images. It is important to notice that our work was voluntarily designed as a compromise between computational efficiency and hardware compatibility. Indeed, our FBA approach was implemented using OpenGL-Cg language on low cost, non vendor-specific graphics hardware. This aspect becomes interesting in a commercial software context because it can run on recent or moderately old hardware (NVIDIA GeForce 6 or ATI 1000 graphic card families).

3. Method

The N-D image is treated as a discrete object and is seen as a graph where each pixel (voxel) is a graph node or vertex. A predefined cost-function is used to characterize the edges abutting two adjacent nodes included in the neighborhood. The cost-function is used to compute multiple shortest-path-trees (sp-tree) where the tree roots are specifically labeled vertices called seeds. The segmentation result is obtained by cutting the graph in order to separate it into two, or more, sets. The algorithm starts by computer or user defined seed groups on the image. Each seed group is characterized by a location and a specific label so that the K-labels belong to K-specific objects in the image. The segmentation algorithm iteratively evolves from these starting labels, so that at the end, the N-D image is segmented in K objects. Each region is then guaranteed to be connected to seed points with the same label.

In the following, we consider an N-D gray scale image as a graph $G=(V,E)$ with a set of V vertices, or nodes, and a set of edges $e \in E \subseteq VxV$ spanning two neighboring vertices, v_i and v_j defined by its neighborhood. The weighted graph assigns a value to each edge e_{ij} called a weight or a cost and is denoted by c_{ij} or $c(e_{ij})$ and defined by $f: E \rightarrow R$, a real-valued weight function. $\lambda(v)$ and $l(v)$ represent the value and the label assigned to a vertex respectively. S^k is a start (or source) vertex with label k or a group of vertices with the same $k-label$. For each unlabeled vertex we can compute the minimal cost to reach the source s as the sum of the weights of the edges in the path. If we compute a sp-tree for k-labeled source vertex S^k, each unlabeled vertex of the graph can be represented by a K-tuple vector which represents the K-cost to reach the K-label. As shown by equation (1), the global cost, noted C_i^k is computed as the cumulative cost of the shortest path P^k between vertices v_i and the S^k seed group.

$$C_i^k = \sum f\left(P_i^k\right) \tag{1}$$

The *i-th* vertex can then be represented by a K-tuple vector $v_i\left\{C_i^j\right\}$ with $j \in 1, K$. The final vertex labeling process, $l(v)$ in (2), is derived from these K-tuples by tagging each vertex with the nearest K-label. The label assigned to vertex v_i is the label of the minimum cost C_j.

$$l(v_i) = label\left\{\min_{j=1}^{j=K}\left(C_i^j\right)\right\} \tag{2}$$

We show that this method can be used to partition a graph in K sub-graphs by performing K-cuts. It means that the image is split in K-specific regions including the k-seed groups. In practice, computing K times the sp-tree is not an optimal approach. We show in the next section that the multiple-sp-tree can be computed in a more effective way.

3.1 Dijkstra's and Ford-Bellman shortest paths algorithm

Dijkstra's shortest paths algorithm (Dijkstra, 1959) is the most popular method to compute the shortest path between two vertices (s,t), or between a start vertex s and all other vertices v_i in a graph. Dijkstra's algorithm is a heap-based method with computational complexity in O (N log N), but this complexity can be reduced by the use of techniques like that presented in (Yatziv et al., 2006). However, generally speaking, priority queues are very difficult to parallelize and we aim to develop other implementation strategies to compute the shortest paths.

A different approach used to calculate the distance of all vertices *vi* from a defined vertex *s* is the Ford-Bellman's algorithm (Bellman, 1956, Ford Jr, 1956) described in Algorithm 1 below. This 50-year-old algorithm gives a concise generalized expression of the cost-minimization problem. The most crucial, unique and unintuitive aspect of this algorithm is that even though the vertices can be processed in any order (even randomly), the algorithm will, in the end, produce the lowest-cost distance from every vertex to the start vertex (Even, 1979). Each vertex is simply relaxed several times until the algorithm converges to the stationary global solution. This important aspect of the Ford-Bellman's algorithm has caught our attention because it allows an efficient parallel implementation that can be achieved by cellular automata as presented in the following section.

```
Algorithm 1: The Ford-Bellman's algorithm
$\lambda(s) \leftarrow 0$ and for every $p \neq s, \lambda(p) \leftarrow \infty$
as long as there is an edge $p \rightarrow q$, such that $\lambda(p) > \lambda(q) + w_{pq}$ then
$\lambda(p) = \lambda(q) + w_{pq}$
$label(p) = label(q)$
```

This algorithm generalizes Dijkstra's algorithm for graphs having negative arc weights but without cycles of the negative weights. Unfortunately, on conventional sequential computers, the algorithm takes *O(n³)* time to generate a complete connected *n*-vertex weighted graph. A hardware specific parallel implementation is proposed by (Nepomniaschaya, 2001). In this paper, the author introduces a natural straightforward matrix representation of the Ford-Bellman algorithm on a STAR-machine. The STAR-machine is an associative parallel system of the SIMD type with vertical processing. The goal of the study was to represent the Ford-Bellman algorithm as a corresponding STAR procedure, to justify its correctness and evaluate time complexity.

3.2 Expression of FBA as a cellular automaton

CA is a collection of cells arranged in an N-D lattice, such that each cell's state changes as a function of time according to a defined set of rules that include the states of neighbouring cells. That is, the state of a cell at a given time depends only on its own state, at the previous time step, and on the states of its neighbourhood cells at the previous time step. All cells on the lattice are updated synchronously. Thus the state of the entire lattice advances in discrete time steps. For a 2D image, the Moore or von Neumann neighbourhoods can be used, while in 3D, the natural extension of these neighbourhoods gives us 6 and 26 neighbours respectively. Following this definition, CA can easily be applied to N-D images (lattice) represented by a graph G, where cells are pixels (or voxels) of this image and vertices of the graph.

We can then write the CA rule that computes, for each time step (t), the Ford-Bellman's algorithm by the following equation :

$$\lambda^{t+1}(p) = \min\left[\lambda^t(p), \min_{q \in N_p}\left\{\lambda^t(q) + w_{pq}\right\}\right] \tag{3}$$

Where N_p is the neighbourhood of *p*.

Equation (3) shows that the Ford-Bellman's CA rule can be written in a very efficient and concise form. The principle of CA is then to apply this transition rule (3) synchronously for all cells and to iterate as long as any cell changes its state. The relation (3) computes the shortest path of all vertices to a set of specific initial seeds. In this case, at the end of the algorithm, all vertices are labeled with the seed label. In a more general case, the user starts to define interactively, on the image, K-group of seeds having K-labels, so that K-labels belong to K-specific objects. In this case, the seeds label must be spread out at the same time as the cell state is updated. We can then write the pseudo-code of the evolution rule of the CA (Algorithm 2) for the FBA when K-labeled seed groups are specified. At least, the user specified 2 labels, one for the object and the other for the background.

For iteration $k+1$, cell labels $label^{k+1}$, and cell states λ^{k+1}, are updated as follows:

```
Algorithm 2: CA rule for K-seeded weighted distance map
$s_l \in V$ Where $l \in [1, K]$ with $K$ the total number of label
$\lambda(s_l) \leftarrow 0$ and for all $p \neq s_l$, $\lambda(p) \leftarrow \infty$
$label(s_l) \leftarrow l$ and for all $p \neq s_l$, $label(p) \leftarrow 0$
for $\forall p \in V$
        $U^k(p) = \min_{q \in N_p} \{\lambda^t(q) + w_{pq}\}$
        $\lambda^{k+1}(p) = \min[\lambda^t(p), U^k(p)]$
        $label^{k+1}(p) = label\{\min[\lambda^t(p), U^k(p)]\}$
end for
```

3.3 Neighbourhood and Cost function

For a 2D image, the Moore or von Neumann neighbourhoods can be used, while in 3D, the relation below gives the 3D edge set connectivity E in the case of 6, 18 and 26-connected for N=1, $\sqrt{2}$ and $\sqrt{3}$ respectively.

$$E = \left\{p, q \,\middle|\, \|C(p) - C(p)\|_2 \leq N\right\} \tag{4}$$

Where $\|\cdot\|_2$ is used to denote the standard Euclidean norm and $C(p)$ maps voxel p to its 3D coordinates.

As for many graph-based segmentation algorithms (Boykov and Funka-Lea, 2006, Chefd'hotel and Sebbane, 2007, Grady and Funka-Lea, 2004, Protiere and Sapiro, 2007), the edge weights encode image intensity changes between two neighboring graph nodes *p, q*. In a general case, the cost w_{pq} can be represented by the following relation:

$$w_{pq} = f(I_p - I_q) + h \bullet \|C(p) - C(q)\|_2 \tag{5}$$

The parameter h is used, if needed, to add a regularization term which represents the geometric distance between two vertices.

3.4 FBA to compute the Watershed transform

Watershed transform is one of the most popular methods for image segmentation. The watershed transform was originally proposed by (Digabel and Lantuejoul, 1977) and later improved in (Beucher and Lantuejoul, 1979). The watershed method can be formulated in a general framework called image labeling, where a label is associated to each pixel from a finite set. The intuitive idea underlying this method comes from geography : when a landscape or topographic relief is flooded with water, watersheds are the dividing lines of the basins of attraction of rain falling over the region. The various formalizations, definitions, algorithms and implementations of the watershed concept can be divided into two classes.

One is based on the specification of a recursive algorithm by (Vincent and Soille, 1991) and the other one is based on distance functions by (Meyer, 1994). Moreover, watershed methods are usually based on sequential algorithms but during the last decade, serious efforts were made to find parallel implementation strategies (Eom et al., 2007, Noguet, 1997, N. Moga et al., 1998). Unfortunately, despite the use of all the techniques and architectures, there is always a stage, in the watershed transform, that remains a global operation. Therefore, only modest speedups are to be expected in the case of parallel implementation (Roerdink and Meijster, 2000). Our approach shows efficient parallel implementation of the watershed transforms based on a cellular automaton (CA) that computes Ford-Bellman's shortest paths (Kauffmann and Piche, 2008).

Our CA algorithm, presented above, does not produce watershed lines. All pixels are merged within some basin, so that the set of basins tessellates the image plane. This is a consequence of the local condition of the CA algorithm that is very advantageous for a parallel implementation of the watershed transform (Roerdink and Meijster, 2000). Unlike other parallel algorithms, our CA algorithm is deterministic and the result does not depend on the order in which the pixels are treated during the execution of the algorithm. This is a consequence of a fundamental aspect of CA, which is that all cells on the lattice are updated synchronously at each time step.

3.5 GPU implementation of FBA

The graphic card is a Single Instruction Multiple Data (SIMD) computer. This type of computer was not commonly available before its introduction into graphic processor, SIMD machines were principally dedicated to signal processing or other tasks. In the 90's there was great interest around these types of computers (for example the famous Connection Machine by Thinking Machines Corporation Inc.) but it was constrained to the research area. Huge improvements in regard to the speed of classical computers and the inherent programming difficulty of SIMD made them less attractive. The expertise needed to program this kind of devices was either lost or not developed. Nowadays SIMD are available at very low cost, so their use became widespread. However, to program a SIMD computer, one often needs to completely reformulate algorithms. Not all algorithms are well suited for GPU; ideal GPGPU applications have large data sets (in regards to the memory available on the graphic card), high parallelism, and minimal dependency between data elements. GPUs are perfect devices to execute CA code: this is due to the fact that CA algorithms only rely on local information (nearest neighbours and local states are usually sufficient). The only complication lies in that we are not aware of the sweeping order of the cells in our CA by the GPU (we do not know if a memory cell has yet to be processed or not). This means that you can have neighbours that are at time (t+1) and others at time (t). In order to update synchronously all the cell states, we need to use some kind of a buffer.

Let's describe how we have programmed these devices to implement the FBA algorithm: To contain the data and the results, we used a RGB 32 bits float graphic texture (double precision floating point will be available in near future on high end graphics cards). In the red Channel (R), we put the image data, in the green channel (G), we put the computed weighted distance map and in the Blue channel (B), we updated the label associated to each vertex. The vertex labeling step could have also been computed from the converged distance map but it is more efficient to update the labels at each iteration because a channel is available to store this information. Since all vertexes are updated synchronously in lockstep, we needed to have some kind of a buffer. The buffering technique used is the famous Ping-pong scheme, also called 'double buffering'. It is a programming technique that uses two buffers to speed up a computer that can overlap I/O with processing. Data in one buffer is being processed while the next set of data is read by the other one. This buffer technique does not need extra video memory. The 'shader' program, that has to be executed on each vertex, is written in OpenGL Cg, which runs on almost all hardware. It is a strait forward implementation of the FBA equation, as described in algorithm 2. The CA is iterated a predefined number of times and the label information are extracted from the Blue channel. In the cases where the algorithm did not fully converge, a few iterations have been added from this non converged state to reach the expected segmentation results. The user can also add or remove seeds and run the 'shader' code for an additional number of iterations.

Our approach offers the great benefits of running on older hardware and of being compatible with new graphic cards. We could have implemented a more efficient algorithm using CUDA, but this avenue is much too restrictive in terms of hardware requirement.

4. Experimental results

Two different studies, based on the FBA-GPU approach, were conducted. In the first study we applied the algorithm to compute the watershed transform on 2D and 3D images. The computational efficiency of our GPU approach was compared to a CPU optimized version of the watershed transform.

The second study concerned the use of the FBA-GPU approach to perform seeded segmentation of organs in medical image data sets. At first, we evaluated the reproducibility and accuracy of our GPU-FBA segmentation approach applied to 3D kidney segmentation on a retrospective study totaling 20 magnetic resonance angiography (MRA) acquisitions. Then, we performed a comparison between the computing performances of our FBA-GPU approach and a CPU implementation of our algorithm. Finally, the FBA-GPU was benchmarked on available graphic hardware's in order to assess the improvement of computation time on different graphic cards.

4.1 Technical specifications

All experimental results presented in the following studies have been performed on an Intel Xeon Dual Core (3 GHz) with 2GB RAM and an ATI Radeon X1950 Pro graphic card with 512 MB of graphical memory.

It is important to notice that our work was voluntarily designed as a compromise between computational efficiency and hardware compatibility. Indeed, our FBA approach was implemented using OpenGL-Cg language on different brands of low-cost graphic cards. This aspect becomes interesting in a commercial software context because it can run on recent or moderately old hardware, even on a NVIDIA GeForce 6 or on ATI 1000 graphic card families.

4.2 Parallel Watershed transform

In order to illustrate CA-Watershed results, the algorithm 2 was applied to different images by using the two following valued functions.

$$f_A = \left|\nabla(I)\right| \quad and \quad f_B = \left|\nabla(G_\sigma * I)\right| \tag{6}$$

where G is a Gaussian smoothing function. The watershed results are represented by mosaic images where each labeled region is filled by the mean value of the pixels inside this region. The starting seeds, s_i , are defined automatically as the local minima of the input image I, such as:

$$I(s_i) < \min_{q \in N_{si}}(I(q)) \tag{7}$$

Where N_s is defined as the neighborhood of s.

The first 2D application represents a magnetic resonance image (MRI) of a kidney along a sagittal plane (figure 1) while the second one is the popular image of Lena (figure 2). These examples show that the watershed regions give a regular partitioning of the image and a coherent and smooth representation of boundaries. On both images we applied the watershed transform by increasing the size of the Gaussian filtering kernel, with the smallest kernel represented in (a) and the largest one in (c). As the kernel size increases, the number of local minima decreases which results in greater watershed regions, as illustrated by figures 1 and 2.

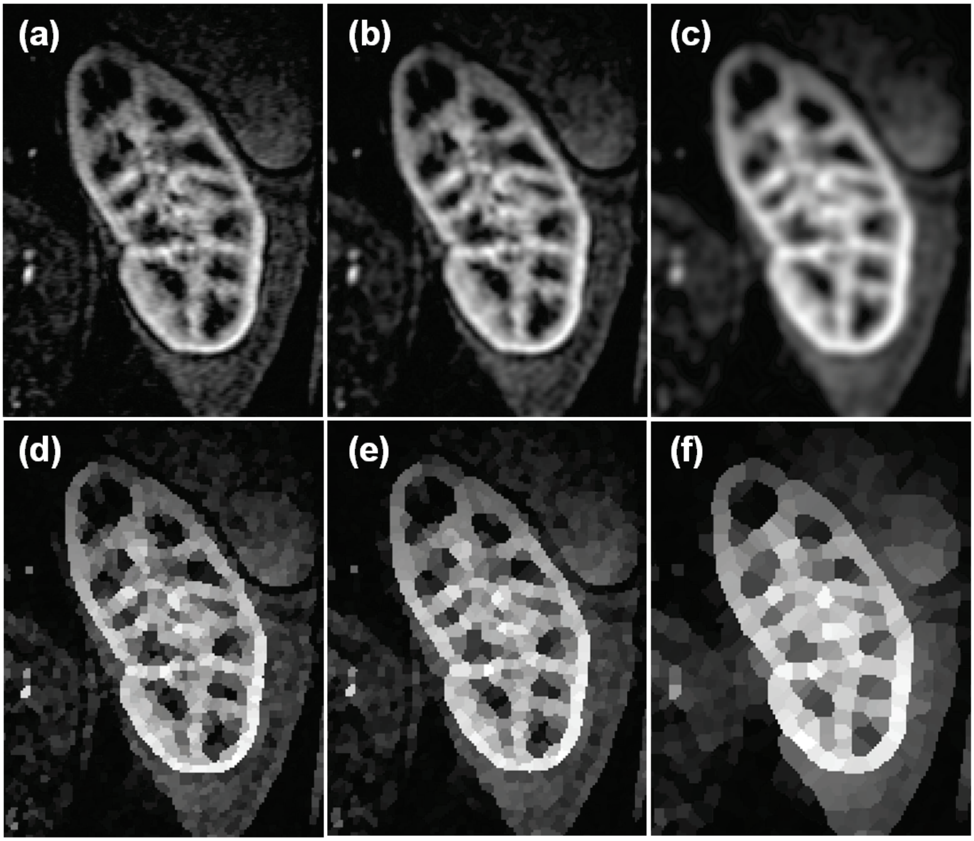


Fig. 1. MRI image of a kidney along the sagittal plane (a), kidney image filtered by a Gaussian (b, c), and respective watershed mosaic images (d, e, f).

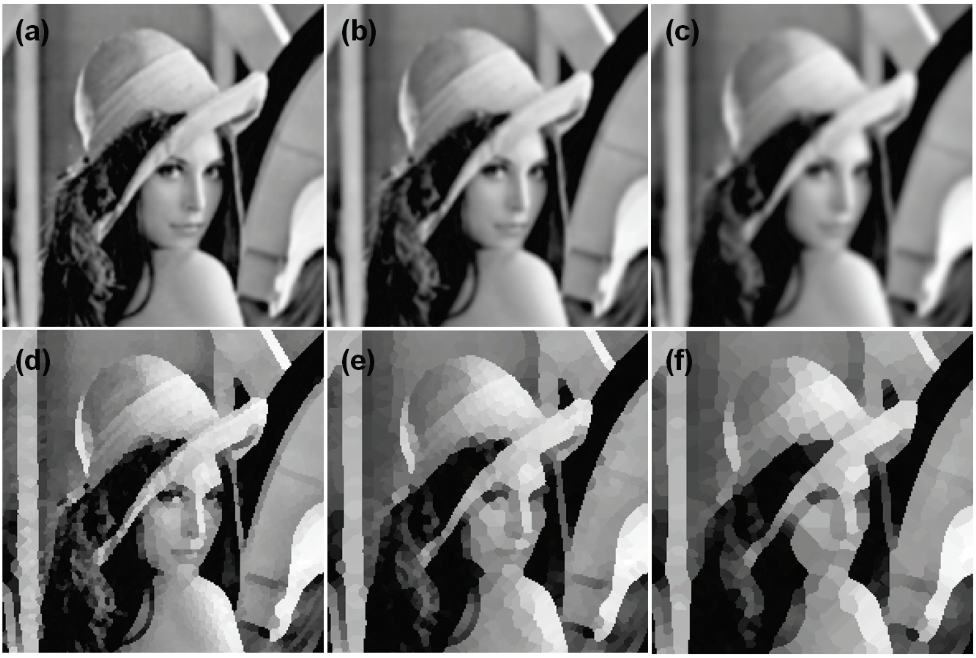


Fig. 2. Image of Lena (a) and smoothed versions (d, c) by a Gaussian function. Watershed mosaic (d, e, f) computed on gradient of respective images (a, b, c).

As we can see in the second study, the number of iterations needed so that the FBA algorithm converged, depends on the Euclidian length of the longest geodesic path between k-labelled vertices. We showed intuitively that computing a watershed transform can be an ideal case for the FBA application because the distance between two k-labelled seeds (two local minima in a specific neighborhood) is generally small, so that only few iterations are needed to reach a converged result. This is especially true in the context of noisy images, such as in medical image datasets, where the local minima are regularly spaced over the whole image dataset. Filtering the image will increase the distance between the local minima, so that more iterations are needed to ensure the convergence of the algorithm. A consequence of this shows that the FBA method is more computationally efficient when the local minima are regularly distributed over the image space and when the distance between the minima is small.

The computing efficiency of our GPU-CA-Watershed was evaluated by comparing its running time to the running time of a C++ implementation of the Tobogganing algorithm described in (Fairfield, 1990, Lin et al., 2006). The initial 3D images used for testing was based on an isotropic CT-scan acquisition of size 512 by 512 by 512 pixels. This dataset was downsampled, using the nearest neighbor method, In order to obtain isotropic datasets sized as a multiple of 64 by 64 by 64 pixels. The CPU and GPU watershed algorithms were applied to all 3D datasets and the respective computing times were recorded.

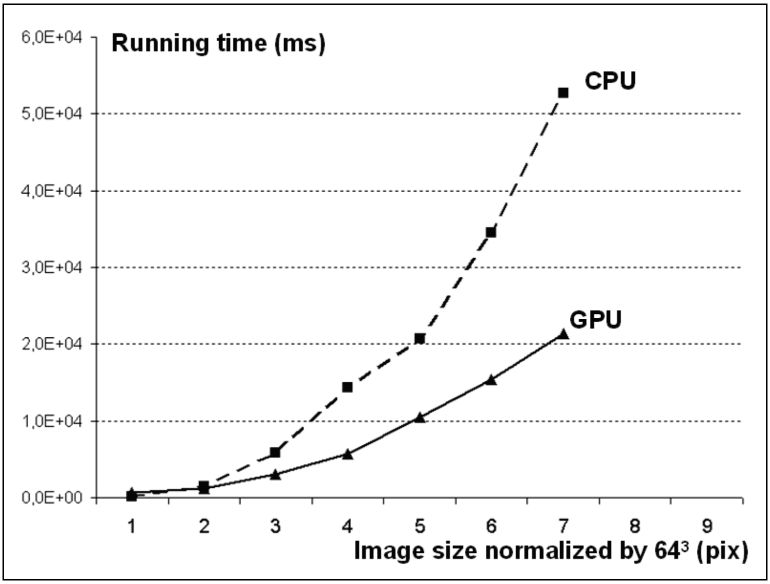


Fig. 3. Comparison between CPU and GPU computing time (ms) of watershed transform on box sized image datasets.

The results of our experiment, illustrated on figure 3, show that the GPU CA-Watershed performs 2.5 times faster than the C++ optimized version of the Tobogganing watershed. Moreover, this speedup factor in favour of GPU can easily be increased by using more recent low cost graphic hardware, as shown in Table 3 in the following section.

4.3 FBA for seeded segmentation

The second study seeked to evaluate the reproducibility and accuracy of the FBA method applied to the segmentation of renal volumes on a retrospective study totaling 20 magnetic resonance angiography (MRA) acquisitions. The computational performances of our FBA-GPU approach were compared with a CPU implementation of the Dijkstra's shortest paths algorithm. Finally, the FBA-GPU implementation was benchmarked on available graphic hardware in order to assess the improvement of computation time on different graphic cards.

The proposed CA-GPU segmentation technique was validated experimentally by measuring the renal volumes on MRA acquisitions of twenty patients affected by symptomatic renovascular disease. The context of validation was to apply our method to perform an automatic 3D segmentation of the kidneys based on user defined labelled seeds. From these segmentation results, the Parenchymal renal volumes were computed.

Study setup : All MRA were performed on a 1.5 T Magnetom Vision unit (Siemens, Erlangen, Germany) using a phased array body coil. The sequence used was a coronal 3D gradient echo centered on the aorta and the kidneys during dynamic IV administration of a contrast agent. The typical image matrix size (XYZ) was 512 by 512 by 150 pixels, with an associated pixel resolution of 0.82, 0.82 and 1.25 mm. A representative sample of our MRA database can be shown in figure 3. In these images, the kidneys are represented in a coronal section where two major structures can be observed: the superficial part is the renal cortex (enhanced signal) and the deep part is the renal medulla (appearing in grey or black). The kidney is a bean-shaped structure which presents concave and convex surfaces. The goal was to segment the 3D kidney Parenchyma (cortex and medulla). There were three main challenges : firstly, the kidney borders are not well defined in the concave surface, because it is the point at which the renal artery enters the organ;, secondly, renovascular disease directly affects the quality of MRA acquisitions which results in poor contrast between the kidney and the background as shown in figure 1b. Finally, the segmentation method had to be able to deal with cortical cysts (appearing, for example, as a black hole on the right kidney of figure 1d) that had to be excluded from the segmentation.

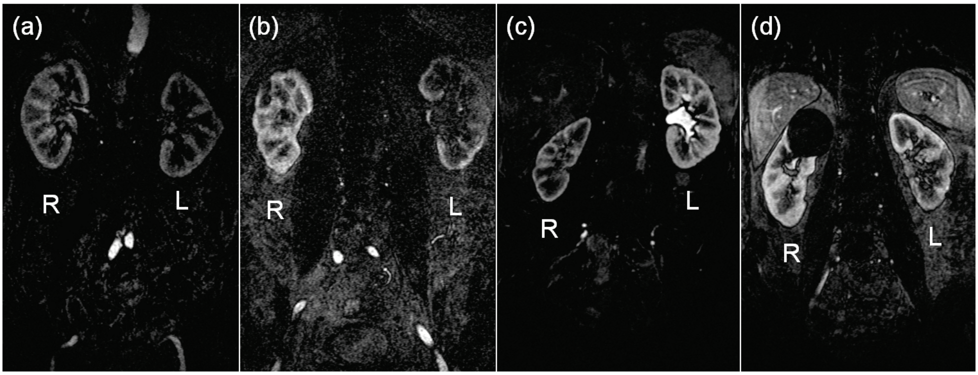


Fig. 3. MRA acquisitions under a coronal 3D gradient echo sequence centered on the aorta and the kidneys during dynamic IV administration of a contrast agent. The kidney is seen as a Bean-Shape structure where two major structures can be observed: the superficial part is the renal cortex (enhanced signal) and deep part is the renal medulla (appearing in grey or black). Significant morphologic and pathologic differences can be observed on the image sample which challenges the segmentation. Poor contrast between the cortex and the background can be observed on the left kidney (L) on figure 1b and 1a. Cortical cysts (appearing, for example, as a black hole on the right kidney of figure 1d) were excluded from the segmentation.

Two readers used the FBA-GPU software tool in a blinded manner to segment the 40 kidneys in MRA images. The automatic segmentation was started from user defined labelled seeds. In order to standardize the segmentation protocol between the readers, the two following steps were established. During the first step, the reader navigated in the 3D MRA dataset through an orthogonal multiplanar reconstruction (MPR) using a mouse-driven synchronized 3D cursor. As illustrated in figure 4, the user moved the cursor near the centre of the kidney so that the renal artery appeared in the axial view.

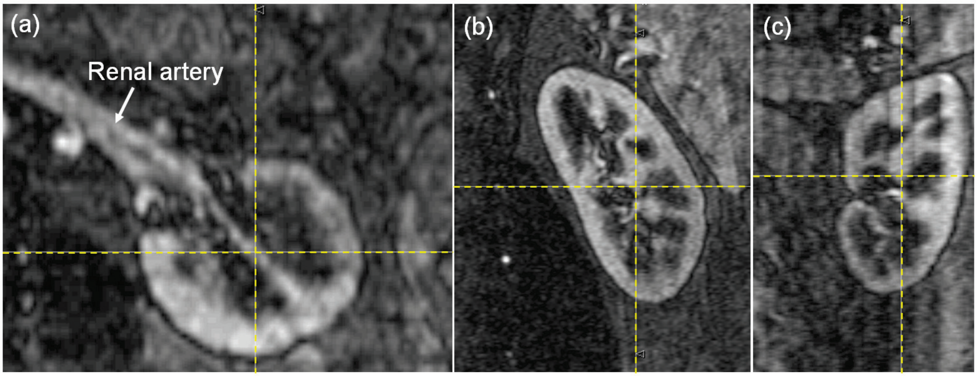


Fig. 4. Multiplanar reformatted images of the kidney as an axial plane (a), a sagittal plane (b) and a coronal plane (c). A synchronized cursor (yellow dashed lines) was used to navigate in the 3D data set. The reader moved the cursor near the centre of the kidney so that the renal artery appeared in the axial view.

In a second step, based on these three selected planes, the user roughly drew two groups of labelled seeds using a painting tool as shown in figure 5. The red seeds needed to be defined inside the kidney while the blue ones were drawn outside the kidney (Fig. 5). The FBA-GPU process was then started and automatically stopped after a fixed number of iterations. The corresponding 3D kidney segmentation result is illustrated in figure 6. The number of iterations was set to 100 (conservative value) for all kidney segmentations in order to ensure the convergence of the algorithm, as will be discussed further.

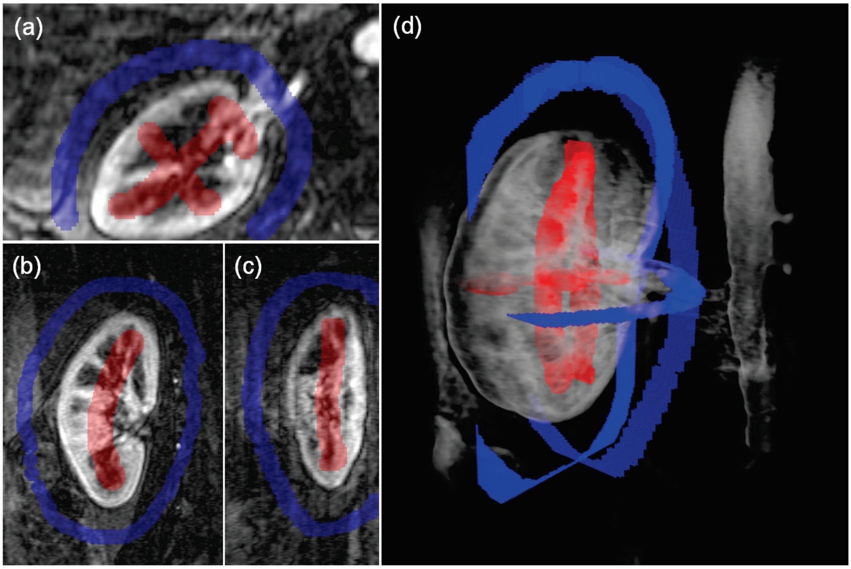


Fig. 5. Initialization of the segmentation algorithm on axial (a), coronal (b) and sagittal (c) planes. The user roughly drew two groups of labeled seeds using a painting tool; the red

seeds were defined inside the kidney while the blue ones were drawn outside the kidney. 3D view of seeds positioning is illustrated on (d).

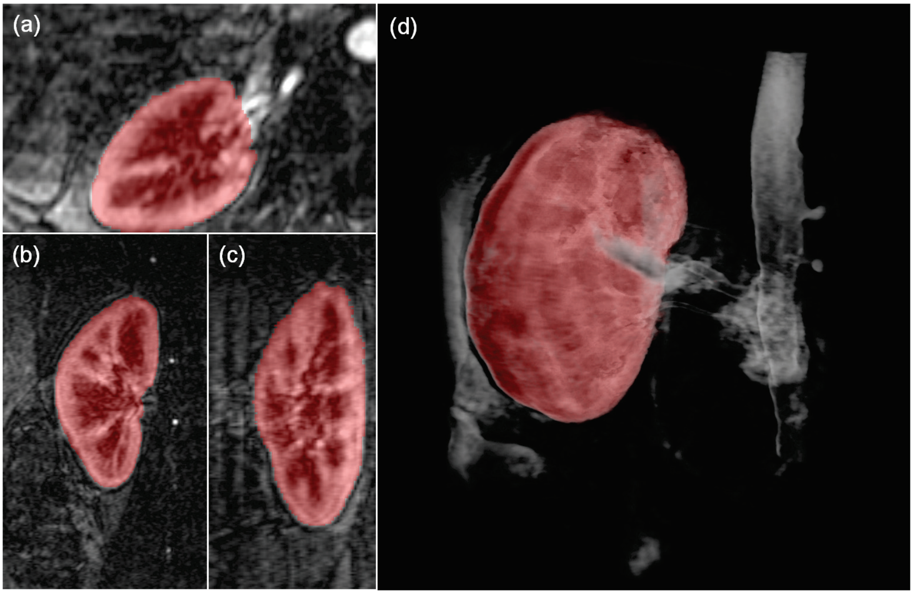


Fig. 6. Example of a segmented kidney. The red overlay represents our automatic segmentation results obtained by our FBAGPU approach after 100 iterations. Kidney VOI is shown in red on axial (a), coronal (b) and sagittal (c) planes, and in 3D view (d).

Reliability: The two readers performed the 3D segmentation of 40 kidneys (from the 20 MRA) using the FBA-GPU approach.The mean and standard deviation of the kidney volumes computed by each reader are summarized in Table 1.

Reader	Volume (mL) : Mean ± SD
Radiologist 1	140.6 ± 37.23
Radiologist 2	140.907 ± 37.222

Table 1. Descriptive statistics of 40 segmented kidney volumes

The inter-observer analysis between reader #1 and #2 is summarized on Table 2 below. These results indicate that inter-observer reproducibility for kidney volume measurement is excellent, ICC=0.998 (0.997-0.999) and show an absolute volume difference (in %of mean kidney volumes) of 1.16% ± 0.86. No statistically significant differences were observed (p=0.43) between reader #1 and #2 for the volume computation by our method.

Method	**Comparison between reader #1 and #2**			
	ICC (95%) CI (two-sided)	Volume diff (%) Mean ± SD	Abs volume diff (%) Mean ± SD	P value for Student T-test
3D CA-GPU	0.998 (0.997-0.999)	-0.21 ± 1.44	1.16 ± 0.86	0.43

Table 2. Statistical results for 40 renal volumes measurements

Measurement time: The whole segmentation time of the FBA-GPU method was subdivided as the initialization time needed to seeds positioning (1 min on average) and as the GPU computation time (3.6 s ± 0.9). This gave us a total estimated segmentation time lower then 1min 04s, on average, per kidney. For complicated MRA cases, more time was needed because it was necessary to add other seeds to ensure good segmentation results. However, the total time never exceeded 3 min.

Computational efficiency: In this section, we validated the hypothesis that our FBA-GPU algorithm was more efficient, in this segmentation context, than a CPU implementation of the method. To perform this comparative analysis we implemented the Dijkstra's all pair shortest path (APSP) algorithm in C++, using a binary heap. The algorithm was adapted to propagate the seed labels needed by the multi-label segmentation approach.

All CPU segmentation tests were performed automatically by using the previously saved labelled seeds, defined by the users during the first validation study.

The fundamental difference between the Dijkstra algorithm (DA) and FBA approach is that the DA implementation uses Priority Queue and defines a sorting function on the nodes. The algorithm converged to an optimal solution, since the search proceeded by expanding the lowest-cost vertices first, and optimal wave fronts, that worked their way out through the search space were generated ; an optimal decision at each step produced a globally optimal solution. In other words, the valid solution was only available once all vertices had been visited. In the FBA approach, each vertex was simply relaxed several times until the algorithm converged to the stationary global solution. This presents the advantage that, at each iteration, the solution is available and becomes closer to the optimal solution until the convergence is reached. The number of iterations needed to converge depends on the length of the longest geodesic path between a k-labelled vertex and all other vertices of the image or between two different k-labelled vertices. This means that the number of iterations needed to converge depends on the geometric complexity of the object to segment. To illustrate this we can say that the FBA approach needs many more iterations to segment an object like a maze than for an organ such as the kidney. Based on different segmentation contexts of human organs (liver, kidney, bones, aortic aneurysm …) we showed that the number of iterations, D, can be estimated by the equations (8) as a constant C that multiplies the maximal shortest Euclidian distance between the seed families defined inside (s) and outside (t) the organ respectively.

$$d_i = \min_{j=1}^{N}\left(\left\| s_i - t_j \right\|\right) \text{ and } D = C \max_{i=1}^{M}\left(d_i\right) \tag{8}$$

Where M and N represent the number of seeds having label s and t respectively.

For our segmentation application, a conservative value for D is given by C > 4.

GPU vs. CPU: The average of the DA-CPU computing time was 38.5s ± 8 while the corresponding FBA-GPU time for 100 iterations was 3.6s ± 0.9. The convergence rate of the FBA-GPU approach was evaluated by comparing the segmentation results obtained after a fixed number of iterations to the fully converged result given by the DA-CPU method. We show in figure 7 that after 10 iterations the difference between the intermediate FBA result and the ideal volume was 30.8% while after 50 iterations, this difference was drastically reduced to 0.12%. We confirmed that for all 40 kidney segmentations, no differences were observed after 60 iterations. This justifies that we fixed the number of iterations at 100 (conservative value) for all segmentation experiments presented in this paper. It should be noted that in the cases where the algorithm did not fully converge, a few iterations were added from this non converged state to reach the expected segmentation results.

GPU Benchmark: We were also interested in the evaluation of the performances of our FBA-GPU approach on different graphics hardware available in our imaging labs. To do this, the same version of the software was installed on seven PCs where we recorded GPU and CPU computation times needed to run the identical kidney segmentation experiment. Table 3 summarizes the measured computing times to run FBA on different graphics hardware and PCs. The GPU time is given separately as the setup time and as the FBA time. The setup-time is the time allocated for data transfer from/to the video memory and the FBA-time represents the time needed to perform 100 iterations of the Ford-Bellman algorithm.

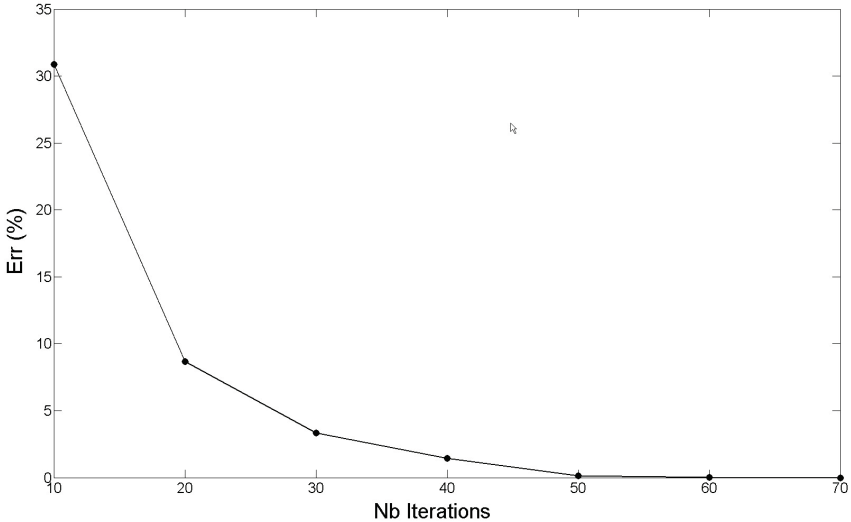


Fig. 7. Difference (Err) between the FBA segmented kidney volume and the converged solution at each 10 iterations. Err is represented in % of the converged volume.

	GPU time (ms)		
Graphic Card family	Setup-time	FBA-time	Speedup factor
ATI Radeon HD 4870	885	563	9,7
Nvidia GeForce GTX 260	812	922	5,9

ATI Radeon HD 3970	n/a	1211	4,5
NVIDIA GeForce 9800 GT	1032	1843	3,0
9800 gtx nvidia laptop	1061	1950	2,8
Radeon X1950 Pro	906	3047	1,8
NVIDIA GeForce 7950 GT	1419	5444	1,0

Table 3. Benchmarking of FBA running times (ms) on different graphics hardware. The speed up factor is given as the ratio of the FBA-time to the slowest FBA-time.

5. Discussion

The purpose of our work was to demonstrate that simple cellular automata algorithms implemented on low cost graphics hardware can perform efficient image processing tasks on N-D medical datasets. The implementation of the Ford-Bellman's algorithm was in a first time applied to perform the watershed transform, and in a second time, to perform seeded organ segmentation.

5.1 Watershed transform

A comparison between our GPU watershed implementation and an efficient CPU implementation of the watershed transform showed that a GPU approach based on a massively parallel implementation of the Ford-Bellman algorithm can outperform efficient CPU implementation of the watershed algorithm. Computing a watershed transform can be an ideal application case for the FBA-GPU approach because the algorithm is initialized by a large number of seeds which are defined as local minima in a specific neighborhood. As the local minimum are defined on the whole image dataset, only few iterations are needed to reach a converged result. However, the number of iterations must be increased in the case of ideal images such these having large plateaus filled with the same gray level, or with images smoothed by a large Gaussian kernel. The reason is that the local minimum are more distant from each other and more iterations are then needed to find the converged watershed lines. We showed that the FBA-GPU approach is particularly efficient over CPU implementation when it is applied to real noisy large images such as medical datasets. In other cases, an hybrid GPU-CPU approach can be considered.

5.2 Seeded segmentation

The second purpose of this study was to demonstrate that our FBA-GPU approach can efficiently be used to perform automatic organ segmentation in 3D medical datasets with a high degree of reliability and accuracy. The high reproducibility of renal volumes segmentation (inter-observer correlation ICC=0.998) combined to its accuracy (mean absolute error < 1.5%) makes this method valid for clinical use. The low difference between renal volumes segmented by two independent readers also indicates that our FBA method is robust to manual seed selection. The maximal absolute error between the two readers was 3.12%, which indicates that the precision of renal volume measurements is weakly affected by the variability of image parameters, such as anisotropic resolution of MRI scans and poor contrast between kidney parenchyma and background in the presence of severe renovascular disease as illustrated on figure 1b. Moreover, the short time required by the whole segmentation process (< 2 min) shows that our method can be used in clinical routine. In a second time we conduct a study to evaluate the computational efficiency of our GPU Ford-Bellman approach compared to a CPU implementation of the Dijkstra's algorithm. For all kidney segmentation (100 iterations) we see a speedup of 10 between the GPU and CPU running time. This speedup factor can easily be increased to 50 by using a more actual graphic hardware as presented in the next GPU benchmark section. Our GPU-based version of the FBA algorithm is simple to implement and presents itself as an alternative and optimized approach to perform graph-based segmentation in regards to other proposed approaches (Vineet and Narayanan, 2008, Bolz et al., 2003, Fung and Mann, 2008, Gernot et al., 2007, Koutis, 2008).

Our work was voluntarily designed as a compromise between computational efficiency and hardware compatibility. Indeed, our FBA approach was implemented using OpenGL Cg language on different brands of low cost graphics cards.

We also benchmarked the FBA on seven different graphics hardware (Table 3), available on standard PCs in our medical imaging department, in order to highlight (non exhaustive) increasing performances and speedups of GPUs between past and newer low cost hardware generations. On one side, by comparing the FBA-time on different GPUs, a speedup factor of 10 was found between the slowest and fastest tested graphics hardware and a speedup greater than 5 between the GPU used in this study and the ATI Radeon HD 4870., The setup-time was overall in the same range for all graphics hardware. On the other side, the CPU computing times were slightly the same for all PCs available in our labs.

5.3 Are GPUs better than CPUs ?

The GPU's rapid increase in both programmability and capability has spawned a research community that has successfully mapped a broad range of computationally demanding, complex problems to the GPU (Owens et al., 2008). While GPUs are a compelling alternative to traditional microprocessors in high-performance computer systems, they can also be seen as a complementary solution to CPU approaches. In our case, using the FBA-GPU approach to compute the shortest path inside a long and tortuous organ, such as vessels or the colon, gives poor performances compared to the DA-CPU approach. Yet, we show that in our study the FBA method is more efficient to perform multi label segmentation of the kidney. For this reason, we suggest that performances comparison between GPU and CPU approaches must be carefully regarded from a global point of view including the application context and the complexity of the algorithms to be implemented.

However, there are some limitations on the use of GPUs. The first limitation concerns the setup-time which represents the minimal time needed for the data transfer from/to the video memory. Since the video memory is accessed through the PCI Express lane it is much slower than the RAM access. It seems clear that the best use case is achieved by loading once the data to the graphics card's memory and to compute as much as possible there. For this reason it is not efficient to test the convergence of the FBA at each iteration or after a low number of iterations. A second limitation concerns the restriction on the type of data that can be used. Indeed, there is no native double precision float in GPU (this could change in a near future) and this limitation can be so important in some computational problems that straightforward usage of GPU is excluded.

6. Conclusion

We have presented a GPU-based Cellular Automaton to perform efficient image processing tasks on large image datasets. Our work is based on the Ford-Bellman's shortest paths algorithm which is first applied to compute the watershed transform and secondly to perform automatic multi label segmentation of organs in N-D medical images with minimal user interaction for initialization. Validation of this method on MRA examinations showed high inter-observer reproducibility and accuracy that allows the method to be used in clinical routine. Our implementation of the FBA in the form of a CA is simple, efficient and straightforward, and can be implemented in low cost vendor-independent graphics cards. Our work was strongly motivated by the fact that the processing power has clearly shifted from the CPU to the GPU. The experimental testing performed on MRA datasets confirms the expected gain in performance with GPU implementation. To our knowledge, we are the first to propose a GPU implementation of FBA as a cellular automaton to perform the watershed transform and seeded segmentation on ND images.

7. References

Aharon, S., Grady, L. & Schiwietz, T. (2005) GPU accelerated isoperimetric algorithm for image segmentation, digital photo and video editing. Google Patents.

Alonso Atienza, F., Requena Carrión, J., García Alberola, A., Rojo Álvarez, J. L., SÁnchez Muñoz, J. J., Martínez SÁnchez, J. & Valdés Chávarri, M. (2005) A Probabilistic Model of Cardiac Electrical Activity Based on a Cellular Automata System. *Revista Española de Cardiología (Internet),* **58,** 41-47.

Bai, X. & Sapiro, G. (2007) A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. pp. 1-8.

Bellman, R. (1956) *ON A ROUTING PROBLEM,* Defense Technical Information Center.

Beucher, S. & Lantuejoul, C. (1979) Use of watersheds in contour detection. In: *International Workshop on Image Processing.* Vol. 17, pp. 2.1–2.12.

Bolz, J., Farmer, I., Grinspun, E. & Schröoder, P. (2003) Sparse matrix solvers on the GPU: conjugate gradients and multigrid. pp. 917-924. ACM New York, NY, USA.

Boykov, Y. & Funka-Lea, G. (2006) Graph Cuts and Efficient NDImage Segmentation. *International Journal of Computer Vision,* **70,** 109-131.

Boykov, Y. & Jolly, M. P. (2000) Interactive Organ Segmentation Using Graph Cuts. *LECTURE NOTES IN COMPUTER SCIENCE***,** 276-286.

Chefd'hotel, C. & Sebbane, A. (2007) Random Walk and Front Propagation on Watershed Adjacency Graphs for Multilabel Image Segmentation. pp. 1-7.

Digabel, H. & Lantuejoul, C. (1977) Iterative algorithms. JL Chermant Eds. In: *Actes du Second Symposium Europeen d'Analyse Quantitative des Microstructures en Sciences des Materiaux, Biologie et Medecine.* Riederer Verlag, Caen, France.

Dijkstra, E. W. (1959) A note on two problems in connexion with graphs. *Numerische Mathematik,* **1,** 269-271.

Dixit, N., Keriven, R., Paragios, N. & Graphique, P. (2005) GPU-Cuts: Combinatorial Optimisation, Graphic Processing Units and Adaptive Object Extraction GPU-Cuts: Segmentation d'Objects.

Eom, S., Shin, V. & Ahn, B. (2007) Cellular Watersheds: A Parallel Implementation of the Watershed Transform on the CNN Universal Machine. *IEICE TRANSACTIONS on Information and Systems,* **90,** 791-794.

Even, S. (1979) Graph Algorithms. Rockville. *MD: Computer Science Press,* **249**.

Fairfield, J. (1990) Toboggan contrast enhancement for contrast segmentation. Vol. 1.

Falcão, A. X., Stolfi, J. & de Alencar Lotufo, R. (2004) The Image Foresting Transform: Theory, Algorithms, and Applications. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE***,** 19-29.

Ford Jr, L. R. (1956) NETWORK FLOW THEORY.

Fung, J. & Mann, S. (2008) Using graphics devices in reverse: GPU-based Image Processing and Computer Vision. pp. 9-12.

Ganguly, N., Sikdar, B. K., Deutsch, A., Canright, G. & Chaudhuri, P. P. (2003) A survey on cellular automata. *Dresden University of Technology, Technical Report Centre for High Performance Computing*.

Gardner, M. (1970) Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game 'Life'. *Scientific American,* **223,** 120-123.

Gernot, Z., Christian, T., Ivo, I., Marcus, M., Art, T. & Hans-Peter, S. (2007) GPU-based light wavefront simulation for real-time refractive object rendering. In: *ACM SIGGRAPH 2007 sketches.* ACM, San Diego, California.

Gobron, S., Devillard, F. & Heit, B. (2007) Retina simulation using cellular automata and GPU programming. *Machine Vision and Applications,* **18,** 331-342.

Grady, L. & Funka-Lea, G. (2004) Multi-label Image Segmentation for Medical Applications Based on Graph-Theoretic Electrical Potentials. *LECTURE NOTES IN COMPUTER SCIENCE***,** 230-245.

Kauffmann, C. & Piche, N. (2008) Cellular automaton for ultra-fast watershed transform on GPU. In: *ICPR 2008.* pp. 1-4. Tampa bay, FL, USA.

Konushin, V., Vezhnevets, V. & Moscow, R. (2006) Interactive Image Colorization and Recoloring based on Coupled Map Lattices. pp. 231–234.

Koutis, I. (2008) Faster Algebraic Algorithms for Path and Packing Problems. In: *Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part I.* Springer-Verlag, Reykjavik, Iceland.

Lin, Y. C., Tsai, Y. P., Hung, Y. P. & Shih, Z. C. (2006) Comparison between immersion-based and toboggan-based watershed image segmentation. *IEEE TRANSACTIONS ON IMAGE PROCESSING,* 15, 632-640.

Meyer, F. (1994) Topographic distance and watershed lines. *Signal Processing,* **38,** 113-125.

Mortensen, E. N. & Barrett, W. A. (1998) Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing,* **60,** 349-384.

N. Moga, A., Cramariuc, B. & Gabbouj, M. (1998) Parallel watershed transformation algorithms for image segmentation. *Parallel Computing,* **24,** 1981-2001.

Nepomniaschaya, A. S. (2001) An Associative Version of the Bellman-Ford Algorithm for Finding the Shortest Paths in Directed Graphs. *LECTURE NOTES IN COMPUTER SCIENCE***,** 285-292.

Noguet, D. (1997) A massively parallel implementation of the watershed based on cellular automata. In: *IEEE International Conference on Application-Specific Systems, Architectures and Processors.* pp. 791-794.

Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E. & Phillips, J. C. (2008) GPU Computing. In: *Proceedings of the IEEE, 96(5), May.*

Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Kruger, J., Lefohn, A. E. & Purcell, T. J. (2007) A survey of general-purpose computation on graphics hardware. Vol. 26, pp. 80-113. Blackwell Publishing Ltd.

Protiere, A. & Sapiro, G. (2007) Interactive Image Segmentation via Adaptive Weighted Distances. *IEEE TRANSACTIONS ON IMAGE PROCESSING,* **16,** 1046.

Qu, Y., Wong, T. T. & Heng, P. A. (2006) Manga colorization. *Proceedings of ACM SIGGRAPH 2006,* **25,** 1214-1220.

Roerdink, J. & Meijster, A. (2000) The watershed transform: Definitions, algorithms and parallelization strategies. *Mathematical Morphology,* **41,** 187-S128.

Rother, C., Kolmogorov, V. & Blake, A. (2004) " GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG),* **23,** 309-314.

Sinop, A. K. & Grady, L. (2007) A Seeded Image Segmentation Framework Unifying Graph Cuts And Random Walker Which Yields A New Algorithm. pp. 1-8.

Vezhnevets, V., Konouchine, V. & Moscow, R. (2005) “GrowCut”-Interactive Multi-Label NDImage Segmentation By Cellular Automata. pp. 150–156.

Vincent, L. & Soille, P. (1991) Watersheds in digital spaces: an efficient algorithm based onimmersion simulations. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* **13,** 583-598.

Vineet, V. & Narayanan, P. J. (2008) CUDA cuts: Fast graph cuts on the GPU. pp. 1-8.

Volkov, V. & Demmel, J. Using GPUs to Accelerate the Bisection Algorithm for Finding Eigenvalues of Symmetric Tridiagonal Matrices.

Von Neumann, J. & Burks, A. W. (1966) *Theory of self-reproducing automata,* University of Illinois Press Urbana.

Wolfram, S. (2002) A new kind of science. . *Champaign, IL: Wolfram Media*.

Xu, N., Ahuja, N. & Bansal, R. (2007) Object segmentation using graph cuts based active contours. *Computer Vision and Image Understanding,* **107,** 210-224.

Yatziv, L., Bartesaghi, A. & Sapiro, G. (2006) O (N) implementation of the fast marching algorithm. *Journal of Computational Physics,* **212,** 393-399.

Yin, L., Jian, S., Chi-Keung, T. & Heung-Yeung, S. (2004) Lazy snapping. In: *ACM SIGGRAPH 2004 Papers.* ACM, Los Angeles, California.

Zhao, Y. (2008) Lattice Boltzmann based PDE solver on the GPU. *The Visual Computer,* **24,** 323-333.

Figure-Ground Discrimination and Distortion-Tolerant Recognition of Color Characters in Scene Images

Toru Wakahara
Faculty of Computer and Information Sciences, Hosei University
3-7-2 Kajino-cho Koganei-shi
Tokyo 184-8584 JAPAN

1. Introduction

Recently, recognition of web documents and characters in natural scenes has emerged as a hot, demanding research field (Doermann et al., 2003). In particular, recognition of characters in scene images with a wide variety of image degradations and complex backgrounds poses the following two key problems.

The first problem is figure-ground discrimination (Herault & Horaud, 1993) or correct binarization of color characters in scene images as a crucial step to the success of subsequent recognition. Most of the binarization methods are based on global, local/adaptive or multi-stage selection of threshold (Trier & Jain, 1995; Wolf et al., 2002; Wu & Amin, 2003). However, color-based binarization has not yet been fully addressed (Miene at al., 2001).

The second problem is distortion-tolerant character recognition under the condition of a small sample size because there is only a limited quantity of data against a wide variety of fonts and image degradations. Hence, we cannot make good use of statistical pattern recognition techniques, including sophisticated discriminant functions, neural networks, support vector machines or kernel methods.

Regarding the first problem we propose three promising approaches. The first approach is application of genetic algorithms (GA) to a combinatorial problem of determining an optimal filter sequence that correctly binarizes an input image (Kohmura & Wakahara, 2006). The filter bank contains a number of typical image processing filters as applied to one of the RGB color planes and logical/arithmetic operations between two color planes. The second approach is selection of a maximum separability axis in the RGB color space and an appropriate threshold on the axis for binarizing an input image as the two-category classification problem (Yokobayashi & Wakahara, 2006). Here, the key idea for solving this problem is application of the Otsu’s criterion (Otsu, 1979) to the distribution of color pixels of the input image projected onto every possible axis in the RGB color space. The third approach is application of K-means clustering in the HSI color space to color pixels of the input image, generation of temporally binarized images via every dichotomization of *K* clusters, and their classification into two categories: character and non-character (Kato &

Wakahara, 2009). Here, a character vs. non-character classification is effectively implemented by support vector machines (SVM).

Regarding the second problem we try to make use of elastic image matching techniques (Uchida & Sakoe, 2005). Here, we apply two kinds of distortion-tolerant template matching based on the deterministic character deformation models. The first one is our global affine transformation (GAT) correlation technique (Wakahara et al., 2001). The GAT correlation absorbs distortion expressible by affine transformation by determining optimal affine parameters that maximize a normalized cross-correlation value between an affine-transformed input image and a template. In particular, image matching by means of normalized cross-correlation was shown to be robust against image blurring and additive random noise (Sato, 2000). The second one is the well-known tangent distance (TD) (Simard et al., 1993). The tangent distance absorbs distortion expressible by a linear combination of predefined geometric and topographical transformations as applied to both an input image and each template.

We show experimental results made on the public ICDAR 2003 robust OCR dataset (ICDAR Datasets, 2003) containing a wide variety of single-character images in natural scenes.

In Section 2, we explain ICDAR 2003 robust OCR dataset. Section 3 proposes three kinds of techniques for figure-ground discrimination or correct binarization of color characters in scene images. In Section 4, we describe two competing techniques of distortion-tolerant image matching for recognizing binarized characters. Section 5 shows experimental results. Section 6 is devoted to discussion and future work.

2. ICDAR 2003 robust OCR dataset

Several datasets used in ICDAR 2003 robust reading competitions (Lucas et al., 2003) are available for download from the website (ICDAR Datasets, 2003). We use the robust OCR dataset containing JPEG single-character images in natural scenes. In particular, we select a total of 698 images from "Sample" subset.

Figure 1 shows examples of images with a variety of image degradations and complex backgrounds.

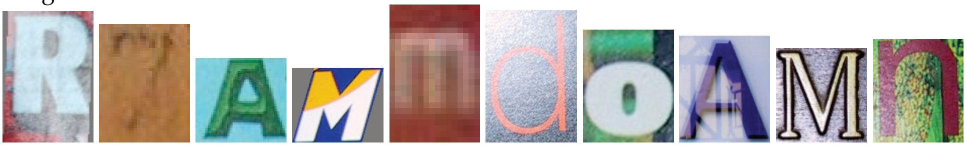


Fig. 1. Examples of images used in our experiments.

3. Figure-ground discrimination of color characters in scene images

In this section, we propose three kinds of techniques for figure-ground discrimination or correct binarization of color characters: determination of an optimal sequence of filters for binarization using GA, binarization using a maximum separability axis in a color space, and K-means clustering in a color space and figure-ground discrimination by SVM.

3.1 Determination of an optimal sequence of filters for binarization using GA

This technique is for binarization of color characters in scene images using genetic algorithms (GA) to search for an optimal sequence of filters through a filter bank. The filter bank contains simple image processing filters as applied to one of the RGB color planes and logical/arithmetic operations between two color planes. First, we classify images extracted from the ICDAR 2003 robust OCR dataset into several groups according to degradation categories. Then, in the training stage, by selecting training samples from each degradation category we apply GA to the combinatorial optimization problem of determining a filter sequence that maximizes the average fitness value calculated between the filtered training samples and their respective target images ideally binarized by humans. Finally, in the testing stage, we apply the optimal filter sequence to binarization of remaining test samples.

3.1.1 Grouping of character images according to degradation categories

By carefully examining a total of 698 images from "Sample" subset we classified them into six groups according to degradation categories: clear, background with pattern, character with pattern, character with rims, blurring, and nonuniform lighting. The criterion upon how to classify degradation categories is rather subjective just to show a wide variety of binarization problems. In practical application degradation categories should be selected automatically, and, also, it is necessary to automatically decide which degradation category a given input image belongs to.

Figure 2 shows examples of images in six degradation categories.

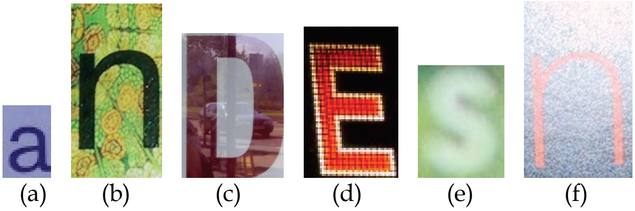


Fig. 2. Examples of images in six degradation categories. (a) Clear. (b) Background with pattern. (c) Character with pattern. (d) Character with rims. (e) Blurring. (f) Nonuniform lighting.

3.1.2 Image transformation by a sequence of filters and filter bank

Figure 3 shows a total flow of image transformation using a sequence of filters as applied to an original image so that a filtered image approximates its target image ideally binarized by humans as closely as possible.

We use GA in search of an optimal sequence of filters, equivalent to the image transformation L^*, while L specifies the ideal binarization. The degree of approximation of L^* to L is evaluated in terms of the fitness value calculated between target and filtered images.

Table 1 shows a list of filters in our filter bank. These filters are not sophisticated but rather primitive ones (Gonzalez & Woods, 2000).

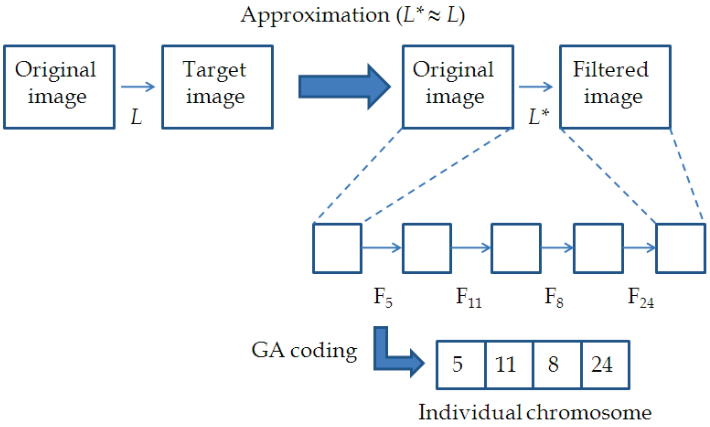


Fig. 3. Total flow of image transformation by a sequence of filters.

No.	Filter name	Function
1	Mean	local mean in a 3×3 window
2	Min	local min in a 3×3 window
3	Max	local max in a 3×3 window
4	Sobel (1)	horizontal differential
5	Sobel (2)	vertical differential
6	Sobel (3)	the norm of differential
7	LightEdge	Laplacian
8	DarkEdge	Laplacian + 255
9	Erosion	morphological erosion
10	Dilation	morphological dilation
11	Inversion	255 – g; g = pixel value
12	Logical sum	max of two color planes
13	Logical product	min of two color planes
14	Algebraic sum	sum of two color planes – their product / 255
15	Algebraic product	product of two color planes / 255
16	Bounded sum	g = sum of two color planes; if g > 255, g = 255
17	Bounded product	product of two color planes – 255; if g < 0, g = 0

Table 1. List of filters in a filter bank.

Each filtering operation is specified in either of the following two ways.
One way is to select one of one-operand filters from no. 1 to no. 11 and one of the RGB color planes to which the selected filter is applied. The other way is to select one of two-operand filters from no. 12 to no. 17 and two of the RGB color planes to which the selected filter is applied, where the filtering result is overwritten onto either of the two color planes.
Hence, the total number of filtering operations is 3 × 11 plus 6 × 6, and equals sixty nine.

3.1.3 Gene encoding and specifications of GA

We use GA (Goldberg, 1989) to search for an optimal filter sequence that transforms an original image so as to yield the maximum fitness value against its target image ideally binarized by humans. Here, the fitness value serves as a similarity measure.

As described in Section 3.1.2, the total number of filtering operations equals sixty nine. We specify each filtering operation by an ID number selected from one to sixty nine. Hence, a filter sequence or a chromosome is encoded as a string of 8-bit integers. Also, we set the maximum number of constituent filters in a chromosome at 80.

The initial population of 300 is randomly generated. We adopt the roulette selection rule based on the fitness values in each generation. We use the modified one-point crossover method that exchanges respective tails with the rate of 80%. Mutation also exchanges every constituent ID number within a chromosome for a different one with the rate of 0.1%.

Finally, we stop the GA process when the maximal fitness value of an elite chromosome exceeds the threshold value of 0.9 or when the number of generations arrives at the predetermined number of 800.

Here, by denoting target and filtered images by $\boldsymbol{T} = \{ T_k(x, y) \}$ and $\boldsymbol{F} = \{ F_k(x, y) \}$ (k = R, G, B), respectively, we calculate a fitness value, $f(\boldsymbol{T}, \boldsymbol{F})$, between target and filtered images by

$$f(\boldsymbol{T},\boldsymbol{F}) = 1 - \frac{\sum_{k=\mathrm{R,G,B}} \sum_{x=1}^{W_x} \sum_{y=1}^{W_y} \left| T_k(x,y) - F_k(x,y) \right|}{3 \times W_x \times W_y \times 255}, \tag{1}$$

where W_x and W_y specify width and height of the image, respectively.

Figure 4 shows examples of binarization of training samples belonging to the degradation category "nonuniform lighting" using an optimal sequence of filters determined via GA.

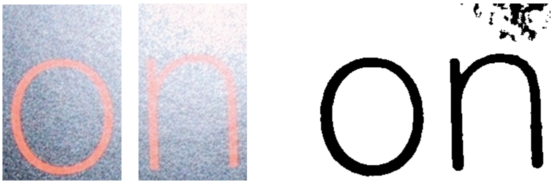

(a) (b)

Fig.4. Examples of binarization of training samples belonging to the degradation category "nonuniform lighting" using an optimal sequence of filters determined via GA. (a) Input images. (b) Binarized images.

It is to be noted that this technique provides us with an optimal sequence of filters for binarization of color characters in each of predetermined degradation categories. In other words, this technique cannot generate a single, all-purpose filter sequence to deal with a wide variety of image degradations and complex backgrounds. In this sense, we can say that this approach is very powerful when we know in advance that all of input images being considered belong to a particular kind of degradation category.

3.2 Binarization using a maximum separability axis in a color space

This technique is for binarization of color characters in scene images following two steps. The first step is temporary binarization by selecting one optimal projection axis with a maximum two-class separability in the RGB color space and an approriate threshold on the axis. Here, we apply Otsu's criterion to a two-class classification problem. The second step is figure-ground determination based on the figure-to-ground ratio on the image periphery and common characteristics that a character pattern should have.

3.2.1 Temporary binarization via Otsu's criterion in the RGB color space

First, color points of all pixels in an input image are projected onto an arbitrarily chosen axis in the RGB color space. Here, we adopt spherical polar coordinates, (r, θ, φ), in 3D color space, and try all axes with angles, (θ, φ), selected at intervals of one degree, respectively. Namely, a total of 180 × 180 axes in 3D color space are considered.

Second, for each point distribution on a chosen axis we calculate maximum between-class separability by setting an optimal threshold according to the Otsu's binarization technique (Otsu, 1979). We know that this idea is also based on the well-known Fisher criterion (Bishop, 2006) as applied to a two-class classification problem. Namely, the between-class separability, S, is defined as the difference of two means normalized by the averaged variance on the chosen axis according to

$$S = \frac{(m_1 - m_2)^2}{\sigma_1^2 + \sigma_2^2} \quad \rightarrow \quad \text{max for a bisection on the axis.} \tag{2}$$

Finally, we select the axis that gives the largest between-class separability and the corresponding threshold for temporary binarization of the input image. Here, from the viewpoint of figure-ground discrimination it is clear that this binarization result is only temporary because there are two possibilities of either class being a character.

Figure 5 shows projection of pixels onto a chosen axis in the RGB color space.

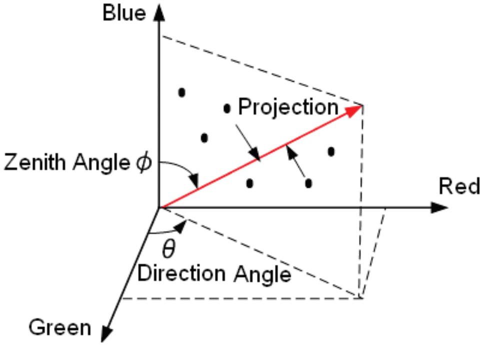


Fig. 5. Projection of pixels onto a chosen axis in the RGB color space.

3.2.2 Figure-ground determination using common characteristics of characters

We assume that an input image contains only one character and a character belongs to alphanumeric characters as shown in Fig. 1.

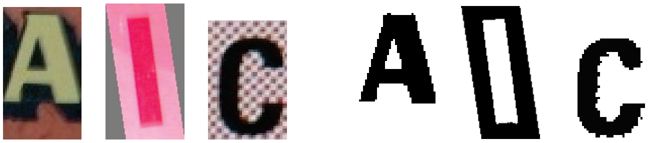
Granting this assumption, we can enumerate common characteristics that such single-character images should have as follows.

(1) The majority of pixels on the image periphery belong not to a character but to a background.
(2) The number of connected components in a character is one except "i" and "j."
(3) The width of a character is narrower than that of a background.

Based on these common characteristics we propose a procedure for figure-ground determination written in a pseudo-code as shown below.

```
If the figure-to-ground ratio on the image periphery is less than a threshold value of Th, then
    consider the present binarized image as the correct one and goto END.
Else if the figure-to-ground ratio on the image periphery is more than the inverse of Th, then
    consider the reversed image as the correctly binarized one and goto END.
Else if the width of a figure is narrower than that of a ground, then consider the present
    binarized image as the correct one and goto END. Here, we define the width of a figure or
    a ground in the image as twice the number of erosion operations (Gonzalez & Woods,
    2000) applied to the corresponding region until it vanishes.
Else consider the reversed image as the correctly binarized one and goto END.
END: Select and save only the maximum connected component of the figure and output the
    resultant image as the final result of figure-ground discrimination.
```

Figure 6 shows examples of binarization using a maximum separability axis in a color space.



(a) (b)

Fig. 6. Examples of binarization using a maximum separability axis in a color space. (a) Input images. (b) Binarized images.

It is to be noted that this technique assumes that a character in an input image is made up of color pixels with similar values in the RGB color space and, hence, binarization is handled correctly as a two-class classification problem using only color information. Therefore, this technique is not well suited to deal with multi-color characters and/or characters with nonuniform backgrounds.

3.3 K-means clustering in a color space and figure-ground discrimination by SVM

This technique is for binarization of color characters in scene images following three steps. The first step applies K-means clustering in the HSI color space to points in an input image, and, then, generates a set of tentatively binarized images by every possible dichotomization of a total of K clusters or subimages. The second step calculates the degree of character-likeness of each tentatively binarized image by SVM in an appropriately chosen feature space. In advance, SVM is trained to determine whether and to what degree each binarized image represents a character or non-character. The third step outputs the binarized image with the maximum degree of character-likeness as an optimal binarization result.

3.3.1 K-means clustering in the HSI color space

First, values of R, G, and B in the RGB color space are converted to values of H, S, and I in the HSI color space, where H, S, and I represent hue, saturation, and intensity, respectively (Gonzalez & Woods, 2000). In particular, we scale each value of H, S, and I to range from 0 to 255 as follows.

$$\begin{aligned}
&I = \max(R,G,B), \quad m = \min(R,G,B),\\
&\mathit{if} \quad I = 0 \ \ \mathit{or} \ \ I = m \quad \mathit{then} \quad S = 0, \quad H = \mathit{indefinite} \ ,\\
&\mathit{else} \quad S = \frac{I-m}{I} \times 255,\\
&\qquad r = \frac{I-R}{I-m}, \ g = \frac{I-G}{I-m}, \ b = \frac{I-B}{I-m},\\
&\qquad \mathit{if} \quad R = I \quad \mathit{then} \quad h = \frac{\pi}{3}(b-g), \qquad \mathit{if} \quad G = I \quad \mathit{then} \quad h = \frac{\pi}{3}(2+r-b),\\
&\qquad \mathit{if} \quad B = I \quad \mathit{then} \quad h = \frac{\pi}{3}(4+g-r), \qquad \mathit{if} \quad h < 0 \quad \mathit{then} \quad h = h + 2\pi,\\
&\quad H = h \times 255.
\end{aligned} \tag{3}$$

When an input image of size $W_x \times W_y$ is given, a total of $W_x \times W_y$ points corresponding to those pixels are scattered in the HSI color space.

Second, K-means clustering is applied to a total of $W_x \times W_y$ points in the HSI color space to generate K clusters, where a number of clusters, K, is determined in advance. The K-means clustering algorithm or nearest mean reclassification algorithm (Bishop, 2006) is as follows.

Step 1: Select K points at random from a total of $W_x \times W_y$ points scattered in the HSI color space as initial cluster centers, $\{ \boldsymbol{\mu}_k^{(\tau=0)} \}$, ($k$ = 1, …, K). τ specifies an iteration number. Then, assign each of $W_x \times W_y$ points to its nearest cluster center among $\{ \boldsymbol{\mu}_k^{(\tau=0)} \}$, ($k$ = 1, …, K), and a set of points assigned to the same cluster center forms one cluster.

Step 2: Compute a mean vector of each cluster and set the mean vector as an update on its cluster center. Then, $\tau = \tau + 1$, and cluster centers thus updated are denoted by $\{ \boldsymbol{\mu}_k^{(\tau)} \}$, ($k$ = 1, …, K).

Step 3: Each point is re-assigned to a new set according to which is the nearest cluster center among $\{ \boldsymbol{\mu}_k^{(\tau)} \}$, ($k$ = 1, …, K), and each new set of points corresponds to a cluster. If there is no further change in the grouping of the data points, output the present K clusters as the clustering result and stop. Otherwise, go to Step 2.

By inverse projection of a set of points forming each cluster in the HSI color space onto a 2D image plane, respectively, we obtain a total of K subimages the sum of which is equivalent to the input image.

3.3.2 Generation of tentatively binarized images by dichotomization of K subimages

We dichotomize K subimages into two groups, and set values of pixels belonging to the one group at 0 (black) and the other group at 255 (white). As a result, we obtain one binarized image, where black pixels represent figure and white pixels represent background.

By considering every possible dichotomization of K subimages we can generate multiple tentatively binarized images the total number of which, N_{binary}, is given by

$$N_{binary} = \sum_{i=1}^{K-1} {}_K C_i = 2^K - 2, \tag{4}$$

where ${}_K C_i$ denotes a binomial coefficient.

Figure 7 shows one example of generation of tentatively binarized images from an input image.

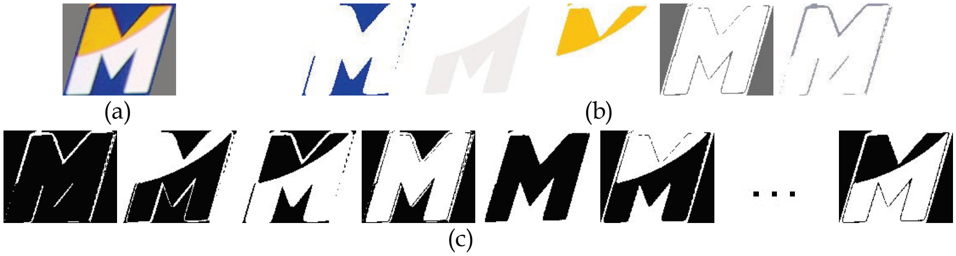


Fig. 7. One example of generation of tentatively binarized images from an input image. (a) An input image. (b) K subimages obtained by K-means clustering (K = 5). (c) (2^K – 2) tentatively binarized images.

From Fig.7, it is seen that a correctly binarized image is included in a set of tentatively binarized images even when a character is represented by multiple colors in the original input image.

It is to be noted that this technique has the possibility of correctly binarizing multi-color characters and/or characters with complex backgrounds. However, it is necessary to devise a means of selecting a correctly binarized image from a set of tentatively binarized images. Also, the total number of clusters, K, should be large enough to just guarantee that a correctly binarized image will be included in a set of (2^K – 2) tentatively binarized images.

3.3.3 Feature extraction from a binary image for estimating character-likeness

We extract a feature vector from a binary image so that a feature vector should represent a kind of character-likness as much as possible. Selection of a good feature vector is a clue to the success of SVM that determines whether and to what degree each binary image represents a character or non-character in the given feature space.

As preprocessing, position and size normalization is applied to each binary image by using moments (Casey, 1970). Namely, the center of gravity of black pixels is shifted to the center of the image, and the second moment around the center of gravity is set at the predetermined value. Here, we set a size of a preprocessed binary image at 80 × 120 pixels.

Then, we extract three kinds of feature vectors all of which are well-known in the field of character recognition: mesh feature, direction code histogram feature, and weighted direction code histogram feature.

Mesh feature:

We divide the input binary image into a total number of 8 × 12 (= 96) square blocks each of which has a size of 10 × 10 pixels and, then, calculate the percentage of black pixels in each of blocks. Finally, those measurements together form the 96-dimensional mesh feature vector.

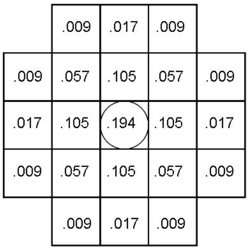
Direction code histogram feature:

One of 4-directional codes, i.e., H (horizontal), R (right-diagonal), V (vertical), and L (left-diagonal), is assigned to every contour pixel of black regions. Then, we divide the input binary image into a total number of 4 × 6 (= 24) square blocks each of which has a size of 20 × 20 pixels. Finally, in each block we count the number of contour pixels assigned to H, R, V, and L, respectively, and their measurements together form the 96-dimensional direction code histogram feature vector.

Weighted direction code histogram feature:

In order to improve robustness against shape distortion we introduce a locally weighted sum of the direction code histogram feature (Kimura et al., 1997). First, we divide the input binary image into a total number of 8 × 12 (= 96) square blocks each of which has a size of 10 × 10 pixels. Hence, we obtain the 384-dimensional direction code histogram feature vector. Then, using a locally weighted sum around each block taken at intervals of two blocks, both horizontally and vertically, the dimension of the feature vector is reduced from 384 to 96. As a result, we obtain the 96-dimensional weighted direction code histogram feature vector.

Figure 8 shows a Gaussian mask for generating the weighted direction code histogram feature.



	.009	.017	.009	
.009	.057	.105	.057	.009
.017	.105	.194	.105	.017
.009	.057	.105	.057	.009
	.009	.017	.009	

Fig. 8. A Gaussian mask for generating the weighted direction code histogram feature. A circle denotes the loci of a block around which a locally weighted sum is calculated.

3.3.4 Discrimination between character and non-character via SVM

The support vector machines, SVM, (Vapnik, 2000) map the input feature vectors, $\boldsymbol{x}$, into a high-dimensional feature space, $\boldsymbol{\Phi}(\boldsymbol{x})$, through some nonlinear mapping, chosen *a priori*. In this space, an optimal separating hyperplane that maximizes the margin is constructed.

The training data set comprises N input feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with corresponding target values $y_1, \ldots, y_N$ where $y_i \in \{ -1, +1 \}$, and new data points $\boldsymbol{x}$ are classified according to the sign of $f(\boldsymbol{x})$ given by

$$\begin{aligned} f(\boldsymbol{x}) &= \sum_{i=1}^{N} \alpha_i \, y_i \left(\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}) \right) - b \\ &= \sum_{i=1}^{N} \alpha_i \, y_i \, K\left(\boldsymbol{x}_i, \boldsymbol{x}\right) - b, \end{aligned} \tag{5}$$

where $(\boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{y}))$ is an inner product in the high-dimensional feature space, and is replaced with the kernel function $K(\boldsymbol{x}, \boldsymbol{y})$ by making use of the kernel trick.

Non negative coefficients $\{ \alpha_i \}$ that maximize the margin are determined by solving a convex quadratic programming problem. The data points $\{ \boldsymbol{x}_k \}$ for which coefficients $\{ \alpha_k \}$ are nonzero are called support vectors because they correspond to points that lie on the maximum margin hyperplanes in the high-dimensional feature space.

We implemented SVM via SVM*light* (Joachims, 1998), and made use of the following three kinds of the kernel functions: linear, polynomial, and radial basis functions.

$$K_1(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}\cdot\boldsymbol{y}), \quad K_2(\boldsymbol{x},\boldsymbol{y}) = \left(s\times(\boldsymbol{x}\cdot\boldsymbol{y})+c\right)^d,$$
$$K_3(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right), \tag{6}$$

where parameter values are set at default ones: $s = c = 1.0$, $d = 3$ and $2\sigma^2 = 1.0$.

Training data were prepared for the training phase of SVM to discriminate between two classes of character and non-character as follows.

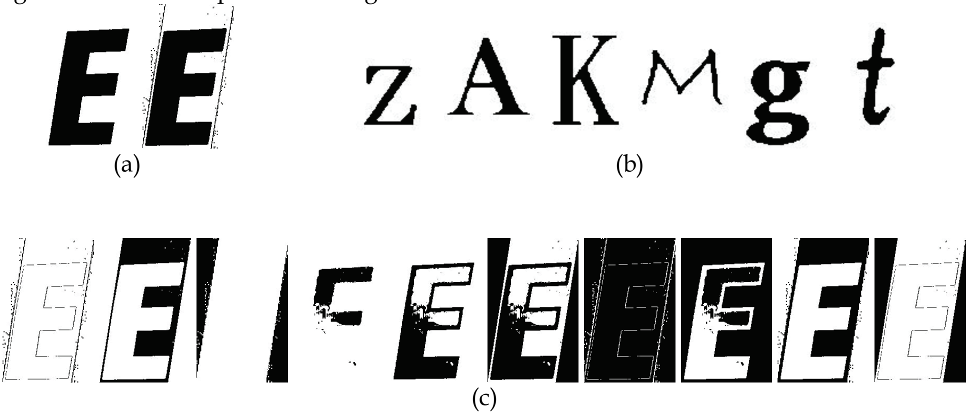
Training data for the *character* class:

First, we selected correctly binarized images from a total of (2^K – 2) tentatively binarized images obtained for each of training samples. Secondly, we added a total of 136 available font sets to the training data.

Training data for the *non-character* class:

We selected incorrectly binarized images from a total of (2^K – 2) tentatively binarized images obtained for each of training samples.

Figure 9 shows examples of training data for the character and non-character classes.



(a) (b)

(c)

Fig. 9. Examples of training data. (a) Character class (correctly binarized images). (b) Character class (available fonts). (c) Non-character class (incorrectly binarized images).

3.3.5 Selection of correctly binarized image via SVM

For a given color character image a total of (2^K – 2) tentatively binarized images are generated by K-means clustering. Then, feature vectors are extracted from each tentatively binarized image. Those feature vectors are fed into the trained SVM. SVM outputs the values of $f(x)$ of Eq. (5), where positive and negative values of $f(x)$ indicate character and non-character classes, respectively.

Here, we regard the value of $f(x)$ as estimating the degree of character-likeness, and, also, assume that the larger the value of $f(x)$ is the more its character-likeness is.

Then, we select a single tentatively binarized image with the maximum value of $f(x)$ among those of $(2^K - 2)$ candidates as an optimal binarization result.

It is to be noted that this technique tackles the problem of how to discriminate between character and non-character using SVM in the high-dimensional feature space based not on deterministic but on probabilistic means. In particular, this technique has a possibility for correctly binarizing both multi-color characters and/or characters with nonuniform backgrounds. Of course, K-means clustering in the HSI color space should generate a sufficient number of subimages for obtaining a successful dichotomization that corresponds to a correctly binarized image.

4. Distortion-tolerant character recognition as elastic template matching

In this section, we compare two competing techniques of distortion-tolerant template matching or elastic image matching. The first one is our global affine transformation (GAT) correlation technique (Wakahara et al., 2001). GAT correlation absorbs distortion expressible by affine transformation by determining optimal affine parameters that maximize a normalized cross-correlation value between an affine-transformed input image and a template. The second one is the well-known tangent distance (TD) (Simard et al., 1993). The tangent distance absorbs distortion expressible by a linear combination of predefined geometric and topographical transformations as applied to both an input image and each template.

First of all, considering that there is only a limited quantity of data against a wide variety of fonts and image degradations we dare to take the position that only a single template is provided for each character category.

Here, we use the "HGP Gothic E" font set for 62 alphanumeric characters as templates. As preprocessing, positon and size normalization together with blurring operation is applied to each template. We set a size of each preprocessed gray-scale template at 28 × 28 pixels.

Figure 10 shows examples of templates.

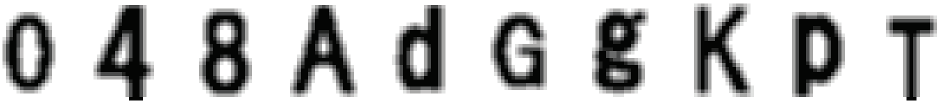


Fig. 10. Examples of templates.

4.1 GAT correlation

This technique provides a computational model for determining optimal affine parameters that deform an original input image, $\boldsymbol{f}$, so as to yield the maximum correlation value against a template image, $\boldsymbol{g}$.

First, both an input image, $\boldsymbol{f} = \{f(\boldsymbol{r})\}$, and each template, $\boldsymbol{g} = \{g(\boldsymbol{r})\}$, are linearly transformed to take the zero mean and the unit variance. As a result, a normalized cross-correlation value is made equal to an inner product (*f*, ***g***). It is to be noted that image matching by means of normalized cross-correlation was shown to be robust against image blurring and additive random noise (Sato, 2000).

We denote the GAT-superimposed input image by *Affine*[*f*]. Here, *Affine*[•] stands for the operation of affine transformation in the 2D space, defined by a 2 × 2 matrix, A, representing rotation, scale-change, and shearing, and a 2D translation vector, ***b***:

$$\mathrm{A} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}. \tag{7}$$

The objective function Φ to maximize the value of (*Affine*[*f*], ***g***) is given by

$$\begin{aligned} \Phi &\equiv (\mathit{Affine}[\boldsymbol{f}], \boldsymbol{g}) = \textstyle\sum_{\boldsymbol{r}} \mathit{Affine}[\boldsymbol{f}](\boldsymbol{r}) \times g(\boldsymbol{r}) \\ &= \textstyle\sum_{\boldsymbol{r}} f(\boldsymbol{r}) g(\tilde{\boldsymbol{r}}) \quad \rightarrow \quad \text{max for A and } \boldsymbol{b}, \\ &\text{where} \quad \tilde{\boldsymbol{r}} = \mathrm{A}\boldsymbol{r} + \boldsymbol{b}. \end{aligned} \tag{8}$$

Then, to avoid an exhaustive search for optimal A and ***b***, we employ another objective function Ψ with Gaussian kernels given by

$$\begin{aligned} &\Psi \equiv \textstyle\sum_{\boldsymbol{r}} \sum_{\boldsymbol{r}'} \gamma\, f(\boldsymbol{r}) g(\boldsymbol{r}') \mathrm{G}(\mathrm{A}, \boldsymbol{b}, \boldsymbol{r}, \boldsymbol{r}') \\ &\quad \rightarrow \quad \text{max for A and } \boldsymbol{b}, \\ &\text{where} \quad \gamma \;: \text{a function of } \nabla f \text{ and } \nabla g, \\ &\mathrm{G}(\mathrm{A}, \boldsymbol{b}, \boldsymbol{r}, \boldsymbol{r}') = \exp\left(-\frac{\| \boldsymbol{r}' - \tilde{\boldsymbol{r}} \|^2}{2D^2} \right), \quad \tilde{\boldsymbol{r}} = \mathrm{A}\boldsymbol{r} + \boldsymbol{b}, \end{aligned} \tag{9}$$

where the weight function γ serves as matching constraints. Also, *D* controls the spread of the Gaussian kernel.

Here, we explain how to practically design the values of γ and *D*.

First, γ of Eq. (9) is a function of ∇*f* and ∇*g* as matching constraints with the aim of promoting matching between pixels with the similar gradients. Here, we propose the concrete form of γ given by

$$\gamma(\nabla f, \nabla g) = \begin{cases} 1 & \text{if } \nabla f = \nabla g, \\ 0 & \text{otherwise}. \end{cases} \tag{10}$$

where gradients ∇*f* and ∇*g* are quantized into eight directions at intervals of π/4. The introduction of γ into Ψ has an effect that optimal affine transformation forces matched pixels to have the same gradient direction.

Second, the parameter *D* of Eq. (9) controls the spread of the Gaussian kernel or the radius of search area for matching pixels by affine transformation. Hence, a suitable selection of *D* is the key to stabilizing the whole matching process. We propose to adaptively determine the value of D proir to GAT application according to the disparity of input and template images in a gradient space as follows.

$$D = \frac{1}{2}\left\{ \underset{r}{Av}\left[\min_{r'} \left(\| \boldsymbol{r} - \boldsymbol{r}' \| ; \gamma(\nabla f, \nabla g) = 1 \right) \right] + \underset{r'}{Av}\left[\min_{r} \left(\| \boldsymbol{r} - \boldsymbol{r}' \| ; \gamma(\nabla f, \nabla g) = 1 \right) \right] \right\}, \tag{11}$$

where Av stands for an averaging operation over either input or template images. Namely, D is the average minimum distance between two points, one in $\boldsymbol{f}$ and the other in $\boldsymbol{g}$, with the same gradient direction.

Now, by setting the derivatives of Ψ with respect to each of six unknown parameters, a_{00}, a_{01}, a_{10}, a_{11}, b_0, and b_1, equal to zero, respectively, we obtain a set of nonlinear equations. Next, by using the 0th order approximation that sets A = I and $\boldsymbol{b} = \boldsymbol{0}$ in the Gaussian kernel, we have a set of simultaneous linear equations. Finally, we solve these simultaneous linear equations by conventional techniques and obtain a sub-optimal solution of A and $\boldsymbol{b}$.

In order to obtain the true optimal GAT of Eq. (8), we use the successive iteration method by iteratively updating the input gray-scale image by sub-optimal affine parameters of Eq. (9) until the value of Φ arrives at a maximum.

4.2 Tangent distance

This technique encourages invariance of distance-based methods to a set of predefined transformations, which realizes distortion-torelant template matching.

Concretely, by using a set of predefined geometric or topographical transformations applicable to an input image, $\boldsymbol{f}$, and each template, $\boldsymbol{g}$, we generate a tangent vector corresponding to each geometric or topographical transformation. Here, it is to be noted that all elements of both input/template images and tangent vectors are gray-scale values in the image plane.

The tangent distance, $D_{\mathrm{T}}(\boldsymbol{f}, \boldsymbol{g})$, is calculated as the minimum distance between two hyper-planes expanded by a set of tangent vectors around input and template images given by

$$D_{\mathrm{T}}(\boldsymbol{f}, \boldsymbol{g}) = \min_{\boldsymbol{\alpha}_f, \boldsymbol{\alpha}_g} \left\| \tilde{\boldsymbol{f}} - \tilde{\boldsymbol{g}} \right\|, \qquad \tilde{\boldsymbol{f}} = \boldsymbol{f} + \mathrm{T}_f \boldsymbol{\alpha}_f, \quad \tilde{\boldsymbol{g}} = \boldsymbol{g} + \mathrm{T}_g \boldsymbol{\alpha}_g, \tag{12}$$

where matrices T_f and T_g have their corresponding tangent vectors as column vectors. Also, $\boldsymbol{\alpha}_f$ and $\boldsymbol{\alpha}_g$ represent expansion coefficient vectors.

Tangent vectors are obtained via convolution between input/template images and Gaussian filters operated in advance by corresponding geometric or topographical transformations. Here, 2D Gaussian filters are given by

$$G_\sigma(\boldsymbol{r}) = \exp\left(-\frac{\|\boldsymbol{r}\|^2}{2\sigma^2}\right), \tag{13}$$

where the value of σ was set at σ = 0.7, and the size of a convolution mask was 19 × 19.

We deal with seven kinds of geometric or topographical transformations: X-translation, Y-translation, rotation, scaling, parallel hyperbolic transformation, diagonal hyperbolic transformation, and thickening (Simard et al., 1993).

Figure 11 shows examples of tangent vectors.

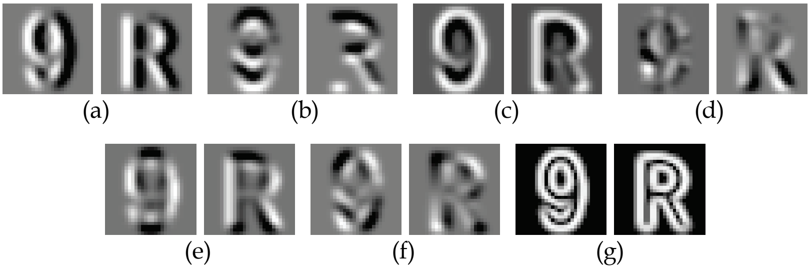


Fig. 11. Examples of tangent vectors. (a) X-translation. (b) Y-translation. (c) Rotation. (d) Scaling. (e) Parallel hyperbolic transformation. (f) Diagonal hyperbolic transformation. (g) Thickening.

5. Experimental results

In this section, we show two kinds of experimental results using ICDAR 2003 robust OCR dataset: figure-ground discrimination of color characters in scene images and distortion-torelant character recognition as elastic template matching.

5.1 Abilities of figure-ground discrimination of color characters in scene images

Determination of an optimal sequence of filters for binarization using GA:

Figure 12 shows examples of binarization results obtained for both training and test samples in all of six degradation categories.

From Fig. 12, it is found that binarization of test samples is remarkably successful even if embedded characters in training and test samples are totally different in shape.

Moreover, In order to evaluate the ability of binarization in a more quantitative manner, we calculated a normalized cross-correlation value between optimally filtered images and their respective target images ideally binarized by humans.

Figure 13 shows relations between average correlation values and image degradation categories obtained from both training and test samples against their target images.

From Fig. 13, it is found that optimal sequences of filters determined by GA have the marked ability to achieve a fairly high correlation value, more than 0.9, between filtered and target images against most of all image degradation categories.

These results show clearly that we can select the optimal filter sequence for binalization of a given image if its degradation category is automatically determined. In other words, when we deal with the case where the cause of degradation is found to be unique and specific, this technique for binarization using the optimal filter sequence is expected to be very powerful.

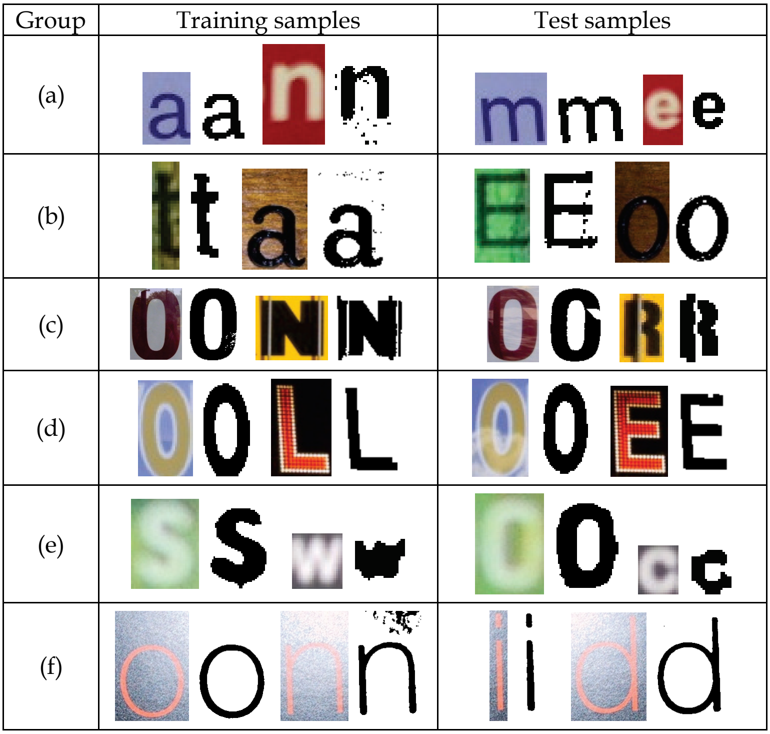


Fig. 12. Examples of binarization results. (a) Clear. (b) Background with pattern. (c) Character with pattern. (d) Character with rims. (e) Blurring. (f) Nonuniform lighting.

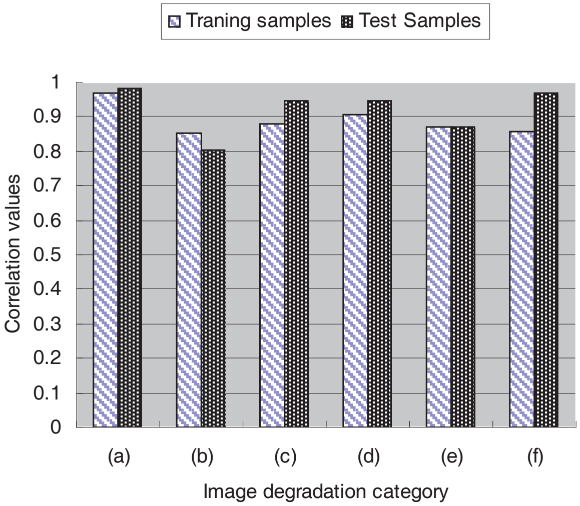


Fig. 13. Relations between correlation values and image degradation categories. (a) Clear. (b) Background with pattern. (c) Character with pattern. (d) Character with rims. (e) Blurring. (f) Nonuniform lighting.

Binarization using a maximum separability axis in a color space:
Table 2 shows rates of successful and unsuccessful binarization.
Figure 14 shows examples of unsuccessful figure-ground discrimination.
From Table 2 and Fig. 14, it is found that the task of temporary binarization poses a more serious problem than that of figure-ground determination does.

Results	Rates
Successful binarization	75.3%
Unsuccessful temporary binarization	17.5%
Unsuccessful figure-ground determination	7.2%

Table 2. Rates of successful and unsuccessful binarization.

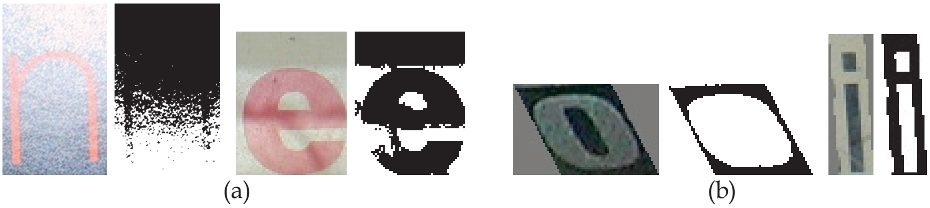


Fig. 14. Examples of unsuccessful figure-ground discrimination. (a) Unsuccessful temporary binarization. (b) Unsuccessful figure-ground determination.

K-means clustering in a color space and figure-ground discrimination by SVM:
The number of clusters, *K*, in the K-means clustering was set at 5, and, hence, a total number of tentatively binarized images was 30 (= $2^5 - 2$).

First, we evaluated the ability of discrimination between character and non-character via SVM using three kinds of feature vectors extracted from tentatively binarized images. Here, we adopted the technique of *S*-fold cross-varidation (Bishop, 2006), which allows a proportion $(S-1)/S$ of the available data to be used for training while making use of all of the data to assess performance. We set at the value of *S* at 10.

Based on evaluation of false reject/acceptance rates (FRR, FAR) according to the sign of $f(x)$ of Eq. (5), we found that the radial basis function (RBF) as a kernel function of SVM achieved the minimum sum of FRR and FAR.

Figure 15 shows the distribution of SVM outputs for test samples using the RBF kernel and the weighted direction code histogram feature.

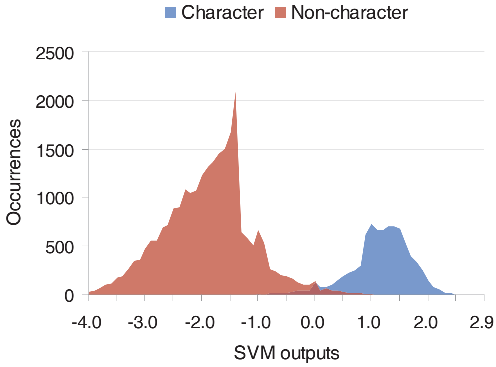


Fig.15. Distribution of SVM outputs for test samples using the RBF kernel and the weighted direction code histogram feature.

Figure 16 shows ROC (Receiver Operating Characteristic) curves obtained by moving a threshold for discrimination between character and non-character on the SVM output.

From Fig. 16, it is found that SVM fed with the weighted direction code histogram feature is the top of the three feature vectors and achieved the minimum equal error rate, EER, of 6.2%.

Next, we investigated the ability of selecting a correctly binarized image from a total of 30 tentatively binarized images based on the values of SVM outputs. Here, we selected the binarized image with the maximum value of SVM outputs as an optimal binarization result. Namely, a total of 30 candidate binary images were arranged in the decreasing order of SVM outputs, and the top one was selected as a correctly binarized image.

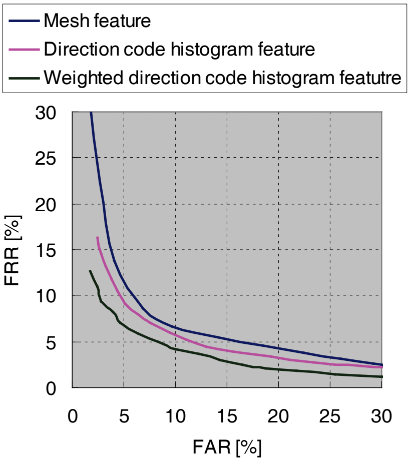


Fig.16. ROC curves obtained for three kinds of feature vectors via SVM with the RBF kernel.

Figure 17 shows cumulative binarization rates via SVM. The *k*th cumulative binarization rate is an average rate at which the top *k* candidate binary images contain a correctly binarized image.

From Fig. 17, it is found that the correct binarization rate or the 1st cumulative binarization rate is 92.2%, and the 7th cumulative binarization rate is over 99.0%.

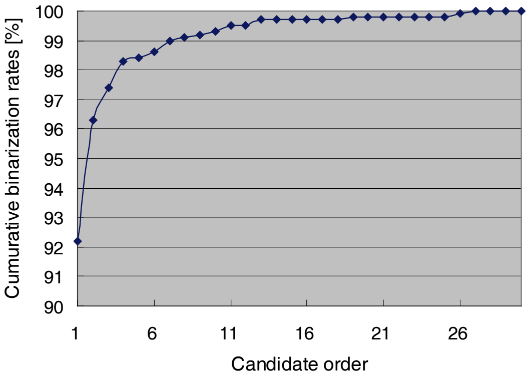


Fig.17. Cumulative binarization rates via SVM.

5.2 Abilities of distortion-tolerant character recognition as elastic template matching

In this subsection, we show results of distortion-tolerant recognition of correlctly binarized characters by the GAT correlation and the tangent distance (Wakahara, 2008).

Input images were normarized with respect to position and size, and were set at a size of 28 × 28 pixels. The matching measure of the GAT correlation is a normalized cross-correlation value, while the matching measure of the tangent distance is a pixelwise distance in gray-scale values, as described in Section 4. The dimension of a feature vector is 28 × 28. It is to be noted that $f(\boldsymbol{r})$ and $g(\boldsymbol{r})$ in the GAT correlation can be any features extracted from images as far as they are a function of 2D loci vectors, $\boldsymbol{r}$. On the other hand, the tangent distance can use no features besides gray-scale values.

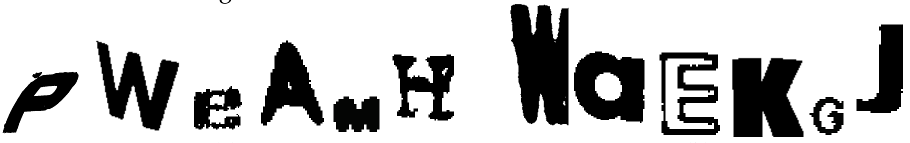
Table 3 shows recognition rates for correctly binarized characters. The matching measure of simple correlation is a normalized cross-correlation value calculated between an input image and each template. Moreover, in the GAT correlation, we tried the well-known gradient features for correlation matching. Here, the dimenstion of a gradient feature vector is 28 × 28 × 8, where an original 2D gray-scale image is decomposed into eight gradient images calculated along the direction at intervals of $\pi/8$.

Methods	Recognition rates (%)
Simple correlation (gray-scale values)	80.4
GAT correlation (gray-scale values)	90.3
GAT correlation (gradient values)	94.1
Tangent distance (gray-scale values)	91.6

Table 3. Recognition rates for correctly binarized characters.

From Table 3, it is first found that both GAT correlation and tangent distance reduced the error rate of the simple correlation more than by half. Secondly, it is found that the use of gradient features in GAT correlation improved the recognition accuracy markedly.

Figure 18 shows examples of correctly recognized and misrecognized images by both of GAT correlation and tangent distance.



(a) (b)

Fig. 18. Examples of correctly recognized and misrecognized images by both of GAT correlation and tangent distance. (a) Correctly recognized. (b) Misrecognized.

Furthermore, in order to evaluate the robustness of GAT correlation and tangent distance against rotation which cannot be compensated by position and size normalization, we fed each of recognizers with artificially rotated templates to be matched against upright templates.

Figure 19 shows relations between rotation angles and mean of normalized cross-correlation values, where elastic image matching was performed between each upright template and their artificially rotated templates from -45 degrees to +45 degrees at intervals of 5 degrees.

From Fig. 19, it is clear that the GAT correlation is superior to the tangent distance in robustness against rotation at an angle of more than 20 degrees.

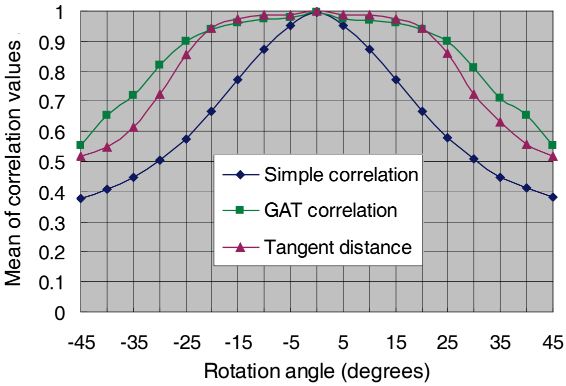


Fig. 19. Relations between rotation angles and mean of normalized cross-correlation values.

6. Discussion and future work

We tackled two challenging problems: figure-ground discrimination or correct binarization of color characters in scene images as a crucial step to the success of subsequent recognition, and distortion-tolerant character recognition under the condition of a small sample size.

Regarding the first problem, we proposed three kinds of techniques. Although each of three techniques showed promising preliminary results, we dare to enumerate their weak points, respectively, as follows.

The first technique of generating an optimal sequence of filters for binarization using GA had the following two disadvantages: not automatic but manual selection of degradation categories and the limited ability against a wide variety of complex backgrounds even if the degradation category is specified.

The second technique of using a maximum separability axis in a color space based on Otsu's criterion had also one major disadvantage: the insufficient adaptability to multi-color characters and/or characters with nonuniform backgrounds.

The third technique of using K-means clustering in a color space and figure-ground discrimination by SVM showed the most promising preliminary results mainly because of its potential ability to deal with multi-color characters and/or characters with nonuniform, complex backgrounds.

Hence, we enumerate several issues concerning the third technique of using K-means clustering in a color space and figure-ground discrimination by SVM still need to be addressed.

(1) Adaptive and stable determination of the optimal number of clusters in K-means clustering,

(2) Selection of more efficient feature vectors for evaluating character-likeness, and

(3) Systematic expansion of training data in SVM using a kind of degradation or deformation models.

Regarding the second problem, we compared two competing techniques as elastic template matching: GAT correlation and tangent distance. Although both of them achieved recognition rates of more than 90% for correctly binarized characters, the recognition accuracy still needs to be much improved to meet the practical demands of the market. From this viewpoint, the following issues remain to be solved.

(1) Appropriate selection of multiple templates per category, and

(2) Cooperation between distortion-tolerant template matching and statistical pattern recognition techniques.

Finally, it is necessary and interesting to extend and apply techniques of recognizing single-character images to recognition of character strings in scene images.

7. Acknowledgments

The author would like to thank his former excellent students, Ms. Hanako Kohmura, Mr. Minoru Yokobayashi, and Mr. Junpei Kato, for their fruitful discussions and long, careful experiments.

8. References

Bishop, C. B. (2006). *Pattern Recognition and Machine Learning*, Springer, 2006

Casey, R. G. (1970). Moment normalization of handprinted characters, *IBM J. Res. Develop.*, Vol. 14, No. 5, pp. 548-557, September 1970

Doermann, D.; Liang, J. & Li, H. (2003). Progress in camera-based document image analysis, *Proceedings of 7th International Conference on Document Analysis and Recognition*, Vol. I, pp. 606-616, Edinburgh, August 2003

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison Wesley, 1989

Gonzalez, R. C. & Woods, R. E. (2000). *Digital Image Processing,* Second Edition, Prentice Hall, 2000

Herault, L. & Horaud, R. (1993). Figure-ground discrimination: a combinatorial optimization approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 15, No. 9, pp. 899-914, September 1993

ICDAR Datasets. (2003). http://algoval.essex.ac.uk/icdar/Datasets.html. 2003

Joachims, T. (1998). Making large-scale SVM learning practical, In: *Advances in Kernel Methods: Support Vector Learning,* Schölkopf, B.; Burges, C., Smola, A., Chap. 11, MIT Press, 1998

Kimura, F.; Wakabayashi, T., Tsuruoka, S., Miyake, Y. (1997). Improvement of handwritten Japanese character recognition using weighted direction code histogram, *Pattern Recognition,* Vol. 30, No. 8, pp. 1329-1337, August 1997

Kato, J. & Wakahara, T. (2009). Binarization of color characters in scene images using K-means clustering and support vector machines, *Proceedings of 12th Meeting on Image Recognition and Understanding*, pp. 351-358, Matsue, July 2009 (in Japanese)

Kohmura, H. & Wakahara, T. (2006). Determining optimal filters for binarization of degraded characters in color using genetic algorithms, *Proceedings of 18th International Conference on Pattern Recognition*, Vol. III, pp. 661-664, Hong Kong, August 2006

Lucas, S. M.; Panaretos, A., Sosa, L., Tang, A., Wong, S. & Young, R. (2003). ICDAR 2003 robust reading competitions, *Proceedings of 7th International Conference on Document Analysis and Recognition*, Vol. I, pp. 682-687, Edinburgh, August 2003

Miene, A.; Hermes, T. & Ioannidis, G. (2001). Extracting textual inserts from digital videos, *Proceedings of 6th International Conference on Document Analysis and Recognition*, pp. 1079-1083, Seattle, September 2001

Otsu, N. (1979). A threshold selection method from gray-level histogram, *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 9, No. 1, pp. 62-69, January 1979

Sato, A. (2000). A learning method for definite canonicalization based on minimum classification error, *Proceedings of 15h International Conference on Pattern Recognition*, Vol. II, pp. 199-202, Barcelona, September 2000

Simard, P.; LeCun, Y. & Denker, J. (1993). Efficient pattern recognition using a new transformation distance, *Advances in Neural Information Processing Systems*, Vol. 5, [NIPS Conference], pp. 50-58, Morgan Kaufmann, 1993

Trier, O. & Jain, A. K. (1995). Goal directed evaluation of binarization methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 17, No. 12, pp. 1191-1201, December 1995

Uchida, S. & Sakoe, H. (2005). A survey of elastic matching techniques for handwritten character recognition, *IEICE Transactions on Information and Systems,* Vol. E88-D, No. 8, pp. 1781-1798, August 2005

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory,* Second Edition, Springer, 2000

Wakahara, T.; Kimura, Y. & Tomono, A. (2001). Affine-invariant recognition of gray-scale characters using global affine transformation correlation, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 23, No. 4, pp. 384-395, April 2001

Wakahara, T. (2008). Figure-ground discrimination and distortion-tolerant recognition of color characters in scene images, *Proceedings of 19th International Conference on Pattern Recognition*, Tampa, December 2008

Wolf, C.; Jolion, J. & Chassaing, F (2002). Text localization, enhancement and binalization in multimedia document, *Proceedings of 16th International Conference on Pattern Recognition*, Vol. 2, pp. 1037-1040, Quebec, August 2002

Wu, S. & Amin, A. (2003). Automatic thresholding of gray-level using multi-stage approach, *Proceedings of 7th International Conference on Document Analysis and Recognition*, Vol. I, pp. 493-497, Edinburgh, August 2003

Yokobayashi, M. & Wakahara, T. (2006). Binarization and recognition of degraded characters using a maximum separability axis in color space and GAT correlation, *Proceedings of 18th International Conference on Pattern Recognition*, Vol. II, pp. 885-888, Hong Kong, August 2006

Segmenting the License Plate Region Using a Color Model

Kaushik Deb and Kang-Hyun Jo
University of Ulsan
South Korea

1. Introduction

Humans can perform usual target recognition without too much effort. However, by computer the task of recognizing specific object in an image, one of the most difficult topics in the field of computer vision or digital image processing. Vehicle license plate detection (VLPD) task is quite challenging from vehicle images due to the multi-style plate formats, view point changes and the nonuniform outdoor illumination conditions during image acquisition (Anagnostopoulos et al., 2008) and (Jiao et al., 2009). In addition, VLPD system should operate fast enough (real time) to satisfy the needs of intelligent transportation systems (ITSs) and not to miss a single interest object from the vehicle image. VLPD is also very interesting in finding license plate area from vehicle image. The VLPD is widely used for detecting speeding cars, security control in restricted areas, in unattended parking zones, for traffic law enforcement and electronic toll collection, etc. With the rapid development of highway and the wide use of vehicles, people have started to pay more and more attention to the advanced, efficient, and accurate ITSs. Recently, the necessity of vehicle license plate recognition (VLPR) has increased significantly. The license plate detection is a crucial and indispensable component of VLPR system. One of the major problems in LP detection is determining LP systems. This system must guarantee robust detection under various weather and lighting conditions, independent of orientation and scale of the plate.

In the recent years developments dealing with simple images have been achieved with acceptable results. However, recent researches have been addressed to processing complex images with unconstrained conditions (Matas, 2005). The proposed license plate detection framework deals with such vehicle images.

In this proposed VLPD method, consists of two main stages. Initially, HSI color model is adopted for detecting candidate regions. According to different colored LP, these candidate regions may include LP regions; geometrical properties of LP are then used for classification. The proposed method is able to deal with candidate regions under independent orientation and scale of the plate. More than one license plate can be detected in the same image. Finally, the decomposition of candidate regions contain predetermined LP alphanumeric characters by using position in the histogram to verify and detect vehicle license plate region.

The focus of this chapter is on the consolidation of a new method to select automatically statistical threshold value in HSI color model for detecting candidate regions. Generally, as a common way of color-based VLPD system, threshold value is defined by predetermined coefficients or by user. It provides stable result, but in poor lighting condition it is too sensitive. Whereas in our experiments we calculate threshold value in a statistical way, 20% of sample data (only green, yellow and white LP areas) are randomly selected for training. After training from those sample data, the mean and standard deviation values of hue are computed for detection of green and yellow LP pixels. Detecting white license plate pixels, the mean and standard deviation values of saturation and intensity are computed to detect green, yellow and white LP from vehicle images.

In addition, the proposed method is able to deal with plates (candidate regions) under independent orientation and scale of the plate. More than one license plate can be detected in the same image. Furthermore, candidate regions may include LP regions; geometrical properties of LP are then used for classification. Finally, the decomposing of candidate region which contains predetermined LP alphanumeric character, by using position in the histogram to verify and detect vehicle license plate region is performed.

2. Relevant work

This section provides a descriptive summary of some methods that have been implemented and tested for VLPD. As far as detection of the plate region is concerned, researchers have found many methods of locating license plate. For example, survey paper (Anagnostopoulos et al., 2008), offers to researchers a link to a public image database to define a common reference point for VLPR algorithmic assessment. In addition, this survey paper discusses about current trends and anticipated research in VLPR system. In (Anagnostopoulos et al., 2006), a method based on image segmentation technique named as sliding windows (SW) has also been proposed for detecting candidate region (LP region). The main thought of image segmentation technique in LP can be viewed as irregularities in the texture of the image and therefore abrupt changes in the local characteristics of the image, manifest probably the presence of an LP. A conventional statistical classifier, based on the k nearest neighbor rule, is used to classify every pixel of a test image to obtain a pixel map where group of positive samples probably indicates the location of a license plate. In this system, time-consuming texture analysis is presented in (Cano & Perez-Cortes, 2003), where a combination of a “kd-tree” data structure and an “approximate nearest neighbor” was espoused. The computational resource demand of this segmentation technique is the main drawback, taking an average of 34 seconds to process of single image. In (Chacon & Zimmerman, 2003), the pulse-coupled neural network (PCNN) is proposed to generate candidate regions that may contain a license plate. If the license plate is not located in the set of candidate regions, the PCNN network parameters are adjusted to generate new regions for LP identification.

Fuzzy logic has been applied in detecting license plates. Authors made some intuitive rules to describe the license plates and gave some membership functions for fuzzy sets e.g. “bright,” “dark,” “bright and dark sequence,” “texture,” “yellowness” to get the horizontal and vertical plate positions (Chang et al., 2004). A technique based on extracts candidate regions by finding vertical and horizontal edges from vehicle region had also been proposed and this segmentation method is named as sliding concentric windows. Finally, vehicle license plate is verified and detected by using HSI color model and position histogram, respectively in (Deb et al., 2008a). Prior knowledge of LP and color collocation has been used to locate the license plate in the image (Gao et al., 2007) as part of the procedure of location and segmentation. In (Hongliang & Changping, 2004), a hybrid license plate localization algorithm based on the edge statistics and morphology for monitoring the highway ticketing system is proposed. This technique can be divided into four sections, which are, vertical edge detection, edge statistical analysis, hierarchical-based license plate location, and morphology-based license plate extraction. The average accuracy of locating license plate is an impressive rate of 99.6%. However, input images were acquired from a fixed distance and view point and therefore, candidate regions in a specific position are devote priority as already depicted. The license plate locations in images are identified by means of integrated horizontal and vertical projections that are scanned using a search window (Huang et al., 2009). Moreover, a character recovery method is exploited to enhance the success rate. A region-based license plate detection method has been presented in (Jia et al., 2007), which firstly applies a mean shift procedure in spatial-range domain to segment a color vehicle image in order to get candidate regions. According to the statistical analysis performed for comparison to other LP like objects; LPs adhere to a unique feature combination of rectangularity, aspect ratio, and edge density. These three features were then estimated to candidate regions to decide whether these regions interpret an LP or not. A usual failure of this method is the failure to detect license plates when vehicle bodies and their license plate have similar colors. In (Jiao et al., 2009), a method for multi-style LP recognition has been presented. This method has introduced the density-based region growing algorithm for LP location, the skew refinement algorithm, the multi-line LP separation algorithm, the optimized character segmentation algorithm and trainable character recognition method for character recognition. Hough Transform (HT) for line detection has been proposed on the assumption that the shape of license plate has been defined by lines in (Kamat & Gansen, 1995).

A modified color texture-based method for detecting license plate in images has been presented in (Kim et al., 2002). A support vector machine (SVM) has been used to analyze the color and texture properties of LPs and to locate their bounding boxes applied by a continuous adaptive mean shift algorithm (CAMShift). The combination of CAMShift and SVMs produces efficient LP detection as time-consuming color texture analysis for less relevant pixels is restricted, leaving only a small part of the input image to be analyzed. In addition, finding candidate areas by using gradient information, it has been verified whether it contains the plate area among the candidates and adjusting the boundary of the area by introducing a template of the LP in (Kim et al., 2002). Other approaches using mathematical morphology method to detect license plate area (Martin et al., 2002) and an approach for segmentation of vehicle plates such as edge image improvement to detect a number of car plates in (Ming et al., 1996) have also been proposed. The proposed method in (Nomura et al., 2005) is committed to the task of character segmentation, describing a morphology-based adaptive approach for degraded plate images.

Moreover, assuming that LP regions are detectable even in noisy low resolution presented, a robust superresolution algorithm for video sequences (Suresh et al., 2007) has been proposed to enhance the LP text of moving vehicles with promising results. In (Wang et al., 2007), a cascade framework, utilizing plate characteristics and developing fast one pass algorithms, has been used for a real-time plate recognition system.

Currently, some researchers prefer a hybrid detection algorithm, where license plate location method based on corner detection, edge detection, characteristics of license shape, character's connection, and projection has been presented in (Xu & Zhu, 2007), (Zhang et al., 2007) and (Yang et al., 2006) is another method which is based on the color collocation of the plate's background and characters combined with the plate's structure and texture to locate the VLP. In (Zhang et al., 2006), a cascade classifier for license plate detection algorithm using both global statistical features and local Haar-like features is proposed. Using Haar-like features makes classifier be invariant to the brightness, color, size and position of license plates. On the other hand, using global statistical features makes the final classifier simple and efficient. Image enhancement and sobel operator to extract out vertical edges and finally search plate region by a rectangular window has been presented in (Zheng et al., 2005).

3. Specific features of Korean VLP

In this section, the color arrangements of the plate and outline of the Korean VLPs that are considered in this study have been discussed.

3.1 Color arrangement of the plate

Korean license plates are well classified as shown in Fig. 1. Each style has a different plate color and/or character color. However, in all, only five distinct colors like white, black, green, yellow, and deep blue are used in these license plates. It is worth paying attention to three different plate colors while searching for LP in an input image. Other types of vehicles, such as diplomatic cars and military vehicles, are not addressed since they are rarely seen. Color arrangements for the Korean VLPs are shown in Table 1.

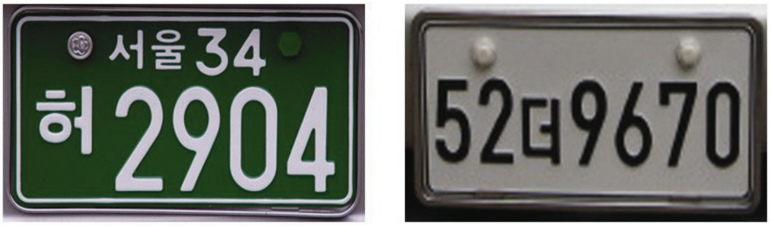


Fig. 1. Outline of the Korean license plate

3.2 Outline of the Korean VLP

Standard LP contains Korean alphabets and numbers which are shown in Fig. 1. Few LPs contain Korean alphabets and numbers in two rows; in future these kinds of LPs are to be converted into single-row types. Where plate color is white and character color is black, they contain seven alphanumeric characters written in a single line. In Fig. 1, where plate color is green and character color is white, they contain Korean LP in two rows. The upper row consists of two small Korean characters of region name followed by one or two numbers of class code or two numbers and one Korean character. The lower row is one Korean character and four big numbers or only four big numbers to indicate the usage and serial number, respectively. When plate color is yellow and character color is black, some LPs contain all alphanumeric characters written in a single line and another type of yellow LP is found that contain Korean LP in two rows. The upper row consists of two small Korean characters of region name followed by one or two numbers of class code. The lower row contains one Korean character and four big numbers to indicate the usage and serial number, respectively.

Vehicle type	Plate color	Character color
Private automobile	White	Black
	Green	White
Taxi, truck, and bus	Yellow	Deep blue
Government vehicle	Yellow	Black

Table 1. Styles of license plates

4. Proposed LP detection framework

In the author's previous work (Deb & Jo, 2008b), HSI color based vehicle license plate detection method was presented. We propose in this chapter an enhanced version of the framework for VLPD as shown in Fig. 2. Like the traditional LP detection method, automatic focus and white balancing of camera often cause the changing illumination. To overcome this problem, we propose an adaptive LP detection method for detecting white license plate pixels; we use it in the case of really high- or low-illumination condition as shown in Fig. 4. And also distinguish with the traditional LP detection method, as license plates can appear at many different angles to the camera's optical axis, each rectangular candidate region is rotated until they are all aligned in the same way before the candidate decomposition. The proposed framework can efficiently determine and adjust the rotated plate as shown in Fig. 8. Measurements such as center of area and the least second moment are employed to solve the rotation adjustment problem. The least second moment provides the principal axis as the orientation with the candidate object. General framework for detecting VLP region is shown in Fig. 2. In the proposed framework, detection is based on color properties of LP, shape-based verification and position histogram.

5. Vehicle license plate detecting module

The VLPD sequence is shown in Fig. 2, which is proposed in this paper, consists of four distinct parts. The first one deals with, by using HSI color model, the detection of the candidate region, i.e., the license plate. The second part allows procedures for refining candidate region by using labeling and filtering. According to different colored LP these candidate regions may include rectangular LP regions; geometrical properties of LP such as area, bounding box, and aspect ratio are then used for classification. The third part includes operations for determining the angle of the candidate – rotation adjustment. Measurements such as center of area and the least second moment are employed to solve the rotation adjustment. The fourth part includes performances for candidate's decomposition and finally, the decomposition of candidate region which contains predetermined LP alphanumeric character by using position in the histogram to verify and detect vehicle license plate (VLP) region.

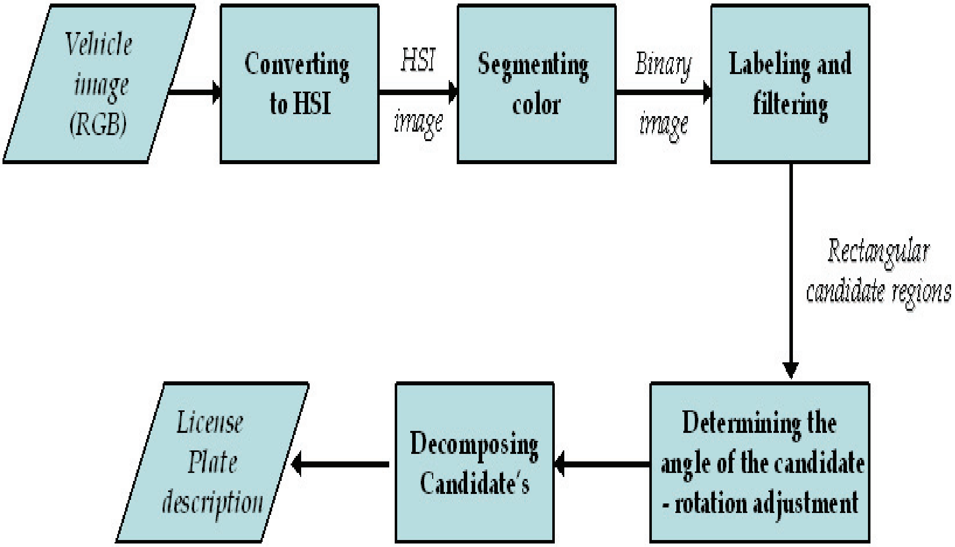


Fig. 2. Vehicle license plate detection framework

5.1 Segmenting color

In the proposed method, input vehicle images are converted into HSI color images. Then the candidate regions are found by using HSI color model on the basis of using hue, saturation and/or intensity. Many applications use the HSI color model. Machine vision uses HSI color space in identifying the color of different objects.

The RGB color model consists of the three additive primaries: red, green, and blue. Spectral components of these colors combine additively to produce a resultant color. Typically, HSI colors are not described on the basis of percentages of primary colors, but rather by their hue, saturation and intensity. The saturation is the "pureness" of the color, the hue is the color itself and intensity describes the brightness of the color. The HSI color model separates all the color information, described by hue and saturation, from the intensity component. The HSI color model is based on color descriptions that are more natural to humans and hence can provide an ideal tool for image processing algorithms. The HSI color space is represented by the diamond, as shown in Figure 3. The hue H is represented as angle 0, varying from $0°$ to $360°$. Adjusting the hue will vary the color from red at $0°$, through yellow at $60°$, green at $120°$, blue at $240°$ and back to red at $360°$. Saturation S corresponds to the radius, varying from 0 to 1. When S =0, color is a gray value of intensity 1. When S =1, color is on the boundary of top cone base. Intensities I vary along Z axis with 0 being black and 1 being white.

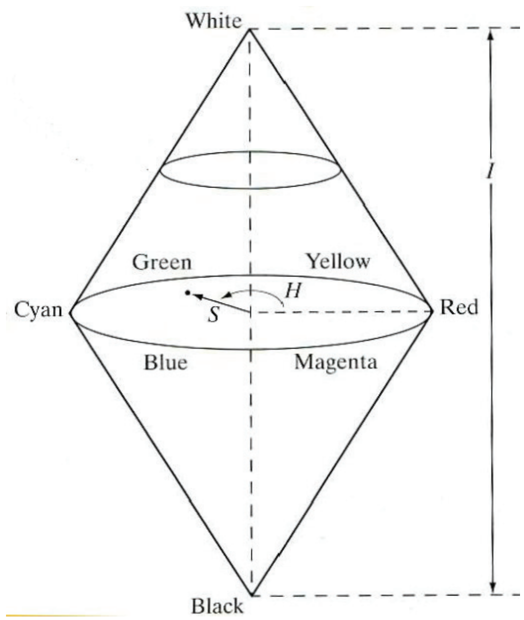


Fig. 3. The HSI color space

The transform from (R, G, B) to (H, S, I) in (Umbaugh, 1998) is

$$H = \cos^{-1} \frac{\frac{1}{2}\left[(R - G) + (R - B)\right]}{\left[(R - G)^2 + (R - B)(G - B)\right]^{\frac{1}{2}}}$$

$$S = 1 - \frac{3}{R + G + B} \min(R, G, B) \tag{1}$$

$$I = \frac{R + G + B}{3}$$

Plate color information is used to detect candidate regions in our experiments, and shape properties of LP allow reducing number of LP-like candidates. One of the common ways of color-based vehicle license plate detection can be formalized as follows:

$$R\ (x, y) > \alpha_R;\ G\ (x, y) > \alpha_G;\ B\ (x, y) > \alpha_B \tag{2}$$

$$R\ (x, y) - G\ (x, y) > \beta_{RG};\ R\ (x, y) - B\ (x, y) > \beta_{RB} \tag{3}$$

where *R, G* and *B* are red, green and blue components of x × y image. α and β are predefined coefficients. Equation (2) sets up limitations for the minimal values of pixel components. Equation (3) formalizes dependencies between pixel components for LP. Generally, common way of using color-based vehicle license plate detection is based on two types of restrictions: first, restriction is based on Eps. (2) and (3). It provides good results in good lighting conditions. However, it is not good for low-contrast images. Pixel belongs to green and yellow LP, respectively like following Eqs. (4) and (5)

$$b_{green} = \begin{cases} 1, & \left[\{R\,(x,y) \le 0.85 \cdot G\,(x,y)\} \;\&\; \{B\,(x,y) \le 0.90 \cdot G\,(x,y)\}\right] \\ 0, & otherwise \end{cases} \tag{4}$$

$$b_{yellow} = \begin{cases} 1, & \left[\{B\,(x,y) \le 0.90 \cdot R\,(x,y)\} \;\&\; \{B\,(x,y) \le 0.80 \cdot G\,(x,y)\}\right] \\ 0, & otherwise \end{cases} \tag{5}$$

where b_{green} and b_{yellow} are green and yellow candidate binary masks. The second restriction is based on Eqs. (4) and (5), and a threshold value is taken heuristically. It provides stable result whereas in bad lighting condition it is too sensitive.

In this proposed method, LP detection is based on its color properties, namely mean and standard deviation values of hue. For detection of green and yellow LP pixels, hue parameter of HSI color is used in our experiment. To detect white LP pixels hue value is meaningless, hence only saturation and intensity parameters are important for this case. To estimate these properties, we used 30 images of LP taken under different lighting and weather conditions. After training from those sample data, the mean and standard deviation values of hue are computed for detection of green and yellow LP pixels. Detecting white license plate pixels, the mean and standard deviation values of saturation and intensity are computed to detect green, yellow and white LP from vehicle images. For detection of green and yellow LP pixels, the binarization process can be formulated as follows:

$$b_{green} = \begin{cases} 1, & \left[\{\mu_H - \delta_H \le H\,(x,y) \le \mu_H + \delta_H\} \;\&\; \{S\,(x,y) \ge 0.08\} \right. \\ & \left. \;\&\; \{0.05 \le I\,(x,y) \le 0.95\}\right] \\ 0, & otherwise \end{cases} \tag{6}$$

$$b_{yellow} = \begin{cases} 1, & \left[\{\mu_H - \delta_H \le H\,(x,y) \le \mu_H + \delta_H\} \;\&\; \{S\,(x,y) \ge 0.12\} \right. \\ & \left. \;\&\; \{0.20 \le I\,(x,y) \le 0.80\}\right] \\ 0, & otherwise \end{cases} \tag{7}$$

where *H(x, y), S(x, y),* and *I(x, y)* are hue, saturation and intensity components of *x*th, *y*th pixel, respectively. μ_H and δ_H are mean hue and hue standard deviation values for green and yellow LP of sample data, respectively.

However, the automatic focus and white balancing of camera often cause the changing illumination. Our proposed LP detection method can work well in normal illumination condition, but it seems not good enough to work in bad illumination conditions. To overcome this problem, we use an adaptive LP detection method; we use it in the case of really high- or low-illumination condition.

For normal, low- and high-illumination conditions of white license plate pixels, the binarization process can be formulated as follows, respectively:

$$b_{white(n)} = \begin{cases} 1, & \left[\{\, S\,(x,y) \le (\mu_S + \delta_S)\,\} \;\&\; \{\, I\,(x,y) \ge \mu_I + 0.25 \cdot \delta_I \,\}\right] \\ 0, & otherwise \end{cases} \tag{8}$$

$$b_{white(l)} = \begin{cases} 1, & \left[\{\, S\,(x,y) \le (\mu_S + \delta_S)\,\} \;\&\; \{\, I\,(x,y) \ge \mu_I - 0.33 \cdot \delta_I \,\}\right] \\ 0, & otherwise \end{cases} \tag{9}$$

$$b_{white(h)} = \begin{cases} 1, \left[\left\{ S\,(x,y) \le \left(\mu_S + \delta_S \right) \right\} \,\&\, \left\{ I\,(x,y) \ge \mu_I + 0.50 \cdot \delta_I \right\} \right] \\ 0, \ otherwise \end{cases} \tag{10}$$

where *S(x, y)* and *I(x, y)* are saturation and intensity components of *x*th, *y*th pixel respectively. μ_S and μ_I are mean values for saturation and intensity, δ_S, δ_I are standard deviation values for saturation, intensities of white LP of sample data, respectively. $b_{white(n)}$, $b_{white(l)}$ and $b_{white(h)}$ are white candidate binary masks. An LP image and its color segmentation results are depicted in Fig. 4(a) – (c) (green, yellow and white back ground LP), respectively.

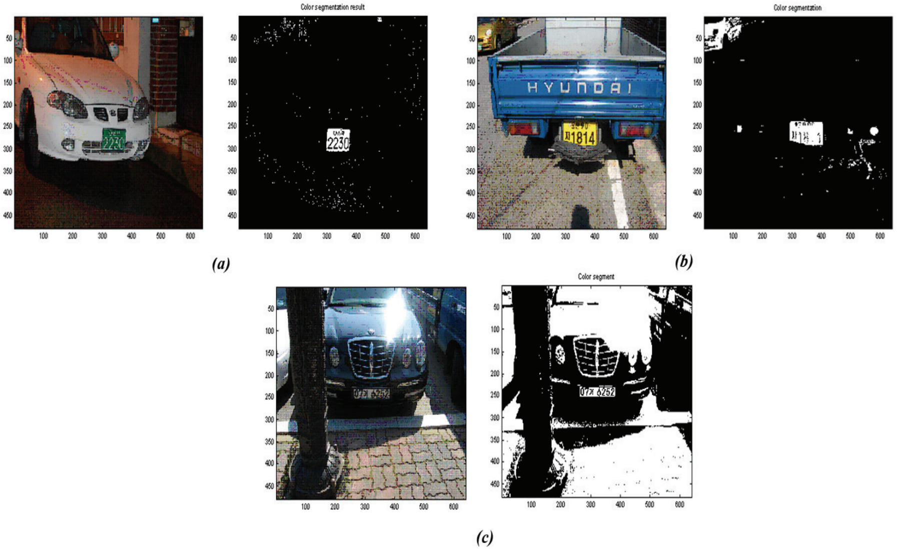


Fig. 4. An LP image *(left)* and its color segmentation results *(right)* using HSI color model

Color segmentation parameters are very sensitive in order to detect as many candidates as possible. All false candidates will be filtered out on the next stages. According to the prior knowledge of vehicle LP inspection, all license plates must be rectangular in shape and have the dimensions and have all alphanumeric characters written in one or two rows, in LP region. After the segmentation, there may still exist noises in the image and that is not ideal. These noises have many types, such as small holes or/and bulges of the target candidate regions. The problem may be resolved by using mathematical morphology processing method. Mathematical morphology is used as a potent tool for image analysis which is based on shapes in the image, not pixel intensities. The two principal morphological operations are dilation and erosion. Dilation allows objects to expand and erosion shrinks objects by etching away (eroding) their boundaries. These operations can be customized by the proper selection of the structuring element, which determines exactly how the objects will be dilated or eroded. Dilation and erosion are combined into other two operations: opening and closing. In this part of the application, we use the closing operation which is dilation followed by erosion to fill in holes and gaps smaller than the structuring element on the plate image. Removal of those holes plays an important role in calculating bounding box region. Implementation of morphological closing operation is depicted in Fig. 5(c) – 7(c), respectively.

5.2 Labeling and filtering

After the candidate regions are obtained by applying color segmentation, features of each region are to be extracted in order to correctly differentiate the LP regions from others. Next step of proposed algorithm is labeling the connected components. In the proposed method, a recursive algorithm is implemented for connected component labeling operation. Recursive algorithm (Shapiro et al., 2001) works on one component at a time, but can move all over the image. In this step we extract candidate regions which may include LP regions from the binary mask obtained in the previous step. During this step, main geometrical properties of LP candidate such as area, bounding box, and aspect ratio are computed.

Following the successful connected component labeling operation in image, measurements such as the area, the bounding box and the aspect ratio for every binary object in the image are performed.

A bounding box is a rectangle whose horizontal and vertical sides enclose the region and touch its topmost, bottommost, leftmost, and rightmost points. Rectangularity is defined as the ratio of the area of candidate object's MER (minimum enclosing rectangle) and the area of the object. Here, the area is measured in pixels and indicates the relative size of the object. The aspect ratio (also called elongation or eccentricity), is defined by the ratio of the bounding box of an object. This can be found by scanning the image and the minimum and maximum values on the row and the columns, where the object lies. This ratio is defined by

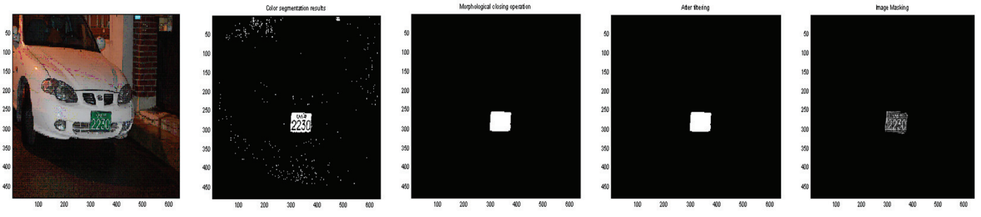
$$\rho_A = \frac{c_{\max} - c_{\min} + 1}{r_{\max} - r_{\min} + 1} \tag{11}$$

where c and r indicate columns and row, respectively. Objects which satisfy ρ_A (aspect ratio) bounds 1 to 3 for green, 1 to 2 for yellow, and 1 to 6 for white LPs are considered as candidate regions. These parameters are used for filtering operation to eliminate LP-like objects from candidate list. Filtering operation is done on geometrical properties of LP regions. Figs. 5 - 7 illustrate the steps for license plate segmentation: (a) an LP image, (b) color segmentation result, (c) implementation of morphological closing operation for removing small holes in candidate region, (d) detected candidate after filtering, and (e) candidate region detection. The most important LP-parameters are grouped in Table 2.

Filtering parameter	CR (green)	CR (yellow)	CR (white)
Bounding box	[0.6, 1.0]	[0.7, 1.0]	[0.7, 1.0]
Aspect ratio	[1.0, 3.0]	[1.0, 2.0]	[1.0, 6.0]
Possible shapes	Rectangle		

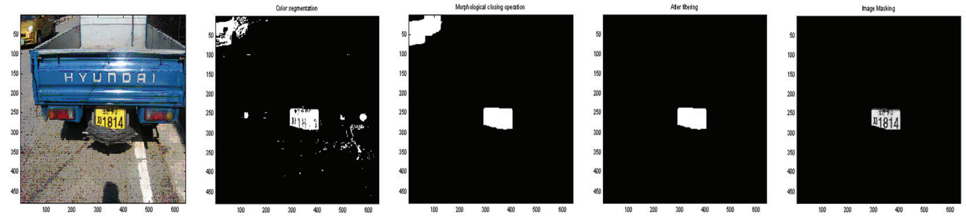
Table 2. Filtering properties

Here CR indicates candidate region.



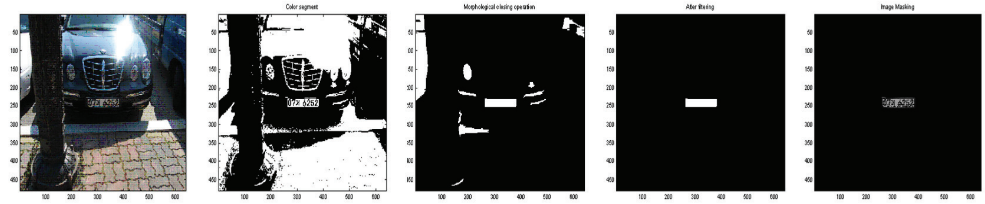
(a) (b) (c) (d) (e)

Fig. 5. Illustration of license plate segmentation: (a) an LP image in a night time, (b) color segmentation result, (c) implementation of morphological closing operation for removing small holes in candidate region, (d) detected candidate after filtering, and (e) candidate region detection



(a) (b) (c) (d) (e)

Fig. 6. Illustration of license plate segmentation: (a) an LP image in a strong sunshine, (b) color segmentation result, (c) implementation of morphological closing operation for removing small holes in candidate region, (d) detected candidate after filtering, and (e) candidate region detection



(a) (b) (c) (d) (e)

Fig. 7. Illustration of license plate segmentation: (a) an LP image in a strong sunshine reflected by vehicle mirror and also a light post located in front of vehicle, (b) color segmentation result, (c) implementation of morphological closing operation for removing small holes in candidate region, (d) detected candidate after filtering, and (e) candidate region detection

5.3 Determining the angle of the candidate - rotation adjustment

As license plates can appear at many different angles (in our experiment is more robust when LP is rotated from -15 to +15 degree) to the camera's optical axis, each rectangular candidate regions is rotated until they are all aligned in the same way before the candidate decomposition. Following the successful filtering operation in image, measurements such as center of area and the axis of least second moment are employed to solve the rotation adjustment problem.

The center of area (centroid), is the midpoint along each row and column axis corresponding to the "middle" based on the spatial distribution within candidate object. This feature used to locate an object in the 2D image plan is defined by the pair $\left(\overline{r_i},\overline{c_i}\right)$:

$$\overline{r_i} = \frac{1}{A_i}\sum_{r=0}^{N-1}\sum_{c=0}^{N-1} r\, I_i\left(r,\, c\right) \tag{12}$$

$$\overline{c_i} = \frac{1}{A_i}\sum_{r=0}^{N-1}\sum_{c=0}^{N-1} c\, I_i\left(r,\, c\right) \tag{13}$$

where $\overline{r_i}$ and $\overline{c_i}$ indicates row and column coordinate of the center of area for the *i*th object. The area, A_i, is measured in pixels and indicates the relative size of the object.

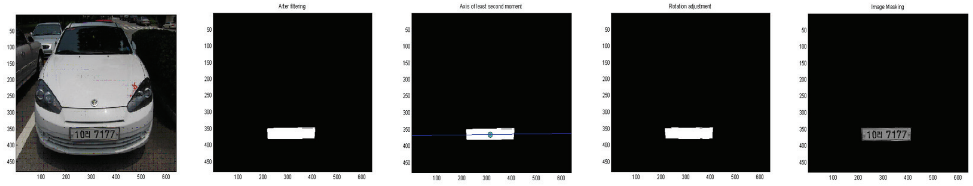
The least second moment provides the principal axis as the orientation with the candidate object. For getting principal axis of detected candidate region, we compute central moments of detected candidate region. The central moments are defined as

$$\mu_{pq} = \sum_{r=0}^{N-1}\sum_{c=0}^{N-1}\left(r - \overline{r}\right)^p \left(c - \overline{c}\right)^q I\,\left(r,\, c\right) \tag{14}$$

We apply this result to obtain a direction of principal axis by centroid of detected candidate region. Angle of principal axis moments is obtained as

$$\theta = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{15}$$

where θ denotes an angle between basis horizontal coordinate and principal axis of region. Figure 8 portrays a sequence of successful license plate identification.



(a) (b) (c) (d) (e)

Fig. 8. Illustration of license plate segmentation: (a) an LP image (b) detected candidate after filtering (c) principal axis (d) rotation adjustment, and (e) extracted candidate

5.4 Decomposing candidate's

Information extracted from image and intensity histograms plays a basic role in image processing, in areas such as enhancement, segmentation, and description. In this section,

verification and detection of the VLP region as well as character segmentation are considered and discussed in this study. The algorithm scheme for candidate decomposition is shown in Fig. 9.
Once the candidate area is binarized, the next step is to extract the information. At first, regions without interest such as border or some small noisy regions are eliminated; the checking is made by height comparison with other plate characters height. Following procedure is performed when LP color is green and yellow: first we proceed by performing horizontal position in the histogram; two objects are found where each object corresponds with one row. Then the rows are isolated and processed separately. As mentioned before in Sect. 3, two types of plate are considered.

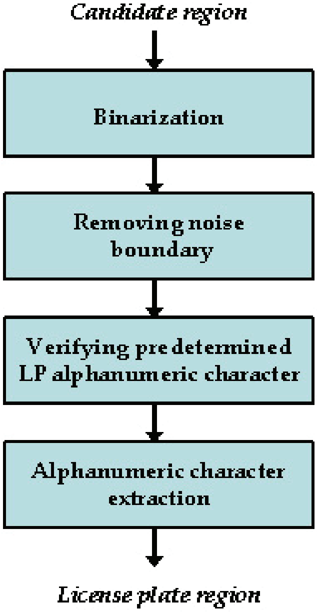

Candidate region
Binarization
Removing noise boundary
Verifying predetermined LP alphanumeric character
Alphanumeric character extraction
License plate region

Fig. 9. Algorithm scheme for candidate decomposition

Processing of the upper row: first filter phase is performed to eliminate the regions without interest. Then vertical position histogram is processed. The upper row also has two different types as we mentioned in Sect. 3. As it can be observed, usually in the upper row we can find two plate-fixing dots as shown in Fig. 1. The right plate-fixing dot does not even appear in the binarization process due to the fact that it is printed green. The left plate-fixing dot is also eliminated. The checking is made by height comparison. From the vertical position in the histogram we can find isolated alphanumeric characters.

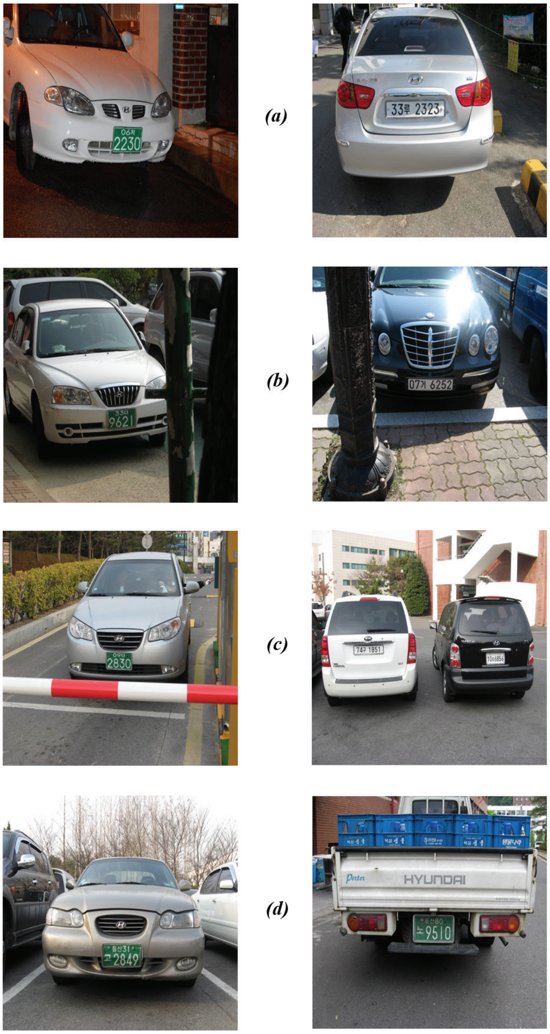


Fig. 10. Example images: (a) different illuminations, (b) complex scenes, (c) various environments, and (d) damaged license plates

Processing of the lower row: first filter phase is performed to eliminate the regions without interest. Then vertical position histogram is performed and from the vertical position histogram the alphanumeric characters are isolated.

According to the prior knowledge of vehicle LP inspection, all white LPs contain seven alphanumeric characters as well as written in a single row. The following procedure is performed for character segmentation: After eliminating border area, vertical position in the histogram is performed for segmenting predetermined alphanumeric characters. As it can be observed, usually we can find also two plate-fixing dots in upper area of plate region. The right plate fixing dot or both plate-fixing dots do not even appear in the binarization process due to the fact that it is printed white. The left plate-fixing dot is also eliminated; this checking is made by height comparison.

Figure 11 shows the following steps for verifying predetermined alphanumeric characters (white back ground LP): (a) extracting candidate region, (b) vertical position histogram with LP border, (c) horizontal position histogram with LP border, (d) horizontal position histogram without LP border,(e) view of normalization candidate region after removing border and noisy area, (f) vertical position histogram (seven peaks for predetermined seven alphanumeric characters in LP region), and (g) character extraction.

6. Experimental results

All experiments have been done on Pentium-IV 2.4 GHz with 1024 MB RAM under Matlab environment. In the experiments, 150 images were used the size is 640 × 480 pixels, some images which are shown in Fig. 10. The images are taken from (a) different illuminations (night time, strong sunshine, and shadow), (b) complex scenes where several objects such as trees, light post in front of vehicles, (c) various environments in campus parking, access areas and more than one license plates in the same image, and (d) damaged LP as bent or old. They were taken in distance from 2 up to 8 m and the camera was focused in the plate region. Under these conditions, success of LP detection has reached to more than 94%. Results of candidate region detection are shown in Table 3.

Image group	Total images	Detected LPs	Success rate (%)
Different illuminations	57	53	92.98%
Complex scenes	15	14	93.34%
Various environmants	73	69	94.52%
Damaged LP	5	5	100%
Total	150	141	94%

Table 3. Detection results

A common drawback of the proposed VLPD system is the failure to detect the boundaries or border of license plates. This occurs when vehicle bodies and their license plate possess similar colors. The average computational time for the color segmentation and filtering operations of the proposed method are shown in Table 4.

Stage	Avg time (s)	Std. deviation (s)
Color segmentation	0.16	0.07
Filtering	0.07	0.02

Table 4. Average computation time

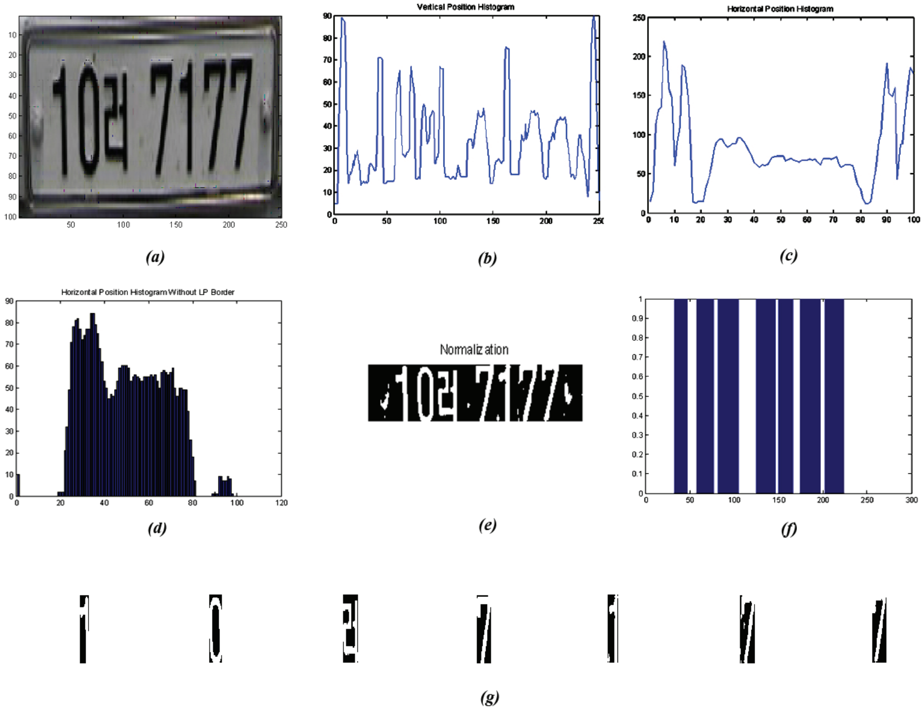


Fig. 11. Steps for verify predetermined alphanumeric characters (white back ground LP): (a) extracting candidate region, (b) vertical position histogram with LP border, (c) horizontal position histogram with LP border, (d) horizontal position histogram without LP border,(e) view of normalization candidate region after removing border and noisy area, (f) vertical position histogram (seven peaks for predetermined seven alphanumeric characters in LP region), and (g) character extraction.

7. Conclusion

In conclusion, a new method is adopted in this paper to select statistical threshold value in HSI color model. In the proposed method candidate regions are found by using HSI color model. These candidate regions may include LP regions; geometrical properties of LP are then used for classification. The proposed method is able to deal with candidate regions under independent orientation and scale of the plate. Finally, VLP regions containing predetermined LP alphanumeric character are verified and detected by using position histogram. Color arrangement and predetermined LP alphanumeric character of the Korean license plate are important features for verification and detection of license plate regions.

While conducting the experiments, different illumination conditions and varied distances between vehicle and camera often occurred. In that case, the result that has been confirmed is very much effective when the proposed approach is used. However, the proposed method is sensitive when vehicle bodies and their license plates possess similar colors. We leave these issues to be considered in future studies.

8. Acknowledgments

The authors would like to thank to Ulsan Metropolitan City, MKE and MEST which have supported this research in part through the NARC, the Human Resource Training project for regional innovation through KOTEF, the Human Resource Development Program for Convergence Robot Specialists through KIAT and post BK21 at University of Ulsan. Also, we express special thanks to IITA for their international graduate students' scholarship.

9. References

Anagnostopoulos, C.; Anagnostopoulos, I.; Loumos, V., & Kayafas, E. (2008). License plate-recognition from still images and video sequences: a survey. IEEE Trans. on Intell. Transp. Systs., Vol. 9, No. 3, September 2008, pp. 377-391

Anagnostopoulos, C.; Anagnostopoulos, I.; Loumos, V., & Kayafas, E. (2006). A license plate-recognition algorithm for intelligent transportation system application, IEEE Trans. on Intell. Transp. Systs., Vol. 7, No. 3, September 2006, pp. 377-392

Cano, J., & Perez-Cortes, J. C. (2003). Vehicle license plate segmentation in natural images. In: Lecture notes on computer science, Vol. 2652, pp 142-149

Chacon, M. I., & Zimmerman, A. (2003). License plate location based on a dynamic PCNN scheme. In: Proceedings of the Int. Joint Conf. on Neural Netw., Vol. 2, pp 1195-1200

Chang, S. L.; Chen, L. S.; Chung, Y. C., & Chen, S. W. (2004). Automatic license plate recognition. IEEE Trans. Intell. Transp. Syst., Vol. 5, No. 1, pp 42-53

Deb, K.; Chae, H. U., & Jo, K. H. (2008a). Vehicle license plate detection method based on sliding concentric windows and histogram. Journal of Computers, Vol. 4, No. 8, August 2009, pp 771-777

Deb, K. & Jo, K. H. (2008b). HSI color based vehicle license detection. In: Proceedings of the IEEE ICCAS, pp 687-691, October 2008, Seoul, Korea

Gao, Q.; Wang, X. & Xie, G.(2007). License plate recognition based on prior knowledge. In: Proceedings of the IEEE Int. Conf. on Automation and Logistics, pp 2964-2968

Hongliang, B. & Changping, L. (2004). A hybrid license plate extraction method based on edge statistics and morphology. In: Proceedings of the IEEE ICPR, pp 831-834

Huang, Y. P.; Chen, C. H.; Chang, Y. T. & Sandnes, F. E. (2009). An intelligent strategy for checking the annual inspection status of motorcycles based on license plate recognition, Expert Syst. with Appl., Vol. 36, No. 5, 2009, pp 9260-9267

Jia, W.; Zhang, H. & He, X. (2007). Region-based license plate detection. Journal of Network and Computer Appl., Vol. 30, No. 4, pp 1324-1333

Jiao, J.; Ye, Q., & Huang, Q. (2009). A configurable method for multi-style license plate recognition, Pattern Recognit., Vol. 42, No. 3, pp 358-369

Kamat, V. & Ganesan, S. (1995). An efficient implementation of the hough transform for detecting vehicle license plates using DSP'S. In: Proceedings of the IEEE Int. Conf. on RTAS, pp 58-59

Kim, K. I.; Jung, K., & Kim, J. H. (2002). Color texture-based object detection: an application to license plate localization. In: lecture notes on computer science, Vol. 2388, pp 293-309

Kim, S.; Kim, D.; Ryu, Y., & Kim, G. (2002). A robust license plate extraction method under complex image conditions. In: Proceedings of the IEEE Int. Conf. on ICPR, pp 216-219

Martin, F.; Garcia, M. & Alba, J. L. (2002). New methods for automatic reading of VLP's (Vehicle License Plates). In: Proceedings of the IASTED Int. Conf. on SPPRA

Matas, J., & Zimmermann, K. (2005). Unconstrained licence plate and text localization and recognition. In: Proceedings of the IEEE Int. Conf. on Intell. Transp. Syst.

Ming, G. H.; Harvey, A. L., & Danelutti, P. (1996). Car number plate's detection with edge image improvement. In: Proceedings of the Int. Symp. on Signal process. and its Appl., Vol. 2, pp 597-600

Nomura, S.; Yamanaka, K.; Katai, O.; Kawakami, H.; & Shiose T. (2005). A novel adaptive morphological approach for degraded character image segmentation, Pattern Recognit, Vol. 38, No. 11, pp. 1961-1975

Suresh, K. V.; Mahesh, K. G., & Rajagopalan, A. N. (2007). Superresolution of license plates in real traffic videos. IEEE Trans. Intell. Transp. Syst., Vol 8, pp 321-331

Wang, S. Z. & Lee, H. J. (2007). A cascade framework for a real-time statistical plate recognition system. IEEE Trans. Inf. Forensics Security, Vol. 2, pp 267-282

Xu, Z., & Zhu, H. (2007). An efficient method of locating vehicle license plate," In: Proceedings of the IEEE Int. Conf. on ICNC, pp 180-183

Yang, Y. Q.; Bai, J.; Tian, R. L. & Liu, N. (2006). Research of vehicle license plate location algorithm based on color features and plate processions. In: Lecture notes on computer science, vol 3930, pp 1077-1085

Zhang, C.; Sun, G.; Chen, D. & Zhao, T. (2007). A rapid locating method of vehicle license plate based on characteristics of characters connection and projection. In: Proceedings of the IEEE Int. Conf. on Industrial and Appl., pp 2546—2549

Zhang, H.; Jia, W.; He, X. & Wu, Q. (2006). Learning-based license plate detection using global and local features. In: Proceedings of the IEEE ICPR, pp 1102-1105

Zheng, D.; Zhao, Y., & Wang, J. (2005). An efficient method of license plate location. Pattern Recognit Lett., Vol 26. No. 15, pp 2431-2438

Umbaugh, S. E. (1998) Computer Imaging : digital image analysis and processing. CRC Press, ISBN 0-8493-2919-1, pp. 45-46

Shapiro, L. G. & Stockman, G. C. (2001). Computer vision. Prentice-Hall, ISBN 0-13-030796-3 New Jersey

Automatic Approaches to Plant Meristem States Revelation and Branching Pattern Extraction: A Review

Hongchun Qu and Qingsheng Zhu
College of Computer Science, Chongqing University
P.R. China

1. Introduction

Plant branching pattern, depends on the nature and on the spatial arrangement of each of plant parts (i.e. botanical entities, metamers or growth units, etc.), at any given time, is the expression of an equilibrium between endogenous genetic controlled growth processes and exogenous stimulations exerted by the nutrients supply and the micro-environmental climate, as well as the competition or cooperation from population (community). From botanical perspective, this expression can be viewed as the result of the repetition of elementary botanical entities through the three main and fundamental morphogenetic processes of growth, branching and reiteration (Barthélémy & Caraglio, 2007). Repetition of these entities induces gradual or abrupt changes reflecting different stages of differentiation in the meristems (Nicolini & Chanson, 1999), which are ordered in time and correspond to the notion of physiological age of meristems (Barthélémy & Caraglio, 2007).

Due to both endogenous control and exogenous effects, the development of meristem leads to some basic branching patterns that make the whole plant exhibits complex structures (Barthélémy & Caraglio, 2007). These basic branching patterns can be roughly divided into four types: 1) terminal or lateral branching, no branching (depends on the position of the active meristem, is the apical or axillary one), 2) monopodial or sympodial branching (depends on the indeterminate or determinate growth pattern of meristem, as shown in Fig. 1, Harris & Woolf, 2006), 3) immediate or delayed branching (depends on immediately or delayed initiation of meristem), and 4) rhythmic or continuous branching (depends on whether all the axillary meristems of a stem develop into lateral axes, or whether lateral axes are grouped as distinct tiers with an obvious regular alternation of a succession of unbranched and branched nodes on the parent stem).

As an intelligent organism (Trewavas, 2005), plant exhibits some kinds of intelligent behavioral capabilities through phenotypic plasticity (e.g. phototropism) other than movement, which is the nature of animals or human beings. This phenomenon demonstrates that the development of plant results from the mutual effect between structure and endogenous physiological process. The branching pattern analysis make it possible to identify these endogenous processes and to separate them from the plasticity of their expression resulting from external influences by means of observation and sometimes experimentation. Applicable to any kind of plant, branching pattern analysis has proved to be one of the most efficient means currently available for the study of the organization of complex arborescent plants. Therefore, the study of plant pattern and revelation of the corresponding meristem states will lead us to get a deeper and better understanding of plant development and also provide a convenient tool for growth rules construction for functional-structural plant modelling (virtual plants), which emerged as a new scientific discipline in the last decades.

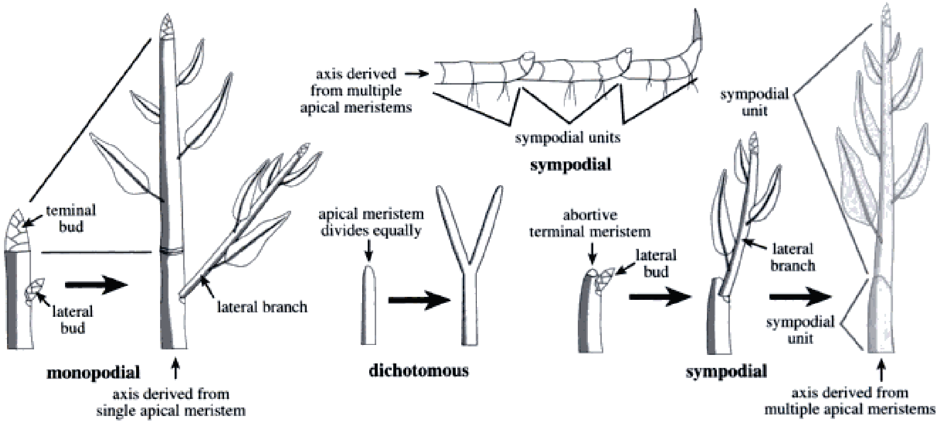

axis derived from multiple apical meristems
sympodial units
sympodial
sympodial unit
teminal bud
lateral bud
lateral branch
monopodial
axis derived from single apical meristem
apical meristem divides equally
dichotomous
abortive terminal meristem
lateral bud
lateral branch
sympodial unit
sympodial
axis derived from multiple apical meristems

Fig. 1. Stem branching patterns (Source: Harris & Woolf, 2006).

Nevertheless, plant branching pattern extraction and the corresponding meristem states revelation by no means a simple task. The extensive methodology used for analyzing the structures produced by meristems needs to be investigated. This can be seen as a methodology that aims to solve an inverse problem in which one tries to infer meristem functioning from the complex structures they produce (Fig. 2). Moreover, this analysis needs to be carried out at different spatial and temporal scales. Generally, the implementation of plant pattern extraction is usually composed of three steps: first, acquiring plant topological and geometrical data via manual work, image processing and pattern recognition, or 3D laser digitizing; second, analyzing these data to reveal hidden relations between plant entities (metamers or growth units) through statistical computing or topological operation; third, extracting the evolutionary rule set that reflecting the variation of meristem states from the second step to validate analysis and to guide the plant modelling. Therefore, the plant branching pattern extraction could be regarded as a complex machine learning system, in which many software and hardware tools as well as artificial intelligence methods are involved.

This chapter reviews detailed methods and approaches in relation to the complex machine learning system of automatic branching pattern extraction. First, we will introduce plant topological and geometrical description, encode database or structure used for storage of measured plant structure. And then, the most important part of this chapter, we will discuss recent methods and theories used for plant topology and geometry acquisition, statistical and structural analysis as well as branching rule extraction for any species of plant. Finally, some unsolved problems and challenges need to be addressed in future research are outlined.

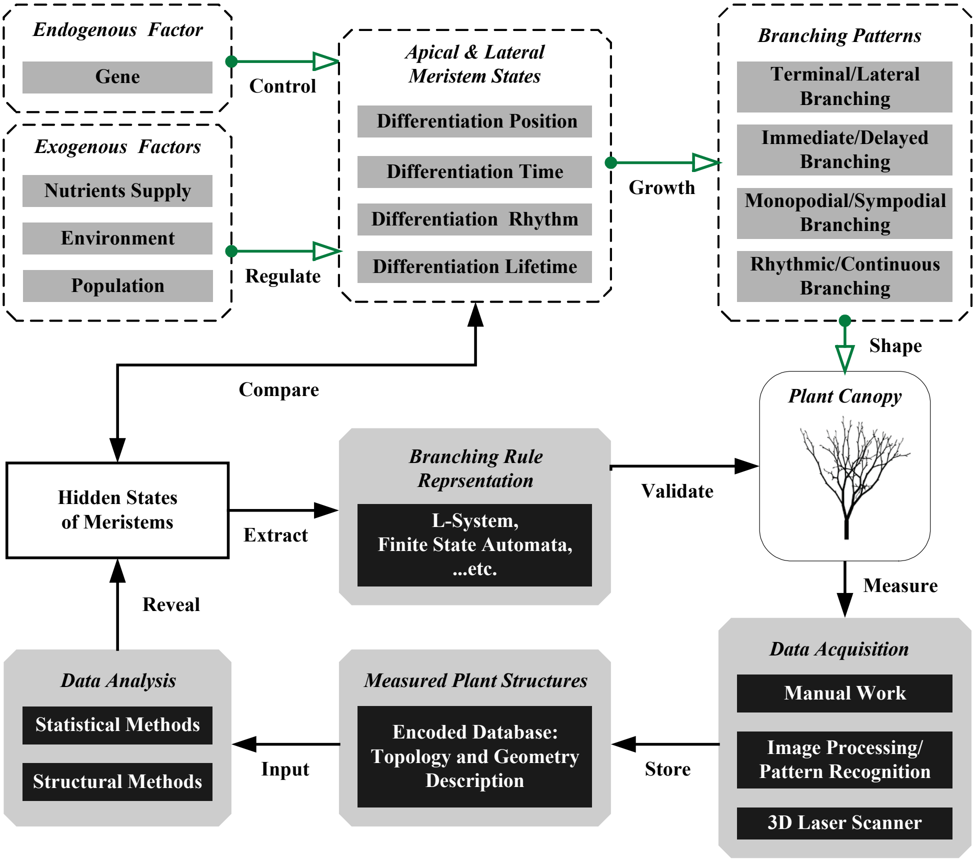


Fig. 2. Schematic description of meristem physiological states revelation and plant branching pattern extraction. Dash-line marked rounded-rectangles and green arrows represent plant branching mechanism and process, while the gray rounded-rectangles and dark arrows represent the branching pattern extraction process.

2. Plant Data Acquisition

Data acquisition is the starting point for extraction of plant branching patterns, yet the type of data used may vary greatly. The description of plant architecture therefore must be investigated and the corresponding architecture model or data structure for recording measured data needs to be established prior to the process of plant architecture measurement.

2.1 Description of plant architecture

As discussed by Prusinkiewicz (1998), on the most qualitative end of the spectrum, the architectural unit (metamer or growth unit) introduced by Edelin (1977) is well-suited to characterize plants within the conceptual framework of architectural models proposed by Hallé et al. (1978). The morphological characteristics incorporated into an architectural unit must be directly observed, estimated or measured. They include: the orientation of branches (e.g. orthotropic or plagiotropic), type of branching (monopodial or sympodial), persistence of branches (indefinite, long or short), degree of lateral shoot development as a function of their position on the parent branch (acrotony, mesotony or basitony), type of meristematic activity (rhythmic or continuous), number of internodes per growth unit, leaf arrangement (phyllotaxis), and position of reproductive organs on the branches (terminal or lateral). An authoritative description of these and other notions used to specify plant architecture was presented by Bell (1991) and Caraglio and Barthélémy (1997). The architectural unit acting as the basic component that make up the canopy consist of a set of these characteristics, and satisfy with all branch orders. Examples of architectural description of specific plants in terms of architectural units also have been investigated by Atger and Edelin (1995).

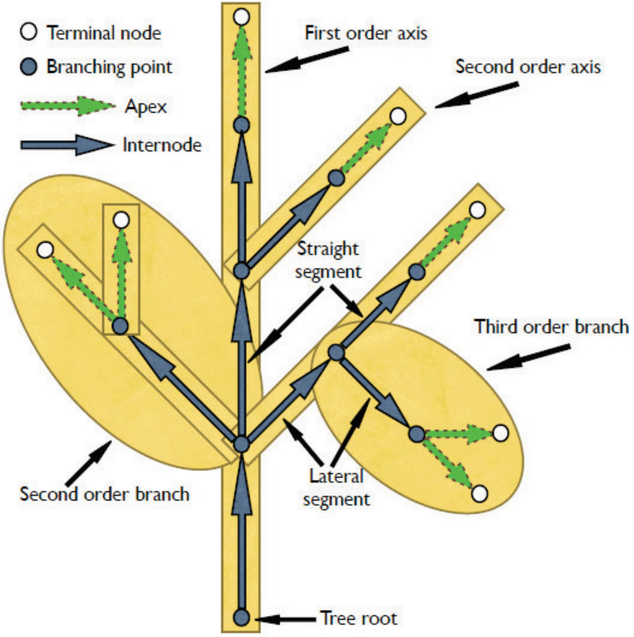


Fig. 3. Plant architecture representation: the multiscale tree graph (MTG). (Source: Godin and Caraglio, 1998).

Nevertheless, this qualitative characterization of architecture unit is insufficient to construct a spatial structure for a plant. The relations among architecture units are exact needed to be taken into consideration. Plant architecture is a dynamic expression of these basic architecture units, in the sense that the observed structural features reflect plant development over time. As stated by Hallé et al. (1978), "The idea of a form implicitly contains also the history of such a form." Correspondingly, the architecture of plant canopy may be viewed as a sequence of branch patterns created over time, rather than merely a set of branch patterns. "In this sequence, leading from axis 1 to the ultimate axes following the specific branching pattern, each branch is the expression of a particular state of meristematic activity and the branch series as a whole can be considered to be tracking the overall activity" (Barthélémy et al., 1991).

Plant maps (McClelland, 1916; Constable, 1991) may be considered the first attempt to characterize the structure for particular plants. This method of description captures the branching topology, i.e. the arrangement of branches, organs, and other features with no respect to the plant's geometry (e.g. the lengths of internodes and the magnitudes of the relative branching angles: the azimuth and the inclination). Plant maps can be recorded using different notations, e.g. Hanan and Room (1996) adapted the idea of plant maps using the bracketed string notation introduced by Lindenmayer (1968), which can be regarded as one of the most notable characters of the L-System (Prusinkiewicz et al., 1990). A different notation was presented by Rey et al. (1997). A refinement of the topological description of plants, named multiscale tree graphs (MTG, as shown in Fig. 3) has been proposed by Godin and Caraglio (1998). This makes it possible to specify plant topology at different scales and levels of detail, and incorporate temporal aspects into a single framework. Multiscale tree graphs form the basis of a coding language implemented in AMAPmod, an interactive program for analyzing the topological structure of plants (Godin et al., 1997a,b). The advantages and detailed descriptions for multiscale representation of plant architecture have also been discussed by Remphrey and Prusinkiewicz (1997).

For each species of plants, at each stage of development and in each environmental condition, the qualitative and quantitative topology and geometry can be measured via manual labour, depending on the complexity of the architecture. Small plants can be observed, manipulated and measured directly but this work is hardly accomplished when plants reach several metres high, furthermore it can be extremely time consuming. Therefore, automatic acquisition methods are preferred, e.g. image processing and pattern recognition, 3D magnetic-scan based digitizing as well as 3D laser scanning techniques.

2.2 Image-based Approaches

Varjo et al., (2006) proposed a digital camera based method for estimating the stem diameters of growing trees for forest inventory purposes. The imaging system consists of a single camera, a laser distance measurement device and a calibration stick placed beside the tree to be measured. To carry out the task, the camera geometry parameters are first determined using linear pinhole camera and nonlinear lens distortion models. In addition of the accurate camera calibration, the viewing geometry has to be determined for 3-D measurement purpose with the help of the calibration stick. The estimation of the stem diameters is carried out by combining the stem curve information from the image with a priori stem form model.

Lin et al. (2001) reported capturing top-view and lateral images taken from two color CCD cameras to measure several geometric features, such as seedling height, average projection area, leaf area index, leaf and stem node number, coordinates of stem nodes and leaf endpoints. The position and approximate shape of overlapped seedling leaves were initially located using elliptical Hough transform. Based on this information, the hidden leaf boundary can be further reconstructed and the total leaf area can be calculated without predetermined calibration relationship. This image-processing algorithm is incorporated into a stereo machine vision system to dynamically measure selected vegetable seedlings. However, this approach is better for small plant such as vegetables and bushes, not well-suited for large woody plants because of the difficult of image capturing.

Biskup et al. (2007) presented an area-based, binocular field stereo system to measure structural canopy parameters such as leaf angle distribution by using techniques such as calibration of cameras and stereo rig, epipolar rectification, colour segmentation of foliage and stereo matching.

Recently, Qu et al. (2009) proposed an image-analyzing-based method to analyze tree structure. In their method, any hand-held cameras with enough resolution (megapixels) can be employed to capture the image sequences of the unfoliaged deciduous plant of interest from a number of different overlapping views. Usually, about 30 to 45 images need to be taken, with coverage 360° around each plant. Then the camera parameters and a collection of 3D cloud points were recovered and extracted from point correspondences and running structure from motion on the captured image sequences. Standard computer vision methods (Hartley & Zisserman, 2000) have been used to estimate the point correspondences across the images and the camera parameters. Moreover, the method proposed by Lhuillier and Quan (2005) was used to compute a semi-dense cloud of reliable 3D points in space. Their image-based process shows a reasonable results for 3D skeleton extraction (Fig. 4).

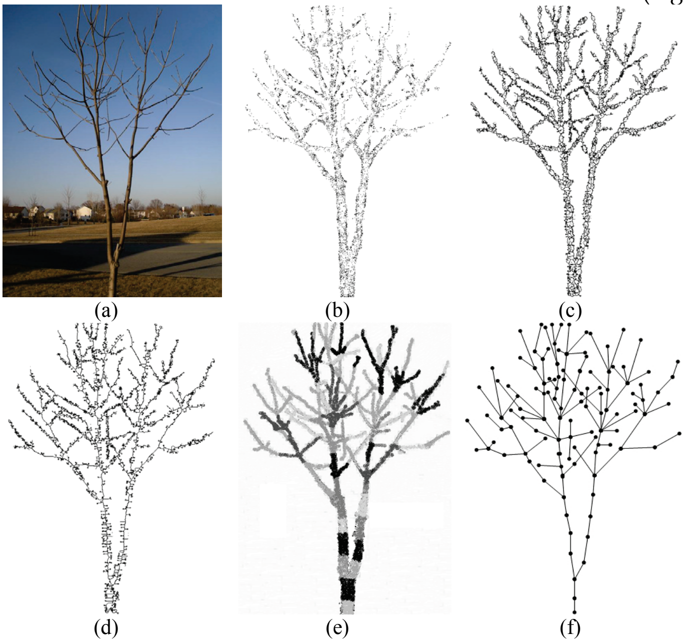


Fig. 4. Image-processing-based plant 3D skeleton recovery. (Source: Qu et al, 2009).

2.3 3D digitizing Approaches

Sinoquet and Rivet (1997) proposed a method for the measurement of the 3D architectural of a 20-year-old and 7-meter-high walnut tree. Their approach combines a 3D digitizing device (3SPACE FASTRAK, Polhemus) associated with the software DiplAmi designed for digitizer control and data acquisition management. It works at the shoot level and simultaneously measures the plant topology, geometry and the shoot morphology. Di iorio et al. (2005) used a low-magnetic-field digitizing device (Fastrak, Polhemus) to measure the geometry and topology of structural root with a diameter of 1cm for a single-stemmed Quercus pubescens tree. In their method, several root architecture characteristics are extracted by macros, including root volume, diameter, length, number, spatial position and branching order.

The algorithms proposed in (Gorte & Pfeifer, 2004) and (Pfeifer et al., 2004) took laser data as input, and created a voxel-based occupancy grid represetentation of the data. Morphological operations were used to find the underlying branching structure and fit cylinders to the branches. Moreover, these algorithms can be used to extract metric parameters of the tree, such as branch length, radius and rotation angles. Xu et al. (2007) proposed a method to reconstruct realistic looking trees from laser scan data. The laser data is first converted to a points cloud, a graph-based technique is used to find the 3D skeleton, and then the 3D information is used to measure the relative geometric parameters of plant branching structures. A similar approach was employed by Tan et al (2008) to find overall branch structures, but images instead of laser range scans are used as input, a structure from motion algorithm is used to create a 3D point cloud from the images. The 3D point cloud and the raw images are then used to find the branching structures.

However, aforementioned methods being developed for digitization of plant architecture are based on direct measurements of position and shape of every plant organ in space. Although they provide accurate results, these methods are particularly time consuming. More automatized methods are now required in order to collect plant architecture data of various types and sizes in a systematic way, i.e. these processes need to be completely implemented by hardware (3D scanner) instead of software.

3. Data Analysis

From botanical perspective, plant architecture is the result of repetitions that occur through growth and branching processes. During plant ontogeny, changes in the morphological characteristics of botanical entities exhibit either similar or very contrasted characteristics, which can be characterized as homogeneous zones. These homogeneous zones were discovered in most plant species with diverse characteristics (length, number of nodes, number of growth units, number of branches, non-flowering/flowering character) attached to the elementary botanical entities, these botanical entities being either built by the same meristem or derived from one another by branching. These results can be related to the notion of “physiological age of a meristem”. The physiological age of a meristem may be defined by a particular combination of morphological, anatomical and functional characteristics of a given botanical entity produced by this meristem (Barthélémy et al., 1997; Barthélémy & Caraglio, 2007). For identifying the physiological age of plant entities, it seems at first sight the most relevant to analyze directly the whole plant structure described at a given scale is to use appropriate analysis methods.

In the last two decades, coupled with precise morphological observations, architectural analysis of several plant species (Caraglio & Edelin, 1990) revealed that, under given environmental conditions, the structure and features of a particular elementary botanical entity are predictable and strongly dependent on both (1) its topological location in the comprehensive architecture of a plant and (2) the ontogenetic stage of the organism. At the level of the whole plant, the "morphogenetic gradients" notion was defined (Barthélémy et al.,1997) in order to take into account the intrinsic organization rules of plant structure and branching pattern and was shown to be a powerful concept (Prusinkiewicz et al., 2001) to explain the observed structure and series of modifications of botanical entities during the ontogeny of any plant species.

In order to enhance the understanding of this filed, some frameworks of investigation are required to reveal the hidden effects (the morphogenetic gradients) of the ontogenetic growth behavior, which should rely on appropriate analysis methods (most being statistical approaches). One challenge of this work is the complexity of the data which are tree-structured with variables of heterogeneous types (binary, count, quantitative, etc.) attached to each botanical entity. In the following section, we will focus on the discussion of statistical approaches to plant architecture and branching pattern, which are organized as the order of structural complexity: from axis to the entire shoot system.

3.1 Statistical Approaches

As discussed by Costes and Guédon (2002), it has been shown that over several growth periods, the growth and consequently the number of lateral, decreases rapidly with ageing (Ouellette & Young, 1995). Such a decrease in the growth and branching characteristics with plant development and ageing has been represented by Gatsuk et al. (1980) and Barthélémy et al. (1997) and has been discovered in most woody plants. As a consequence, when growing conditions are keeping optimal, the first annual growth of the stem developing from the grafted bud is the longest in the tree and bears the limbs which will later make up the plant architecture. This make it possible that evaluate plant growth and branching habits by analyzing the branching pattern of the first annual shoot of the trunk.

To test the aforementioned assumption, Costes and Guédon (2002) proposed a method of branching pattern analysis on 1-year-old trunks of six apple cultivars (Malus domestica Borkh.) using the AMAPmod software (developed by Godin et al., (1997,1999). Before the analysis procedure, the number of metamers (White, 1979), the location and the length of the sylleptic shoots were recorded from the shoot that had developed from the retained bud at the end of the first year of growth. Furthermore, at the end of second year of growth, three other types of axillary bud fate which led to proleptic development were recorded, they include: 1)spur or short shoot consisting of preformed organs only, with no or little elongation of the internodes, 2)long shoot, where the corresponding internodes are elongated and 3) bourse, resulting from the differentiation of the meristem into an inflorescence after the development of a few preformed leaves. Then the branching model on the trunks of these 1-year-old apple cultivars was established using the Hidden semi-Markov chain (HSMC, as shown in Fig. 5), which is particularly useful for identifying homogeneous zones within sequences and detecting transitions between zones (Ephraim & Merhav, 2002). In the model each state corresponding to a branching zone is denoted as a circle and the possible transitions between states are represented by arcs associating with probabilities. The occupancy distributions are listed above the corresponding states, as are the possible lateral types (denoted by symbol: 0, latent bud; 1, proleptic spur; 2, proleptic long shoot; 3, bourse; 4, sylleptic shoot) observed in each state.

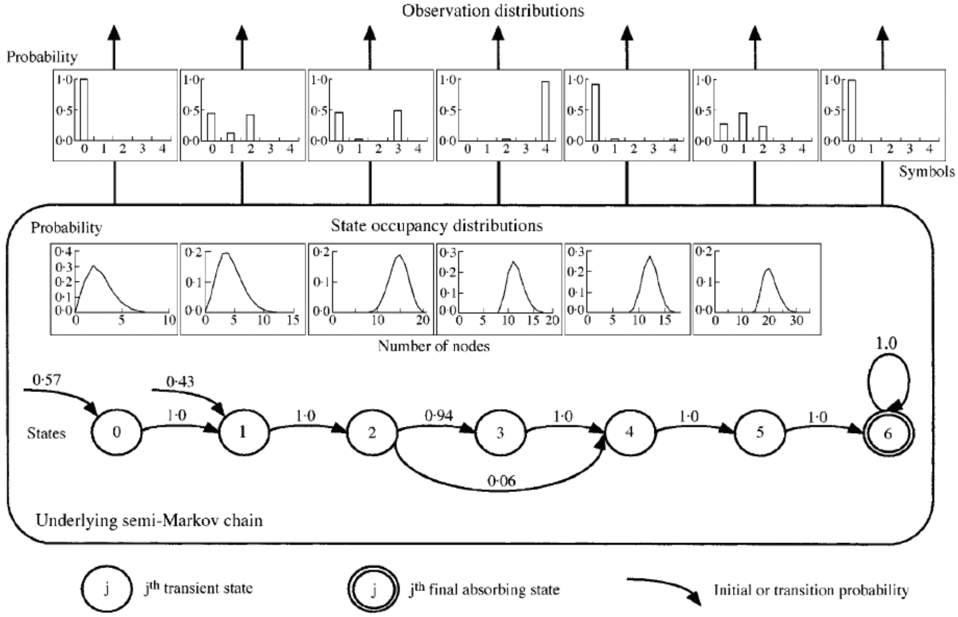

Observation distributions
Probability
1·0
0·5
0·0
0 1 2 3 4
Symbols
Probability
State occupancy distributions
Number of nodes
1.0
0·57
0·43
1·0
0·94
0·06
States
0
1
2
3
4
5
6
Underlying semi-Markov chain
j
jth transient state
jth final absorbing state
Initial or transition probability

Fig. 5. Modelling branching on 1-year-old trunks using a hidden semi-Markov chain. (Source: Costes and Guédon, 2002)

Their analysis results show that the succession of lateral types along trunks as discrete sequences highlighted the existence of successive zones within which the lateral type composition was homogeneous, but changed between zones. The five zones that were common to and located in the same position in all cultivars studied demonstrate that successive developmental stages occur in the same order over a growing season and can be used to explain the fate of meristems.

Although plant architecture are composed of repetition of growth units (White, 1979; Barlow, 1994), these growth units always show diversity on length with plant age and branching order (Gatsuk et al., 1980; Barthélémy et al, 1997). Plant branching patterns are likely to change with the length of growth units depending on plat growth stage (Costes et al. 2003) . To investigate this phenomenon, Renton et al. (2006) use the Hidden semi-Markov chain model to explore the similarities and gradients in growth unit branching patterns during plant ontogeny. Their experimental data (two 6-year-old Fuji apple trees) were encoded into a database corresponding to entire plant described at the node scale. Within this database, four types of growth units were measured: short (length < 5cm), medium (length between 5 and 20 cm), long (length > 20 cm) and floral growth units. And accordingly, five types of lateral growth were identified: latent buds, short lateral growth units, medium lateral growth units, long lateral growth units and floral growth units. Their Hidden semi-Markov chain model relies on three assumptions: 1) the branching types within a different zones (i.e. hidden states) are independent of growth unit length, year of growth and branch order, 2) each branching zone may be present or absent depending on growth unit length, year of growth and branch order, and 3) some branching zones may be longer or shorter, depending on growth unit length.

Analysis results of Renton et al. (2006) show that growth branching patterns exhibited both similarities and gradients during plant ontogeny. The degree of similarity of growth units over the years depends on their sharing certain zones, the floral and the short-lateral zones. Complex branching structures with more than one median branching zone tended to decrease in number towards the periphery, while the percentage of unbranched medium growth units progressively increased. Two phenomena also have been discovered: first, the two median zones disappeared with increasing plant age and branch order and second, the floral zone length decreased with the parent growth unit length.

The aforementioned statistical analysis of sequential data from plant architecture are mainly based on Markovian model, for instance the Hidden semi-Markov chains for investigating homogeneous zones of botanical entities (e.g. growth units). These models, although accurately accounting for the structure contained along remarkable paths in the plant (e.g. a plant trunk), are not suited for identifying tree-structured zones, because the dependencies among botanical entities of disjoint sequences are eluded. The complete topology has to be included in the investigation for the existence of multiple dependent successors or descendants to be considered in the distribution of zones. The statistical framework of the Hidden Markov tree (HMT) introduced by Crouse et al. (1998) in the signal-processing engineering just provides the appropriate solution for the analysis of tree-structured data.

Durand et al. (2005) proposed the Hidden Markov tree model to label the homogeneous zones in plant, which architecture is modelled by assigning one hidden state to each growth unit. The hidden state represents the class of the growth units. Each class contains growth units that have similar statistical properties or attributes such as the number of internodes, connection type (succession or branching), etc. Although the HMT is quite close to the Hidden Markov chains, both of which have the same parameter set and are based on local dependency assumptions between hidden states, the parameter estimation (EM algorithm, refer to Arthur Dempster et al. 1977) for HMT is different from Hidden Markov chains (Durand et al. 2004). Two successive steps: the parameter estimation from the measured entities and state tree restoration are executed in Durand et al. (2005) proposed approach. The state tree restoration makes the underlying zones (i.e. the hidden states) directly apparent : different zones in a same state have equivalent attribute distributions. The different distributions can be interpreted as an underlying stage of differentiation: the physiological age of the meristems. The plant is therefore automatically segmented into comparable parts, whereas states changes highlight where the ruptures (physiological states of meristems) are (as shown in Fig. 6). This HMT model assume that the transitions of hidden states conform to the first-order semi-Markov dynamics, because the first-order model is enough to reflect the statistical properties of plant and is easy to be learned. However, from a biological point of view, it is as yet a simplified assumption.

Plant development is controlled by the combined effect of gene activity and environmental constraints, which in turn combine with ontogenetic gradients. At a given date, a plant architecture is thus the outcome of three complex combination: 1) an endogenous component which is assumed to be structured as a succession of roughly stationary phases separated by marked change points asynchronous between individuals (Guédon et al., 2007), while the 2) environmental component which regulate the plant development are mainly of climatic origin such as light, rainfall or temperature, 3) the individual component corresponds to the local environment of each individual such as pathogen infestation or competitions between trees for light or nutrient resources. These factors are rarely measurable and not considered by aforementioned approaches.

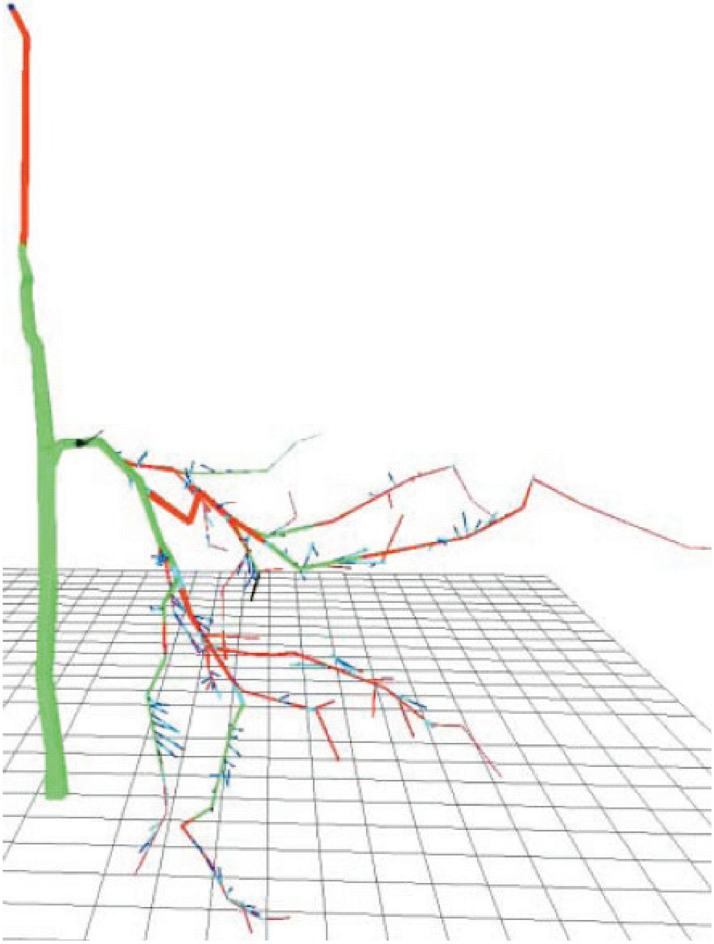


Fig. 6. Restored hidden state tree for the apple tree data set. Each growth unit is coloured according to its hidden state. (Source: Durand et al. 2005).

Incorporating both the influence of environmental variables and inter-individual heterogeneity in a hidden Markovian model is a challenging problem. Guédon et al. (2007) proposed a set of methods for analyzing the endogenous and the exogenous components. In particular, hidden semi-Markov chains with simple observation distributions were applied to plant growth data. In this case, the underlying semi-Markov chain represents the succession of growth phases and their lengths while the environmental component is characterized globally. Hidden semi-Markov chains (Guédon, 2003) generalize hidden Markov chains (Ephraim and Merhav, 2002) with the distinctive property of explicitly modeling the sojourn time in each state. Based on above works, Chaubert-Pereira et al.

(2008) introduced semi-Markov switching linear mixed models that generalize both Markov switching linear mixed models and hidden semi-Markov chains. These models can be regarded as a finite mixture of linear mixed models with semi-Markovian dependencies and make it possible to identify and to characterize the different growth components (e.g. endogenous, exogenous effects and competition or cooperation from population) of plants. The utilization of climatic covariates and individual-state-wise random effects renders the endogenous growth component more synchronous between individuals than with a simple Gaussian hidden semi-Markov chain. Moreover, the behavior of each plant within the population can be explored on the basis of the predicted individual-state-wise random effects.

Up to now, approaches we discussed merely focus on the topological relations among botanical entities of plant. However, the geometry of plant entities and spacial architecture of them are equally important to the revelation of meristems' hidden states, and furthermore, these information also make great help to plant architecture 3D modelling and reconstruction. Wang et al. (2006) proposed a novel tree modeling approach, efficiently synthesizing trees based on a set of tree samples captured from the real world. They designed a two-level statistical model for characterizing the stochastic and specific nature of trees. At the low level, the plantons, which are a group of similar organs, to depict tree organ details statistically. At the high level, a set of transitions between plantons is provided to describe the topological and geometrical relations between organs. The authors designed a maximum likelihood estimation algorithm to acquire the two-level statistical tree model from single samples or multi- samples.

3.2 Structural Analysis-based Approaches

As reviewed as Barthélémy and Caraglio (2007), most plants repeat their architectural unit during their development, late in ontogeny. Oldeman (1974) named this process "reiteration" and defined it as a morphogenetic process through which the organism duplicates its own elementary architecture, i.e. its architectural unit at different scale (node, metamer, growth unit, etc.). The result of this process is called a "reiterated complex" (Hallé et al., 1978; Barthélémy et al., 1988, 1991) or a "reiterate" (Millet et al.,1998). This property of plant architecture can be also called the "self-similarity" and consequently, provides an alternative to investigate the plant architecture and branching pattern.

Plant structures are usually represented by either ordered or unordered rooted tree (Prusinkiewicz & Lindenmayer, 1990; Godin & Caraglio, 1998). The intrinsic property of self-similarity make plant structure has some kind of redundancy, in some sense, that is the tree structure (graph) can be reduced to a minimum structure (graph) with the isomorphic structure to the previous one. The graph isomorphism can be defined as the edit-distance between two structures, as stated by Ferraro and Godin (2000). To study the redundancy of structure embedded at various levels in tree architectures, Godin and Ferraro (2009) investigated the problem of approximating trees by trees with particular self-nested structures. Self-nested trees are such that all their subtrees of a given height are isomorphic. Their investigation show that these trees present remarkable compression properties, with high compression rates. In order to measure how far a tree is from being a self-nested tree, a quantitative measure of the degree of self-nestedness for any tree has been introduced. For this, a self-nested tree has been constructed to minimize the distance of the original tree to the set of self-nested trees that embed the initial tree. To solve this optimization problem, a polynomial-time algorithm has been designed to make it quantify the degree of self-nestedness of a tree in a precise manner. The distance to this nearest embedding self-nested tree (NEST) is then used to define compression coefficients that reflect the compressibility of a tree.

From the view point of the structural analysis of botanical plants, one therefore can give a biological interpretation of the NEST of a tree based on the hypothesis that isomorphic tree structures at macroscopic levels are actually produced by meristems in identical physiological states. This makes it possible to show that the reduction graph of the NEST of a plant may be interpreted as the maximum sequence of differentiation states that any meristem of a plant may go through. Analysis results showed that the NEST of one plant may be interpreted in biological terms and reveals important aspects of the plant growth (Barthélémy and Caraglio, 2007).

4. Results Utilization

The statistical and topological analysis approaches discussed above make it possible to formally reveal the sequences of meristem physiological state differentiation corresponding to each axis of a given plant. These open up the perspective to use such an analysis on various plant species as a guiding principle to develop some applications or functional-structural plant models, and even to further explore the notion of meristem state and differentiation at a bio-molecular and genetic levels, in the spirit of the pioneering work described in (Prusinkiewicz et al., 2007).

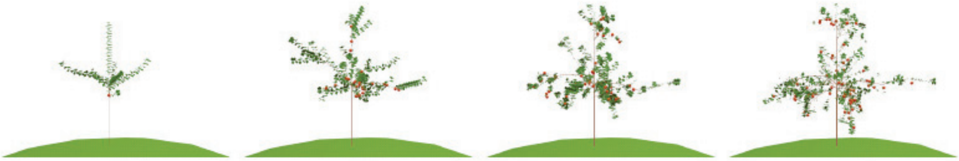


Fig. 7. Simulated apple trees using the MAPPLET model. (Source: Costes et al. 2008).

A new type of structure-function model named MAPPLET (Markov Apple Tree) has been developed by Costes et al. (2008). In MAPPLET, the statistical approach, which is inspired by the hierarchical Hidden Markov model has been carried out to model the development of apple trees (over the first six years of the growth). The tree topology of MAPPLET, i.e. both the succession of growth units along axes and the branching structures of growth units at node scale are controlled by the hidden states and spatial transitions between them, which are the results of the statistical approach: the Hidden Markov model. Moreover, the biomechanical model of MAPPLET simulates the bending of branches under fruit and branch weight. Therefore, from the global perspective, the MAPPLET is an integrated developmental framework can capture both the apple tree topology and its form (the shape of the branches, as determined dynamically by the gravity and the wood properties). The core simulation of MAPPLET is implemented using a L-system implemented with the L+C language (Karwowski & Prusinkiewicz, 2003) with which the statistical analysis module of

V-Plants (Renton et al., 2006) has been integrated. The simulation results of MAPPLET are shown in Fig. 7.

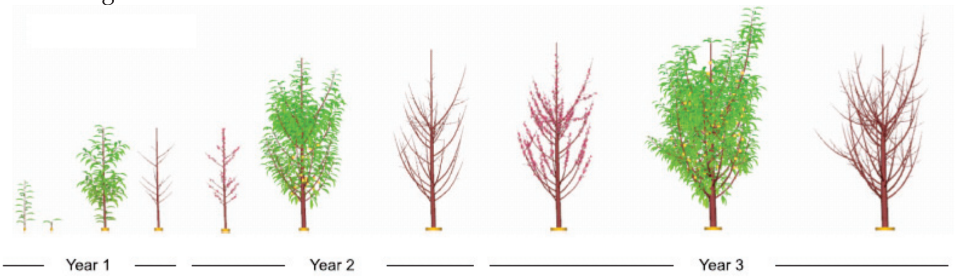


Fig. 8. L-Peach model output shows the development of the structure of a 3-year-old unpruned peach tree. (Source: Lopez et al., 2008).

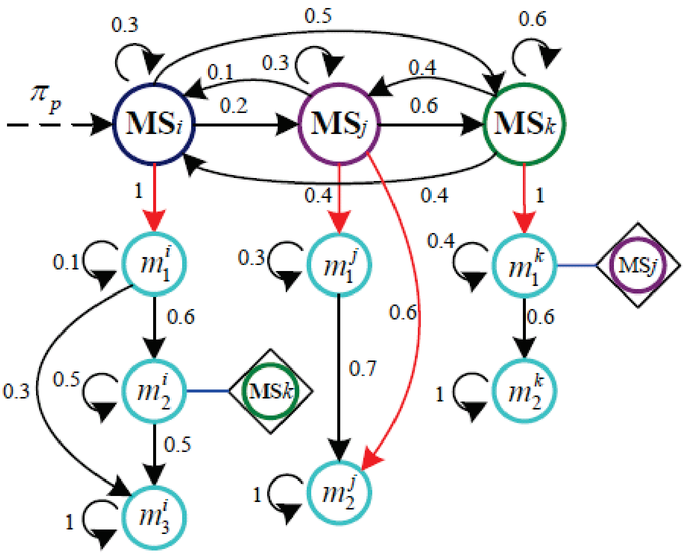


Fig. 9. Schematic structure of the Bidimensional Hierarchical Automaton (BHA), where cycles marked with MS$_{i,j,k}$. etc. represent the physiological states of meristems (hidden states of the Hidden Markov tree model), while the light blue cycles denote the different types of metamers (with different length of internodes). (Source: Qu et al., 2008).

Following this first integration of advanced stochastic processes for modeling tree topology with mechanistic processes, the approach was extended through the integration of Markovian models with the carbon-based source-sink model L-Peach (Lopez et al., 2007), which was developed from the original version of L-Peach (Allen et al., 2005). In the newest version of L-Peach, the Hidden semi-Markov chain is used to control the branching structure, it successfully reproduced peach trees that were similar to the peach trees observed in the real world (Fig. 8).

The branching patterns of plants not only can make convenience to the functional-structural plant model, but also open up the new perspective to plant architecture 3D reconstruction (Wang et al., 2006) as well as growth rule extraction (Qu, et al., 2008), which emerges a new scientific attempt. Over the last decades, L-System has been widely used as a powerful tool in plant modeling, in particular the plant branching control. However, it is really a difficult work to manually develop a L-System for a given plant species depending only on imagination or experience. Qu, et al. (2008) proposed a novel approach of automatic L-System discovery via branching pattern analysis of unfoliaged trees. In their approach, three steps are involved for L-System extraction: 1) image processing as well as pattern recognition methods are employed to recover topological and geometrical information for growth units and metamers from multiple images of unfoliaged trees, 2) Markovian methods are used to further analyze data which have been extracted in the first step for capturing the hidden relations between plant entities and, 3)the L-System has been generated via the runtime of a Bidimensional Hierarchical Automaton (BHA), which is constructed from the analysis result of the second step for describing plant branching structure, as shown in Fig. 9.

5. Conclusion

This chapter reviewed the approaches and theories in relation to the plant branching pattern extraction, those include plant architecture description, measurement and acquisition for topology and geometry of plant botanical entities, statistical and structural analysis for the revelation of physiological states of meristem as well as the utilization of these analysis results for plant modelling and 3D structure reconstruction.

The study of plant branching pattern requires detailed metric data about the plant architecture. Acquiring these metric information can be extremely time conuming when using manual labor. To address this issue, many researches contributed to the theoretical and applied approaches to automatically acquire plant topology and geometry, such as 3D laser scanning as well as image recognition. However, there are still some deficits in data acquisition need to be overcome. For instance the image processing based approaches, in which usually several images have been taken, if branch is not seriously occluded, a reasonable 3D branching structure can still be generated, but it will be obviously failed when a branch is fully occluded by other branches or leaves. Moreover, regarding the 3D digitizing approach, more automatized methods are now required in order to collect plant architecture data of various types and sizes in a systematic way, i.e. these processes (including laser scanning, 3D cloud computation and branching skeleton extraction) need to be completely implemented by hardware (3D scanner) instead of software.

Plant branching structure can be interpreted as the indirect transformation of different physiological states of the meristems, thus connected entities may exhibit either similar or very contrasted characteristics. During the last decades, some statistical models (e.g. Hidden semi-Markov chain, Hidden Markov tree, semi-Markov switching linear mixed model, etc.) have been employed by botanists and statisticians to discover and characterize homogeneous entity zones and transitions between them in different temporal scales within plant topological and geometrical data. These analytical methods and models lead to a clustering of the entities into classes sharing the similar statistical properties that help to find the tendency of the differentiation of meristems. One limitation of these stochastic methods must be mentioned is that one assume that the transitions of botanical entities conform to the first-order Markov dynamics, because the first-order model is enough to reflect the statistical properties of plants and also is easy to be learned. However, from the perspective of botany, it is as yet a simplified assumption. As an alternative approach, analysis of structural similarity has been explored to reduct a complex structure to a simplest one that may be interpreted as the maximum sequence of differentiation states that any meristem of a plant may go through.

In addition, Computer scientists proposed theoretical methods to integrate these hidden relations as growth rules into some classic complex systems such as parametric probabilistic L-System, Bidimensional Hierarchical Automaton, etc. Naturally, the mapping between plant growth process and these complex systems used for plant branching rules description are built. Moreover, these complex systems provide an open interface so that any virtual plants models can access it easily as long as they are compatible with this interface.

6. Acknowledgement

The authors are grateful to the Virtual Plants team (a joint team between INRIA, CIRAD and INRA, France, European Union) for their valuable comments. This research is supported by the National Natural Science Foundation of China (60773082), the National “863” Hi-Tech Research and Development Project of China (2006AA10Z233), the National Study-abroad Project for Joint PhD Program of Key Constructed High Level Universities in China (CSC-2008605048).

7. References

Allen, M. T.; Prusinkiewicz, P. & DeJong, T. M. (2005). Using L-systems for modeling source-sink interactions, architecture and physiology of growing trees: the L-PEACH model. *New Phytologist* 166, 869-880.

Atger, C. & Edelin, C. (1995). Un cas de ramification sympodiale à déterminisme endogène chez un système racinaire. *Platanus hybrida Brot. Acta. Bot.* Vol. 142, 23-30, Gallica.

Barlow, P. (1994). From cell to system: repetitive units of growth in the developement of roots and shoots. In: *Growth patterns in vascular plants* (I. M., ed.), pp. 19-58. Dioscorides Press, Portland.

Barthélémy (1988). Architecture et sexualité chez quelques plantes tropicales: le concept de floraison automatique, University Montpellier II.

Barthelemy, D. (1989). Levels of Organization and Repetition Phenomena in Seed Plants. *Seminar of the French Soc of Theoretical Biology*, pp. 309-323, Solignac Abbaye, France.

Barthélémy, D.; Edelin, C. & Hallé, F. (1991). Canopy architecture. *Physiology of trees* (A. S. Raghavendra, ed.), pp. 1-20, Wiley, London.

Barthélémy, D. & Caraglio, Y. (2007). Plant architecture: A dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny. *Annals of Botany* 99, 375-407.

Barthélémy, D.; Caraglio, Y. & Costes, E. (1997). Architecture, gradients morphogénétiques et âge physiologique chez les végétaux. In: *Modélisation et simulation l'architecture* *des végétaux* (d. R. P. Bouchon J, Barthélémy D, ed.), pp. 89-136. Science Update, Paris: INRA Editions.

Bell, A. D. (1991). Plant Form: An Illustrated Guide to Flowering Plants. Oxford Univ. Press, Oxford.

Biskup, B.; Scharr, H.; Schurr, U. & Rascher, U. (2007). A stereo imaging system for measuring structural parameters of plant canopies. *Plant Cell and Environment* 30, 1299-1308.

Caraglio, Y., Barthélémy, D. (1997). Revue critique des termes relatifs à la croissance et à la ramification des tiges des végétaux vasculaires. In: *Modélisation et simulation de l'architecture des végétaux* (D. R. P. Bouchon Jean, Barthélémy Daniel, ed.), pp. 11-87. INRA, Paris.

Caraglio, Y. & Edelin, C. (1990). Architecture et dynamique de la croissance du platane, Platanus hybrida Brot. (Platanaceae) fsyn. Platanus acerifolia (Aiton) Willd.g. In: *Bulletin de la Société Botanique de France, Lettres Botaniques* 137, 279-291.

Constable, G. A. (1991). Mapping the Production and Survival of Fruit on Field-Grown Cotton. *Agronomy Journal* 83, 374-378.

Costes, E. & Guédon, Y. (2002). Modelling branching patterns on 1-year-old trunks of six apple cultivars. *Annals of Botany* 89, 513-524.

Costes, E.; Sinoquet, H.; Kelner, J. J. & Godin, C. (2003). Exploring within-tree architectural development of two apple tree cultivars over 6 years. *Annals of Botany* 91, 91-104.

Costes, E.; Smith, C.; Renton, M.; Guedon, Y.; Prusinkiewicz, P. & Godin, C. (2007). MAppleT: simulation of apple tree development using mixed stochastic and biomechanical models. *5th International Workshop on Functional Structural Plant Models*, pp. 936-950, Napier, New Zealand.

Crouse, M. S.; Nowak, R. D. & Baraniuk, R. G. (1998). Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing* 46, 886-902.

Dempster, A. P.; Laird, N.M. & Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B* (*Methodological*) 39, 1-38.

Di Iorio, A.; Lasserre, B.; Scippa, G. S. & Chiatante, D. (2005). Root system architecture of Quercus pubescens trees growing on different sloping conditions. *Annals of Botany* 95, 351-361.

Durand, J. B.; Goncalves, P. & Guédon, Y. (2004). Computational methods for hidden Markov tree models - An application to wavelet trees. *IEEE Transactions on Signal Processing* 52, 2551-2560.

Durand, J. B.; Guédon, Y.; Caraglio, Y. & Costes, E. (2005). Analysis of the plant architecture via tree-structured statistical models: the hidden Markov tree models. *New Phytologist* 166, 813-825.

Edelin, C. (1977). Images de l'architecture des coniferes, Université de Montpellier II.

Ephraim, Y. & Merhav, N. (2002). Hidden Markov processes. *IEEE Transactions on Information Theory* 48, 1518-1569.

Ferraro, P. & Godin, C. (1998). A distance measure between plant architectures. *2nd International Workshop on Functional-Structural Tree Models*, pp. 445-461, Clermont Ferra, France.

Florence Chaubert-Pereira, Y. G.; Christian Lavergne & Catherine Trottier (2008). In: *Markov and semi-Markov switching linear mixed models for identifying forest tree growth components*.

Gatsuk, L. E.; Smirnova, O. V.; Vorontzova, L. I.; Zaugolnova, L. B. & Zhukova, L. A. (1980). Age States of Plants of Various Growth Forms - a Review. *Journal of Ecology* 68, 675-696.

Godin, C. & Caraglio, Y. (1998). A multiscale model of plant topological structures. *Journal of Theoretical Biology* 191, 1-46.

Godin, C.; Costes, E. & Caraglio, Y. (1997a). Exploring plant topological structures with the AMAPmod software: an outline. *Silva Fennica* 31, 357-368.

Godin, C.; Costes, E. & Sinoquet, H. (1999). A method for describing plant architecture which integrates topology and geometry. *Annals of Botany* 84, 343-357.

Godin, C. & Ferraro, P. (2009). In: *Quantifying the degree of self-nestedness of trees. Application to the structural analysis of plants* .

Godin, C.; Guédon, Y. & Costes, E. (1999). Exploration of a plant architecture database with the AMAPmod software illustrated on an apple tree hybrid family. *Agronomie* 19, 163-184.

Godin, C.; Guédon, Y.; Costes, E. & Caraglio, Y. (1997b). Measuring and analyzing plants with the AMAPmod software. In: *Advances in computational life sciences sciences, Vol I : Plants to ecosystems* (P. Michalewicz Maillard, ed.), pp. 63-94, Australia: CSIRO.

Godin, C.; Guédon, Y.; Costes, E. & Caraglio, Y. (1997). measuring and analyzing plants with the amapmod software. In: *Plants to ecosystems-advances in computational life sciences* (M. MT., ed.), Vol. 1, pp. 53-84. CSIRO Publishing, Collingwood.

Godin, C. & Sinoquet, H. (2005). Functional-structural plant modelling. *New Phytologist* 166, 705-708.

Gorte, B. & Pfeifer, N. (2004). Structuring laser-scanned trees using 3d mathematical morphology. *Proceedings of 20th ISPRS Congress*, pp. 929-933.

Guédon, Y. (2003). Estimating hidden semi-Markov chains from discrete sequences. *Journal of Computational and Graphical Statistics* 12, 604-639.

Guédon, Y.; Caraglio, Y.; Heuret, P.; Lebarbier, E. & Meredieu, U. (2007). Analyzing growth components in trees. *Journal of Theoretical Biology* 248, 418-447.

Hallé, F. (1978). Architectural variation at specific level of tropical trees. In: *Tropical trees as living systems* (Z. M. Tomlinson PB, ed.), pp. 209–221. Cambridge University Press, Cambridge.

Hallé, F. O.; R.A.A. & Tomlinson, P.B. (1978). Tropical trees and forests : an architectural analysis Springer Verlag, Heidelberg.

Hanan, J. S. & Room, P.M. (1996). *Virtual plants. A hypertext document and digitizing software distribution*. Cooperative Research Centre for Tropical Pest Management, Brisbane.

Harris, J. & Woolf, M. (2006). Systematic evidence and descriptive terminology. In: *Plant systematics* (M. G. Simpson, ed.). Amsterdam ; Boston : Elsevier/Academic Press, Burlington MA. U.S.A.

Hartley, R. I. a. Z., A. (2000). *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521623049.

Jari Varjo, H. H.; Juha Lappi; Jukka Heikkonen and & Juujärvi (2006). In: *Digital horizontal tree measurements for forest inventory*.

Lhuillier, M. & Quan, L. (2005). A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 418-433.

Lin, T. T.; Liao, W.C. & Chien, C.F. (2001). 3D graphical modeling of vegetable seedlings based on a stereo machine vision system. *ASAE Annual Meeting*, Sacramento, California,U.S.A. Paper No:01-3137.

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development, Parts I and II. *J. Theor.Biol.* 18, 280-315.

Lopez, G.; Favreau, R. R.; Smith, C.; Costes, E.; Prusinkiewicz, P. & DeJong, T. M. (2007). Integrating simulation of architectural development and source-sink behaviour of peach trees by incorporating Markov chains and physiological organ function submodels into L-PEACH. *5th International Workshop on Functional Structural Plant Models*, pp. 761-771, Napier, New Zealand.

McClelland, C. K. (1916). On the regularity of blooming in the cotton plant. *Science* 44, 578-581.

Millet J, B. A. & Edelin C. (1998). Plagiotropic architectural development and successional status of four tree species of the temperate forest. *Canadian Journal of Botany* 76, 2100-2118.

Nicolini E, C. B. (1999). La pousse courte feuillée, un indicateur du degré de différenciation chez le Hêtre (Fagus sylvatica L.). *Canadian Journal of Botany* 77, 1539-1550.

Oldeman, R. (1974). L'architecture de la forét guyanaise. *Mémoire no*. Vol. 73, Paris: O.R.S.T.O.M.

Pfeifer, N.; Gorte, B. & Winterhalder, D. (2004). Automatic reconstruction of single trees from terrestrial laser scanner data. *Proceedings of 20th ISPRS Congress*, pp. 114-119.

Prusinkiewicz, P. (1998). Modeling of spatial structure and development of plants: a review. *Scientia Horticulturae* 74, 113-149.

Prusinkiewicz, P.; Erasmus, Y.; Lane, B.; Harder, L. D. & Coen, E. (2007). Evolution and development of inflorescence architectures. *Science* 316, 1452-1456.

Prusinkiewicz, P.; Karwowski, R.; Mech, R. & Hanan, J. (1999). L-studio/cpfg: A software system for modeling plants. *International Workshop on Applications of Graph Transformations with Industrial Relevance* (ACTIVE 99) (M. Nagl, A. Schurr and M. Munch, eds.), pp. 457-464, Kerkrade, Netherlands.

Prusinkiewicz, P. & Lindenmayer, A. (1990). In: *The algorithmic beauty of plants*, Springer Verlag, New York.

Prusinkiewicz, P.; Mundermann, L.; Karwowski, R. & Lane, B. (2001). The use of positional information in the modeling of plants. *Siggraph 2001*, pp. 289-300, Los Angeles, Ca.

Prusinkiewicz, P. & Aristid, L. (1990). In: *The Algorithmic Beauty of Plants*, Springer-Verlag.

Qu, H. C.; Zhu, Q. S.; Guo, M. W. & Lu, Z. H. (2009). An Intelligent Learning Approach to L-Grammar Extraction From Image Sequences of Real Plants. *International Journal on Artificial Intelligence Tools*, To be published.

Qu, H. C.; Zhu, Q. S.; Zeng, L. Q.; Guo, M. W. & Lu, Z. H. (2008). Automata-Based L-Grammar Extraction from Multiple Images for Virtual Plants. *3rd International Conference on Bio-Inspired Computing - Theories and Applications* (D. Kearney, V. Nguyen, G. Gioiosa and T. Hendtlass, eds.), pp. 89-96. IEEE, Adelaide, Australia.

Quellette DV, Y. E. (1995). lateral shoot development in six diverse seedling populations of apple. *Fruit Varieties Journal* 49 248-254.

Radoslaw Karwowski & Prusinkiewicz, P. (2003). Design and Implementation of the L+C Modeling Language. *Electronic Notes in Theoretical Computer Science* 86, 1-19.

Radoslaw Karwowski & Prusinkiewicz, P. (2004). The L-system-based plant-modeling environment L-studion 4.0. *Proceedings of the 4th InternationalWorkshop on Functional-Structural PlantModels*, pp. 403-405, Montpellier, France.

Remphrey, W. R.; Prusinkiewicz, P. (1997). Quantification and modeling of tree architecture. In: *Plants to Ecosystems. Advances in Computational Life Sciences I*. (M. T. Michalewicz, ed.), pp. 45-52, CSIRO Publishing,Melbourne.

Renton, M.; Guedon, Y.; Godin, C. & Costes, E. (2006). Similarities and gradients in growth unit branching patterns during ontogeny in 'Fuji' apple trees: a stochastic approach. *Journal of Experimental Botany* 57, 3131-3143.

Rey, H.; Godin, C. & Guédon, Y. (1997). Vers une représentation formelle des plantes. In: *Modélisation et Simulation de l' Architecture des végétaux* (J. Bouchon, Reffye, P.D., Barthélémy, D., ed.), pp. 139-171. INRA Editions,Paris, .

Room, P. M.; Maillette, L. & Hanan, J. S. (1994). Module and Metamer Dynamics and Virtual Plants. *Advances in Ecological Research*, Vol. 25, 105-157.

Sinoquet, H. & Rivet, P. (1997). Measurement and visualization of the architecture of an adult tree based on a three-dimensional digitising device. *Trees-Structure and Function* 11, 265-270.

Tan, P.; Fang, T.; Xiao, J. X.; Zhao, P. & Quan, L. (2008). Single Image Tree Modeling. *ACM SIGGRAPH Conference 2008*, Singapore.

Trewavas, A. (2005). Plant intelligence. *Naturwissenschaften* 92, 401-413.

Van den Berg, C.; Willemsen, V.; Hage, W.; Weisbeek, P. & Scheres, B. (1995). Cell fate in the Arabidopsis root meristem determined by directional signalling. *Nature* 378, 62-65.

Wang, R.; Hua, W.; Dong, Z. L.; Peng, Q. S. & Bao, H. J. (2006). Synthesizing trees by plantons. *Visual Computer* 22, 238-248.

White, J. (1979). the plant as a metapopulation. *Annual Review Ecological Systems* 10, 109-145.

Xu, H.; Gossett, N. & Chen, B. Q. (2007). Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics* 26.

An Approach to Textile Recognition

Kar Seng Loke
Monash University Sunway Campus
Malaysia

1. Introduction

Batik and Songket motifs (Ismail, 1997) are traditional Malaysian-Indonesian cloth designs, with intrinsic artistic value and a rich and diverse history. Despite having a history spanning centuries, they are still valued today for their beauty and intricacy, commonplace amongst today's fashion trends. These patterns and motifs, however, defy a simple means of systematic cataloguing or indexing, and categorization. Linguistic terms are not accurate enough to identify or categorize, with sufficient accuracy, a particular textile motif, save for a few common design patterns due to the diversity of patterns.

The motifs themselves are usually highly stylised abstract designs derived from nature or mythology. The interesting thing about them, from the point of pattern recognition, is that the patterns are non-repeating but unmistakably belong to the same category; that is, according to the general theme of the non-repeating motifs, they belong to the same textile. Therefore, the pattern identification would have to be by example; making this ideal for content-based image retrieval and recognition.

While there are other approaches that try to classify individual patterns within the textile motifs, we approach problem as a form of "macro textures". Texture can be described as patterns of "non-uniform spatial distribution" of pixel intensities, that is to say that, intensity patterns are varying across space. In a similar manner, the Batik and Songket individual patterns vary across the textile but maintaining a similar theme. Therefore we adapt the approach for texture recognition and expand it to account for macro level variation as opposed to at pixel level. We are able to get good results on it, and considered among the best results reported.

In this paper, we will be using test images will be from a collection of traditional Batik and Songket design motifs. They will be used as input for performing classification and recognition by extending previous research on textile and texture recognition. The collection consists of 180 different samples (Ismail, 1997), sourced from 30 different texture classes (6 samples per class). Refer to Figure 1 for samples of the classes used in this paper.

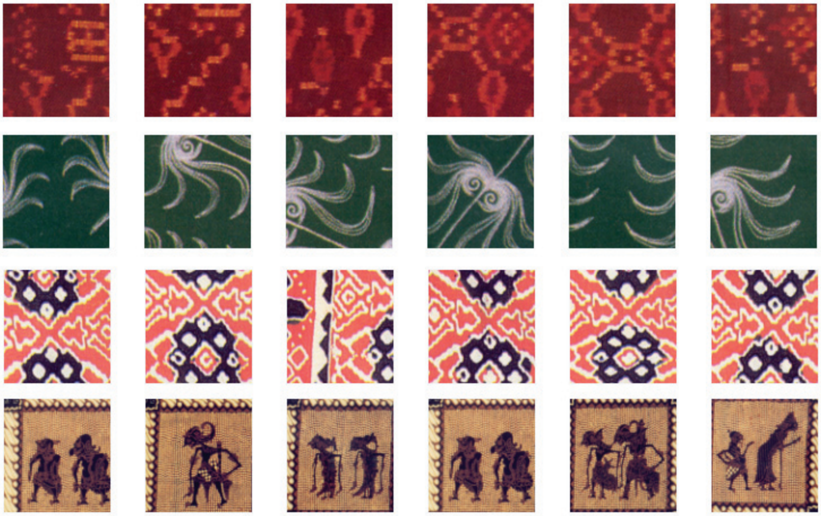


Fig. 1. Samples of texture motifs from 4 different classes used as sample data for this research.

2. Related Work

Grey Level Co-occurrence Matrix or GLCM (also known as Spatial-dependence Matrix) has been known as a powerful method (Davis, 1981; Walker et al., 1997) to represent the textures. Textures can be described as patterns of "non-uniform spatial distribution" of gray scale pixel intensities. Allam et. al (1997), citing Wezka et al. (1976) , and Conners and Harlow (1980) found that co-occurrence matrices yield better results than other texture discrimination methods. Haralick (1973) achieved a success rate of approximately 84% by using the extraction and calculation of summary statistics of the GLCM found in grayscale images, having an advantage in speed compared with other methods. Based on the good acceptance of GLCM approaches to texture recognition, in this research, we have adopted the use of GLCM as the basis for our textile motifs recognition. GLCM-based texture recognition have been used in combination with other techniques, including combining its statistical features with other methods, such as genetic algorithms (Walker et al., 1997). Practical applications of GLCM in image classification and retrieval include iris recognition (Zaim et al., 2006), image segmentation (Abutaleb, 1989) and CBIR in videos (Kim et al., 1999).

For use in colour textures, Arvis et al. (2004) have introduced a multispectral variation to the GLCM calculation that supports multiple colour channels, by separating each pixel's colour space into RGB components, and uses pairings of individual colour channels to construct multiple co-occurrence matrices.

We will be using the six RGB multispectral co-occurrence matrices – generated by separating each colour pixel into its Red, Green, and Blue components. RGB colour space is selected as opposed to others such as YUV and HSV, as it yields a reasonable (Chindaro et al., 2005) rate of success. The orthogonal polynomial moments for these six matrices are used as descriptors for the matrices in place of the summary statistics such as Haralick's measures (Davis et al., 1981). Allam et al. (1997) have also devised a method using orthonormal descriptors in their work on texture recognition on a 2-class problem, with a less than 2% error rate. Jamil et al. (2006a, 2006b) have worked on retrieval of Songket patterns based on their shapes using geometric shape descriptors from gradient edge detectors. Their method achieved their best "precision value of 97.7% at 10% recall level and 70.1% at 20% recall level" (Jamil et al., 2006a). Other approaches to textile recognition include using regular texel geometry (Han et al., 2009)

3. Description of approach

3.1 Co-occurrence Matrices in Image Representation

A source image with 256 possible colours is defined as I(x, y), with (x, y) determining the pixel coordinates, and the restriction of pixel values overlapping print: given by $0 \le I(x, y) \le 255$. The multispectral co-occurrence matrix (Arvis, 2004) represents the total number of pixel pairs in I(x, y) having a colour value i (from the channel a), and value j (from channel b).

A vector **T** may separate the pixel pairs where:

$$(x_2, y_2) = (t_x+x_1, t_y+y_1) \tag{1}$$

given (x_1, y_1) as coordinate of the first pixel, (x_2, y_2) for the second pixel. The reason that we introduce the vector **T** is to provide some degree of freedom when dealing with textures of a different scale (macrotextures have a larger **T**, microtextures on the other hand need a smaller **T**). To yield a co-occurrence matrix with rotation-invariance (to deal with all possible orientations of neighbouring pixels), the set of all possible t_x and t_y values must satisfy $x^2 + y^2 = r^2,\ r \in Z$, representing a fixed distance from the centre pixel.

Therefore, a co-occurrence matrix from channels *a* and *b* (a, b ∈ {R, G, B}) in I(x, y), separated by a vector **T** is represented mathematically as:

$$C_{ab} = \begin{bmatrix} c_{ab}^{\ 00} & \cdots & c_{ab}^{\ 0j} \\ \vdots & \ddots & \vdots \\ c_{ab}^{\ i0} & \cdots & c_{ab}^{\ ij} \end{bmatrix} \tag{2}$$

An elements of the above matrix, $c_{ab}(i_a, j_b)$ has the mathematical definition:

$$c_{ab}(i_a, j_b) = \sum\nolimits_{x,y} \sum\nolimits_{tx,ty \in U} \delta[I(x,y)-i] \times \delta[I(x+t_x, y+t_y)-j] \tag{3}$$

i_a and j_b are intensity values from channels a and b respectively, **T** is the distance vector as defined in (1) and $x, y \in I$. δ is is the Kronecker Delta.

Each of the six individual multispectral matrices, C_{ab} (a, b ∈ {R, G, B}) is converted to a grayscale image (having 256 possible shades of gray), $G_{ab}(i, j)$, such that $0 \le i, j \le 255$. The pixel intensity at any given position (i, j) correlates directly with the value in the co-occurrence matrix $C_{ab}(i, j)$, through the following equation:

$$g_{ab}(i,j) = \frac{c_{ab}(i,j)}{\max(c_{ab}(i,j))} \times 255 \tag{4}$$

After normalization, *min(c*$_{ab}$*(i, j))* will have g_{ab} = *0*, while *max(c*$_{ab}$*(i, j))* has g_{ab} = *255*. Histogram equalization is applied to improve the contrast of the generated matrices, which will improve visibility of outlying values in the graphical representation of the matrices.

Images in the same texture class will have a similar combination of six matrices which is distinct to a particular class (figure 2).

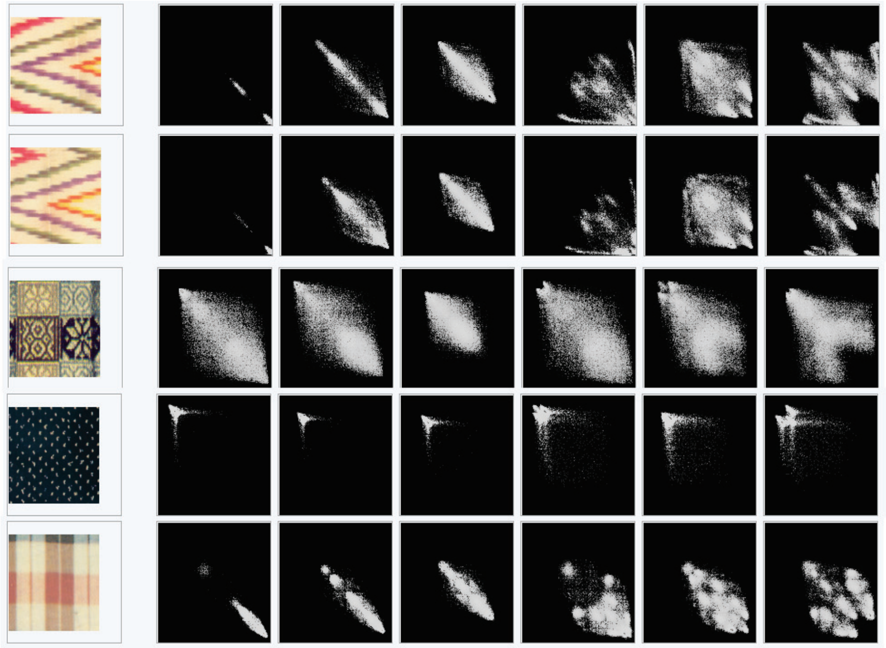


Fig. 2. Multispectral RGB co-occurrence matrices for 'Batik' motifs. Each row shows: 'Batik' motif and its corresponding matrices from the RR, GG, BB, RG, GB, and BR channels.

3.2 Orthogonal polynomial decomposition

We use orthogonal polynomials as a means of representing the information found in the co-occurrence matrices. Most GLCM or multispectral co-occurrence matrix-based methods of texture recognition uses a set of 'summary statistics' summarizing important textural features found in a particular image's matrix. Examples would be the five common features (Arvis, 2004) are derived from Haralick's (1973) original set of thirteen.

See et al. (2008) have shown that discrete orthogonal polynomials such as the Tchebichef discrete orthogonal polynomial can be an effective way of representing any 2D function. Various orthogonal polynomial moments, such as Zernike (Wang et al., 1998) and Hermite (Krylov et al., 2005) have been applied to texture classification. However, our approach differs in that we apply the orthogonal polynomial moments on the co-occurrence matrix image, not on the image directly.

Our approach require that the multispectral co-occurrence matrices to be treated as an image, and hence can be represented as a series of image moments (See et al., 2008; Kotoulas et al., 2005). We propose the usage of "shape" information from the multispectral matrices, by means of orthogonal polynomial decomposition, as a basis in texture recognition and classification. The decomposition coefficients would be larger but they contain more textural information as compared to the summarized set of 5 common Haralick features.

The Tchebichef orthogonal polynomial is used for the purposes of decomposition of the multispectral matrices. In the research of See et al. (2008), the Tchebichef orthogonal polynomial outperforms other polynomials in general, second only to the Discrete Cosine Transform which is used as the basis for comparison. Other orthogonal polynomials have limitations which render them unsuitable for decomposing our multispectral co-occurrence matrices. Specifically, Krawtchouk moments only work for binary images, Hahn only work for specific cases in which the foreground is significantly whiter than the background, and Poisson-Charlier generally yields unsatisfactory results [15]. Hence, the Tchebichef orthogonal polynomial is ideal for decomposing the six generated multispectral co-occurrence matrices and using the resulting moment coefficients as basis for texture discrimination.

The limited finite expansion of the moments allow only prominent features to be preserved while discarding those moments which carry little or no information. The first few moments encode gross overall shape and other moments carry finer details; thus, by discarding higher moments, we are able to save on complexity while preserving the entire set of second-order textural statistics in the multispectral matrix.

The transformation of image intensity into moment orders is defined mathematically as M_{pq} (See et al,. 2008):

$$M_{pq} = \frac{1}{\rho(p)\rho(q)} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} m_p(x) w(x) m_q(y) w(y) f(x,y) \tag{5}$$

$0 \le p, q, x, y \le N-1$; $m_n(x)$ is a set of finite discrete orthogonal polynomials, w(x) the weight function, and ρ(n) the rho function.

The Tchebichef polynomial is defined mathematically as (See et al, 2008):

$$m_n(x) = n! \sum_{k=0}^{n} (-1)^{n-k} \binom{N-1-k}{n-k} \binom{n+k}{n} \binom{x}{k} \tag{6}$$

$$\rho(n) = (2n)! \binom{N+n}{2n+1} \tag{7}$$

$$w(x) = 1 \tag{8}$$

where m_n is the n-th Tchebichef polynomial, ρ(n) the rho function and w(x) the weight function. Further details can be found in See et al (2008).

4. Classification

Firstly we compared the compared the results obtained from the Batik/Songket database and the VisText database. Images used are 100-by-100 pixel samples. For the 'Batik'/'Songket' database, 132 images from 16 different classes are used; for the VisTex database, 200 images from 40 different classes are used. A pixel radius of one unit, i.e. $r = 1$ is used for the construction of the multispectral matrices as it has been identified from prior research in GLCM to give optimum results. This results in 256 x256 GLCM matrices per image. We also need to compare the results obtained from varying orders from the discrete orthogonal polynomials.

For comparing two sample images S and T, we need to calculate the distance the visual representation S_{ab} and the visual representation T_{ab} ($a, b \in \{R, G, B\}$). The distance between the N^2 pairs of coefficients would have to be calculated. The distance(S_{ab}, T_{ab}) is defined as the Euclidean distance in the N^2 dimension between coefficients of S_{ab} and T_{ab}. Once the six distances for each of the six multispectral representations have been obtained, the final difference score, diff(S, T) is then obtained from the Euclidean distance (in the 6th dimension) of these six values. Hence, the smaller diff(S, T), the more similar S and T are ,diff(S, T) is 0 iff S=T; and diff(S, T) is symmetrical.

The k-nearest neighbor classifier is used to evaluate our findings, where $k = 3$. In order to estimate the moment order to use, we tested it from order 5 to 20. The percentage of correct classifications for the DCT and Tchebichef methods applied to our two data sets, versus the number of moment orders used in the process, is given in Figure 2 below. The best success rate was found using the Tchebichef orthogonal polynomial, with 10 as the best order of moments used. Some of these results have appeared in Cheong & Loke (2008a, 2008b). Overall the results using Vistex is better than using the Batik image database.

The Tchebichef orthogonal polynomial the reconstructed multispectral matrices strike a balance between preserving the shape of the matrices' visual representation and a good degree of variance when matching with other samples. DCT also creates a good approximation of the matrix pattern, however its reconstructions create a more rigid pattern while discarding certain outlying values visible in the matrix; this rigidity allows little room for error and will sometimes reject similar patterns. An order of 10 seems to allow for adequate intra-class variance. Lesser orders fail to capture the matrix shape well; greater orders result in a detailed reconstruction lacking in variance, causing certain samples to be rejected as false negatives.

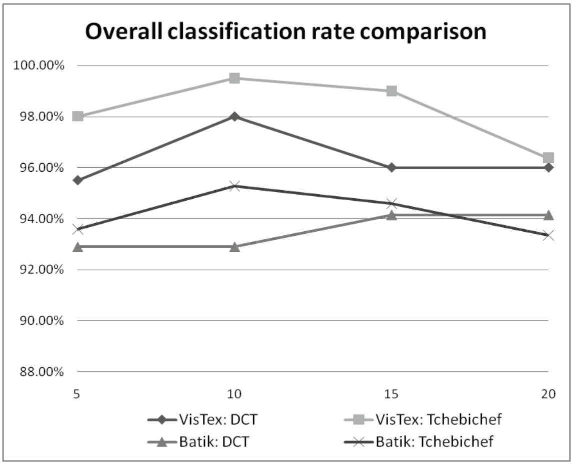


Fig. 2. Graph of average classification rate vs. number of moment orders used, for all sample data with both DCT and Tchebichef methods.

The best results we obtained (Figure 2) was using the 3kNN at 99.5% for VisTex textures and 95.28% for the Batik/Songket motifs.

Using Weka (Witten et al., 2005), we tested with various other classification methods to see if further improvements can be obtained using the best Tchebichef polynomial decomposition moment order set of 10. We also increased the number of samples to 180 encompassing 30 classes.

We used two unsupervised clustering algorithms and two supervised classifiers to classify our sets of generated moment coefficients.

The unsupervised clusterers are IBk (k-means with the k value set to the number of expected classes, i.e. 30), FarthestFirst (an optimized implementation of the k-means method); while the two supervised classifiers are BayesNet and kNN (k-nearest neighbour, with the k value set to 5). All of them use default parameters as defined in Weka. For the supervised classifier, we use 10-fold cross-validation to automatically partition the test and training data: the collection of sample data is partitioned into 10 mutually-exclusive partitions (called folds) (Kohavi, R., 1995).

The k-means algorithm by McQueen (1967) works to partition our sample data (unsupervised) into k distinct clusters. The naïve K-means algorithm does so by minimizing total intra-cluster variance; in the context of our methods, it tries to identify the samples which minimize the variance within a particular texture class, thereby properly grouping these samples by texture class. FarthestFirst (Hochbaum et al., 1985) is an implementation of an algorithm by Hochbaum and Shmoys, cited in Dasgupta and Long (2005). It works "as a fast simple approximate clusterer" modeled after the naïve k-means algorithm. kNN (the k-nearest neighbour) classifier works by assigning a texture (whose class is yet unknown) to the class in which the majority of its k neighbours belong to. In this case, we compare the linear distance between a texture sample and each of its k (we fix the value of k=5) neighbors, finally assigning it a class based on the majority of its 5 neighbours. The BayesNet Bayesian network learning algorithm in Weka uses the K2 hill-climbing strategy to construct a Bayesian network from the given coefficient data; by constructing a model to determine Bayesian probability of a single sample image as belonging to a class (Korb et al. 2004). The results have improved from earlier results, increasing the classification rate from 95.28% to 99.44% using the BayesNet and to 97.78% using the 5kNN classifier. Even the FarthestFirst returned better results compared to earlier classification runs.

The results are presented below.

Method	Samples	Correct	Incorrect	Percentage
Supervised: BayesNet	180	179	1	99.44%
Supervised: 5NN (kNN)	180	176	4	97.78%
Unsupervised: FarthestFirst	180	173	7	96.11%
Unsupervised: k-means (IBk)	180	167	13	92.78%

Table 1. Experimental results as determined in Weka for each of the four methods.

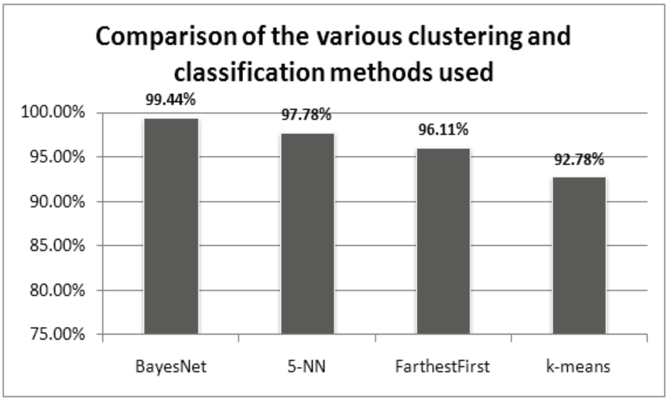


Fig. 3. Graph comparing the correct classification percentage for each of the four methods used

5. Dimension Reduction

The number of attributes generated, in order of 700, prompted us to study if the dimensionality can be reduced. As previously mentioned, using the Tchebichef orthogonal polynomial decomposition (with 10 moment orders) on 6 co-occurrence matrices yields a total of 726 attributes. The high number of attributes increases the complexity in storing the pattern descriptors. Another issue is the extended runtimes, deteriorating performance of the classification algorithms, and inefficiency of the knowledge discovery process due to irrelevant or redundant attributes, which could be compounded by the existence of a large number of samples in the knowledge base. Occam's Razor - in our case, the principle of using only the features that are necessary for textile classification - is the basis for our motivation to counter the 'curse of dimensionality'. Therefore, it is necessary to examine the effects of the reduced number of attributes on the accuracy of the classification.

If N is the number of moment orders used for the decomposition process then the total number of coefficients resulting from the decomposition process for each matrix is N^2. For the 6 matrices involved, the total number of coefficients per sample image is therefore $6N^2$. The Tchebichef orthogonal polynomial used in the decomposition of the 6 generated multispectral matrices resulted in the many attributes in the order of 700. This resulted in 6(10+1)2 = 726 moment coefficients because 10th order moments were used.

We first generate the 726 moment coefficients on our set of 180 texture samples (each being 16.7 million colours, of size 100-by-100 pixels). The total number of classes is 30 with 6 samples in each class. The coefficient data obtained is fed into Weka (Witten et al., 2005) for classification (see 4.1). Five-fold cross validation was used for testing. The co-occurrence matrix coefficients are generated from the database of 180 sample images of 30 classes of 'Batik' and 'Songket' textures. The coefficients are then stored in CSV format and imported into Weka for further analysis.

InfoGain (Dumais et al., 1998), one of the simplest attribute ranking methods, work by determining the Shannon information After testing using the entire raw coefficients, we further tested with dimension reduction on the coefficients. Different attribute-selection filters are applied on the data to reduce the dimensions of the coefficient. For each filter, 'maximum attribute' parameter is set to values ranging from 2 to 16, i.e. reducing to dimension to 2 to 16. Experiments were performed using 5-fold cross-validation to classify the data.

The Weka FilteredClassifier and AttributeSelectionFilter options are used for this purposes to ensure that the same attribute selections are applied for training set and test set. The attribute-selection filters used are independent of the classification algorithms used. The list of filters selected were the ones used in Hall et al (2003) and (Deegalla et al. 2007). They are the Information Gain Attribute Ranking method (InfoGain), the RELIEF method, and finally Principal Components Analysis (PCA).

InfoGain (Dumais et al., 1998), one of the simplest attribute ranking methods, work by determining the Shannon information gain between an attribute and its class; the higher the information gain, the more relevant the attribute is. RELIEF (Kira et. al 1992; Konoeneko, 1994) randomly samples an instance from the data, locates its nearest neighbours and uses their attribute values in turn to update relevance scores for each attribute. The underlying principle behind RELIEF is that useful attributes are similar for instances of the same class, and vice versa.

Random projection (RP) (Bingham et al., 2001) uses a random matrix to project the original data set into a lower dimensional subspace. RP depends on the Johnson and Lindenstrauss theorem (Dasgupta et al., 2003) which states that any points in a d-dimensional Euclidean space can be mapped to a smaller k-dimensional Euclidean space while maintaining all pair-wise distance within an arbitrarily small factor.

PCA uses a linear transform to project the original attribute space to a lower dimensional subspace. Both PCA and RP are unsupervised in that class information is not required, whereas InfoGain and RELIEF are supervised, i.e. it uses class information for attribute selection.

For classification testing, we used the k-nearest neighbor lazy classifier (Aha et al., 1991), with k=1 (IB1) and k=3 (IB3), and the Bayesian Network (BayesNet) classifier. The k-nearest neighbor classifier works by assigning a sample to the class in which the majority of its k neighbors belong to. BayesNet in Weka constructs a Bayesian network from the data; by constructing a model to determine Bayesian probability of a single sample as belonging to a class. The advantage of BayesNet is that it can take into consideration the conditional dependency of attributes.

5.1 Dimension Reduction Results

The experimental results obtained via Weka are presented in the following figures 4-7.

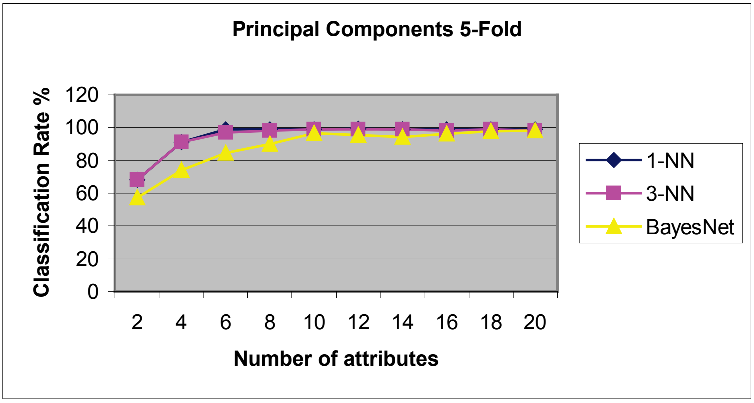


Fig. 4. Plot of classification rate versus number of attributes using Principal Components Analysis with 5-fold cross-validation.

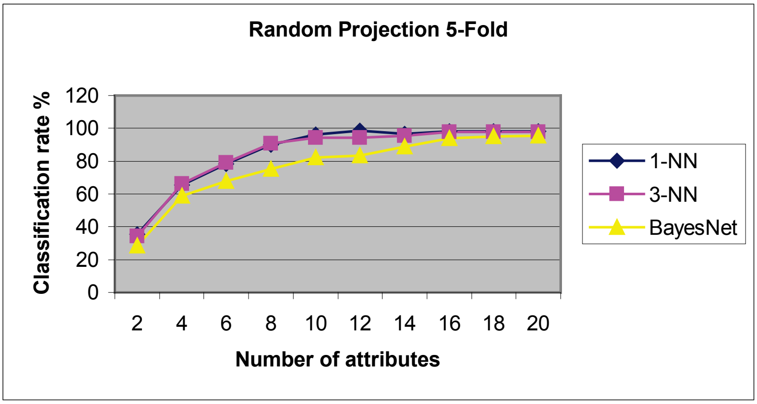


Fig. 5. Plot of classification rate versus number of attributes using Random Projection with 5-fold cross-validation.

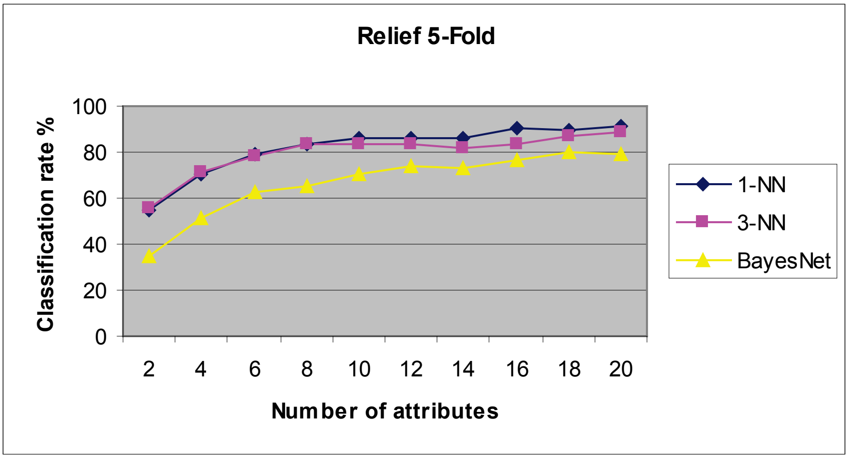


Fig. 6. Plot of classification rate versus number of attributes using RELIEF Attribute Evaluation with 5-fold cross-validation.

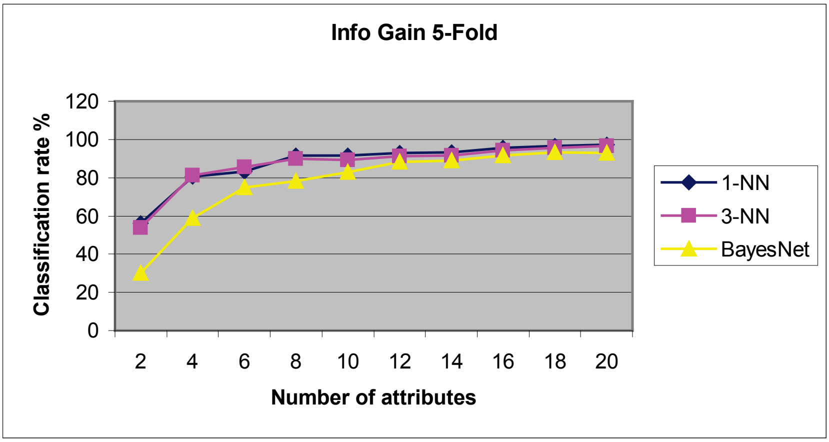


Fig. 7. Plot of classification rate versus number of attributes using Info Gain Attribute Evaluation with 5-fold cross-validation.

Results in figures 4-7 were obtained on the same data set using 5-fold cross validation. Results in figure 8 were obtained using a new test set of 60 samples, each class represented by 2 samples, and trained entirely on the data set used in the 5-fold cross validation test.

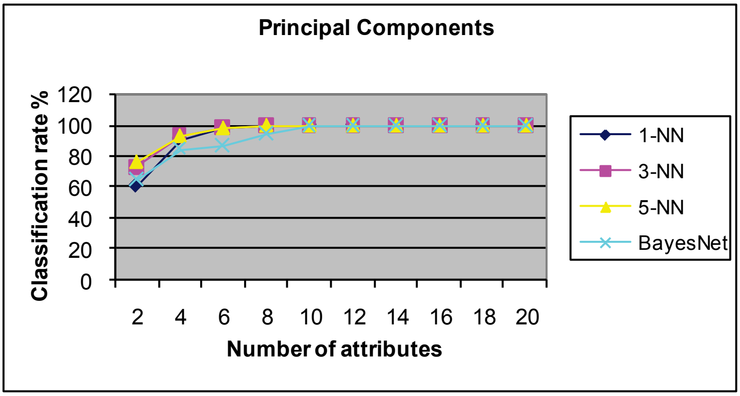


Fig. 8. Plot of classification rate versus number of attributes using Principal Components Analysis with new test data set.

The computation of the model using RP took the least time, ranging from 0.02-0.08 seconds. This was followed by Info Gain which took around 0.6s, RELIEF about 4s, and finally PCA which ranges around 8s.

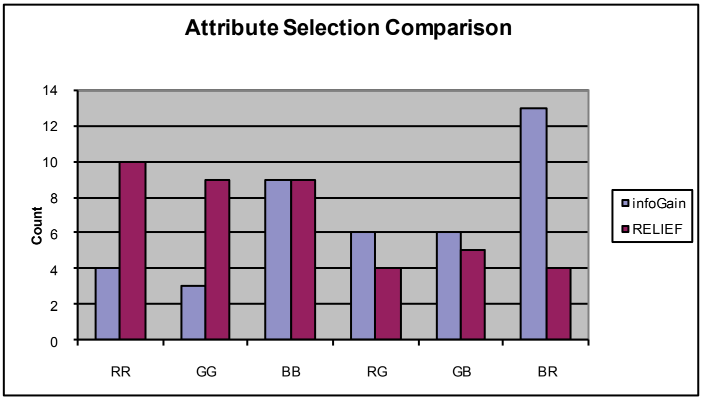


Fig. 9. Comparison of attribute selection of InfoGain and RELIEF

The results from dimension reduction are rather mixed. The supervised method of attribute reduction InfoGain and RELIEF returned the worse results. BayesNet did not particular work well on RELIEF. However the 1-NN returned respectable results using InfoGain attributes. Analyzing the attribute top 40 rankings from RELIEF indicate that it preferred attributes selected from the RR, BB and GG co-occurrence matrix whereas InfoGain preferred BB and BR attributes higher (see figure 9). There was considerable overlap in the BB, RG and CB.

The best results are obtained on unsupervised methods using subspace projection namely PCA and RP method. PCA returned the best results peaking at 8 attributes for 1-NN, 10 attributes for 3-NN, and 20 attributes for BayesNet, the results returned respectively are

98.9%, 98.9% and 98.3% correct classification rates. Random projection required twice the amount of attributes to reach their best values, for 1-NN and 3-NN at 16 attributes, and 20 attributes for BayesNet. The returned results are 98.3%, 97.8% and 95.6% correct classification rates respectively. The results on PCA reduction using 20 attributes only differs by only 1% using the full attribute set. The results using nearest neighbour classification in fact is better than obtained using the full attribute set.

The results obtained for PCA and RP are in agreement with the results obtained in (Deegalla et al., 2007), that is random projection requires a larger number of dimensions compared to PCA to achieve comparable results. In this case 1-NN and 3-NN classification peaked at around 8 attributes for PCA compared to 16 for RP. This is a significant reduction from 726 attributes needed in the original method. Even though nearest neighbour classification is not efficient for large datasets, this reduction in the number of attributes will increase the computational efficiency.

To confirm the results, we did additional testing on a new set of test data not included earlier. The data set consists of 30 classes with 2 samples in each class. This will also allow us to test if using a larger training set will increase the accuracy of the model. The results are in figure 8. The results showed that the results have improved when trained on a larger training set. 100% correct classification was achieved using 8 PCA attributes for 1-NN and 3-NN, while BayesNet reached 100% classification using 10 PCA attributes.

6. Discussion and Analysis

Prior research on the GLCM has focused predominantly on textures. Arvis et al. (2004) with their multispectral co-occurrence matrix method, with a 5-Nearest Neighbours classifier yielding a 97.9% percentage of good classification for VisTex textures. Previous research work involving color texture analysis using a combination of Gabor filtering and the multispectral method on the Outex (Ojala et al., 2002) database has yielded a rate of success of 94.7%. Allam's (1997) result of a 2% error rate differs in the fact it is only applied to a 2-class problem, restricted to grayscale texture. This differs in our motivation of using the "shape" of the co-occurrence pattern, and we achieved between 98%-100% classification on Batik/Songket.

The results for 'Batik' and 'Songket' achieved here are among the best for such kinds of irregular textile patterns based on the limited prior research found. Our experimental tests on co-occurrence matrices using summary statistics suggest that summary statistics may not always capture the full representation of the co-occurrence matrix. The rationale being that it is possible for many similar distributions to have the possibility of producing a similar value (Walker et al., 1995). An illustration of such a case is as follows, whereby the 5 common Haralick features combined with the 6 multispectral matrices yield a very low Euclidean difference even though the two samples (below, figure 10) are of visually different texture, highlighting the inadequacy of the statistical measure especially in non-uniform and colored texture images.

However, for development of a successful end-user application, some issues still need to be addressed, namely lighting variation and scale.

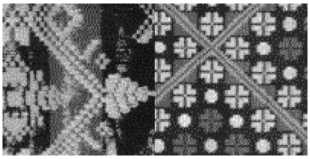


Fig. 10. Sample of two images with similar Euclidean distance using Haralick features.

Textile design motifs such as those found in these textiles tend to have a more non-uniform distribution in the GLCM as opposed to textures. This also makes it difficult to be captured by Haralick's summary statistics as the "shape" information is not adequately represented. Our method has the best success rate using the Tchebichef orthogonal polynomial, with 10 order of moments used (Cheong et al., 2008). This is due to the fact that with Tchebichef, the reconstructed matrices strike a balance between preserving the shape of the matrices' visual representation and a good degree of variance when matching with other samples.

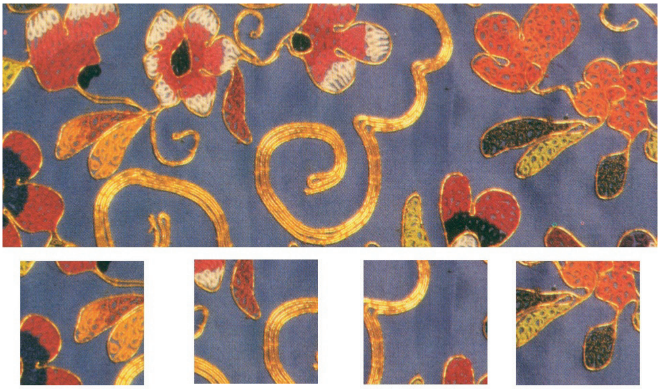


Fig. 11. (Top) The full Batik cloth. (Bottom) Sampled regions used for training are not identical but bear "family resemblance".

The results indicate that nearest neighbour classification perform slightly better than BayesNet. This is probably because the textile patterns that we used are not identical, nor are they, considered on the whole, statistically uniform. They are more akin in a "family resemblance" manner (Wittgenstein, 1953). This can be explained by that no sample within the same class shares all the features, but each sample in the class shares overlapping features with each neighbours (see figure 11). Or as Wittgenstein puts it: "Something runs through the whole thread - namely the continuous overlapping of those fibres".

Reduction from 726 to 8 attributes means that only 1.1% of the original information is significant for classification. Arvis (2004), which achieved 97.9% on VisTex textures, still required 30 attributes based on 5 measures specified on each colour pair co-occurrence matrix. Based on the results presented here a reduction of down to 2% from the original attributes is adequate for classification.

7. Potential Applications

A simple non-textual access to textile patterns is capable of opening up a wealth of applications. For designers it could unlock their creative potential, by using access to textile pattern collection for inspiration or to stimulate innovation. One such project is Fashion and Apparel Browsing for Inspirational Content (FABRIC) (Ward et al., 2008). They could use it compare the designs, or to study them by browsing, or to survey the trend. It could also be used to avoid copyright issues, and to distinguish one's work by stamping their uniqueness on it.

There are also cultural aspects to it, because an accessible collection of patterns could be used for archival purposes, storing the narrative, the times and trends of particular groups of people through history, as well as charting the changes. It would be useful for understanding historical trends, as many of the patterns may shed different narratives throughout their history. In the Malay Archipelago, textiles, apart from artistic expression are also linked to religious and cultural beliefs. The patterns in the textile are a means of communication between the human and spirit world, and play a significant role in birth and death rites, whereby it is thought that the more powerful patterns, the more potent protection they offer (Hout, 1999).

Commercially an efficient method of comparing and recognizing textile patterns could spur the application of visual comparison shopping for fashion and clothing. A visual-based search engine could let shoppers select similar items based on colour, shape and pattern, in addition to price. Useful categories for such comparison shopping include shoes, handbags, and clothing. A usage scenario would that a shopper has some clothes of a particular pattern, but would like a matching pattern for the shoes, or sees a pattern that he or she likes, and wants possibly a matching pattern for shoes or clothing. This could be extended to mobile-based comparison shopping. In this scenario, the mobile phone camera snaps an image of the pattern, and the online store searches for similar items available.

8. Conclusion

We have successfully demonstrated the multispectral co-occurrence matrices method for use in the recognition of Batik and Songket design motifs and introduced the use of the Tchebichef orthogonal polynomial to decompose each of these matrices into a series of moments as a means to capture more complete second-order pixel statistics information.

The advantage to this method is having a good degree of accuracy as compared to the use of summary statistics which is commonly used in GLCM research. We have also shown that this method is viable in matching non-uniform design motifs as opposed to only textures. This makes our approach suitable to be used in image retrieval applications for not only traditional Batik and Songket textile motifs but other design motifs. While Haralick's measures (1973) have been successfully applied to texture recognition, it is not so good for non-uniform patterns like textile motifs.

We have shown that a significant reduction in attributes down to about 2% of the original attributes contributed only slight deterioration of classification rate.

This makes this approach, combined with an appropriate attribute selection scheme, suitable for fast content-based retrieval applications, not only for traditional Batik and Songket textile motifs, but other design motifs where the patterns are overlapping in similarity. In particular, the application of principal components attribute reduction provided the highest accuracy at a higher computational cost. If computational cost is an issue, then the random projection method returned respectable results, the next best compared to other methods tested.

9. References

Abutaleb, A.S. (1989). Automatic thresholding of gray-level pictures using two-dimensional entropies. Computer Vision Graphics Image Processing, Vol 47, pp. 22-32.

Aha, D.W., D. Kibler, and M.K. Albert. (1991). Instance-Based Learning Algorithms. Machine Learning, 6, p. 37-66.

Allam, S., M. Adel, and P. Refregier. (1997). Fast algorithm for texture discrimination by use of a separable orthonormal decomposition of the co-occurrence matrix. Applied Optics, Vol 36, No 32, pp. 8313-8321.

Arvis, V., et al. (2004). Generalization of the cooccurrence matrix for colour images: application to colour texture classification. Image Analysis and Stereology, Vol 23, pp. 63-72.

Bingham E. & Mannila, H.(2001). Random projection in dimensionality reduction : applications to image and text data,In: Knowledge Discovery and Data Mining, pp. 245-250.

Cheong, M. & K.S. Loke. (2008a). An Approach to Texture-Based Image Recognition by Deconstructing Multispectral Co-occurrence Matrices using Tchebichef Orthogonal Polynomials. Proceedings of ICPR, Tampa, Florida, Dec 8-11, 2008.

Cheong, M. & K.S. Loke. (2008). Textile Recognition Using Tchebichef Moments of Co-occurrence Matrices, In: Lecture Notes in Computer Science LNCS 5226, pp. 1017-1024. 2008. Springer-Verlag Berlin Heidelberg.

Chindaro, S., K. Sirlantzis, and F. Deravi. (2005). Texture Classification System using Colour Space Fusion. Electronics Letters, 2005, 41, 10.

Cohen, F.S., Z. Fan, S. Attali. (1991). Automated Inspection of Textile Fabrics Using Textural Model. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 8, pp. 803-808, August.

Conners, R.W. & C.A. Harlow.(1980). A theoretical comparison of texture algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1980. Vol 3: pp. 204-222.

Dasgupta, S. and A. Gupta. (2003). An elementary proof of a theorem of Johnson and Lindenstrauss. Random Structures & Algorithms, 2003. 22(1): p. 60-65.

Deegalla S., Bostrom, H. (2006). Reducing High-Dimensional Data by Principal Component Analysis vs Random Project for Nearest Neighbor Classification. 5th Internatioal Conference on Machine Learning and Applications20 06, pp. 245-250, Orlando, Florida, Dec 2006.

Davis, L.S. (1981). Image Texture Analysis Techniques - a Survey, In: Digital Image Processing, J.C. Simon and R.M. Haralick, Editors, D. Reidel: Dordrecht, The Netherlands.

Hall, M.A., Geoffrey, H. (2003). Benchmarking Attribute Selection Techniques for Discrete Class Data Mining. IEEE Transaction on Knowledge and Data Engineering, 2003. 15(3): p. 1-16.

Han J, McKenna S J. (2009). Classifying and Comparing Regular Textures for Retrieval Using Texel Geometry. International Conference on Computer Vision Theory and Applications , Portugal, 5-8 February 2009

Haralick, M., K. Shanmugam, and I. Dinstein. (1973). Textural Features for Image Classification. IEEE Trans. on Sys. Man, and Cyber., 1973. 3(6): pp. 610-621.

Hochbaum & Shmoys. (1985). Farthest First Traversal Algorithm: A best possible heuristic for the k-center problem. Mathematics of Operations Research, 10(2):pp. 180-184.

Hout, v I. (1995). Indonesian weaving between heaven and earth: Religous implications of bird motifs on textiles, Royal Tropical Institute, ISBN 9068328360, Amsterdam.

Witten, I.H. & Frank, E. (2005). Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

Ismail, S.Z. (1997). Malay woven textiles: The beauty of a classic art form, Dewan Bahasa dan Pustaka, Malaysia.

Jamil, N., Z.A. Bakar, & T.M.T. Sembok (2006). Image Retrieval of Songket Motifs using Simple Shape Descriptors. Geometric Modeling and Imaging— New Trends (GMAI'06). 2006.

Jamil, N. and Z.A. Bakar (2006). Shape-Based Image Retrieval of Songket Motifs. The 19th Annual Conference of the NACCQ. 2006.

MacQueen, J.B. (1967). Some Methods for classification and Analysis of Multivariate Observations. The 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press.

Ojala, T., et al. (2002). Outex - new framework for empirical evaluation of texture analysis algorithms. The 16th International Conference on Pattern Recognition, 2002.

Korb, K.B. and A.E. Nicholson (2004). Bayesian Artificial Intelligence, London: Chapman & Hall/CRC Press.

Krylov, A.S., A. Kutovoi, and W.K. Leow (2003). Texture parameterization with Hermite functions. Computer Graphics and Geometry, 5(1): pp. 79-91.

Kotoulas, L. and I. Andreadis (2005). Image analysis using moments. The 5th Int. Conf. on Tech. and Automation.

Kohavi, R (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Fourteenth International Joint Conference on Artificial Intelligence, San Mateo, CA., 1995.

Kim, K. et al. (1999). Efficient video images retrieval by using local co-occurrence matrix texture features and normalised correlation. Proceedings of The IEEE Region 10 Conf., 2: pp. 934-937.

Kira, K., Rendell, L. (1992). A practical approach to feature selection. Proceedings of the Ninth International Conference on Machine Learning, 1992.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. Proc. ECML-94. 1994. Catania, Sicily.

See, K.W., et al. (2008). Image reconstruction using various discrete orthogonal polynomials in comparison with DCT. Applied Mathematics and Computation, Vol 193, no 2, pp. 346-359.

Ward, A. A., McKenna, S. J., Buruma, A., Taylor, P,, Han J. (2008). Merging technology and users: applying image browsing to the fashion industry for design inspiration. 6th International Workshop on Content-based Multimedia Indexing (CBMI), London, 18-20 June 2008

Walker, R.F., P.T. Jackway, and I.D. Longstaff. Recent developments in the use of the co-occurrence matrix for texture recognition. in DSP 97: 13th International Conf. on DSP. 1997.

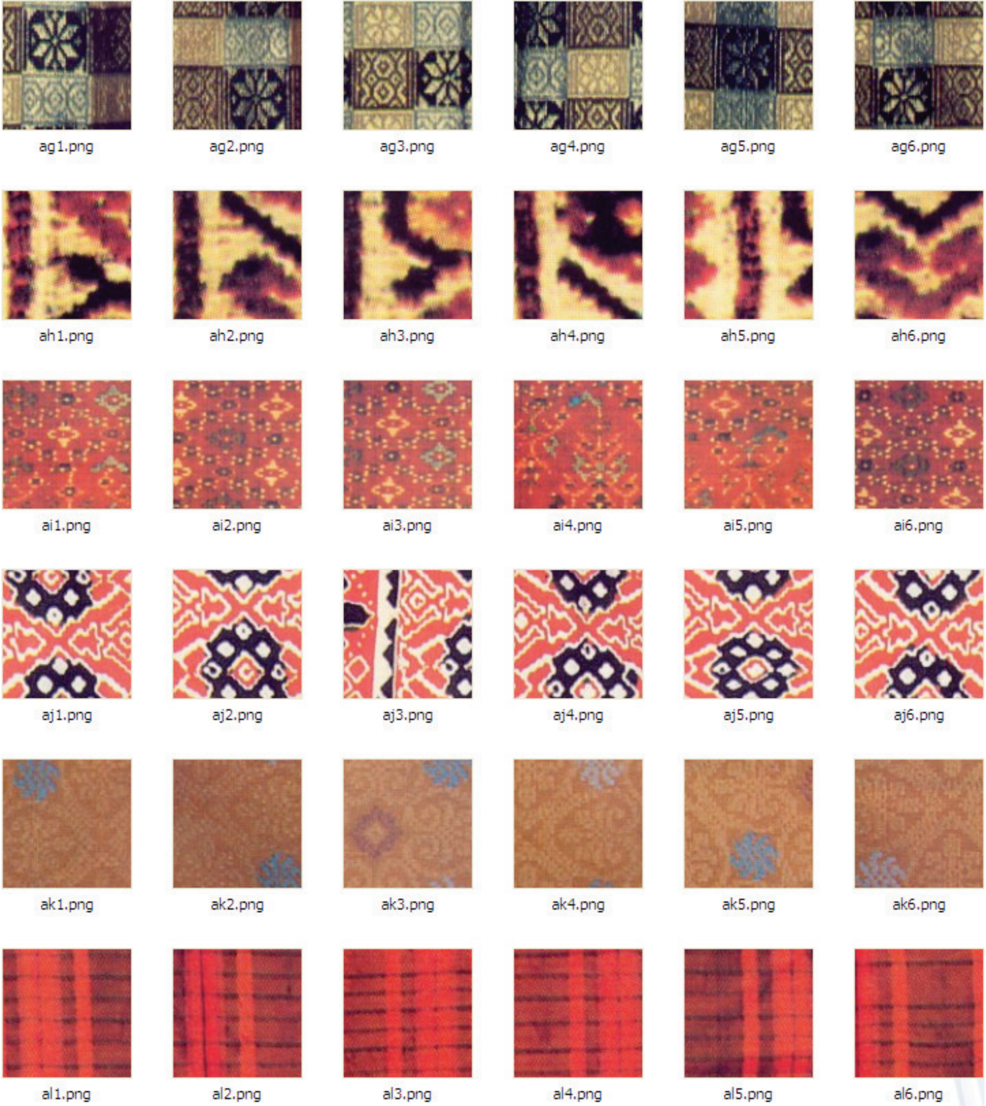
Wang, L. and G. Healey (1998). Using Zernike moments for the illumination and geometry invariant classification of multispectral texture. IEEE Trans. on Image Processing, 7(2): pp. 196-203.

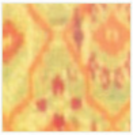
Weszka, J.S., C.R. Dyer, and A. Rosenfeld (1976), A comparative study of texture for terrain classification. IEEE Trans. on Sys., Man, and Cyber., Vol 6, pp. 269-265.

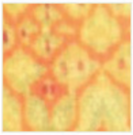
Wittgenstein, L. (1953). Philosophical Investigations. Blackwell Publishing.

Zaim, A., et al. (2006). A new method for iris recognition using gray-level cooccurence matrix. IEEE International Conf. on Electro/information Technology, 2006.

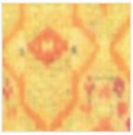
Appendix

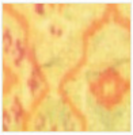

ag1.png
ag2.png
ag3.png
ag4.png
ag5.png
ag6.png
ah1.png
ah2.png
ah3.png
ah4.png
ah5.png
ah6.png
ai1.png
ai2.png
ai3.png
ai4.png
ai5.png
ai6.png
aj1.png
aj2.png
aj3.png
aj4.png
aj5.png
aj6.png
ak1.png
ak2.png
ak3.png
ak4.png
ak5.png
ak6.png
al1.png
al2.png
al3.png
al4.png
al5.png
al6.png


as1.png


as2.png

as3.png


as4.png


as5.png

as6.png

at1.png

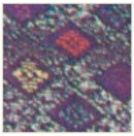
at2.png

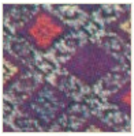
at3.png

at4.png

at5.png

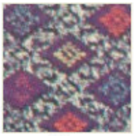
at6.png


au1.png


au2.png

au3.png

au4.png

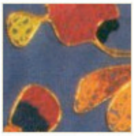

au5.png

au6.png

av1.png

av2.png

av3.png


av4.png

av5.png

av6.png

aw1.png

aw2.png

aw3.png

aw4.png

aw5.png

aw6.png

ax1.png

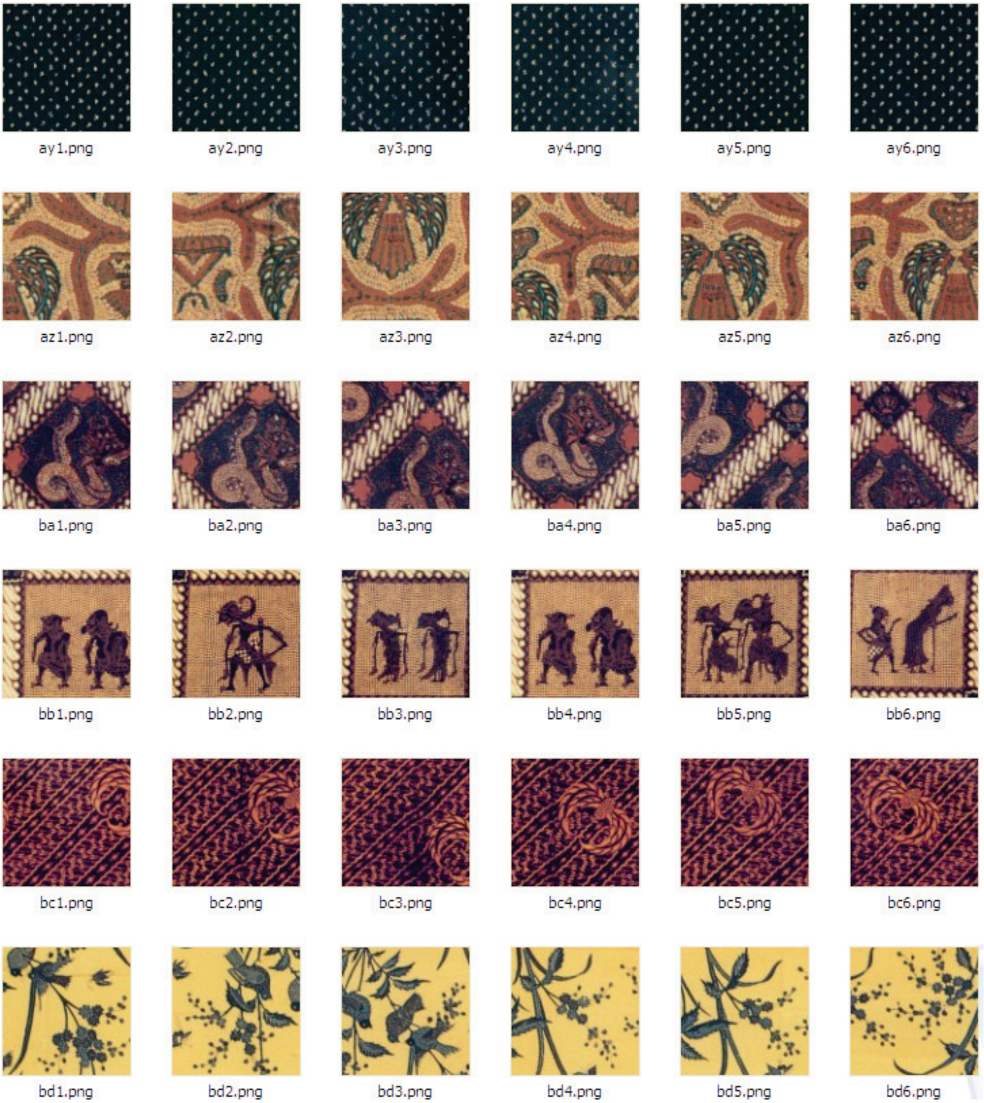
ax2.png

ax3.png

ax4.png

ax5.png

ax6.png


ay1.png
ay2.png
ay3.png
ay4.png
ay5.png
ay6.png
az1.png
az2.png
az3.png
az4.png
az5.png
az6.png
ba1.png
ba2.png
ba3.png
ba4.png
ba5.png
ba6.png
bb1.png
bb2.png
bb3.png
bb4.png
bb5.png
bb6.png
bc1.png
bc2.png
bc3.png
bc4.png
bc5.png
bc6.png
bd1.png
bd2.png
bd3.png
bd4.png
bd5.png
bd6.png

25

Approaches to Automatic Seabed Classification

Enrique Coiras, David Williams
NATO Undersea Research Centre
19126 La Spezia. Italy

1. Introduction

The latest advances in sonar technology permit to observe the underwater environment with improved resolution and coverage, with the latest side-scan and synthetic aperture sonar (SAS) systems able to produce images with a resolution of a few centimetres. This capability enables the seabed configuration to be determined relatively easily by visual inspection of the sonar images —rocks, sand ripples and underwater plants can be clearly identified visually, for instance. This process of seafloor characterization can be automated by analyzing the characteristics of the image texture at any given point. The resulting classification is useful for diverse applications, including environmental monitoring, security, and defence.

Seabed classification is the process by which one divides (or *segments*) an image of a typically large area of seabed into different regions based on their local characteristics. The characteristics (or *features*) used for classification and the specific classes chosen will depend on the application.

For mine countermeasures (MCM), for instance, an area may be segmented into flat, rippled and complex regions, which roughly divide the surveyed area into sub-regions of increasing mine-hunting difficulty. Another example could be the estimation of fishing density in an area over time, which can be determined by examining the number of trawl marks on the seabed.

Regardless of the particular application, the basic desired output of a seabed classification algorithm is a 'map' of the seabed indicating where each seabed type is present. Furthermore, as with more traditional remote sensing modalities, analyzing collected data in an automated manner is vital when dealing with vast quantities of data.

In this chapter, approaches to automatic classification of seabed into different types or classes are discussed. The chapter is organized as follows. Section 2 gives an overview of the different general approaches for seabed classification. Section 3 discusses more detailed aspects of these general approaches, such as feature extraction, classifiers, and fusion of multiple views. In Section 4, a few specific classification approaches, which span the variety of possible methods, are described. Example results on real sonar imagery are also shown for these particular approaches. Finally, conclusions are summarized in Section 5.

2. Overview

Many different approaches for addressing the task of automatic seabed classification exist. One useful categorization of these approaches is based on the knowledge that is possessed about the problem. From this point of view, all seabed classification approaches can be divided into three broad categories.

In the most ideal case, *a priori* knowledge about specific seabed types that one will encounter is available. This information in turn permits a model-based approach, in which a mathematical model that characterizes certain specific seabed types can be constructed. One can then use the degree to which a portion of seabed matches the defined model as a metric for performing classification. The main advantage of a model-based approach is that the algorithm can be tailored specially for a seabed type of particular interest.

Unfortunately, it is usually difficult or impossible to accurately model most types of seabed. Because of the characteristic complexity or natural variations within a given type of seabed, one must therefore typically employ an alternative seabed classification approach that assumes less specific knowledge of the problem. Another disadvantage of model-based approaches is their poor flexibility, being very specific and requiring new models for any new classes that may have to be added to the system.

Rather than knowing detailed information about certain seabed types to be encountered, one may instead know only about general classes of seabed types. Often, this general knowledge is provided in the form of labelled training data. That is, data for which the true identity of different seabed types from some location is possessed. When such information is available and is exploited, the resulting approach is referred to as supervised.

The advantage of employing a supervised approach is that the resulting classification of the seabed will be in terms of types or classes known *a priori*. Thus, the resulting segmentation of the seabed will be in terms of specific seabed types that are important for the application at hand. As briefly mentioned in the introduction, MCM operations typically seek to segment an area of seabed into one of three basic types -- flat, benign seabed; seabed characterized by sand ripples; and complex seabed that contains rocks or other clutter objects. By possessing labelled training data (or *ground-truth*) of these three classes, an algorithm that seeks to discriminate among these seabed types can be developed.

One drawback to the use of supervised approaches for seabed classification is that the process of acquiring labelled seabed data is expensive and time-consuming. In addition to acquiring the data at sea, a human would be required to tediously hand-label the data as specific seabed types. Moreover, this ground-truthing process is inherently subjective and could vary from one human operator to another.

Another potential source of concern regarding supervised methods is the implicit assumption that the underlying statistics that generated the training and test data are the same. In real applications, this assumption is often violated, leading to what is known as covariate shift (Sugiyama et al., 2007), sample selection bias (Zadrozny, 2004), or concept drift (Widmer & Kubat, 1996; Liao et al., 2005). For example, if one collects training data and learns a classifier from one site, but then attempts to classify test data collected at a different location, a fundamental mismatch in class statistics can lead to poor classification performance.

This scenario motivates the use of the third general approach to seabed classification, in which no labelled training data is required. This unsupervised approach is appropriate when no labelled training data exists or when little *a priori* information about the seabed types to be encountered is available.

Since in principle any information known in advance about the seabed classification task at hand should be exploited, the relevance of unsupervised classification algorithms arises when no such knowledge is possessed. This lack of knowledge about the problem is reflected by the numerous drawbacks that plague unsupervised approaches. For one, the resulting groups into which the seabed is segmented may not be valuable divisions for the application. Moreover, the number of seabed types must often be specified, though it is not always known *a priori*. Lastly, with an unsupervised approach, human intervention is required to associate the groups into which the data is segmented with distinct seabed types. Despite these drawbacks, when no information or training data is available, there is little alternative to unsupervised methods.

3. Features, Classifiers, Outputs, and Fusion

Model-based seabed classification approaches are specially tailored for specific seabed types and certain scenarios. More commonly, seabed classification approaches fall under the purview of supervised or unsupervised methods. In both supervised and unsupervised approaches, features must be extracted from the seabed and a subsequent classifier must be built. The number of possible combinations of choices for these two requirements is endless. Thus, rather than attempting to provide an exhaustive list of such approaches, a representative sample of commonly used techniques will be presented in this chapter.

The purpose of the feature extraction stage is to represent an image of a given area of seabed in a succinct manner. The features that are extracted should be such that they are capable of discriminating among the different seabed types of interest. That is, the feature values of one particular seabed type should be differentiable from those of other seabed types.

A few examples of types of features useful for seabed classification include moment-based features, wavelet-based features, and features derived from eigendecompositions. Moment-based features calculate certain properties of the distributions of the gray-level pixel intensities for a given area of seabed. These features are motivated by the fundamental sonar scattering physics of the seabed. For example, the amount of acoustic energy scattered back to the sonar receiver from areas of seabed characterized by sand ripples or rocks is larger than the amount scattered from flat, benign seabed.

Features based on wavelet decomposition are popular because they can be used to characterize textural properties of seabed images. Namely, the wavelet coefficient energy will be large when the orientation and scale match the orientation and scale of high-energy texture components in an image (Mallat, 1999).

Feature sets based on spectral clustering (Meila & Shi, 2000; Ng et al., 2001), which exploits the eigenvectors of a matrix composed of distances between data points, have also been used with success. Spectral clustering will effectively transform a feature vector into a new feature vector in a lower dimensional space by retaining only the eigenvectors corresponding to the largest eigenvalues.

After feature extraction, classification must be performed on the resulting set of feature vectors. The objective of the classification stage is to develop a rule that will successfully discriminate among the various seabed types by discriminating their corresponding feature vectors in feature space. When an unsupervised approach is employed, no training data is available to perform the classification and the classification stage typically only clusters the data. Common unsupervised approaches include *k*-means clustering and methods based on modelling the data as mixtures of Gaussians.

When a supervised approach is employed, labelled training data is used to build a classifier to which unlabeled test data is subsequently submitted. Many different discriminative classification approaches exist, such as decision trees (Breiman, 1993), support vector machines (Shawe-Taylor & Cristianini, 2000), relevance vector machines (Tipping, 2001), and Gaussian processes (Rasmussen & Williams, 2006).

Depending on the classification approach employed, the result of the algorithm can be in different forms. For example, each area of seabed may be assigned a probability or score of belonging to each seabed type. Alternatively, each area of seabed may be classified as a certain seabed type directly via a hard decision.

When multiple views of an area of seabed are available, different data fusion approaches can be used to combine them. There exist two general approaches to perform such fusion. In one approach, each view of the seabed is considered independently and then the multiple decisions or seabed type scores are combined in some particular manner. In the second approach, all views of the seabed are considered jointly so that a single decision regarding seabed type is produced. An example of the former fusion approach is Dempster-Shafer theory (Shafer, 1976), while an example of the latter fusion approach would be the result of a fully Bayesian framework (Berger, 1993).

4. Example Classification Approaches and Results

To provide more specific examples of seabed classification methods, three particular algorithms are described in greater detail in this section. Two of them (model-based and unsupervised) have been applied to SAS images acquired by the MUSCLE autonomous underwater vehicle (AUV), while the other method (supervised) is demonstrated on side-scan data collected by a Remus AUV.

MUSCLE images were obtained by the NATO Undersea Research Centre (NURC) during the Colossus II sea trial in the Baltic Sea off the coast of Latvia. In that trial, high-resolution sonar data was collected by the MUSCLE AUV, which is equipped with a 300 kHz sonar capable of constant image resolution of approximately 3 cm x 3 cm at ranges up to 200 m.

The Remus images were collected during a NURC exercise in Capo Teulada, Sardinia, where an area of five square kilometres was observed by multiple sensors. The Remus vehicle is equipped with a 900 kHz Marine Sonic sonar capable of forming images of about 8.6 cm x 12 cm in resolution at ranges up to 30 m.

4.1 Model-based Classification Example

Model-based classification algorithms are useful when the classification target can be described mathematically. The case of sand ripples exemplifies this situation, given the similarity between sand ripples and a striped pattern or sinusoidal function. By applying image correlation (Lewis, 1995) the response of the image to a bank of predefined filters based on those patterns can be used to quantify the "ripplicity" of every image pixel.

Since a single sinusoidal crest will correlate also with lines or any proud object on the seafloor (showing a highlight followed by a shadow, just like the ripples), the filters are designed to contain three sinusoidal periods—the idea being that only rippled areas will respond to them fully. The filters cover a range of orientations and scales so that rippled areas of various sizes and directions can be detected. Figure 1 shows the configuration of the filter bank using six orientations. The angle and size of the individual filter that produces the highest response will give the approximate orientation and size of the ripples in that area. The estimated angles and sizes, however, will only be meaningful when ripples are actually present in the region.

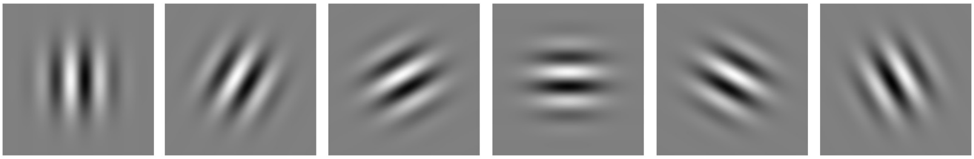
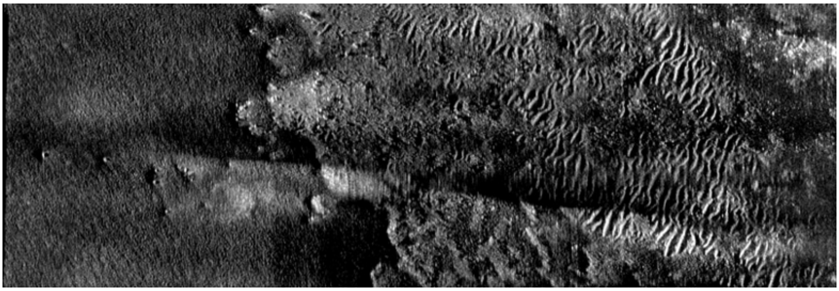


Fig. 1. Filter bank containing six orientations that is used by the model-based ripple classification algorithm.

In order to determine whether a given response value is to be considered high enough to indicate the presence of ripples, a calibration step is used to derive lower and upper thresholds for the responses to the filter bank. In this way the values returned when convolving with the filters can be scaled from 0 to 1, providing a continuous measure that ranges from “no ripple” to “perfect ripple”. The thresholds are obtained by filtering two predetermined images: one containing a synthetic image of alternate black and white bands (which will determine the high threshold for the filters) and an image containing random pixels (which will determine the lower threshold). Both images are tuned to the scale of the filters—the bands being exactly half a period of the sinusoidal component, and the random image being median-filtered to produce blobs of the relevant scale. To ensure the lower threshold is safely set, it is actually obtained as the median response from twenty of those random images.

The specificity of this filter-based method can be further increased by targeting the differences in response to orthogonal directions. The underlying idea being that a rippled area will produce a high response to a ripple filter of the adequate orientation and a low response to the filter tuned to the orthogonal direction, whereas an area showing no ripples will produce responses of similar strength to any filter orientation. This is the approach used to produce the results shown in Figure 2, where ripples have been detected in a MUSCLE SAS image using six orientations and three filter sizes of 0.5, 0.6 and 0.7 meters.



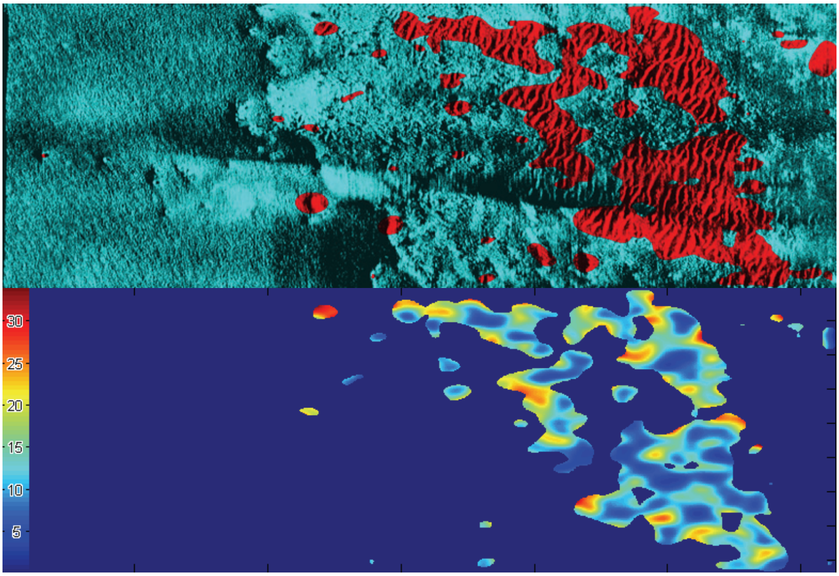


Fig. 2. Results of model-based sand ripple characterization. (a) A MUSCLE SAS seabed image. (b) Segmentation into rippled (red) and non-rippled (cyan) areas. (c) Local ripple orientation in degrees.

4.2 Supervised Classification Example

When model-based classification is not feasible or practical, a more general approach is to use a supervised system. In a supervised approach, an algorithm is trained to recognize particular characteristics that permit the assignment of objects to different classes.

A supervised system is more flexible in the sense that there is no need to devise a new model every time a previously unseen class has to be considered. For seabed classification this is especially important when using data from different sensors (Coiras, 2007). The multi-sensor seabed classification system presented here uses an independent supervised binary classifier (detector) for every class considered. The detectors are trained using a given set of ground-truth samples in sensor space (that is, not yet geo-referenced), and their classification performances are estimated in order to determine their individual confusion matrices.

The training samples are manually ground-truthed with an image editing application (in this case, Photoshop). A small set of images from the mission that are representative of the classes to consider are selected, and a binary map is created for each of them. In the training example shown in Figure 3, three binary segmentations have been manually produced for flat, posidonia and rock classes.

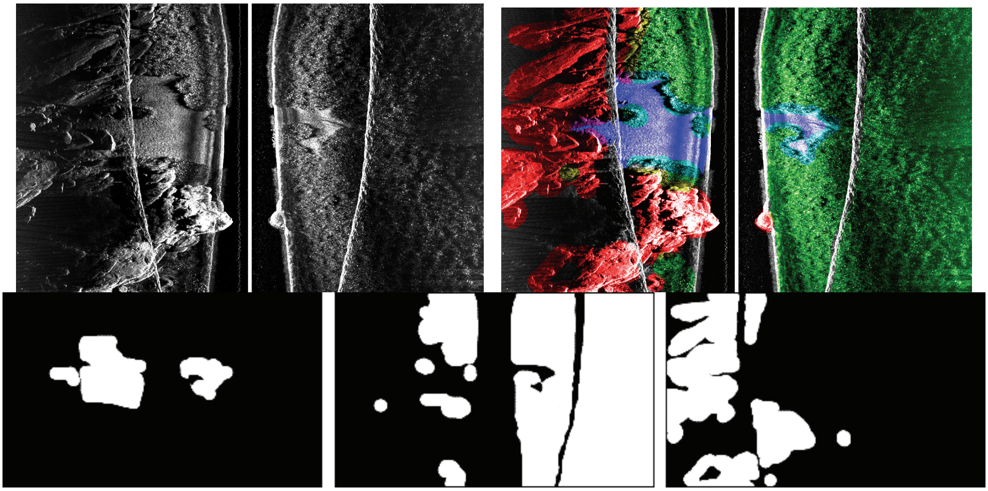


Fig. 3. Manual segmentation of a side-scan image to train a supervised classifier. Counter-clockwise from top-left: (a) side-scan image; (b) segments for flat, posidonia and rock areas; (c) colored side-scan image using the flat, posidonia, and rock segment images as blue, green, and red channels.

The subsequent classification is based on texture analysis (Chang & Kuo, 1993) and uses wavelet decomposition to generate the feature vectors. Four bi-orthogonal wavelets (Mallat, 1999) in two scales generate feature vectors of 16 components that are then classified using a decision tree. This results in three maps for each class, as shown in Figure 4, which correspond to the probability of each image pixel belonging to each of the classes considered.

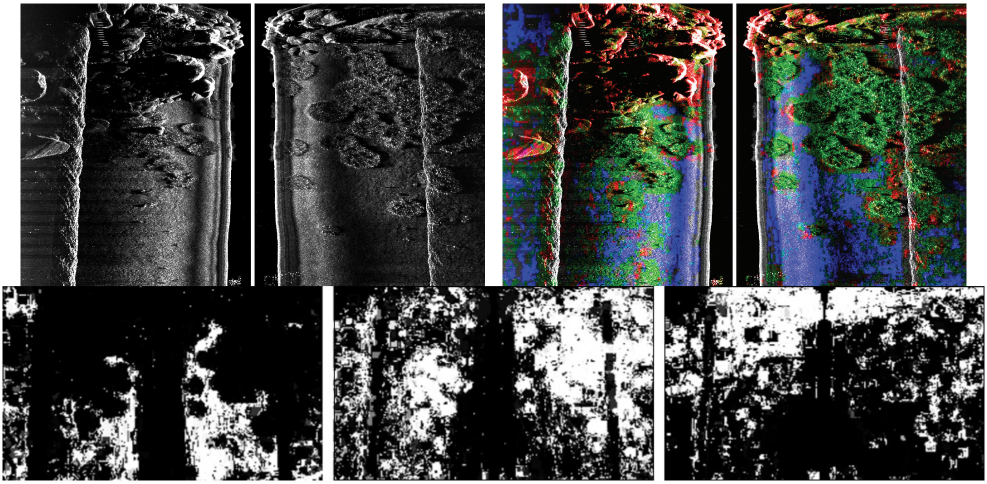


Fig. 4. Supervised segmentation of a side-scan image. Counter-clockwise from top-left: (a) side-scan image; (b) probability images for flat, posidonia and rock classes; (c) colored side-scan image using the flat, posidonia, and rock segment images as blue, green, and red channels.

A decision regarding the actual class of a pixel can be taken at this point by selecting, for instance, the class that has the highest probability. In our case, the decision is delayed until after geo-referencing. This choice is made because in our seabed observation missions, each seafloor point is observed more than once, and ideally all gathered evidence should be used to make a more informed classification decision.

At this stage, the performance of the sensor for each class can be taken into account and used to modulate the probabilities given by the decision tree, as described in (Coiras, 2007) for the multi-sensor case.

After all images have been processed in sensor space, they are geo-referenced into an area mosaic. The different observations available for every seabed point are combined using data fusion. In our case, the Dempster-Shafer theory of evidence (Shafer, 1976) is used because of its ability to cope with conflicting information, which is particularly important for the multi-sensor case. The result of the data fusion stage is a set of three mosaics of the area composed of the belief that each pixel corresponds to the flat, posidonia, and rock classes.

The final classification map for the observed area is determined by the maximum-belief decision rule, which selects a single class for each of the image pixels. Figure 5 shows the final classification result for the Capo Teulada survey, in south-eastern Sardinia.

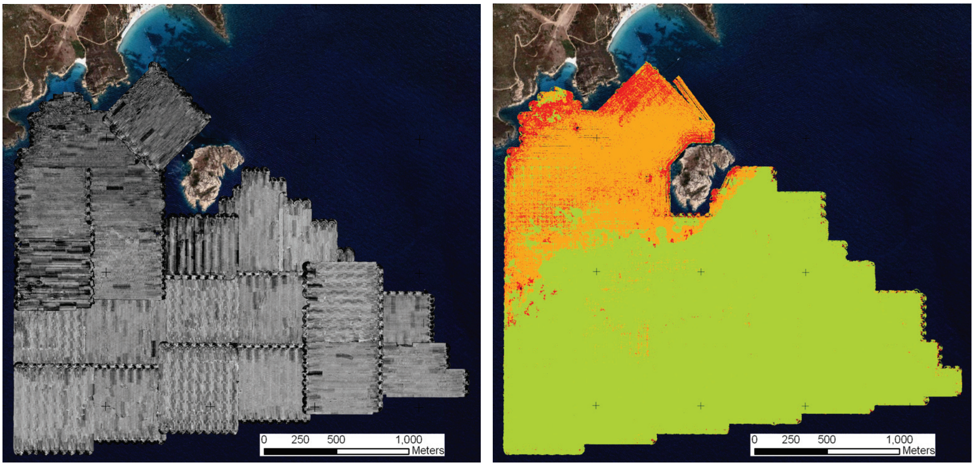


Fig. 5. Results of supervised segmentation for a full side-scan mission covering an area of five square kilometres. (a) Overview of the covered area showing all side-scan tracks. (b) Result of the supervised classification of the seabed into the following three classes: flat seabed (green), posidonia formations (orange) and rocky seabed (red).

4.3 Unsupervised Classification Example

Next we consider an unsupervised seabed classification algorithm (or more appropriately, seabed *segmentation* algorithm, since the seabed classes are not known). In this particular approach, the "atomic" unit for seabed classification is assumed to be a 2 m x 2 m area of seabed. That is, each 2 m x 2 m area of seabed corresponds to one data point. This particular size was chosen as a compromise among several factors. The larger the area chosen, the more likely that a single data point will have the unfavourable property of containing multiple types of seabed. However, if the area is too small, the distinguishing characteristics of the seabed that indicate a certain seabed type may be lost.

We consider four different unsupervised segmentation approaches, which differ both in the set of features employed and in the clustering method used. Specifically, segmentation is performed when using (i) moment features with *k*-means clustering, (ii) moment features with spectral clustering, (iii) wavelet features with *k*-means clustering, and (iv) wavelet features with spectral clustering.

The wavelet-based features consist of 16 features that are derived from the coefficients of a bi-orthogonal wavelet decomposition (Mallat, 1999) of each SAS image block (i.e., data point). The moment-based are the mean, variance, skewness, and kurtosis of the distribution of pixel values of an image block (i.e., data point). Finally, spectral clustering can be performed on either set of features to effectively transform a feature vector into a new feature vector, in a lower dimensional space. These 16 wavelet features and 4 moment features are extracted for each data point (2 m x 2 m area of seabed).

We perform unsupervised seabed segmentation on a MUSCLE SAS image that spans an area of 56 m x 56 m of seabed. To allow an assessment of the segmentation results, we manually ground truth this SAS image, shown in Figure 6, into three seabed types (namely, flat, rippled, and rocky seabed). The result of this ground-truthing is also shown in Figure 6.

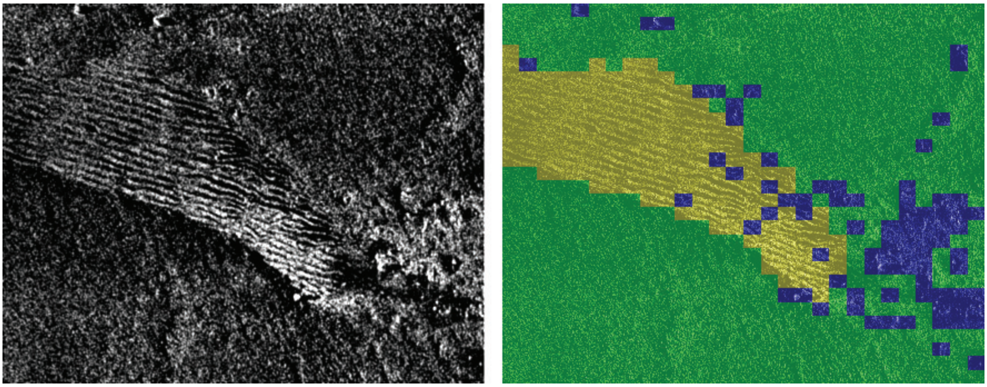


Fig. 6. (a) A SAS image, and (b) its corresponding ground truth (where green, yellow, and blue correspond to flat, rippled, and rocky seabed, respectively).

In the four feature and clustering combinations considered, the segmentation process is completely unsupervised, since it is assumed that no training data is available. The number of clusters to be learned is fixed at k=4. Because the k-means algorithm is not guaranteed to result in the globally optimal clustering, 100 random cluster-centroid initializations are considered for each case. The clustering for which the distortion — defined as the sum of distances from each point to its assigned cluster centroid — is a minimum is selected as the final clustering (and by extension, the final segmentation).

The results of the unsupervised seabed segmentation on the SAS image shown in Figure 6 are shown in Figure 7. Because the methods are unsupervised, no explicit correspondence between clusters and seabed types exists. However, for purposes of evaluating the segmentation results here, one can easily assign a correspondence between ground-truth seabed types and clusters.

5. Conclusion

In this chapter, an overview of different automatic seabed classification approaches has been provided. Although in some cases a completely automated model-based method is possible (when the phenomenon to identify can be modelled mathematically), in most situations some degree of human intervention is required. In the supervised case, initial training is required to drive the system's focus to the classes of interest, whereas in the unsupervised case the operator should ratify the soundness of the class division suggested by the system. It can even be argued that the model-based case also requires human intervention, since the model itself must be created in the first place.

In any case, it has been shown that the three general classification approaches are extremely useful for the automatic processing of data collected in AUV sonar missions, which frequently consist of hundreds or thousands of images that would otherwise require manual processing. This automation allows operators to focus on higher-level tasks, such as mission planning or model design, which could make seabed surveillance operations simpler and more effective.

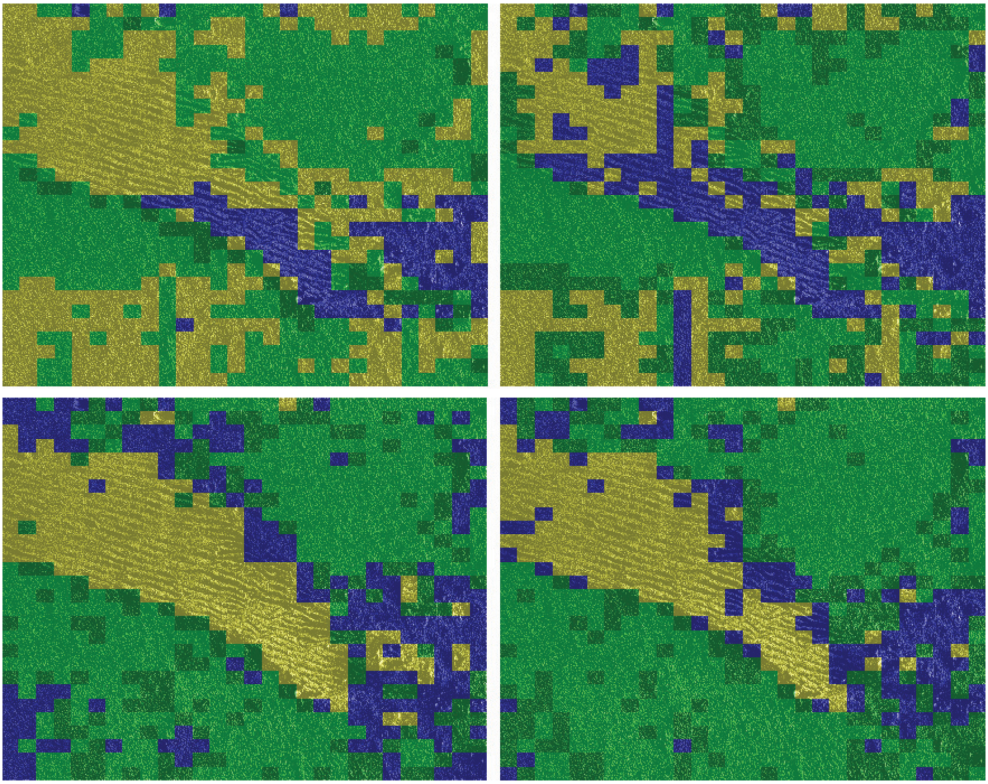


Fig. 7. Results of unsupervised seabed segmentation algorithms. Clockwise from upper left: results from using (a) moment features with *k*-means, (b) moment features with spectral clustering, (c) wavelet features with spectral clustering, and (d) wavelet features with *k*-means.

6. References

Berger, J. (1993). Statistical Decision Theory and Bayesian Analysis, Springer.

Breiman, L. (1993). Classification and Regression Trees, Chapman & Hall.

Chang, T., Kuo, C.-C.J. (1993). Texture Analysis and Classification with Tree-structured Wavelet Transform, IEEE Transactions on Image Processing, Vol. 2, No. 4, pp. 429-441.

Coiras, E., Myers, V. & Evans, B. (2007). Reliable Seabed Characterization for MCM Operations, Proceedings of the IEEE/MTS Oceans'07 Conference.

Lewis, J.P. (1995). Fast Template Matching, Vision Interface, p. 120-123.

Liao, X., Xue, Y., and Carin, L. (2005). Logistic Regression with an Auxiliary Data Source, Proceedings of the 22nd International Conference on Machine Learning, pp. 505-512.

Mallat, S. (1999). A Wavelet Tour of Signal Processing, Academic Press.

Meila, M. & Shi, J. (2000). Learning Segmentation by Random Walks, Advances in Neural Information Processing Systems, pp. 873-879, MIT Press.

Ng, A., Jordan, M., and Weiss, Y. (2001). On Spectral Clustering: Analysis and an Algorithm, Advances in Neural Information Processing Systems, pp. 849-856, MIT Press.

Rasmussen, C. & Williams, C. (2006). Gaussian Processes for Machine Learning, MIT Press.

Shafer, G. (1976). A Mathematical Theory of Evidence, Princeton University Press.

Shawe-Taylor, J. & Cristianini, N. (2000). An Introduction to Support Vector Machines: and Other Kernel-based Learning Methods, Cambridge University Press.

Sugiyama, M., Krauledat, M., and Muller, K. (2007). Covariate Shift Adaptation by Importance Weighted Cross Validation. Journal of Machine Learning Research, Vol. 8, pp. 985-1005.

Tipping, M. (2001). Sparse Bayesian Learning and the Relevance Vector Machine. Journal of Machine Learning Research, Vol. 1, pp. 211–244.

Widmer, G. & Kubat, M. (1996). Learning in the Presence of Concept Drift and Hidden Contexts. Machine Learning, Vol. 23, pp. 69-101.

Zadrozny, B. (2004). Learning and Evaluating Classifiers Under Sample Selection Bias, Proceedings of the 21st International Conference on Machine Learning.

26

Pattern recognition methods for improvement of differential protection in power transformers

Abouzar Rahmati
University Of Ilam
Iran

1. Introduction

Differential protection was already applied towards the end of the 19th century, and was one of the first protection systems ever used.

Faults are detected by comparison of the currents flowing into and out of the protected plant item. As a result of the fast tripping with absolute selectivity it is suited as main protection of all important items of plant, i.e. generators, transformers, busbars as well as cables and overhead lines and feeders at all voltage levels.

The power transformer protection is of critical importance in power systems. Since minimization of frequency and duration of unwanted outages is very desirable there is a high demand imposed on power transformer protective relays. This includes the requirements of dependability associated with no mal-operations, security associated with no false tripping, and operating speed associated with short fault clearing time.

One of the main concerns in differential protection of this particular component of power systems lies in the accurate and rapid discrimination of magnetizing inrush current from different internal faults currents. This is because the magnetizing inrush current, which occurs during the energizing the transformer, generally results in several times full load current and therefore can cause mal-operation of the relays. Such mal-operation of differential relays can affect both the reliability and stability of the whole power system.

The principle of differential protection is initially described in this chapter. Subsequently different protection schemes are covered in the next sections. At last an algorithm based on pattern recognition of current signals using wavelet transform which is a power signal processing tool is proposed.

2. Mode of operation of differential protection

The differential protection is 100% selective and therefore only responds to faults within its protected zone. The boundary of the protected one is uniquely defined by the location of the current transformers. Due to simple current comparison, the principle of differential protection is very straight forward.

Generators, motors and transformers are often protected by differential protection, as the high sensitivity and fast operation is ideally suited to minimize damage. On feeders the differential protection is mainly used to protect cables, particularly on short distances where distance protection cannot be readily applied.

The prime objective of busbar differential protection is fast, zone selective clearance of busbar faults to prevent large system outages and to ensure system stability. Mal-operation must be avoided at all cost as these could result in extensive supply interruption.

3. Principles of differential protection

The basic principles which have been known for decades are still applicable and independent of the specific device technology.

The differential protection compares the measured values of signals with regard to magnitude and phase. This is possible by direct comparison of instantaneous values or by vector (phasor) comparison. In each case the measurement is based on Kirchhoff's laws which state that the geometric (vector) sum of the currents entering or leaving a node must add up to 0 at any point in time. The convention used in this context states that the currents flowing into the protected zone are positive, while the currents leaving the protected zone are negative. The current differential protection is the simplest and most frequently applied form of differential protection. The measuring principle is shown in Fig. 1. X is the winding of the protected machine. The relay compares an operating current with a restraining current. The operating current (also called differential current), I_O, and the restraining current, I_R, are obtained as below:

$$I_O = |I_1 + I_2| \qquad (1)$$
$$I_R = K|I_1 - I_2|$$

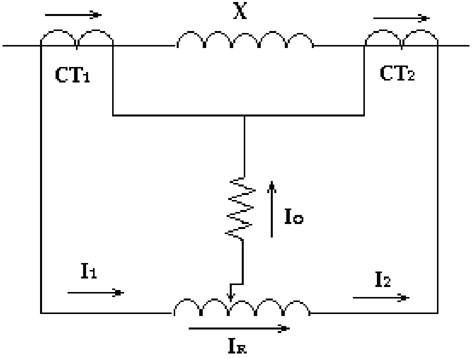


Fig. 1. Differential relay connection diagram

When there is no internal fault, the current entering in X is equal in phase and magnitude to current leaving X. The CT's are of such a ratio that during the normal conditions or for external faults (Trough faults) the secondary currents of CT's are equal. The relay generates a tripping signal if the operating current, I_O, is greater than a percentage of the restraining current, I_R,, according to:

$$I_O > KI_R + I_P \tag{2}$$

where K is the relay operating characteristic, that consists of a straight line having a slope equal to K. Intersection of this characteristic with vertical axis (I_O) define the relay minimum pickup current, I_P. The relay percentage restraint characteristic typically has an excellent behavior, but it has problems discriminating fault currents from false differential currents caused by magnetizing inrush and transformer over excitation.

4. Main difficult in differential protection

Differential protection is established as the main protection for transformer due to its simple principle of operation and sensitivity. However, a key problem of differential protection is accurate and rapid discrimination of magnetizing inrush current from an internal fault current.

Initial magnetizing due to switching a transformer in is considered the most severe case of an inrush. When a transformer is de-energized (switched-off), the magnetizing voltage is taken away, the magnetizing current goes to zero while the flux follows the hysteresis loop of the core. This results in certain remanent flux left in the core. When, afterwards, the transformer is re-energized by an alternating sinusoidal voltage, the flux becomes also sinusoidal but biased by the remanence. The residual flux may be as high as 80-90% of the rated flux, and therefore, it may shift the flux-current trajectories far above the knee-point of the characteristic resulting in both large peak values and heavy distortions of the magnetizing current.

Figure 2 shows a typical inrush current. The waveform displays a large and long lasting dc component, is rich in harmonics, assumes large peak values at the beginning (up to 30 times the rated value), decays substantially after a few tenths of a second, but its full decay occurs only after several seconds (to the normal excitation level of 1-2% of the rated current).

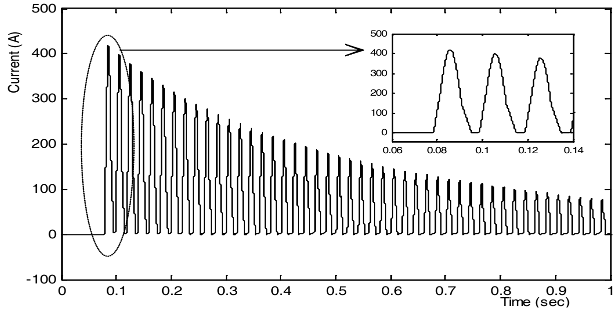


Fig. 2. Typical inrush current

It is evident that relaying protection should be initiated in response to internal fault but not to inrush current. To avoid the needless trip by magnetizing inrush current, many different restrain methods are proposed in recent years.

Since the magnetizing branch representing the core appears as a shunt element in the transformer equivalent circuit, the magnetizing current upsets the balance between the currents at the transformer terminals, and is therefore experienced by the differential relay as a "false" differential current. The relay, however, must remain stable during inrush conditions. In addition, from the standpoint of the transformer life-time, tripping-out during inrush conditions is a very undesirable situation (breaking a current of a pure inductive nature generates high overvoltage that may jeopardize the insulation of a transformer and be an indirect cause of an internal fault).

5. The proposed schemes for differential protection

5.1 Recognition of type of signals using Fourier Transform

Magnetizing inrush current generally contains a large second harmonic component in comparison to an internal fault. As a result conventional transformer protection systems are designed to block during inrush transients by this large second harmonic. The ratio of the second harmonic of differential current in excess of a preset threshold is interpreted as a present of magnetizing inrush.

Let us calculate the harmonic component of a typical inrush current waveform. We will assume a simplified waveform for the inrush current. Let the magnetizing characteristic be a vertical line in the $\emptyset - i$ plane, and be a straight line with a finite slope in the saturated region. This makes the current waveform of Figure 3 acquire the shape shown in Figure 4.

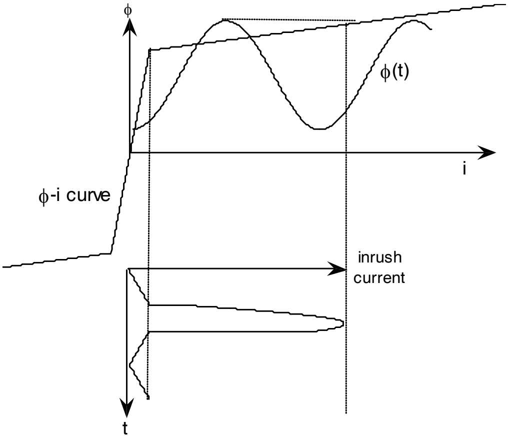


Fig. 3. Magnetizing current during energizing of a transformer

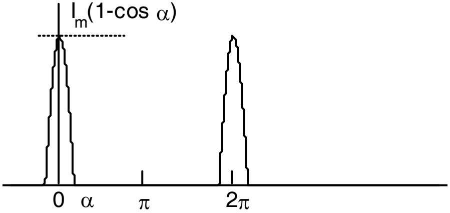


Fig. 4. Idealized inrush current waveform

The flux in the core is above the saturation knee point for a total angular span of 2 ∝ radians, and the corresponding current is a portion of a sine wave. For the remainder of the period, the current is zero. Although this is an approximation, it is quite close to an actual magnetizing current waveform. We may use Fourier series analysis to calculate the harmonics of this current. Consider the origin to be at the center of a current pulse, as shown in Figure 4. Then, the approximation for the current waveform is:

$$i(\theta) = \begin{cases} I_m(\cos\theta - \cos\alpha) & 0 \leq \theta \leq \alpha \ \ , \ \ (2\pi - \alpha) \leq \theta \leq 2\pi \\ 0 & \alpha \leq \theta \leq (2\pi - \alpha) \end{cases} \tag{3}$$

Since this choice of the origin gives a symmetric waveform about $\theta = 0$, we may use the cosine Fourier series for the current. The nth harmonic is given by:

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_0^{2\pi} i(\theta)\cos(n\theta)\,d\theta = \frac{2}{\pi}\int_0^{\alpha} I_m\,(\cos\theta\cos n\theta \ - \ cos\alpha\cos n\theta)\,d\theta \\ &= \frac{I_m}{\pi}\left[\frac{1}{n+1}\sin(n+1)\alpha + \frac{1}{n-1}\sin(n-1)\,\alpha - \frac{2}{n}\cos\alpha\sin n\alpha\right] \end{aligned} \tag{4}$$

The peak of the current wave is $I_m(1 - \cos\alpha)$, and the fundamental frequency component a_1 is given by:

$$a_1 = \frac{I_m}{\pi}\left[\alpha - \frac{1}{2}\sin 2\alpha\right] \tag{5}$$

The relative magnitude of various harmonic components with respect to the fundamental frequency component, as calculated from equation (4) and (5), is tabulated in Table 1 up to the 13th harmonic, and for saturation angles of $60^\circ, 90^\circ$ and 120°. It should be noted that when the saturation angle is 90° there are no odd harmonics present. As the angle of saturation increases, the harmonic content decreases: indeed, if α becomes π there will be no harmonics at all. However, in most cases, α is much less than π, and a significant amount of harmonics are present in the magnetizing inrush current. Of all the harmonic components, the second is by far the strongest.

Harmonic	$^{a_n}/_{a_1}$		
	$\alpha = 60^\circ$	$\alpha = 90^\circ$	$\alpha = 120^\circ$
2	0.705	0.424	0.171
3	0.352	0.000	0.086
4	0.070	0.085	0.017
5	0.070	0.000	0.017
6	0.080	0.036	0.019
7	0.025	0.000	0.006
8	0.025	0.029	0.006
9	0.035	0.000	0.008
10	0.013	0.013	0.003
11	0.013	0.000	0.003
12	0.020	0.009	0.005
13	0.008	0.000	0.002

Table 1. Harmonics of the magnetizing inrush current

Magnetizing inrush current generally contains a large second harmonic component in comparison to an internal fault. As a result conventional transformer protection systems are designed to block during inrush transients by this large second harmonic. The ratio of the second harmonic of differential current in excess of a preset threshold is interpreted as a present of magnetizing inrush. However, the second harmonic due to CT saturation component may also be generated during internal faults. Moreover, it was found that in certain cases, the second harmonic generated during internal faults in transformers is relatively large, which impairs the ability of this kind of the criterion. Consequently, the commonly used conventional differential protection technique based on the second harmonic restraint will thus have difficulty in distinguishing between internal fault currents and inrush currents.

5.2 Wave shape recognition of signals in time domain

Wave shape recognition techniques represent another alternative for discriminating internal faults from inrush current signals. In this kind inrush restraining methods pays attention to the periods of low and peaks values of the inrush current signal in the time domain.

It has been observed from Fig. 2 that the inrush wave is distinguished from a fault wave (which is sinusoidal wave shape) by a period in each cycle during which very low magnetizing currents (i.e. the normal exciting currents) flow, when the core is not in saturation. This property of the inrush current can be used to distinguish this condition from an internal fault. The condition to declare inrush would be that during a power system frequency cycle, there should always be an interval of time when an instantaneous differential current is equal to the normal magnetizing current, which is close to zero (below 0.5 %). This interval must be at least about 1/4 of the period, that is, about 5 ms in 50 Hz power systems.

Generally, there are basically two inrush restraining methods of wave shape recognition:

The first, and more common approach, pays attention to the periods of low and flat values in the inrush current (“dwell-time” — criterion 1), and the second algorithm pays attention to the sign of the peak values and the decaying rate of the inrush current (criterion 2).

A. Criterion 1

The hypothesis of magnetizing inrush may be ruled out if the differential current does not show in its every cycle a period lasting no less than 1/4 of a cycle in which the shape of the waveform is both flat and close to zero (see Figure 5).

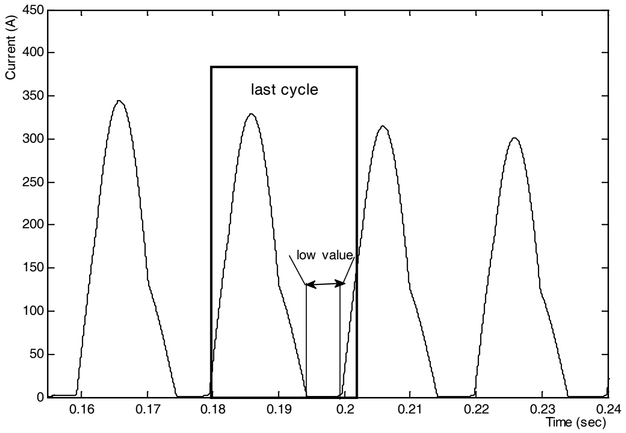


Fig. 5. Illustration of the direct waveform recognition of inrush (criterion 1)

This form of direct waveform restraining regardless of its implementation shows weaknesses:

(a) the recognition of an internal fault versus magnetizing inrush takes one full cycle

(b) the CTs, when saturated during inrush conditions (very likely due to the dc component in the current), change the shape of the waveform within the dwell periods (Figure 6) and may cause a false tripping

(c) during severe internal faults, when the CTs saturate, their secondary currents may also show periods of low and flat values exposing the relay to missing operations

B. Criterion 2

The hypothesis of magnetizing inrush may be ruled out if the differential current has its peaks displaced by half a cycle, and any two consecutive peaks are not of the same polarity (see Figure 6). This method needs robust detection of the peak values. Timing between two consecutive peaks must be checked with some tolerance margin accounting for the frequency deviations.

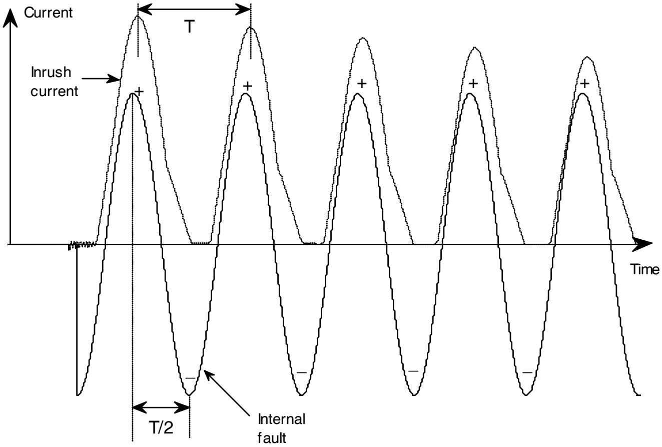


Fig. 6. Illustration of internal fault and magnetizing inrush currents (criterion 2)

Theoretically, this method needs three quarters of a cycle to distinguish between internal faults and inrush conditions. The first peak of the fault current appears after a quarter of a cycle, the next one - half a cycle later. With the second peak arriving, the criterion rejects the inrush hypothesis and sets the tripping permit.

The main disadvantage of this algorithm is the need of cross polarization between the phases. Not always all three phases show the typical inrush uni-polar waveform. Also, during very smooth energization of a protected transformer (what may accidentally happen owing to the adequate relation between the switching angle and the remanent flux), this criterion will fail.

This criterion may be also used in its indirect form as a modifier for the instantaneous differential overcurrent element. Defining the overcurrent principle as:

$$TRIP = |i_D| > \delta \tag{6}$$

and specifying one threshold, one needs to adjust this threshold very high to prevent false trippings (above the highest inrush current). One may, however, re-define the operating principle (Figure 7):

$$TRIP = (i_D > \delta_+) \;\&\; (i_D < \delta_-) \tag{7}$$

and use two thresholds to detect the uni-polarity/bi-polarity of the signal (Figure 7). When using the modified overcurrent principle, the setting may be adjusted as low as one third of the traditional threshold. This allows much more internal faults to be quickly detected by the unrestrained overcurrent algorithm.

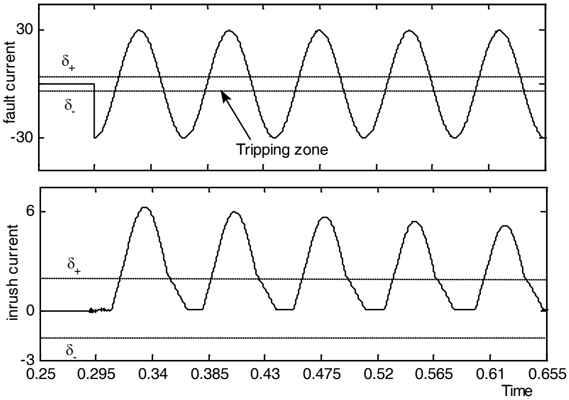


Fig. 7. Illustration of the double-threshold overcurrent principle

5.3 Mathematical morphology for recognition of signals

Mathematical morphology (MM) is a relatively new tool for image and signal processing. It is based on theoretic set concept, extracting object features by choosing a suitable structuring shape as a probe. Morphological operations, based on set transformation, are used to convert an image or signal into a quantitative description of its geometrical structure. The applications of MM are mainly focused on image processing, nonlinear filtering, machine vision, and pattern recognition.

Some schemes are proposed to identify inrush current signals from other conditions by using a morphological decomposition scheme (MDS) based on the morphological wavelet. In these approaches differential currents decompose into a series of components by designed morphological operators; then the extracted features of these components will employ for inrush signals identification.

MM is a nonlinear approach and has been widely used in many signal/image processing applications due to its simple and robust performance. Dilation and erosion are the two basic operators of MM, which are defined as

$$(f \oplus b)(x) = max\{f(x-s)+b(s)\}, x \in D_f, s \in D_b \tag{8}$$

$$(f \ominus b)(x) = min\{f(x+s)-b(s)\}, x \in D_f, s \in D_b \tag{9}$$

Where f is the signal under processing, b is the SE, and D_f and D_b represent the field of definition of f and b, respectively.

The morphological wavelet is a nonlinear multiresolution signal decomposition scheme. A formal definition of the morphological wavelet is presented as follows. Assume that sets V_j and W_j exist. V_j is referred to as the signal space at level j, and W_j is the detail space at level j. The morphological wavelet has two analysis operators which together decompose a signal in the direction of increasing j. The signal-analysis operator $\psi_j^{\uparrow}$ maps a signal from V_j to V_{j+1} (i.e., $\psi_j^{\uparrow} : V_j \longrightarrow V_{j+1}$), while the detail analysis operator maps it from V_j to W_{j+1} (i.e., $\omega_j^{\uparrow} : V_j \longrightarrow W_{j+1}$). On the other hand, a synthesis operator proceeds in the direction of decreasing j, denoted as $\Psi_j^{\downarrow} : V_{j+1} \times W_{j+1} \longrightarrow V_j$.

In order to yield a complete signal representation, the mappings $(\psi_j^{\uparrow}, \omega_j^{\uparrow})$ and $\Psi_j^{\downarrow}$ should be inverses of each other, i.e.,

$$\begin{gathered} \Psi_j^{\downarrow}\left(\psi_j^{\uparrow}(x_j), \omega_j^{\uparrow}(x_j)\right) = x_j \\ \begin{cases} \psi_j^{\uparrow}\left(\Psi_j^{\downarrow}(x_{j+1}, y_{j+1})\right) = x_{j+1} \\ \omega_j^{\uparrow}\left(\Psi_j^{\downarrow}(x_{j+1}, y_{j+1})\right) = y_{j+1} \end{cases} \end{gathered} \tag{10}$$

Here, x is called the approximation signal and y is the detail signal. Therefore, decomposing an input signal $x_0 \in V_0$ with the following recursive analysis scheme is:

$$x_0 \longrightarrow \{x_1, y_1\} \longrightarrow \{x_2, y_2, y_1\} \longrightarrow \cdots \longrightarrow \{x_j, y_j, y_{j-1}, \dots, y_1\} \longrightarrow \cdots$$

Where $x_{j+1} = \psi_j^{\uparrow}(x_j)$, $y_{j+1} = \omega_j^{\uparrow}(x_j)$, and x_0 can be exactly reconstructed from x_j and $y_j, y_{j-1}, \dots, y_1$ by means of the following recursive synthesis scheme:

$$x_{j-1} = \Psi_{j-1}^{\downarrow}(x_j, y_j) \tag{11}$$

Let us analyse the scheme in detail which was proposed (Z. Lu et al, 2009) based on morphological decomposition scheme. This work proposes MDS based on the concepts of morphological wavelet. The operators in the scheme are specifically designed by using fundamental morphological operators-dilation and erosion-and are able to decompose differential currents signals into a series of components for the purpose of inrush identification.

In MDS, the analysis operators $\psi_j^{\uparrow}$ and $\omega_j^{\uparrow}$ and the synthesis operator $\Psi_j^{\downarrow}$ are defined as

$$\begin{aligned} \psi_j^{\uparrow}(r_j) &= r_{j+1} = \gamma(r_j) \\ \omega_j^{\uparrow}(r_j) &= s_{j+1} = r_j - \gamma(r_j) \\ \Psi_j^{\downarrow}\left(\psi_j^{\uparrow}(r_j)\ \omega_j^{\uparrow}(r_j)\right) &= r_j = r_{j+1} + s_{j+1} \end{aligned} \tag{12}$$

Where $r_1 = I$ is the transformer differential current. $s_1 = \emptyset$ and $j = 1,2, \ldots$. With the analysis operators, the signal I is decomposed into a set of components $\{s_2, \ldots, s_j, r_j\}$; and by the synthesis operator, it can be reconstructed from $I = r_j + \sum_j s_j$. $\gamma(r_j) = (r_j \ominus g_j) \oplus g_j$, where $\ominus$ and $\oplus$ denote the morphological erosion and dilation, respectively, and g_j is the structuring element (SE) at the decomposition level j. With such a scheme, the signal I is decomposed into a set of segments which reveal the shape information of the signal. Each half cycle of the current signal is decomposed into several fractions, the width of which is determined by the length of the corresponding SE. The height of these fractions can therefore be viewed as the increment of the current signal.

Prior to applying the decomposition scheme for inrush identification, the current I is translated into two signals as

$$\begin{aligned} I' &= I + c_1 \\ I'' &= -I + c_2 \end{aligned} \tag{13}$$

Where c_1 and c_2 are predetermined constants, so that I' and I'' are calculated by the morphological operators.

I' is an input signal applied to deal with half cycles which contain the peaks of I, while I'' is the other input signal to process half cycles which contain the valleys of I. SEs g_j are simple zero-valued flat lines with length of l_j. Assume that f_r represents the sampling frequency of the system, then $l_j = j/f_r$ ($j = 1,2, \ldots, N/4$), in which N is the number of sampling points per cycle.

The process of the decomposition scheme is illustrated in Fig. 8. It can be seen from this figure that current I' is the additional of I and 20 A, and its mirror I'' is the additional of $-I$ and 50 A, which make $I' \geq 0$ and $I'' \geq 0$. The decomposition scheme beings from level 1 ($j = 1$) and ends at level $j = N/4$. A group of components s_j can be extracted from the currents I' and I'' by using (12), and the height of the current increment for each component is measured and denoted as I_j. Assuming that the currents in Fig. 8 are sampled with 12 points per cycle, the decomposition scheme will iteratively run three times and six components of s_j, in total, are extracted from I' and I'', respectively. I_1, I_2 and I_3 are the current increments of s_j, which are extracted from the current I'. Another group of current increments I_1, I_2 and I_3 are obtained from I'', respectively.

For identification magnetizing inrush current from other conditions by above approach a feature criterion can define to quantify the features of the current waveform, based on the measured current increments I_j in comparison with the values calculated from a standard sinusoidal wave.

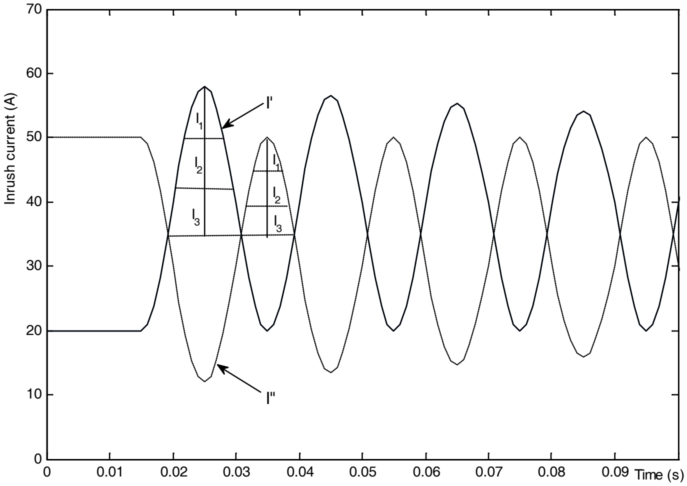


Fig. 8. Morphological decomposition of a current waveform

5.4 Neural network method for pattern recognition

Artificial Intelligence (AI) based techniques are well developed in the areas of pattern classification and recognition.

Neural networks (NNs) are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the connections between elements largely determine the network function. You can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The Fig. 9 illustrates such a situation. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are needed to train a network.

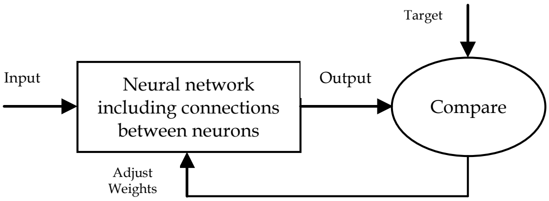


Fig. 9. Learning block diagram

Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems.

The NN applications were extended to variety of power system protective relaying and fault analysis problems.

The review indicates that most of the development efforts were related to the most common relaying functions such as distance protection for transmission lines and transformer current differential protection. This is expected due to the importance, complexity and wide application of the mentioned protection principles. However, it is increasing to note that NN applications were mostly related to the fault detection and classification, which confirms the unique NN capability to act as a pattern classifier.

As per other non-relaying applications, it appears that fault analysis and detection of equipment incipient failure had created a lot of attention for NN applications. The ability of the NN architecture to process data in parallel and in a hierarchical fashion has been exploited in the fault analysis applications. The NN ability to learn from historical data was quite useful in the equipment diagnostic area. More developments are expected in both the fault analysis and equipment diagnostic areas.

Of the three most common types of ANNs, namely multi-layer, perceptron (MLP), Kohonen network (KN) and Hopfield network (HN), the MLP has hitherto been the mainstream of applications in power systems; this is principally because the supervised learning associated with the MLP is superior in terms of accuracy compared with either the KN or HN.

There are now widespread applications of ANNs in power systems. However, this part deals with only one problem, fault classification in double-circuit transmission lines using combined unsupervised/supervised in some detail (Aggarwal & Yonghua, 1998).

Parallel transmission lines which can significantly increase transmission capacity on existing systems are finding more widespread usage. However, there is difficulty in classifying the fault types on such lines using conventional techniques, principally because faulted phase(s) on one circuit have an effect on the phases of the healthy circuit due to mutual coupling between the two circuits. The problem is compounded by the fact that this coupling is highly non-linear in nature and is dependent on a complex interplay amongst a number of variables. As a consequence, the coupled phase(s) on the healthy circuit may sometimes be wrongly diagnosed as being faulted phase(s) under certain fault conditions. Thus conventional classifiers based on logical comparison techniques or linear algorithms are not well suited for such circuits. In this respect, neural computing has the very important attribute of being able to solve non-linear system identification problems through using neurons, links and learning algorithms, and hence ANNs are ideally suited to deal with complex non-linear fault classification problem.

ANNs have to be trained to learn and, in this respect, the training algorithms can be divided into supervised, unsupervised and combined unsupervised/supervised as shown in Fig. 10. Classifiers trained with supervision require data labels that specify the correct class during training. Clustering algorithms use unsupervised training and group unlabelled training data into internal clusters. Classifiers that use combined unsupervised/supervised training firstly use unsupervised training with unlabelled data to form internal clusters; labels are then assigned to clusters during the supervision stage. Different ANNs with, different training techniques have their own advantages and disadvantages. A typical supervised error back-propagation (EBP) network is a non-linear regression technique which attempts to minimise the global error. An EBP network can provide very compact distributed representations of complex data sets, and is smaller in size compared with a combined unsupervised/supervised ANN with the same inputs and outputs. However, training of an EBP network is very slow (time consuming), needs much larger training sets and it very easily gets stuck on local minima. Furthermore, it can be difficult to retrain the ANN with new training data.

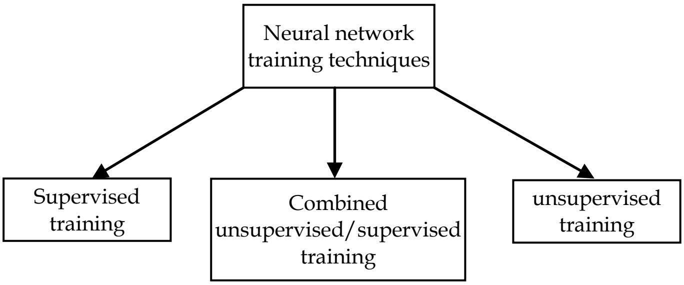

Neural network training techniques
Supervised training
Combined unsupervised/supervised training
unsupervised training

Fig. 10. ANN training techniques

Unsupervised learning means learning examples without teaching, i.e. there are no desired outputs. A typical unsupervised learning network is the KN. The network attempts to learn a topological map from an N-dimensional input space into a two dimensional feature space. Thus the network has many advantages over the EBP network, such as fast learning, a much smaller amount of training data etc. But, in view of the fact the network is without an output layer, it is not recommended to be used on its own for either pattern classification or other decision-making processes. Rather, it is used as the front end to an output layer with supervised learning and becomes a combined unsupervised/supervised (CUS) learning network, the subject of the technique described here.

This part proposes a fault type classification technique for double-circuit transmission lines using a CUS network.

The CUS-based classifier is a technique that separates object recognition into two parts: (i) feature extraction with unsupervised learning in the first stage, and (ii) classification with supervised learning sitting on the top, subsequently. An important basic principle is that the features must be independent of class membership, since the latter is not yet known at the feature extraction stage by definition. This implies that, if any learning methods are used for developing the feature extractors, they should be unsupervised in a sense, because the target class for each object is unknown. Fig. 11 typifies a CUS-based network.

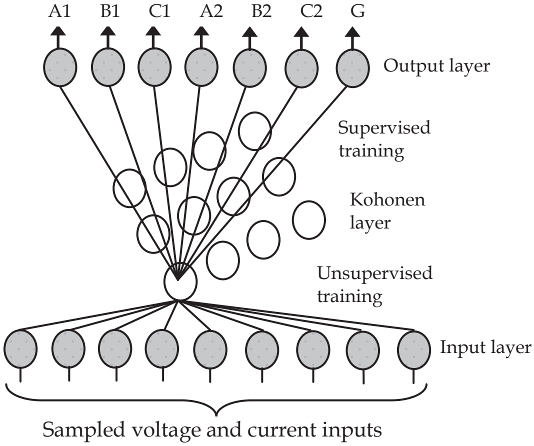


Fig. 11. CUS-based architecture network

The input vector for the CUS network comprises nine variables associated with the three voltage and six current signals in a double-circuit line. The feature extraction is based on time-domain windows, each window length being of three samples. The outputs are composed of seven variables Al, Bl, C1, PL2, B2, C2 and G; of these, '1' and '2' signify circuits 1 and 2, respectively, and variable G indicates whether ground is involved in a fault. As an example, if we get an output 1,0,0,0,0,0,1 this would indicate an 'a'-phase-earth fault on circuit 1.

In order to ascertain the attributes of the CUS network over a MLP network utilising a standard EBP training algorithm, a comparison in performance was made between the two ANNs. In this respect, the latter ANN also had nine variables but, unlike the CUS network, each window length used had four samples; a smaller window length made the convergence of the network to the requisite value extremely difficult. The hidden layer had 18 neurons and the output vector again comprised seven variables.

An extensive series of case studies showed that the MLP-based network converged after about 100 000 iterations (in approximately 45 minutes on a 133 MHz Pentium PC) and reached root-mean-square (RMS) error of 0.1. On the other hand, the CUS-based network converged after only 4000 iterations, i e in approximately 2 minutes, and reached a much lower RMS error of 0 03 Furthermore (although not shown here), of the 100 test cases considered, the misclassification rate was 6% and 1% for the MLP and CUS networks, respectively.

6. Wavelets and signal processing

6.1 Wavelet theory

Wavelet theory provides a unified framework for a number of techniques which had been developed independently for various signal processing applications. For example, multiresolution signal processing, used in computer vision; sub band coding, developed for speech and image compression; and wavelet series expansion, developed in applied mathematics, have been recently recognized as different views of a signal theory. In fact, wavelet theory covers quite a large area. It treats both the continuous and the discrete-time cases. It provides very general techniques that can be applied to many tasks in signal processing, and therefore has numerous potential applications. In particular, the Wavelet Transform (WT) is of interest for the analysis of non-stationary signals, because it provides an alternative to the classical Short-Time Fourier Transform (STFT). In contrast to the STFT, which uses a single analysis window, the WT uses short windows at low frequencies. This is in the spirit of constant relative bandwidth frequency analysis. The WT is also related to time-frequency analysis.

6.2 Signal Construction Using Wavelets

Wavelet theory establishes that a general transient signal can be constructed by the superposition of a set of special signals (different structures occurring at different time scales and at different times). These special signals may be selected as wavelets. For a set of wavelets to be admissible as a basic building block, they must satisfy two basic conditions: they must be oscillatory, and they must decay to zero quickly. If these conditions are combined with the condition that the wavelets must also integrate to zero, then these conditions are the non-rigorous admissibility criteria that must be satisfied to be a wavelet.

The selection of the best wavelets is a function of the characteristics of the signal to be processed. For example, a musical tone can be described by four basic parameters: intensity, frequency, time duration, and time position. Thus, the key to the process is to select a wavelet to realize the signal in terms of the best basis and most efficient superposition. The best and most efficient wavelet set is also a function of the objective of the reconstruction. Typical applications are compactation for storage purposes, fast reconstruction for signal identification, and efficient reconstruction for signals analysis. The selection of the best wavelet basis is a function of the characteristics of the original signal to be reconstructed or analyzed. It also depends on the compact support and/or fast reconstruction required by the process.

For image processing, for example, and due to improved resolution and efficiencies, the best wavelet basis is usually found to be in a family of multiresolution functions that are orthogonal or biorthogonal. However, these bases exploit (for efficiency reasons) a specialized spacing in the wavelet parameters that specify position (shift) and dilation (width) which requires the scale and translation parameters to be spaced by integer powers of 2. The spacing that is usually used is called a dyadic lattice. The nature of power system signals seems to point towards trigonometric based wavelets. For power system electromagnetic transient signals, the wavelet basis should have two desirable characteristics:

1. Reduce the number of wavelet components that describe the signals
2. Reveal the natural (physical) transient oscillatory components of the signal.

6.3 Differential protection based on wavelet transform

Traditional digital protective relays present several drawbacks; for instance, they are usually based on algorithms that estimate the fundamental component of the current and voltage signals neglecting higher frequency transient components. Moreover, phasor estimation requires a sliding-window of a cycle that may cause a significant delay. Furthermore, accuracy is not assured. The Fourier transform is a very useful tool for analyzing the frequency content of stationary processes. When dealing with non-stationary processes, however, other methods for determining the frequency content must be applied.

For this reason wavelet decomposition is ideal for studying transient signals and obtaining a much better current characterization and a more reliable discrimination. Wavelets allow the decomposition of a signal into different levels of resolution (frequency octaves). The basis function (Mother Wavelet) is dilated at low frequencies and compressed at high frequencies, so that large windows are used to obtain the low frequency components of the signal, while small windows reflect discontinuities.

Wavelet transform has a special feature of variable time-frequency localization which is very different from windowed Fourier transform.

Differential protection algorithms based on FFT have disadvantages including the neglecting of high frequency harmonics. Furthermore, different windowing techniques should be applied to calculate the current and voltage phasors and this causes significant time delay for the protection relay. In this case, accuracy is not assured completely. Due to increased standards of the delivered energy quality such as IEEE 519, high performance algorithms should be taken into account.

The Grossmann & Morlet (1984) definition of the continuous wavelet transform (CWT) for a 1-D signal $f(x) \in L^2(R)$ is:

$$\begin{aligned} W(a,b) &= k(a) \int_{-\infty}^{+\infty} f(x)\overline{\Psi}\left(\frac{b-x}{a}\right) dx \\ &= k(a) \int_{-\infty}^{+\infty} f(x)\widetilde{\Psi}\left(\frac{x-b}{a}\right) dx \end{aligned} \tag{14}$$

Where $\widetilde{\Psi}(x) = \overline{\Psi}(x)$; $a \in R^+$ and $b \in R$, are the scale and the position parameters, respectively, with R^+ being the set of positive real numbers; $L^2(R)$ denotes the Hilbert space of square integrable functions, and the bar denotes the conjugated complex. The constant $k(a)$ can be taken to be $1/\sqrt{a}$ in order to insure normalization in energy of the set of wavelets $\Psi(\frac{x-b}{a})$ obtained by a translation and dilation of the "mother wavelet" Ψ. The first formula permits the interpretation of the wavelet transform as a convolution product; the second as a correlation function. If the wavelet is symmetric and real $\widetilde{\Psi} = \overline{\Psi}$ (as in the case of the Poission wavelet) both notions coincide (Moreau et al. 1997).

The main advantage of the CWT is that it reveals the signal content in far greater detail than either Fourier analysis or the discrete wavelet transform (DWT). The continuous nature of the wavelet function is kept up to the point of sampling the scale-translation grid used to represent the wavelet transform is independent of the sampling of the signal under analysis. In this case, the discrete wavelet transform is:

$$\widetilde{W}_{l,k} = \int_{-\infty}^{+\infty} f(x)\Psi_{l,k}(x)dx \tag{15}$$

and the inverse discrete transform (IDWT) is

$$f(x) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \widetilde{W}_{l,k} \ \Psi_{l,k}(x) \qquad (16)$$

In the L^2 sense.

The advantage of analysing a signal with wavelets is that it enables one to study the local features of the signal with a detail matched to their characteristic scale. In the temporal domain such a property allows for an effective representation of transient signals. We can say that the DWT enables one to make a multiresolution analysis of a signal. It is possible to have both smooth wavelets with compact support and symmetry of the associated scaling functions and this avoids bias for the locations of maxima and minima of the signal.

The Wavelet Transform is well suited to the problem in this study. It is similar to the Fourier transform, but uses a basis function that decays rapidly from a central feature rather than the infinite sine function. For this reason wavelet decomposition is ideal for studying transient signals and obtaining a much better current characterization and a more reliable discrimination.

The application areas of wavelets cover time and frequency analysis, electromagnetic analysis, filters, integral equations, transient analysis, picture processing, and data compressing techniques.

In Figure 12, time and frequency localization are shown for the short time fast Fourier Transform. This approach was presented by Gabor in 1945.

In Figure 13, time and frequency localization are shown for the continuous wavelet transform. This approach was presented by Morlet in 1980.

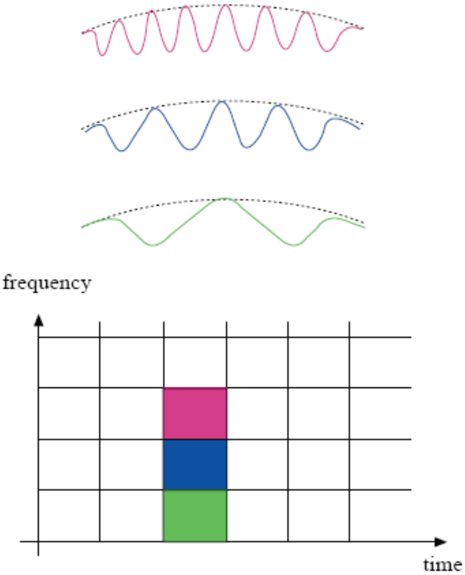


Fig. 12. Gabor localizes the short time fast Fourier Transform (STFFT) in 1945

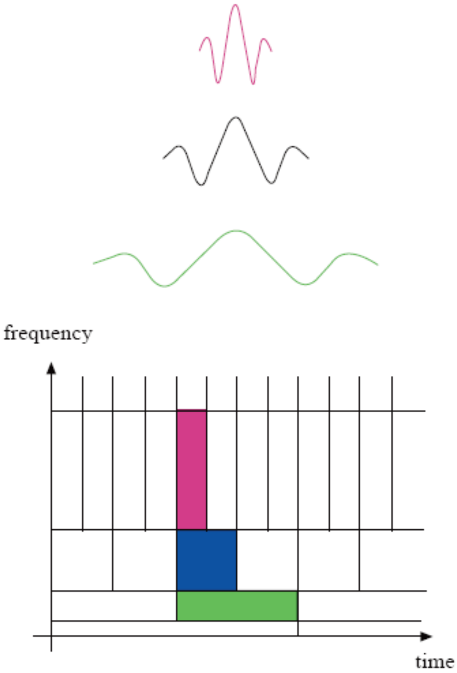


Fig. 13. Morlet proposes the continuous wavelet transform in 1980

6.4. Identification of signals using wavelet transform

A wavelet-based signal processing technique is an effective tool for power system transient analysis and feature extraction. An application of wavelet analysis to identify various types of currents flowing through a power transformer.

Recently, several new protective schemes have been proposed to deal with problem in power transformer protection based on wavelet transforms (WT). Some of wavelet based methods use voltage and current signals for identification magnetizing inrush current from internal fault currents. The drawback of these methods is that those require the measurement of voltage in addition to current that increases the cost of hardware implementation. Another methods use combination of WT and neural network. Generally, in these methods a WT use as a pre-processor and the output of the wavelet is the input of the artificial neural network. The required algorithm training is some drawbacks of these algorithms.

This work proposes a new algorithm based on the Wavelet Transform (WT) to identify magnetizing inrush current from internal fault current in three phase power transformers. To discriminate between various cases, the developed method uses different features of fault and inrush currents. At first the wavelet transform technique is applied to decompose transformer differential currents into approximated and detailed wavelet components (i.e. An and D1 − Dn). Each of these levels are time domain signals cover specific frequency band. Then, a diagnosis criterion by using the specific wavelet coefficients is defined. This criterion discriminate internal faults from inrush currents accurately and in short time (less than

5 ms) after the disturbance. The proposed algorithm is evaluated using various simulated different inrush and internal fault current signals on a power transformer. Magnetizing inrush currents and fault currents has been developed using the ATP-EMPT software. Then using wavelet and defined criteria, the transient fault current and magnetizing inrush current are differentiated. The results proved that the proposed technique is able to offer the desired responses and could be used as a very fast and accurate method.

6.4.1 proposed algorithm

The new proposed algorithm is based on waveform analysis of the fault and inrush currents. Fig. 14 shows the features of these waveforms. As is shown, the inrush current has a non-sinusoidal shape and there is a dead period per cycle in magnetizing inrush during which the current will be near zero because of the saturation characteristic of the transformer. Magnetizing inrush also exhibit a characteristic peaked wave which is caused by asymmetric saturation of the transformer core. The inrush current at the switching time increases very slowly and is near zero; while the progress of slope variation is increasing and after a few samples it amplifies in rapidly. However, when a fault occurs, slope of the differential current at the fault time is high, and slope variation decrease as time passes. These different behaviors could be used for discrimination of various cases.

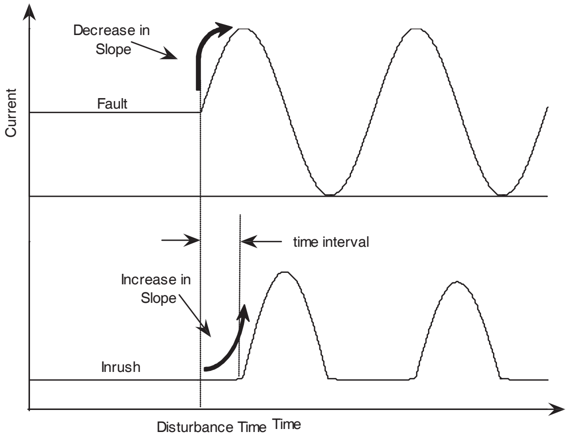


Fig. 14. Features of fault and inrush waveforms

The proposed algorithm is based on this fact that the location of rapid slope variation, for inrush current occurs after internal fault by a time interval. A large slope in the time domain means that there are higher frequencies in the frequency domain. Therefore, following the internal fault, the amplitude of the high frequencies at the initial instants has larger values than the other times. However, following the inrush current the amplitude of the high frequencies components at the initial instants is lower than the other times. The differential current and resultant frequency levels (A7 and D1-D7) from WT due to inrush current and AB-G internal fault at t=0.1 s are shown in Figs. 15 and 12 respectively.

Through various simulations, it is found that the mentioned features appear in D4 wavelet (Table 2). The time duration between the time of disturbance and the maximum peak of the differential current in D4 is considered as the diagnosis criterion, and called t_p.

Wavelet Coefficients	Frequency band (Hz)
A7	0-39.06
D7	39.06-78.125
D6	78.125-156.25
D5	156.25-312.5
D4	312.5-625
D3	625-1250
D2	1250-2500
D1	2500-5000

Table 2. Wavelet frequency levels for sample rate 10 khz

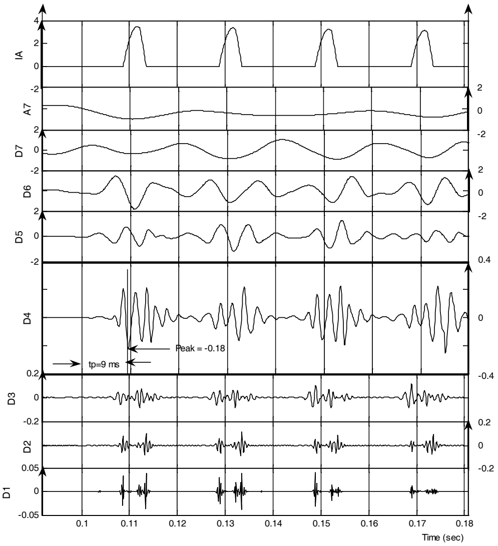


Fig. 15. Differential current and related frequency levels, inrush

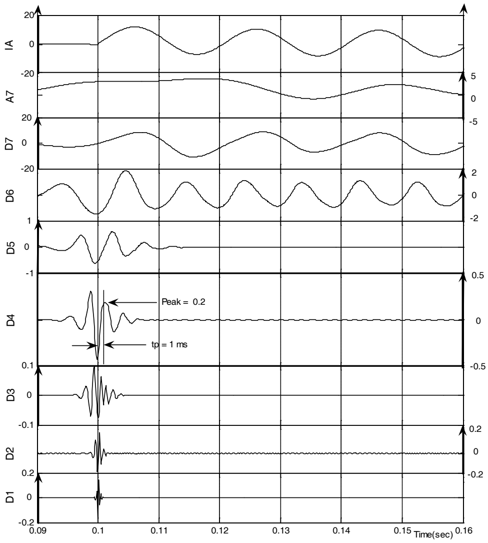


Fig. 16. Differential current and related frequency levels, AB-G fault

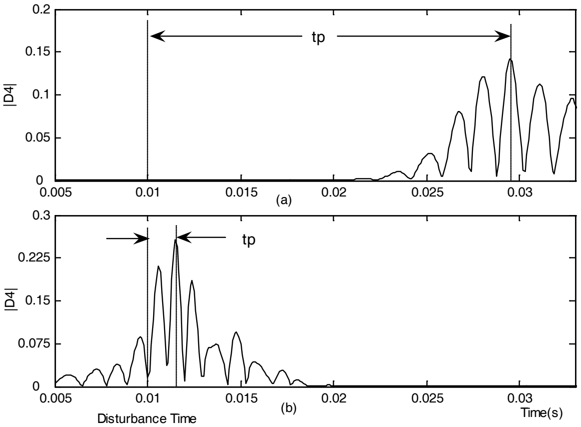


Fig. 16. D4 and t_p for (a) inrush (b) internal fault

Fig. 16 shows the interval time t_p and the absolute values of the differential current waveforms for the fault current and inrush current at frequency level D4. In the case of inrush current, t_p is higher than a setting $(t_p > t_{setting})$, and in the case of internal fault, t_p is lower than a setting $(t_p < t_{setting})$. Comparison of t_p with $t_{setting}$ is considered for three phases and if at least in one phase $t_p < t_{setting}$, a fault is occurred and the trip command is issued and else, there is no any trip command. As shown in Fig. 16, the above criterion can be used to discriminate the internal fault from the inrush current in about a quarter a cycle. It provides a very quick and simple algorithm.

The performance of the proposed algorithm was evaluated for different types of fault and inrush currents. Different cases of fault currents are simulated where some factors affecting the characteristics of the current, such as type of fault and load condition are considered. Different cases of inrush current are also simulated by varying some parameters that influence the characteristics of this current (e.g. residual core flux and voltage angle). Moreover, different cases for simultaneous inrush and fault conditions are simulated.

A) Inrush Current

For the case of magnetizing inrush current, the no-load transformer at a supply line voltage of 400 kV is considered. Fig. 17 shows the three-phase differential currents and the frequency range D4 for I_{diff-a}, I_{diff-b} and I_{diff-c}. Switching time is $115\ ms$ and residual core flux and phase angle of the supply is chosen $B_{rA} = 50\%$ and $\theta_A = 0^{\circ}$respectively. In this figure the frequency range D4 and relevant differential current (doted curve) in each phase are shown after initiation of disturbance. As it is seen from the Fig. 18 $t_{p-a} = 4.6\ ms$, $t_{p-b} = 4.6\ ms$ and $t_{p-c} = 6.4\ ms$ are obtained.

Investigation of various simulations reveals that values of t_p for various inrush currents are usually greater than $4 - 5\ ms$ (i.e., at least there is a period that for it t_p is greater than this range). Also for internal fault currents there is a period that for it t_p is less than $0.5 - 1.5\ ms$. Therefore we can choose $t_{setting}$equal to $2 - 3\ ms$. In this paper $t_{setting}$ is chosen as $2\ ms$. As seen from Fig. 18 $t_{p-a} > t_{setting}$, $t_{p-b} > t_{setting}$ and $t_{p-c} > t_{setting}$ which shows

that there is no fault and the maltrip is not issued. Obtained results for different conditions of inrush currents are shown in Table 3. This table shows the value of $\mathrm{t_p}$for different phases due to different inrush current. The first column shows phase a voltage angle at the instant of switching. In the second column differential current of various phase are shown. In both no-load and full-load cases shown in next columns, influence of remnant fluxes in the power transformer core at instant of switching as percent of the rated flux have been studied.
As shown in this table for all of the studied cases, the obtained value of t_p is greater than 2 ms. As a result all of these cases are correctly classified as inrush cases.

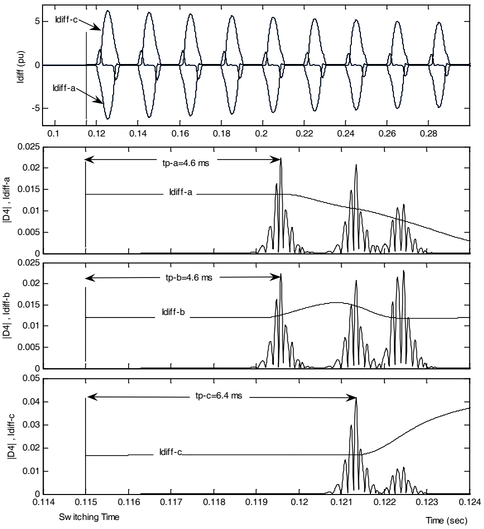


Fig. 18. Frequency range |D4| for I_{diff-a}, I_{diff-b} and $I_{diff-c}$for unloaded magnetizing inrush

θ_A	Phase	No load		Full load	
		$B_{rA}=0$ $B_{rB}=0$ $B_{rC}=0$	$B_{rA}=58\%$ $B_{rB}=0$ $B_{rC}=-58\%$	$B_{rA}=0$ $B_{rB}=0$ $B_{rC}=0$	$B_{rA}=58\%$ $B_{rB}=0$ $B_{rC}=-58\%$
0	A	5.3	6.1	5.2	4.8
	B	4.8	6	4.8	4.8
	C	4.8	5.2	4.8	6.1
80	A	4.7	6.8	4.8	3
	B	4.6	6.8	4.7	6.2
	C	6.1	6.8	5.3	3.8
120	A	4.8	5.2	4.8	4.8
	B	5.3	6.1	5.2	4.8
	C	4.8	6	4.8	4.8

Table 3. Values of $\mathrm{t_p}$ (ms) for inrush currents

B) Fault Current

To obtain the simulation data for internal fault, different faults such as single line-to-ground fault, line-to-line fault, line-to-line-to-ground fault and three phase fault simulated on the inside of the transformer zone with a balanced Y-connected load of phase connected to the secondary side. The resistance at the fault location is chosen as zero and transformer is assumed to have rated load (in the case of on-load simulations).

Fig. 19 shows three-phase differential currents and the frequency range D4 due to $\mathrm{I_{diff-a}}$, $\mathrm{I_{diff-b}}$ and $\mathrm{I_{diff-c}}$. In this case, the line-to-line-to-ground fault (fault AB-G) on the secondary side of the transformer is occurred, the transformer is full-load and the fault time is $20\ \mathrm{ms}$. As it is seen from the Fig. 19 the time duration between of the initiate instant of disturbance (fault time= 20 ms) to the maximum peak of the differential current in D4, for each phases are computed as: $\mathrm{t_{p-a}} = 0.5\ \mathrm{ms}$, $\mathrm{t_{p-b}} = 0.5\ \mathrm{ms}$ and $\mathrm{t_{p-c}} = 2.2\ \mathrm{ms}$. These values in each phase will be compared with $\mathrm{t_{setting}}$. As seen Fig. 19 $\mathrm{t_p}$ for phase a and phase b are lesser than $\mathrm{t_{setting}}$ ($\mathrm{t_{p-a}} < \mathrm{t_{setting}}$, $\mathrm{t_{p-b}} < \mathrm{t_{setting}}$ and $\mathrm{t_{p-c}} > \mathrm{t_{setting}}$) which shows the disturbance is a fault. Respect to obtained results it is founded that the proposed method discriminate fault from inrush current quickly, less than a quarter a cycle after the disturbance.

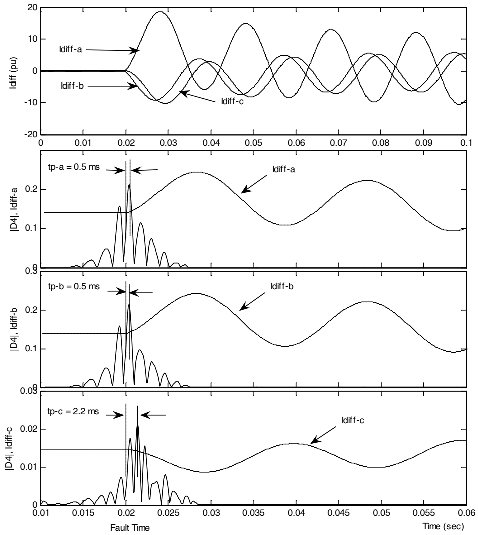


Fig. 19. Frequency range |D4| for I_{diff-a}, I_{diff-b} and I_{diff-c} for AB-G internal fault

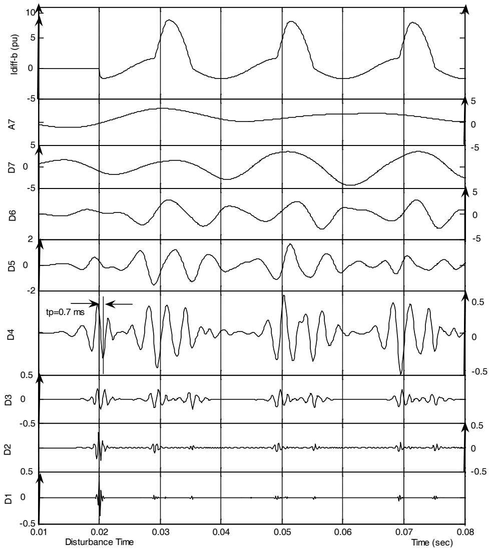


Fig. 20. Differential current and related frequency levels for simultaneous internal fault AB-G and inrush

The results of application of the proposed algorithm for internal fault conditions have been summarized in Table 4. Simulations have been carried out for different faults in no-load and on-load of power system.

C) Simultaneous Internal Fault and Inrush Current

After studying fault and inrush currents cases separately, some more complicated cases, i.e. simultaneous internal fault and inrush current are considered. In Table 5, four different cases have been studied and in all cases the fault has been properly diagnosed fast and reliably. Fig. 20 shows the differential current (phase B) for simultaneous inrush and fault (AB-G) on the primary side at $\mathrm{t} = 0.02\ \mathrm{s}$ as well as WT coefficients in D1-D7 and A7. Fig. 21 shows the three phase differential currents and D4 for $\mathrm{I_{diff-a}}$, $\mathrm{I_{diff-b}}$ and $\mathrm{I_{diff-c}}$. As it is seen from the Fig. 21 the $\mathrm{t_p}$ amount for differential current in D4 for three phases, is $\mathrm{t_{p-a}} = 0.07\ \mathrm{ms}$, $\mathrm{t_{p-b}} = 0.07\ \mathrm{ms}$ and $\mathrm{t_{p-c}} = 4.8\ \mathrm{ms}$. As seen Fig. 21 $\mathrm{t_p}$ for phase a and phase b are less than $\mathrm{t_{setting}}$ ($\mathrm{t_{p-a}} < \mathrm{t_{setting}}$, $\mathrm{t_{p-b}} < \mathrm{t_{setting}}$ and $\mathrm{t_{p-c}} > \mathrm{t_{setting}}$). Thus, the occurrence of the fault is detected accurately shorter than a quarter of a cycle ($\mathrm{t_{setting}} = 2$ ms).

θ	phase	No load				Full load			
		a-g	a-b	a-b-g	a-b-c	a-g	a-b	a-b-g	a-b-c
0	a	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	b	0.7	0.1	0.1	1.5	0.7	0.12	0.1	1.5
	c	0.1	0.1	0.1	0.1	0.1	0.1	0.12	0.11
80	a	0.1	0.13	0.1	0.1	0.12	0.1	0.1	0.1
	b	4.8	0.11	0.1	0.1	5	0.13	0.1	0.11
	c	0.1	0.1	1.5	0.1	0.11	0.1	1.5	0.1
120	a	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	b	1.5	0.1	0.1	0.1	3.1	0.1	0.1	0.1
	c	0.1	0.1	0.1	1.5	0.1	0.1	0.1	1.5

Table 4. Values of t_p (ms) for internal faults

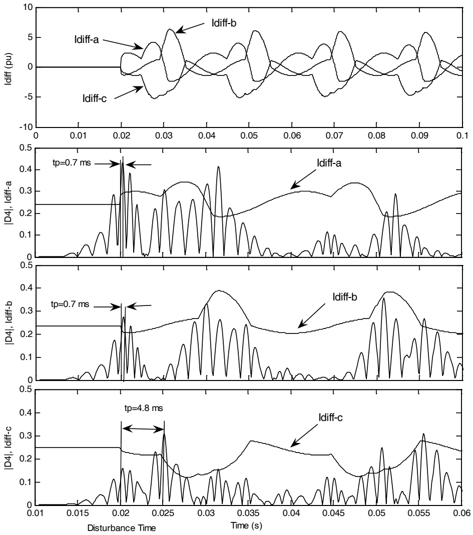


Fig. 21. Differential currents and |D4| for three phases for simultaneous internal fault AB-G and inrush current

$\boldsymbol{\theta_A}$ (Deg.)	$\boldsymbol{B_r}$	Phase	No load				Full load			
			A-G	A-B	A-B-G	A-B-C	A-G	A-B	A-B-G	A-B-C
80	$\boldsymbol{B_{rA} = 0}$	A	0.2	0.1	0.1	2.5	0.1	0.1	1.5	1.5
	$\boldsymbol{B_{rB} = 0}$	B	0.1	0.2	0.1	0.1	0.1	0.2	0.1	0.1
	$\boldsymbol{B_{rC} = 0}$	C	5.1	0.1	2.5	0.1	4.6	0.1	0.1	0.1
	$\boldsymbol{B_{rA}}$ $= \mathbf{58}\%$	A	0.2	0.1	2.5	0.2	0.1	0.2	0.1	0.1
		B	0.2	0.1	0.1	0.2	3.5	0.2	1.5	0.1
	$\boldsymbol{B_{rB} = 0}$ $\boldsymbol{B_{rC}}$ =-58%	C	4.6	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Table 5. Values of $\mathbf{T_P}$ $(\mathbf{ms})$ for simultaneous inrush currents and internal faults

7. Conclusion

This manuscript invetigates the coomon approachs for pattern recognition of current signals for identification of differential currets which flow into the differential relays. Final it presents a successful technique to distinguishing between internal faults and inrush currents in power transformers using wavelet transform. The diagnosis process in this method is based on the different characteristics of differential currents waveforms. A diagnosis criterion by quantifying the extracted features is defined in terms of time difference of amplitude of wavelet coefficients over a specific frequency band. By using this criterion function for three phases, internal faults can be accurately discriminated from inrush current. Several cases are used for testing the proposed algorithm. The simulation results show fast, accurate and reliable capabilities of the algorithm to identify different types of currents flowing in a power transformer under various inrush currents and internal faults conditions. The proposed scheme is a powerful yet simple way of assigning transformer differential current to inrush and fault groups.

8. References

ABB Company (2003). Technical Reference Manual Of RET 521*2.5-Transformer Protection Terminal, ABB Company, 1MRK 504 036-UEN, December 2003.

Aggarwal and Yonghua Song (1998). Artificial neural networks in power systems, power engineering journal december 1998.

A. Guzman, S. Zocholl, G. Benmouyal, and H. J. Altuve (2001). A currentbased solution for transformer differential protection__part I: Problem statement, IEEE Trans. Power Del., vol. 16, no. 5, pp. 485–491, Oct. 2001.

A. Guzman, S. Zocholl, G. Benmouyal, and H. J. Altuve (2002). A current-based solution for transformer differential protection__part II: Relay description and evaluation, IEEE Trans. Power Del., vol. 17, no. 5, pp. 886–893, Oct. 2002.

A. Rahmati, and M. Sanaye-Pasand (2008). New Method for Discrimination of Transformers Internal Faults from Magnetizing Inrush Currents Using Wavelet Transform, IEEE POWERCON2008 Conference, New Delhi, India, 12-15 October 2008.

H. Mortazavi and H. Khorashadi-Zadeh (2004). A new inrush restraint algorithm for transformer differential relay using wavelet transform, in Proc. Int. Conf. Power System Technology-Powercon, Singapore, Nov. 21–24, 2004, pp. 1705–1709.

M.R. Zaman and M.A. Rahman (1998). Experimental testing of the artificial neural network based protection of power transformers, IEEE Trans. Power Del., vol. 13, no. 2, pp. 510-517, Apr. 1998.

Omar A. S. Youssef (2003). A wavelet-based technique for discrimination between faults and magnetizing inrush currents in transformers, IEEE Trans. Power Del., vol. 18, no. 1, pp. 170–176, Jan. 2003.

P. L. Moa and R. K. Aggarwal (2001). A novel approach to the classification of the transient phenomena in power transformer using combined wavelet transform and neural network, IEEE Trans. Power Del. ,vol. 16, no. 4, pp. 654-660, Oct. 2001.

S. A. Saleh and M. A. Rahman (2005). Modeling and Protection of a Three-Phase Power Transformer Using Wavelet Packet Transform, IEEE TRANSACTIONS ON POWER DELIVERY, VOL. 20, NO. 2, APRIL 2005.

S.Sudha and A. Ebenezer Jeyakumar (2007). Wavelet and ANN Based Relaying for Power Transformer Protection, Journal of Computer Science 3 (6): 454-460, 2007, ISSN 1549-3636.

Y. Sheng and M. Steven (2002). Decision trees and wavelet analysis for power transformer protection, IEEE Trans. Power Del., vol. 17, no. 2, pp. 429–433, Apr. 2002.

Z. Lu, W. H. Tang, T. Y. Ji and Q. H. Wu (2009). A morphological scheme for inrush identification in transformers protection, IEEE Trans. Power Del., Vol. 24, no. 2, pp. 560-568, April 2009

Forecasting Air Quality Data with the Gamma Classifier

Itzamá López-Yáñez[1], Cornelio Yáñez-Márquez[1],
Víctor Manuel Silva-García[2]
[1] *IPN Computing Research Center,*
[2] *IPN Computing Innovation and Technological Development Center*
Mexico

1. Introduction

Environmental topics have gained the attention of increasingly large portions of global population. In different languages and through diverse means, civil associations launch campaigns for people to realize the importance of protecting the environment (Toepfer *et al.*, 2004; Hisas *et al.*, 2005), even attracting the active participation of governments (United Nations, 1992; United Nations, 1997; Secretaría de Comercio y Fomento Industrial, 1986; Web del Departamento de Medio Ambiente y Vivienda de la Generalitat de Cataluña, 2007).

Computer Sciences have not been immune to the awareness dawn. In this sense, several techniques of artificial intelligence have been applied to the analysis and forecasting of environmental data, such as artificial neural networks (Sucar *et al.*, 1997; Dutot *et al.*, 2007; Salazar-Ruiz *et al.*, 2008) and Support Vector Machines (Wang *et al.*, 2008). One particular technique which has been recently used in the prediction of environmental data —in particular, air quality data— is the Gamma classifier (López, 2007). This relatively new algorithm has shown some promising results.

In this work the Gamma classifier is applied to forecast air quality data present in public databases measured by the Mexico City Atmospheric Monitoring System (*Sistema de Monitoreo Atmosférico*, SIMAT in Spanish) (Sistema de Monitoreo Atmosférico de la Ciudad de México, 2007).

The rest of the chapter is organized as follows: the air quality data and SIMAT are described in section 2, while section 3 is dedicated to the Gamma classifier. Section 4 contains the main proposal of this work, and in section 5 the experimental results are discussed. Conclusions and future work are shown in section 6.

2. SIMAT

The Mexico City Atmospheric Monitoring System (*Sistema de Monitoreo Atmosférico de la Ciudad de México*, SIMAT in Spanish) is tightly coupled with the evolution of the Mexican

capital, and with the problems inherent to its development. The information herein presented is taken from (Sistema de Monitoreo Atmosférico de la Ciudad de México, 2007).
SIMAT is committed to operate and maintain a trustworthy system for the monitoring of air quality in Mexico City, as well as analyzing and publishing this information in order to fulfil the current requirements and legislation. The objective of SIMAT is to watch and evaluate the air quality in Mexico City, as a pre-emptive measure for health protection of its inhabitants, in order to promptly inform the populace as well as enable decision making in prevention and air quality improvement programs. SIMAT is made up by four specialized subsystems, one Atmospheric Monitoring Mobile Unit, and a Calibration Standards Transfer Laboratory. The four subsystems are:

- RAMA (Automatic Atmospheric Monitoring Network, *Red Automática de Monitoreo Atmosférico* in Spanish) takes continuous and permanent measurements of several contaminants: ozone (O_3), sulphur dioxide (SO_2), nitrous oxides (NO_x), carbon monoxide (CO), particulate matter less than 10 microns in diameter (PM_{10}), and particulate matter less than 2.5 microns in diameter ($PM_{2.5}$); each measurement is taken automatically every hour.
- REDMA (Manual Atmospheric Monitoring Network, *Red Manual de Monitoreo Atmosférico* in Spanish) monitors particulate matter suspended in the air — particulate matter less than 10 microns in diameter (PM_{10}), particulate matter less than 2.5 microns in diameter ($PM_{2.5}$), and total suspended particulate matter (PST)—, as well as their concentration and composition; each measurement is taken manually every six days.
- REDMET (Meteorological and Solar Radiation Network, *Red de Meteorología y Radiación Solar* in Spanish) monitors meteorological parameters —such as wind direction and speed— and solar radiation, in order to elaborate meteorological forecasting and dispersion models; it also records and monitors the UV index.
- REDDA (Atmospheric Deposit Network, *Red de Depósito Atmosférico* in Spanish) measures both dry and wet deposit, whose analysis allows the study of rain properties and the flow of toxic substances from the atmosphere to the surface.

IMECA	**Condition**	**Effects on Health**
0-50: green	Good	Suitable for conducting outdoor activities
51-100: yellow	Regular	Possible discomfort in children, the elderly and people with illnesses
101-150: orange	Bad	Cause of adverse health effects on the population, particularly on children and older adults with cardiovascular and / or respiratory illnesses such as asthma
151-200: red	Very Bad	Cause of greater adverse health effects on the population, particularly on children and older adults with cardiovascular and / or respiratory illnesses such as asthma
>200: purple	Extremely Bad	Cause of adverse health effects in the general population. Serious complications may present in children and older adults with cardiovascular and / or respiratory illnesses such as asthma

Table 1. IMECA and its implications for health

The Air Quality Metropolitan Index (*Índice Metropolitano de la Calidad del Aire*, IMECA in Spanish) is a reference value for people to be aware of the pollution levels prevalent in any zone, in a precise and timely manner, in order to take appropriate protection measures. When the IMECA of any pollutant is greater than 100 points, its concentration is dangerous for health and, as the value of IMECA grows, the symptoms worsen, as can be seen in table 1.

Generating the IMECA is one of the primordial tasks of SIMAT. Since July 1st, 1998, the IMECA has been transmitted 24 hours every day to different electronic and printed communication media. Currently, the hourly value of IMECA can be consulted online in (Sistema de Monitoreo Atmosférico de la Ciudad de México, 2007) and also by telephone at the IMECATEL service, which started operations on March 22, 2001. On both of these services, information is available 24 hours a day.

In November 2006, the *Gaceta Oficial del Distrito Federal* published the Federal District Environmental Norm (*Norma Ambiental para el Distrito Federal*) NADF-009-AIRE-2006 (Gobierno del Distrito Federal, 2006), which states the specifications for elaborating the IMECA for the criteria pollutants, such as: O_3, NO_2, SO_2, CO, PM_{10} and $PM_{2.5}$.

For each of the criteria pollutants, the norm states equations for calculating the corresponding IMECA, from the concentration data. Tables 2, 3, and 4 show these equations for CO, O_3, and SO_2, respectively. With these equations, the IMECA value and IMECA level (condition) can be easily computed from the concentration of each of the pollutants, in parts per million (ppm).

IMECA Interval	Concentration Intervals (ppm)	Equations
0-50: green	0-5.50	$\text{IMECA[CO]} = \text{con}\left[\text{CO}\right] \times 50/5.50$
51-100: yellow	5.51-11.00	$\text{IMECA[CO]} = 1.82 + \text{con}\left[\text{CO}\right] \times 49/5.49$
101--150: orange	11.01-16.50	$\text{IMECA[CO]} = 2.73 + \text{con}\left[\text{CO}\right] \times 49/5.49$
151--200: red	16.51-22.00	$\text{IMECA[CO]} = 3.64 + \text{con}\left[\text{CO}\right] \times 49/5.49$
>200: purple	>22.00	$\text{IMECA[CO]} = \text{con}\left[\text{CO}\right] \times 201/22.01$

Table 2. IMECA calculation equations for carbon monoxide (CO)

IMECA Interval	Concentration Intervals (ppm)	Equations
0-50: green	0-0.055	$\text{IMECA}[O_3] = \frac{\text{con}\left[O_3\right] \times 100}{0.11}$
51-100: yellow	0.056-0.110	
101--150: orange	0.111-0.165	
151--200: red	0.166-0.220	
>200: purple	>0.220	

Table 3. IMECA calculation equation for ozone (O_3)

IMECA Interval	Concentration Intervals (ppm)	Equations
0-50: green	0-0.065	$\text{IMECA}[\text{SO}_2] = \dfrac{\text{con}\left[\text{SO}_2\right]\times 100}{0.13}$
51-100: yellow	0.066-0.130	
101--150: orange	0.131-0.195	
151--200: red	0.196-0.260	
>200: purple	>0.260	

Table 4. IMECA calculation equation for sulphur dioxide (SO_2)

3. The Gamma classifier

This pattern classifier, of recent proposal, has shown some very promising results. The following discussion is strongly based on (López, 2007).

The basis of the Gamma classifier is the gamma operator, hence its name. In turn, the gamma operator is based on the *α*, *β*, and u_β operators and their properties, in particular when dealing with binary patterns coded with the modified Johnson-Möbius code. Also, it is important to define the sets A and B, since they are used throughout this work. Thus, let there be the sets A = {0, 1} and B = {0, 1, 2}.

3.1 Preliminaries

The alpha and beta operators are defined in tabular form, taking into account the definitions of the sets A and B, as shown in table 5.

$\alpha \;:\; A\times A \to B$		
x	y	$\alpha(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

$\beta \;:\; B\times A \to A$		
x	y	$\beta(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

Table 5. Definition of the Alpha and Beta operators

Now, the unary operator u_β, which receives as input an n dimensional binary vector **x**, and outputs a non-negative integer number, is calculated as shown in equation 1.

$$u_\beta(\mathbf{x}) = \sum_{i=1}^{n} \beta(x_i, x_i) \tag{1}$$

On the other hand, the modified Johnson-Möbius code allows to convert a set of real numbers into binary representations by following these steps:

1. Substract the minimum (of the set of numbers) from each number, leaving only non-negative real numbers.

2. Scale up the numbers (truncating the remaining decimals if necessary) by multiplying all numbers by an appropriate power of 10, in order to leave only non-negative integer numbers.
3. Concatenate e_m–e_j zeros with e_j ones, where e_m is the greatest non-negative integer number to be coded, and e_j is the current non-negative integer number to be coded.

Finally, the generalized gamma operator γ_g, which takes as input two binary patterns $\mathbf{x} \in A^n$ and $\mathbf{y} \in A^m$, with $n, m \in Z^+$, $n \leq m$, and a non-negative integer number θ; and gives a binary number as output; can be calculated as in equation 2.

$$\gamma_g(\mathbf{x}, \mathbf{y}, \theta) = \begin{cases} 1 & \text{if } m - u_\beta[\alpha(\mathbf{x}, \mathbf{y}) \mathrm{mod}\, 2] \leq \theta \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

With these tools we are now ready to present the algorithm for the Gamma classifier.

3.2 The Gamma classifier algorithm

Let $k, m, n, p \in Z^+$; $\{\mathbf{x}^\mu | \mu = 1, 2, \ldots, p\}$ be the fundamental pattern set with cardinality p, where $\forall \mu\, \mathbf{x}^\mu \in R^n$, and let $\mathbf{y} \in R^n$ be an n dimensional real-valued pattern to be classified. It is assumed that the fundamental set is partitioned into m different classes, each class having a cardinality k_i, $i = 1, 2, \ldots, m$, thus $\sum_{i=1}^{m} k_i = p$. In order to classify $\mathbf{y}$, these steps are followed:

1. Code the fundamental set with the modified Johnson-Möbius code, obtaining a value e_m for each component. The e_m value is calculated as defined in equation 3.
$$e_m = \bigvee_{i=1}^{p} x_j^i \tag{3}$$
2. Compute the stop parameter, as expressed in equation 4.
$$\rho = \bigwedge_{j=1}^{n} e_m(j) \tag{4}$$
3. Code $\mathbf{y}$ with the modified Johnson-Möbius code, using the same parameters used with the fundamental set. If any y_j is greater than the corresponding $e_m(j)$, the γ_g operator will use such y_j instead of m.
4. Transform the index of all fundamental patterns into two indices, one for the class they belong to, and another for their position in the class (i.e. $\mathbf{x}^\mu$ which belongs to class i becomes $\mathbf{x}^{i\omega}$).
5. Initialize θ to 0.

6. Do $\gamma_g\left(\mathbf{x}_j^{i\omega}, \mathbf{y}_j, \theta\right)$ for each component of the fundamental patterns in each class, following equation 2.
7. Compute a weighted sum c_i for each class, according to equation 5.

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^{n} \gamma_g\left(\mathbf{x}_j^{i\omega}, \mathbf{y}_j, \theta\right)}{k_i} \tag{5}$$

8. If there is more than one maximum among the different c_i, increment θ by 1 and repeat steps 6 and 7 until there is a unique maximum, or the stop condition $\theta \geq \rho$ is fulfilled.
9. If there is a unique maximum, assign $\mathbf{y}$ to the class corresponding to that maximum:

$$C_y = C_j \quad \text{such that } \bigvee_{i=1}^{m} c_i = c_j \tag{6}$$

10. Otherwise, assign $\mathbf{y}$ to the class of the first maximum.

The Gamma classifier is inspired on the Alpha-Beta associative memories, taking the alpha and beta operators as basis for the gamma operator. As such, the Gamma classifier is a member of the Associative Approach to Pattern Recognition, in which the algorithms and models use concepts and techniques derived from associative memories in order to recognize and classify patterns.

As can be seen, this classifier is relatively simple, requiring simple operations. Its complexity is polynomial, as was shown in (López, 2007). Also, notice that while being iterative, the classifier will reach a solution in finite time: at best in one iteration, at worst in the same amount of iterations as the stop parameter indicates (see equation 4).

Although the gamma classifier is not old, it has already been applied to several different problems: classification of the Iris Plant database, localization of mobile stations, software development effort estimation of small programs, and of course environmental data prediction. In these problems, some quite different from each other —and even unfulfilling of the basic premises of the classifier—, the Gamma classifier has shown competitive experimental result.

4. Proposed application

In the current work, the authors apply the Gamma classifier to environmental data obtained from databases derived from SIMAT, in order to forecast air quality data. In particular, the RAMA database was used. The experiments were conducted on three pollutants: carbon monoxide (CO), ozone (O_3), and sulphur dioxide (SO_2). The fundamental set was built with all the samples taken at a particular monitoring station during 2006, while the testing set was built with the samples taken during two non-consecutive months of 2007: February and May. Given that not all stations sample all pollutants, different stations were selected for each pollutant: IMP (*Instituto Mexicano del Petróleo*) for CO, CES (*Cerro de la Estrella*) for O_3, and TLI (*Tultitlán*) for SO_2. This can be seen also in table 6.

		Fundamental set			Testing set		
Experiment	**Pollutant**	**Period**	**Station**	**Size**	**Period**	**Station**	**Size**
1	CO	2006	IMP	8710	2007-Feb	IMP	651
2	CO	2006	IMP	8710	2007-May	IMP	723
3	O3	2006	CES	8749	2007-Feb	CES	651
4	O3	2006	CES	8749	2007-May	CES	723
5	SO2	2006	TLI	8749	2007-Feb	TLI	641
6	SO2	2006	TLI	8749	2007-May	TLI	711

Table 6. Composition and sizes of fundamental and testing sets for each experiment

Each pattern is made up by *n* successive samples, concatenated each after the other. As the class for such pattern, the *n*+1–th sample is used. Thus, patterns are built from the samples as mentioned above, and then these patterns are grouped together in fundamental and testing set for each experiment. The composition and size of each of these set, for each experiment, can be seen in table 6.

5. Experimental Results

As mentioned in the previous section, both the fundamental set and the testing set were formed with data taken from the RAMA database for each pollutant, containing hourly samples of concentration measured in parts per million (ppm).

With these data, input patterns of 10 samples were formed; that is, *n*=10. While the value of *n* can be arbitrarily chosen, 10 gave good results in preliminary tests. The output patterns (i.e. the class) were taken from the sample following the last sample in the pattern.

Once trained with the fundamental set, the Gamma classifier is presented with the testing set, obtaining the pollutant concentration forecast for the next hour (see figures 1 through 6).

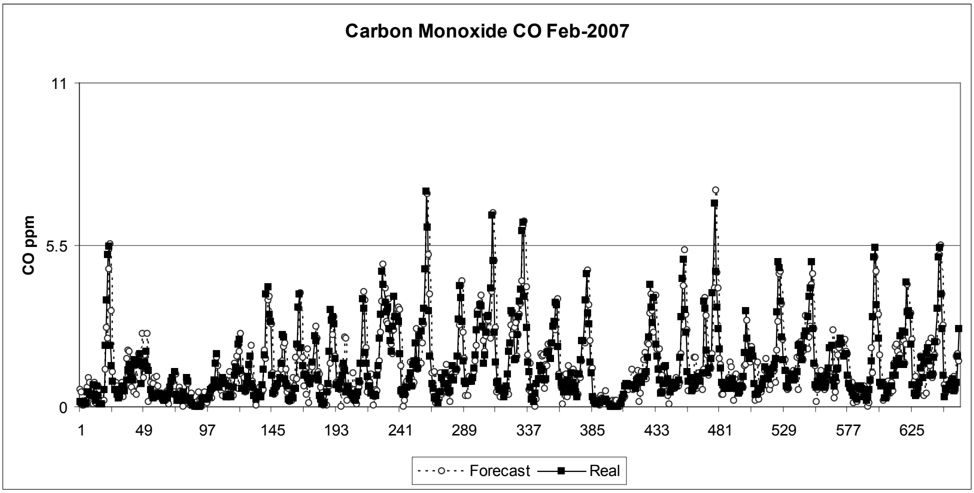


Fig. 1. Predicted values *vs* real values for carbon monoxide (CO), February 2007

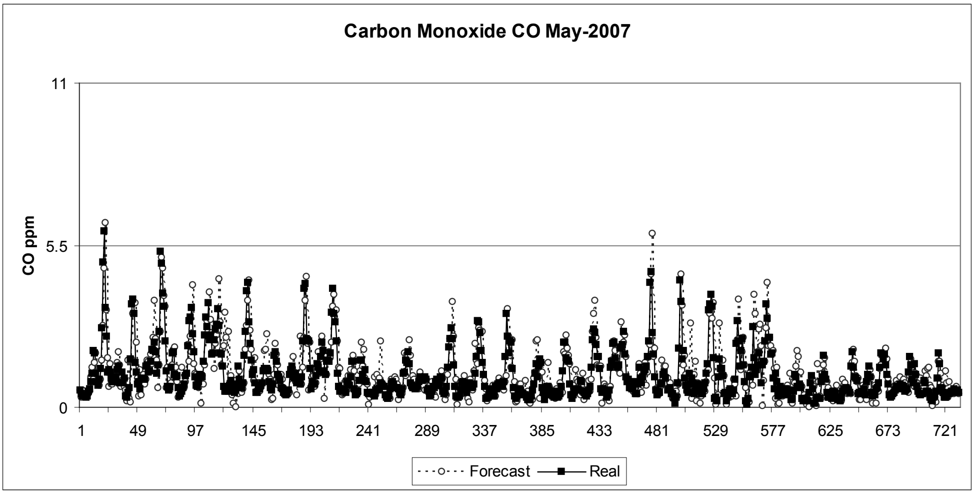


Fig. 2. Predicted values *vs* real values for carbon monoxide (CO), May 2007

Here are some examples of the results obtained: for experiment 1 (CO Feb 2007) on February 3rd at 18:00, the measured (real) CO concentration was 0.42 ppm, while the Gamma classifier predicted 0.42 ppm, which gives an error of 0.00 ppm. While this is clearly the best result, some error can be found too. For experiment 4 (O_3 May 2007) on May 12 at 17:00 the system predicted 0.034 ppm of O_3 concentration, while the observed value was 0.048 ppm, for an error of -0.014 ppm. Yet larger errors can be seen; for instance, on experiment 5 (SO_2 Feb 2007) February 19 at 1:00, the forecast was 0.059 ppm while the real concentration of SO_2 was 0.251, which amounts to an error of -0.192 ppm. These examples can be seen in table 7.

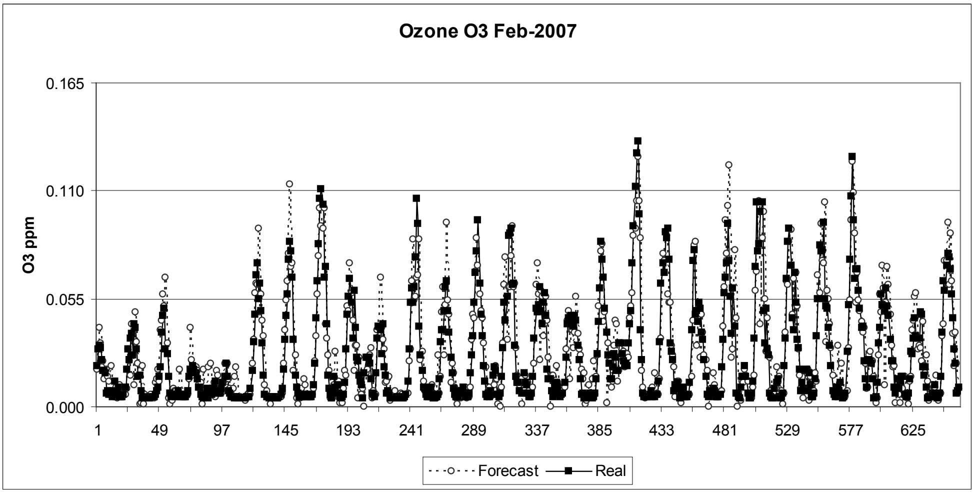


Fig. 3. Predicted values *vs* real values for ozone (O_3), February 2007

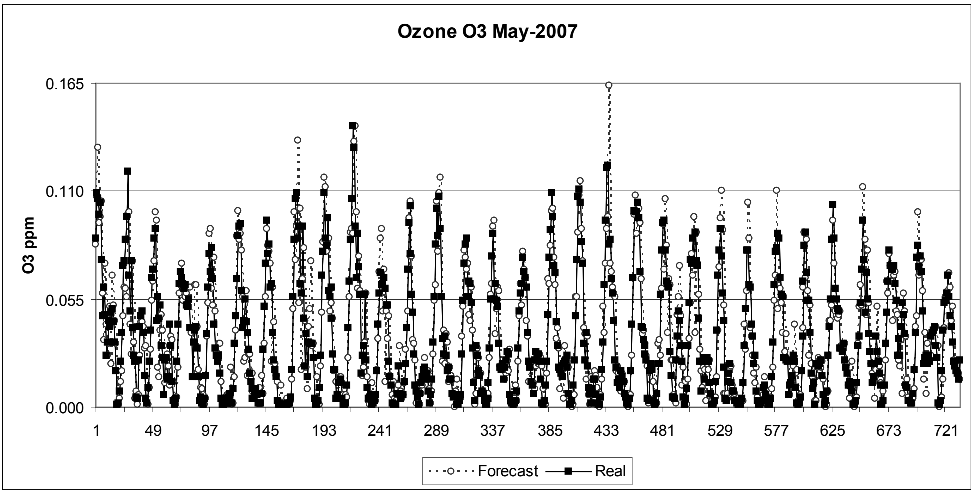


Fig. 4. Predicted values *vs* real values for ozone (O_3), May 2007

Pollutant	Date	Hour	Forecast	Observation	Error
CO	February 3	18:00	0.42 ppm	0.42 ppm	0.00 ppm
O_3	May 12	17:00	0.034 ppm	0.048 ppm	-0.014 ppm
SO_2	February 19	1:00	0.059 ppm	0.251 ppm	-0.192 ppm

Table 7. Examples of results

An interesting characteristic of these pollutants, which can be observed in the figures (1 through 6), is that CO and O_3 have a periodic behaviour according to the hour of the day,

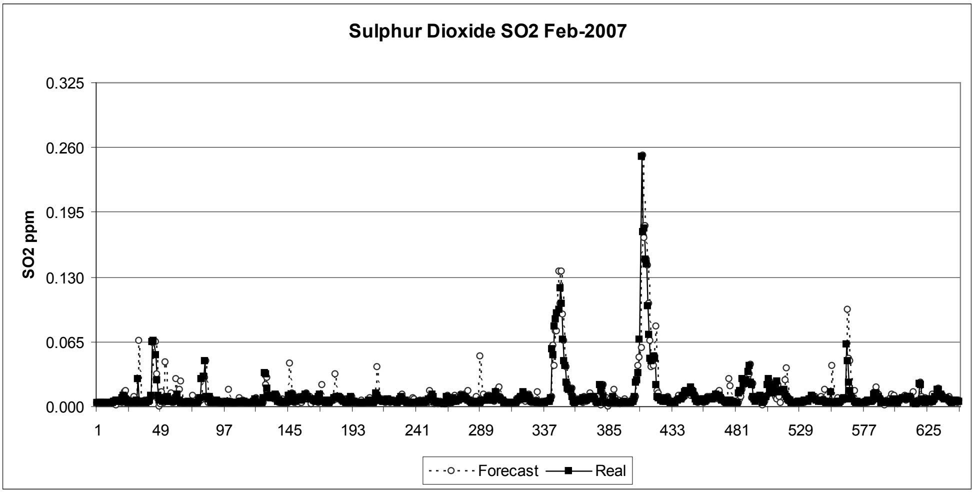


Fig. 5. Predicted values *vs* real values for sulphur dioxide (SO_2), February 2007

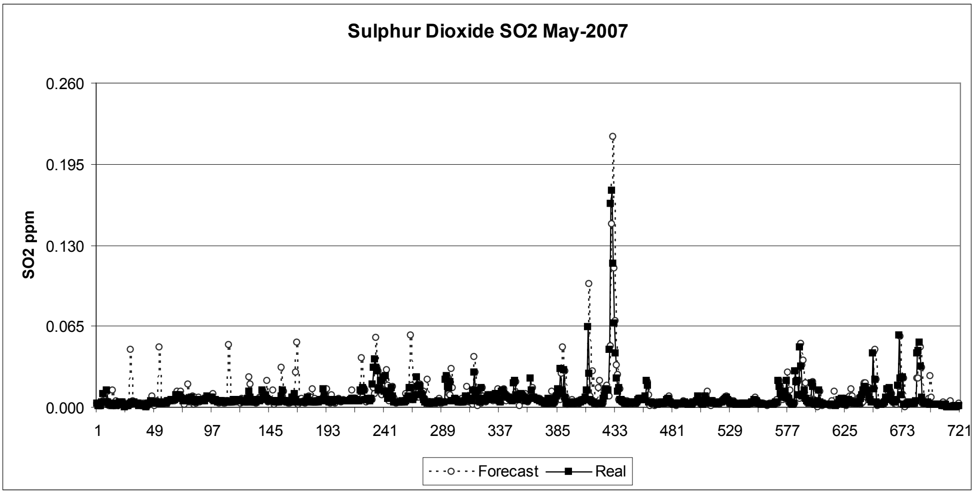


Fig. 6. Predicted values *vs* real values for sulphur dioxide (SO_2), May 2007

while SO_2 does not present such an easily discernible periodic behaviour. In the case of CO, the peaks usually happen in the second quarter of the day (6:00 to 12:00 hours), while in the case of O_3, the peaks occur more commonly around the third quarter of the day (12:00 to 18:00 hours). It is also noteworthy that the greater errors usually appear close to a peak, either positive or negative; this last observation is true for all three pollutants.

An example of the latter observation is that on February 15 and 18, there was a sharp change in the behaviour of SO_2: while during the rest of the month the concentration of this pollutant was low (it remained in the 0-0.65 ppm range, indicating a good IMECA condition), in those days the concentration reached 0.119 ppm (Feb. 15) and 0.251 (Feb. 18) for an IMECA condition of very bad. Again, on May 19, the SO_2 concentration reached exceptional levels: 0.174 ppm when the mean for that month was 0.007 ppm. It is clear that these were exceptional situations, which lie out of the ordanary situations. If the Gamma classifier (or any other algorithm for that matter) is not trained with such exceptional ciscumstances, it is to be expected that the corresponding forecast will not be accurate.

Two quantitative measures of the performances shown by the Gamma classifier on this application were used. On one side, Rooted Mean Square Error (RMSE), which is a widely used measure of performance and is calculated as shown in equation 7. On the other side, the bias, which can be calculated by following equation 8, is used to describe how much the system is underestimating or over estimating the results. For both equations, P_i is the *i*–th predicted value and O_i is the *i*–th observed (real) value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(P_i - O_i\right)^2} \tag{7}$$

$$Bias = \frac{1}{n}\sum_{i=1}^{n}\left(P_i - O_i\right) \tag{8}$$

The RMSE and bias exhibited by the results of each experiment are shown in table 8. Notice that the bias is small in all cases, especially if the size of the testing sets (641 patterns in the smallest, 723 in the largest) is taken into account.

Experiment	Pollutant	Station	Testing Period	RMSE	Bias
1	CO	IMP	2007-Feb	0.726013	7.96
2	CO	IMP	2007-May	0.611769	45.58
3	O_3	CES	2007-Feb	0.012302	0.607
4	O_3	CES	2007-May	0.014443	0.306
5	SO_2	TLI	2007-Feb	0.012096	0.573
6	SO_2	TLI	2007-May	0.010487	0.439

Table 8. RMSE and bias for each experiment

The RMSE exhibited by the three pollutants is also small. Again, that of CO is comparatively larger, although when the order of the data processed is taken into account (the mean of CO concentration during 2006 was 1.237 ppm), having a RMSE of 0.726 ppm for February and one of 0.612 ppm for May is relatively small. Ozone exhibited smaller values for RMSE: 0.0123 ppm for February and 0.0144 ppm for May. If these values are compared to the average O_3 concentration for 2006, 0.0262 ppm, it is clear that the error shown is not too high. The smallest RMSE of the three pollutants was presented by SO_2: 0.0121 ppm on February and 0.0105 ppm on May. The annual mean for this pollutant in 2006 was 0.0099 ppm, which is smaller than the RMSE for both experiments.

These comparisons between the RMSE and the previous annual mean for each pollutant is not a particularly good measure of how good the prediction was. They only indicate that the errors are close to said mean, preferably smaller. A better measure would be to compare these results with those offered by using other methods.

However, there is a problem with such comparison too. Most authors use data taken from databases close to them. Therefore, most results are obtained by processing data different for each method; it even happens that different databases use different units to measure the same pollutants [!]. Thus, a direct comparison is not appropriate, even though the same measure of error is used. It is due to this lack of a standard, benchmarking database that comparisons should be done carefully.

One example is the comparison between the results presented in (Sucar *et al.*, 1997) and those shown in the current work. The experiments on both publications were done with the same database (SIMAT – RAMA), with the same pollutant (in the case of experiments 3 and 4: O_3), with the same unit of sample measure (ppm), and with the same measure of error (absolute average error). However, the data used for experimentation was taken from different stations and different years. Although the results are comparable, they are not directly comparable. And this is the best match of data; greater caution should be taken when comparing the results obtained by experiments done on data taken from different databases.

Table 9 presents a comparison between the results obtained in this work and those presented in other publications, with the restriction of using the same database: SIMAT; in particular, the RAMA database. Notice that the results presented in (Sucar *et al.*, 1997) are greatly surpassed.

Experiment	Algorithm Used	Pollutants Considered	Size of Training / Testing Sets	Performance (Abs. Avg. Error)
	Bayesian network (Sucar *et al.*, 1997)	O_3 (ppm)	400 / 200	0.221000
	Neural network (Sucar *et al.*, 1997)	O_3 (ppm)	400 / 200	0.160000
	C4.5 (Sucar *et al.*, 1997)	O_3 (ppm)	400 / 200	0.176400
	Gamma classifier (Yáñez-Márquez *et al.*, 2008)	SO_2 (ppm)	8749 / 709	0.000408
1	Gamma classifier (current work)	CO (ppm)	8710 / 651	0.012042
2	Gamma classifier (current work)	CO (ppm)	8710 / 723	0.062183
3	Gamma classifier (current work)	O_3 (ppm)	8749 / 651	0.000918
4	Gamma classifier (current work)	O_3 (ppm)	8749 / 723	0.000417
5	Gamma classifier (current work)	SO_2 (ppm)	8749 / 641	0.000676
6	Gamma classifier (current work)	SO_2 (ppm)	8749 / 711	0.000795

Table 9. Comparison of related results (SIMAT database) in absolute average error given for pollutant concentration

Experiment	Algorithm Used	Pollutants Considered	Size of Training / Testing Sets	Performance (RMSE)
	Neural network (Dutot *et al.*, 2007)	O_3 (μg/m^3)	613 / 105	15
	Neural network (Salazar-Ruiz *et al.*, 2008)	O_3 (ppb)	NA / 1343	9.43
			NA /2367	13.79
	Online SVM (Wang *et al.*, 2008)	SO_2 (μg/m^3)	240 / 168	12.96, 10.90
	CALINE3 (Gokhale & Raokhande, 2008)	PM_{10}, $PM_{2.5}$ (μg/m^3)	~120	88, 55
	Gamma classifier (Yáñez-Márquez *et al.*, 2008)	SO_2 (ppm)	8749 / 709	0.009218
2	Gamma classifier (current work)	CO (ppm)	8710 / 723	0.611769
3	Gamma classifier (current work)	O_3 (ppm)	8749 / 651	0.012302
6	Gamma classifier (current work)	SO_2 (ppm)	8749 / 711	0.010487

Table 10. Comparison of related results (diverse databases) in RMSE, given for pollutant concentration; NA indicates a not available value, ppb means parts per billion

Even taking into account the above mentioned discussion, the error exhibited by the Gamma classifier is several orders of magnitude smaller: the average error of experiment 3 (0.000918 ppm) is more than 150 times smaller than that of the neural network (0.160000 ppm); and experiment 3 did not give the best results [!]. Also, the results of experiments 3 and 4 (0.000417 ppm) are coherent with those of (Yáñez-Márquez *et al.*, 2008) (0.000408 ppm), in the sense that they are quite similar.

On the other hand, table 10 shows the results of several experiments, done with data taken from different databases (one experiment was chosen for each pollutant among those presented in the current work). Taking these differences into consideration, as well as the fact that the results reported are based on different units (some in parts per million, other in parts per billion, and yet others in $\mu g/m^3$), it can be said that the Gamma classifier exhibits competitive performance.

6. Conclusions and Future Work

In this work, the utility of applying the Gamma classifier to forecasting air quality data has been experimentally shown. More specifically, the hourly concentration of three pollutants: carbon monoxide, ozone, and sulphur dioxide, as taken from the RAMA database, was analyzed. Six experiments were done, two on each pollutant: year 2006 was learned, and the hourly concentration values for the months of February 2007 and May 2007 were predicted.

The experimental results show a small error when compared to the data being predicted. However, it is noteworthy that most significant errors occur when the graph of the data changes direction (i.e. starts decreasing after increasing, or vice versa), implying a quite likely venue of improvement.

It is also clear that there were exceptional situations present in the data from which the testing sets for experiments 5 and 6 were built. In particular, the days February 15 and 18, and again in May 19, presented SO_2 concentration values which were out of the ordinary. Although these exceptional situations caused the Gamma classifier to incur on large errors for those days, the RMSE for both experiments was still competitive.

One possibility to improve the forecast results when such situations arise is to train the system with data taken during events similar to these, when the conditions were anomalous.

A different approach to improve the results, and not just for these experiments, would be to take into account several variables at the same time. For instance, learn from the data taken at several monitoring stations, or learning from several (related) pollutants at the same time.

It is also worthy of mention that direct comparisons with results reported in other works are difficult, since the same data is seldom used for experimentation on different works. It would greatly improve these comparisons to have a standard database to serve as a benchmark.

8. References

Dutot, A.-, Rynkiewicz, J., Steiner, F.E., Rude, J. (2007). A 24-h forecast of ozone peaks and exceedance levels using neural classifiers and weather predictions. *Environmental Modelling and Software*, Vol. 22, No. 9, (September 2007) 1261-1269, ISSN: 1364-8152.

Gobierno del Distrito Federal (2006). Norma Ambiental para el Distrito Federal (in Spanish). *Gaceta Oficial del Distrito Federal*, XVI Epoch.

Gokhale, S., Raokhande, N. (2008). Performance evaluation of air quality models for predicting PM10 and PM2.5 concentrations at urban traffic intersection during winter period. *Science of the Total Environment*, Vol. 394, No. 1 (May 2008) 9-24, ISSN: 0048-9697.

Hisas, Liliana *et al*. (2005). *A Guide to the Global Environmental Facility (GEF) for NGOs*, UNEP-United Nations Foundation.

López Yáñez, Itzamá (2007). *Clasificador Automático de Alto Desempeño* (In Spanish). M.Sc. Thesis. National Polytechnics Institute, Computers Research Center, Mexico

Salazar-Ruiz, E. *et al.* (2008). Development and comparative analysis of tropospheric ozone prediction models using linear and artificial intelligence-based models in Mexicali, Baja California (Mexico) and Calexico, California (US). *Environmental Modelling and Software*, Vol. 23, No. 8, (August 2008) 1056-1069, ISSN: 1364-8152.

Secretaría de Comercio y Fomento Industrial (1986). *Determiación de Neblina de Ácido Fosfórico en los Gases que Fluyen por un Conducto* (in Spanish), Mexican Norm NMX-AA-090-1986, Mexico.

Sistema de Monitoreo Atmosférico de la Ciudad de México (2007). *IMECA* (in Spanish). Available at www.sma.df.gob.mx/simat/pnimeca.htm

Sucar, L.E., Pérez-Brito, J., Ruiz-Suárez, J.C., Morales, E. (1997). Learning Structure from Data and Its Application to Ozone Prediction. *Applied Intelligence*, Vol. 7, No. 4, (November 1997) 327-338, ISSN: 0924-669X.

Toepfer, Klaus *et al.* (2004). *Aliados Naturales: El Programa de las Naciones Unidas para el Medio Ambiente y la sociedad civil* (in Spanish), UNEP-United Nations Foundation.

United Nations (1992). *Rio Declaration on Environment and Development*.

United Nations (1997). *Kyoto Protocol to The United Nations Framework Convention on Climate Change*.

Wang, W., Men, C., Lu, W. (2008). Online prediction model based on support vector machine. *Neurocomputing*, Vol. 71, No. 4-6 (January 2008) 550-558, ISSN: 0925-2312.

Web del Departamento de Medio Ambiente y Vivienda de la Generalitat de Cataluña (in Spanish) (2007). Available at: http://mediambient.gencat.net/cat

Yáñez-Márquez, C., López-Yáñez, I., Sáenz-Morales, G. de la L. (2008). Analysis and Prediction of Air Quality Data with the Gamma Classifier. *Lecture Notes in Computer Science* (ISI Proceedings), LNCS 5197, Springer-Verlag Berlin Heidelberg, 651-658 ISBN: 978-3-540-72394-3.

Spam Recognition using Linear Regression and Radial Basis Function Neural Network

Tich Phuoc Tran[1] , Min Li[1], Dat Tran[2] and Dam Duong Ton[3]
[1]*Centre for Quantum Computation and Intelligent Systems (QCIS)*
University of Technology, Sydney, NSW 2007, Australia
{tiptran, minli }@it.ut.edu.au

[2]*Faculty of Information Sciences and Engineering*
University of Canberra, ACT 2601, Australia
dat.tran@canberra.edu.au

[3]*Faculty of Computer Science*
University of Information Technology, VNU-HCMC, Vietnam
damdt@uit.edu.vn

Keywords: Spam Recognition, Redial Basis Function, Linear Regression

Abstract:

Spamming is the abuse of electronic messaging systems to send unsolicited bulk messages. It is becoming a serious problem for organizations and individual email users due to the growing popularity and low cost of electronic mails. Unlike other web threats such as hacking and Internet worms which directly damage our information assets, spam could harm the computer networks in an indirect way ranging from network problems like increased server load, decreased network performance and viruses to personnel issues like lost employee time, phishing scams, and offensive content. Though a large amount of research has been conducted in this area to prevent spamming from undermining the usability of email, currently existing filtering methods' performance still suffers from extensive computation (with large volume of emails received) and unreliable predictive capability (due to highly dynamic nature of emails). In this chapter, we discuss the challenging problems of Spam Recognition and then propose an anti-spam filtering framework; in which appropriate dimension reduction schemes and powerful classification models are employed. In particular, Principal Component Analysis transforms data to a lower dimensional space which is subsequently used to train an Artificial Neural Network based classifier. A cost-sensitive empirical analysis with a publicly available email corpus, namely Ling-Spam, suggests that our spam recognition framework outperforms other state-of-the-art learning methods in terms of spam detection capability. In the case of extremely

high misclassification cost, while other methods' performance deteriorates significantly as the cost factor increases, our model still remains stable accuracy with low computation cost.

1. Introduction

Email is widely accepted by the business community as a low cost communication tool to exchange information between business entities which are physically distant from one another. It minimizes the cost of organizing an in-person meeting. It is reported by a recent survey SurePayroll (Surepayroll, 2007), over 80% of small business owners believe email is a key to the success of their business and most people today spend between 20% to 50% of their working time using email, including reading, sorting and writing emails. Due to the very low cost of sending email, one could send thousands of malicious email messages each day over an inexpensive Internet connection. These junk emails, referred to as *spam,* can severely reduce staff productivity, consume significant network bandwidth and lead to service outages. In many cases, such messages also cause exposure to viruses, spyware and inappropriate contents that can create legal/compliance issues, loss of personal information and corporate assets. Therefore, it is important to accurately estimate costs associated with spam and evaluate the effectiveness of countermeasures such as spam-filtering tools. Though such spam prevention capability is implemented in existing email clients, there are some barriers that discourage users from utilizing this feature including error-prone and labor-intensive maintenance of filtering rules. Many researchers have developed different automatic spam detection systems but most of them suffer from low accuracy and high false alarm rate due to huge volume of emails, the wide spectrum of spamming topics and rapidly changing contents of these messages, especially in the case of high misclassification cost (Bayler, 2008). To deal with such challenges, this chapter proposes an anti-spam filtering framework using a highly performing *Artificial Neural Network* (ANN) based classifier. ANN is widely considered as a flexible "model-free" or "data-driven" learning method that can fit training data very well and thus reduce *learning bias* (how well the model fits the available sample data). However, they are also susceptible to the overfitting problem, which can increase generalization variance, i.e. making the predictive model unstable for unseen instances. This limitation can be overcome by combining ANN with a simple Linear Regression algorithm which makes the resulting classification model a stable semi-parametric classifier. Such model combination aims at stabilizing non-linear learning techniques while retaining their data fitting capability. Empirical analysis with the Ling-Spam benchmark confirms our superior spam detection accuracy and low computation cost in comparison with other existing approaches.

This chapter is organized as follow. Firstly, an overview of the spam problem is presented with associated negative impacts and protection techniques. This is followed by the application of Machine Learning (ML) to spam recognition, related works and details of the Ling-Spam corpus. Next, a brief review of several commonly used classification models and our proposed framework is given. The subsequent section compares the performance of our method with other learning techniques using the benchmark corpus under different cost scenarios. Finally, we provide some conclusion remarks for this chapter and future research directions.

2. Spam Recognition and Machine Learning techniques

2.1. Spam Recognition as a Challenging Task

2.1.1. Overview of Spamming

Spamming is one of the biggest challenges facing Internet consumers, corporations, and service providers today. Email spamming, also known as *Unsolicited Bulk Email* (UBE) or *Unsolicited Commercial Email* (UCE), is the practice of sending unwanted email messages, frequently with commercial content, in large quantities to an indiscriminate set of recipients (Schryen, 2007). Due to the increasing popularity and low cost of email, there are more and more spam circulating over the Internet. Spam emails were found to account for approximately 10% of the incoming message to corporate networks (L. F. Cranor & LaMacchia, 1998) and currently costs businesses US $ 13 billion annually (Swartz, 2003). Not only wasting time and consuming bandwidth, undesired emails are extremely annoying to most users due to their unsuitable contents, ranging from advertising vacations to pornographic materials.

Spam is usually classified into two categories which have different effects on Internet users. *Cancellable Usenet spam* is a single message sent to many Usenet newsgroups (Wolfe, Scott, & Erwin, 2004). This spamming attack can overwhelm the users with a barrage of advertising or other irrelevant posts. It also subverts the ability of system administrators and group owners to manage the topics they accept on their systems. The second type of spam is *Email spam* which targets individual users with direct mail messages (Bayler, 2008). Not only causing loss of productivity for email users, it also costs money for Internet Service Providers (ISP) and online services to transmit spam, and these costs are transferred directly to other subscribers. Though there are different types of spam, they all share some common properties. First, the sender's identity and address are concealed. Second, spam emails are sent to a large number of recipients and in high quantities. In fact, spam is economically viable to its senders not only because it is low cost to send an email, but also because spammers have no operating costs beyond the management of their mailing lists. This attracts numerous spammers and the volume of unsolicited mail has become very high. Finally, spam messages are unsolicited, that is, the individuals receiving spam would otherwise not have opted to receive it.

To successfully send a spam message, spammers usually undertake two steps: (1) collecting target email addresses and (2) bypassing anti-spam measures (Schryen, 2007). The later task involves cleverly disguising an unsolicited message as a non-spam message with normal appearing subject lines and other ways of getting around anti-spam software. The first task seems easier but in fact could be very challenging. To collect valid email addresses of potential target victims, the following techniques can be deployed (Bayler, 2008):

- Buying lists of addresses from some companies or other spammers.
- Harvesting email addresses from web sites or UseNet News posts with automated programs.
- Stealing users address books on compromised computers.
- Collecting addresses via Internet Relay Chat (IRC) programs.
- Guessing email addresses, then sending email to see if it goes through (Directory Harvest attacks).
- Using false reasons to trick a user into giving up their email address (Social Engineering attacks).

Though spam emails are troublesome, most of them can be easily recognized by human users due to their obvious signatures. For example, spam emails normally relate to specific topics such as prescription drugs, get-rich-quick schemes, financial services, qualifications, online gambling, discounted or pirated software. However, with a huge volume of spam messages received every day, it would not be practical for human users to detect spam by reading all of them manually. Furthermore, spam sometimes comes disguised, with a subject line that reads like a personal message or a non-delivery message. This makes highly accurate spam detection software desirable for encountering spam.

2.1.2 Impacts of Spamming and Preventive Techniques

Even though spam does not threaten our data in the same way that viruses do, it does cause businesses billions of lost dollars worldwide. Several negative impacts of spam are listed as follow (Schryen, 2007):

- Spam is regarded as privacy invasion because spammers illegally collect victim's email address (considered as personal information)
- Unsolicited emails irritate Internet users.
- Non-spam emails are missed and/or delayed. Sometimes, users may easily overlook or delete critical emails, confusing them with spam.
- Spam wastes staff time and thereby significantly reduce enterprises' productivity.
- Spam uses a considerably large bandwidth and uses up database capacity. This causes serious loss of Internet performance and bandwidth.
- Some spam contains offensive content.
- Spam messages can come attached with harmful code, including viruses and worms which can install backdoors in receivers' systems.
- Spammers can hijack other people's computers to send unwanted emails. These compromised machines are referred to as "zombie networks", networks of virus- or worm-infected personal computers in homes and offices around the globe. This ensures spammers' anonymity and massively increases the number of spam messages can be sent.

Various counter-measures to spam have been proposed to mitigate the impacts of unsolicited emails, ranging from regulatory to technical approaches. Though anti-spam legal measures are gradually being adopted, their effectiveness is still very limited. A more direct counter-measure is software-based anti-spam filters which attempt to detect spam from legitimate mails automatically. Most of the existing email software packages are equipped with some form of programmable spam filtering capability, typically in the form of blacklists of known spammers (i.e. block emails that come from a black list; check whether emails come from a genuine domain name or web address) and handcrafted rules (i.e. block messages containing specific keywords and unnecessary embedded HTML code). Because spammers normally use forged addresses, the blacklist approach is very ineffective. Handcrafted rules are also limited due to their reliance on personal preferences, i.e. they need to be tuned to characteristics of messages received by a particular user or groups of users. This is a time consuming task requiring resources and expertise and has to be repeated periodically to account for changing nature of spam messages (L. F. Cranor, LaMacchia, B.A. , 1998).

Spam detection is closely related to *Text Categorization* (TC) due to their text-based contents and similar tasks. However, unlike most TC problems, spamming is the act of blindly mass-mailing an unsolicited message that makes it spam, not its actual content (Schryen, 2007): any otherwise legitimate message becomes spam if blindly mass-mailed. From this point of view, spamming becomes a very challenging problem to the sustainability of the Internet, given the content of emails the only foundation for spam recoginition. Nevertheless, it seems that the language of current spam messages constitutes a distinctive genre, and that the topics of most current spam messages are rarely mentioned in legitimate messages, making it possible to train successfully a text classifier for spam recognition.

2.2. Machine Learning for Spam Recognition

Recent advances of *Machine Learning* (ML) techniques in *Text Classification* (TC) have attracted immense attention from researchers to explore the applicability of learning algorithms in anti-spam filtering (Bayler, 2008). In particular, a collection of messages is input to a learning algorithm which infers underlying functional dependencies of relevant features. The result of this process is a model that can, without human intervention, classify a new incoming email as spam or legitimate according to the knowledge collected from the training stage. Apart from automation which frees organizations from the need of manually classifying a huge amount of messages, this model can be retained to capture new characteristics of spam emails. To be most useful in real world applications, the anti-spam filters need to have a good generalization capability, that is, they can detect malicious messages which never occur during the learning process. There has been a great deal of research conducted in this area, ranging from simple methods such as propositional learner Ripper with "keyword-spotting rules" (Cohen, 1996) to more complicated approaches such as Bayesian networks using *bags of words* representation and binary coding (Sahami, 1998). In (Androutsopoulos, Koutsias, Chandrinos, Paliouras, & Spyropoulos, 2000), a system implementing Naïve Bayes and a k-NN technique is reported to be able to outperform the keyword-based filter of Outlook 2000 on the Ling-Spam corpus. Ensemble methods also prove their usefulness in filtering spam. For example, staked Naïve Bayes and k-NN can achieve good accuracy (Drucker, Wu, & Vapnik, 1999), and Boosted trees were shown to have better performance than individual trees, Naïve Bayes and k-NN alone (Carreras & Marquez, 2001). A support vector machine (SVM)(Drucker et al., 1999) is also reported to achieve a higher detection rate as well as lower false alarm rate for spam recognition compared with other discriminative classification methods. It is suggested that email headers play a vital role in spam recognition, and to get better results, classifiers should be trained on features of both email headers and email bodies (Androutsopoulos et al., 2000).

2.3 Ling-Spam Benchmark

The Ling-Spam corpus (Androutsopoulos et al., 2000) is used as a benchmark to evaluate our proposed algorithm with other existing techniques, the anti-spam filtering task. Using this publicly available dataset, we can conduct tractable experiments and also avoid complications of privacy issues. While spam messages do not pose this problem as they are blindly distributed to a large number of recipients, legitimate email messages may contain personal information and cannot usually be released without violating the privacy of their recipients and senders.

The corpus contains legitimate messages collected from a moderated mailing list on profession and science of linguistics and the spam messages collected from personal mailboxes:

- 2412 legitimate messages with text added by the list's server removed.
- 481 spam messages (duplicate spam messages received on the same day excluded)

The headers, HTML tags, and attachments of these messages are removed, leaving only the subject line and body text. The distribution of the dataset (16.6% is spam) makes it easy to identify legitimate emails because of the topic-specific nature of the legitimate mails. This dataset is partitioned into 10 stratified subsets which maintain the same ratio of legitimate and spam messages as in the entire dataset. Though some research (Androutsopoulos et al., 2000; Carreras & Marquez, 2001; Hsu, Chang, & Lin, 2003; Sakkis et al., 2003) has been conducted on this data showing their comparative efficiency, most of them suffer from high a false alarm rate which results in a degraded performance when the misclassification cost is high. Overcoming this problem is our major objective in this chapter.

3. Spam Recognition Methods

This section discusses commonly used learning algorithms for spam recognition problems.

3.1. Naïve Bayes

Naive Bayes is a well-known probabilistic classification algorithm which has been used widely for spam recognition (Androutsopoulos et al., 2000). According to Bayes' theorem, we can compute the probability $Prob(C = c|\vec{X} = \vec{x})$ that a message with vector $\vec{x} = \{x_1, \dots, x_n\}$ belongs to a class $c \in \{legit, spam\}$:

$$P\left(C = c\middle|\vec{X} = \vec{x}\right) = \frac{P(C = c).Prob(\vec{X} = \vec{x}|C = c)}{\sum_{k\in\{spam,legit\}} P(C = k).P(\vec{X} = \vec{x}|C = k)}$$

The calculation of $P(\vec{X} = \vec{x}|C = c)$ is problematic because most all novel messages are different from training messages. Therefore, instead of calculating probability for messages (a combination of words); we can consider their words separately. By making the assumption that $x_1, \dots, x_n$ are conditionally independent given the class c, we have:

$$Prob\left(C = c\middle|\vec{X} = \vec{x}\right) = \frac{P(C = c).\prod_{i=1}^{n} P(X_i = x_i|C = c)}{\sum_{k\in\{spam,legit\}} P(C = k).\prod_{i=1}^{n} P(X_i = x_i|C = k)}$$

3.2. Memory Based Learning

In (Androutsopoulos et al., 2000), an anti-spam filtering technique using Memory-Based Learning (MBL) that simply stores the training messages. The test messages are then classified by estimating their similarity to the stored examples based on their *overlap* metric which counts the attributes where the two messages have different values. Given two instances $\vec{x}_i = \{x_{i1}, \dots, x_{in}\}$ and $\vec{x}_j = \{x_{j1}, \dots, x_{jn}\}$, their overlap distance is:

$$d\left(\vec{x}_i, \vec{x}_j\right) = \sum_{r=1}^{n} \delta(x_{ir}, x_{jr})$$

Where $\delta(x, y) = 0$ if $x = y$ or *1* otherwise.

The confidence level that a message $\vec{x}$ belongs to a class c is calculated based on the classes of other neighbor instances $C(\vec{x}_i)$:

$$W_c(\vec{x}) = \sum \left(1 - \delta(c, C(\vec{x}_i))\right)$$

MBL's performance can be significantly improved by introducing some weighting schemes.

3.3. Distance Weighting

Depending on how far a test instance is away from its neighborhood, its' confidence level is estimated:

$$W_c(\vec{x}) = \sum \frac{1}{d^3(\vec{x}, \vec{x}_i)} \left(1 - \delta(c, C(\vec{x}_i))\right)$$

3.3.1 Attribute Weighting

Unlike the basic k-neighborhood classifiers where all attributes are treated equally, MBL assigns different weights to the attributes; depending on how well they discriminate between the categories, and adjust the distance metric accordingly. In particular, an attribute X_i has a weight of M_i which is the reduction of entropy *H(C)* (uncertainty on any category *C* of a randomly selected instance) and the expected value of entropy $H(C|X = x)$ (uncertainly on any category *C* given the value of attribute *X*). This means an attribute would have a higher weight if knowing its value reduces uncertainty on category *C*.

$$M_i = H(C) - \sum_{x \in \{0.1\}} P(X = x).H(C|X = x)$$

Where

$H(C) = \sum_{c \in \{spam,legit\}} P(C = c).log_2 P(C = c)$

$H(C|X = x) = \sum_{c \in \{spam,legit\}} P(C = c|X = x).log_2 P(C = c|X = x)$

The distance between two instances is recalculated as below:

$$d(\vec{x}_i, \vec{x}_j) = \sum_{r=1}^{n} M_r \delta(x_{ir}, x_{jr})$$

3.4. Boosted Decision Tree

Boosted Tree (BT) is a popular method implemented in many anti-spam filters with great successes (Carreras & Marquez, 2001). It uses the ADA-Boost algorithm (Schapire & Singer, 2000) to generate a number of Decision Tress classifiers which are trained by different sample sets drawn from the original training set. Each of these classifiers produces a hypothesis from which a learning error can be calculated. When this error exceeds a certain level, the process is terminated. A final composite hypothesis is then created by combining individual hypotheses.

3.5. Support Vector Machine

Support Vector Machines (SVM) (Drucker et al., 1999) have become one of the popular techniques for text categorization tasks due to their good generalization nature and the ability to overcome the curse of dimensionality. SVM classifies data by a set of representative support vectors. Assume that we want to find a discriminant function *f(x)* such that $y_i = f(x_i)$. A possible linear discriminant function can be presented as $f(x) =$

$sgn((w.x)+b)$ where $(w.x)+b=0$ is a separating hyperplane in the data space. Consequently, choosing a discriminant function is to find a hyperplane having the maximum separating margin with respect to the two classes. A SVM model is constructed by solving this optimization problem.

3.6. Artificial Neural Network

Artificial neural network (ANN) has gained strong interests from diverse communities due to its ability to identify the patterns that are not readily observable. Multi-Layer Perceptron (MLP) is the most popular neural network architecture in use today. This network uses a layered feed-forward topology in which the units each perform a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their output (Rumelhart & McClelland, 1986). Though many applications have implemented MLP for superior learning capacity, its performance is unreliable when new data is encountered. A recently emerging branch of ANN, the RBF networks, is also reported to gain great successes in diverse applications. In this chapter, MLP is implemented as typical ANN models for spam recognition.

4. Classification Framework for Spam Recognition

Although letting undetected spam pass through a filter is not as dangerous as blocking a legitimate message, one can argue that among one million incoming emails, a few thousand unsolicited message that are misclassified as normal is still very costly. Hence, anti-spam filters really need to be accurate, especially when they are used in large organizations. Advanced ML techniques have been used to improve performance of spam filtering systems. Amongst those methods, *Artificial Neural Network* (ANN) has been gaining strong interests from diverse communities due to its ability to identify the patterns that are not readily observable. Despite recent successes, ANN based applications still have some disadvantages such as extensive computation and unreliable performance. In this study, we use a Modified Probabilistic Neural Network (MPNN) which is developed by Zaknich (Zaknich, 1998).

If there exists a corresponding scalar output y_n for each local region (cluster) which is represented by a center vector $\underline{c}_i$, MPNN can be modeled as follow (Zaknich, 2003):

$$\hat{y}(\underline{x}) = \frac{\sum_{i=0}^{M} Z_i y_i f_i(\underline{x} - \underline{c}_i, \delta)}{\sum_{i=0}^{M} Z_i f_i(\underline{x} - \underline{c}_i, \delta)}$$

With Gaussian function $f_i(\underline{x}) = exp\frac{-(\underline{x}-\underline{c}_i)^T(\underline{x}-\underline{c}_i)}{2\delta^2}$

Where

$\underline{c}_i$= center vector for cluster i in the input space

y_i= scalar output related to $\underline{c}_i$

Z_i= number of input vectors x_j within cluster $\underline{c}_i$

δ = single smoothing parameter chosen during network training

M = number of unique centers $\underline{c}_i$

Though MPNN is reported to provide acceptable accuracy and affordable computation, it, just like other ANN, cannot classify reliably when an unusual input which differs from their training data emerges. As a result, it is essential that some degree of generalization capacity must be incorporated in the MPNN based classifiers. A possible approach to this problem is to incorporate MPNN with a linear model which offers stability, analyzability and fast adaptation (Hayes, 1996).

4.1 Description

Figure 1 shows the overall filtering framework proposed for spam recognition problem. There are 4 main phases.

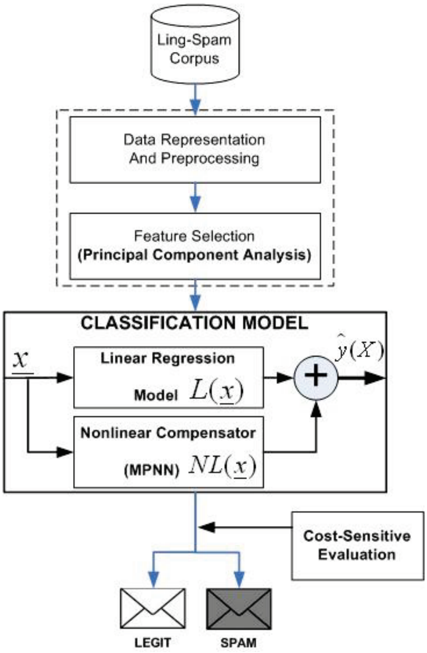

Ling-Spam Corpus
Data Representation And Preprocessing
Feature Selection (Principal Component Analysis)
CLASSIFICATION MODEL
$\underline{x}$
Linear Regression Model $L(\underline{x})$
Nonlinear Compensator (MPNN) $NL(\underline{x})$
+
$\hat{y}(X)$
Cost-Sensitive Evaluation
LEGIT
SPAM

Fig. 1. Proposed anti-spam filtering framework

4.1.1. Phase 1: Data Representation and Preprocessing

The purpose of data preprocessing is to transform messages in the mail corpus into a uniform format that can be understood by the learning algorithms. Features found in mails are normally transformed into a vector space in which each dimension of the space corresponds to a given feature in the entire corpus. Each individual message can then be viewed as a feature vector. This is referred to as the "*bag of words*" approach. There are two methods to represent elements of the feature vector: (1) *multi-variate presentation* assigns a binary value to each element showing that the word occurs in the current mail or not and (2) *multi-nomial presentation* represents each element as a number that shows the occurrence frequency of that word in the current mail. A combination of "bag of words" and multi-variate presentation is used in our experiments (the order of the words is neglected). To construct the feature vectors, the important words are selected according to their *Mutual Information (MI)* (Sakkis et al., 2003):

$$MI(X,C) = \sum_{x\in\{0,1\},c\in\{spam,legit\}} Prob(X=x,C=c).\,log_2 \frac{Prob(X=x,C=c)}{Prob(X=x).\,Prob(C=c)}$$

The words with the highest *MI* values are selected as the features. Assume that there are n features to be chosen, each mail will be represented by a feature vector $\vec{x} = \{x_1, \dots, x_n\}$ where $x_1, \dots, x_n$ are the values of binary attributes $X_1, \dots, X_n$, indicating the presence or absence of an attribute (word) in current message.

Moreover, w*ord stemming* and *stop-word removal* are two important issues that need to be considered in parsing emails. Word stemming refers to converting words to their morphological base forms (e.g. "gone" and "went" are reduced to root word "go"). Stop-word removal is a procedure to remove words that are found in a list of frequently used words such as "and, for, a". The main advantages of applying the two techniques are the reduction of feature space dimension and possible improvement on classifiers' prediction accuracy by alleviating the data sparseness problem(Androutsopoulos et al., 2000). The Ling-Spam corpus has four versions, each differs from each other by the usage of a *lemmatizer* and a *stoplist* (removes the 100 most frequently used words). We use the version with lemmatizer and stoplist enabled because it performs better when different cost scenarios are considered (Androutsopoulos et al., 2000). Words that appear less than 4 times or longer than 20 characters are discarded.

Also, it is found that phrasal and non-textual attributes may improve spam recognition performance (Androutsopoulos et al., 2000). However, they introduce a manual configuration phase. Because our target was to explore fully automatic anti-spam filtering, we limited ourselves to word-only attributes.

Finally, some data cleaning techniques are required after converting raw data into appropriate format. In particular, to deal with missing values, the simplest approach is to delete all instances where there is at least one missing value and use the remainder. This strategy has the advantage of avoiding introducing any data errors. Its main problem is that discard of data many damage the reliability of the resulting classifier. Moreover, the method cannot be used when a high proportion of instances in the training set have missing values. Together, these weaknesses are quite substantial. Although it may be worth trying when there are few missing values in the dataset, this approach is generally not recommended. Instead, we use an alternative strategy in which any missing values of a categorical attribute are replaced by its most commonly occurring value in the training set. For continuous attributes, missing values are replaced by its average value in the training set.

4.1.2. Phase 2: Feature Transformation

The tremendous growth in computing power and storage capacity has made today's databases, especially for text categorization tasks, contain very large number of attributes. Although faster processing speeds and larger memories may make it possible to process these attributes, this is inevitably a losing struggle in the long term. Besides degraded performance, many irrelevant attributes will also place an unnecessary computational overhead on any data mining algorithm. There are several ways in which the number of attributes can be reduced before a dataset is processed. In this research, a *dimension reduction* (also called *feature pruning* or *feature selection*) scheme called *Principal Component Analysis* (PCA) (Jolliffe, 2002) is performed on the data to select the most relevant features. This is necessary given the very large size and correlated nature of the input vectors. PCA eliminates highly correlated features and transforms the original data into lower dimensional data with most relevant features. From our observation, the selected features are words that express the distinction between spam and non-spam groups, i.e. they are either common in spam or legitimate messages, not in both. Several punctuation and special symbols (e.g. "$", "@") are also selected by PCA, and therefore, they are not eliminated during preprocessing.

4.1.3. Phase 3: Email Classification

The data after being processed by the Feature selection module is input to train the Classification Model. The resulting model is then used to label emails as either "legit" or "spam", indicating whether a message is classified as legitimate or a spam email. To implement the Classification Model, we propose an intelligent way of combining the linear part of the modeling with a simple non-linear model algorithm. In particular, MPNN is adapted in the nonlinear compensator which will only model higher ordered complexities while linear model will dominate in case of data far away from training clusters. It is described in the following equation.

$$\hat{y}(X) = L(\underline{x}) + NL(\underline{x}) = [w_0 + \underline{Wx}] + [\frac{\sum Z_i \alpha_i y_i f_i(\underline{x} - \underline{c_i}, \delta)}{\sum Z_i f_i(\underline{x} - \underline{c_i}, \delta)}]$$

Where
$L(\underline{x})$ = Linear Regression Model
$NL(\underline{x})$= Nonlinear Residual Compensator (MPNN)
w_0 = initial offset
w_i = weights of the linear model
$\alpha_i = \frac{y_{Ni}}{d_1}$ = Compensation factor
$y_{Ni} = y_i - [w_0 + W_i \underline{x_i}]$ = difference between the linear approximation and the training output
$d_i = ||\underline{x} - \underline{c_i}||$ = distance from the input vector to cluster *i* in the input space

The combination of linear regression model and MPNN is referred to as Linear Regression – Modified Probabilistic Neural Network (LR-MPNN). The piecewise linear regression model is firstly approximated by using all available training data in a simple regression fitting analysis. The MPNN is then constructed to compensate for higher ordered characteristics of the problem. Depending on different portions of the training set and how far the test data is away from the training data, the impact of nonlinear residual function is adjusted such that the overall Mean Square Error is minimized. This adjustment is formulated by the *compensation factor*α_i. In particular, $\alpha_i$is computed based on how well the linear model performs on a training examples and distances from a test vector to clusters of training data. Firstly, the goodness of the linear model $L(\underline{x})$ on a particular training data is measures by y_{Ni} which is defined as the difference between the linear approximation and the actual output of the training data. A very small value of y_{Ni} means that $L(\underline{x})$ fits the data well in this case and therefore it should have higher priority or the impact of the nonlinear model $NL(\underline{x})$ is minimized. In contrast, large value of y_{Ni} indicates that $NL(\underline{x})$ should compensate more for the degraded accuracy of $L(\underline{x})$.

Secondly, to determine how far a given test vector is away from the available training data, a distance d_i from that vector to each training cluster $\underline{c_i}$ is calculated. For any data which is far away from the training set, i.e. d_i is large, the value of α_i will be minimized. As the result, $NL(\underline{x})$will have minimal residual effect and $L(\underline{x})$ will dominate. This is because $L(\underline{x})$ has more stable generalization than $NL(\underline{x})$ for new instances.

4.1.4. Phase 4: Evaluation

To evaluate the overall performance of the framework, the Cost-sensitive Evaluation module computes several performance metrics and also takes into consideration different cost scenarios.

4.2. Performance Evaluation

4.2.1. Performance Measures

To measure the performance of different learning algorithms, the following measures are used:

$$SPAM\ RECALL\ (SR) = \frac{n_{S\to S}}{n_{S\to S} + n_{S\to L}}$$

$$SPAM\ PRECISION\ (SP) = \frac{n_{S\to S}}{n_{S\to S} + n_{L\to S}}$$

$$FAR(False\ Alarm\ Rate) = \frac{n_{S\to L}}{N_S}$$

$$Miss\ Rate(MR) = \frac{n_{L\to S}}{N_L}$$

From the above equations, Spam Recall (SR) is, in fact, the percentage of spam messages ($N_S = n_{S\to S} + n_{S\to L}$) that are correctly classified ($n_{S\to S}$) while Spam Precision (SP) compares the number of correct spam classifications ($n_{S\to S}$) to the total number of messages classified (correctly and incorrectly) as spam ($n_{S\to S} + n_{L\to S}$). As the Miss Rate (MR) increases, the number of misclassifications of legitimate emails increases while the False Alarm Rate (FAR) increases, the number of misclassifications of spam emails (passing from the filter) increases. Therefore, both of FAR and MR should be as small as possible for a filter to be effective (should be 0 for a perfect filter).

4.2.2. Cost-Sensitive Analysis

a) Cost Scenarios

Depending on what action is taken by a spam filter in response to a detected spam message, there are three major misclassification cost scenarios. The *no-cost* case is when the filter merely flags a detected spam message. This notification of spam does not risk losing any legitimate mail due to misclassification error (no misclassification cost), but it still takes time for the human users to check and delete the spam messages manually. To minimize the user efforts on eliminating spam, the filter can automatically detect and remove the suspicious messages. However, the total cost of misclassification in this case can be extremely high due to the seriousness of falsely discarding legitimate mails. This refers to the *high-cost* scenario.

Beside the above approaches, the filter may not either flag or completely eliminate the detected spam messages. Instead, it might resend the message to the sender. This approach, referred to as *moderate-cost*, combats spamming by increasing its cost via *Human Interactive Proofs* (HIP) (L. F. Cranor & LaMacchia, 1998). That is, the sender is required to give a proof of humanity that matches a puzzle before his message is delivered. The puzzles could be, for example, images containing some text that is difficult to automatically analyze by pattern recognition software. Alternatively, for anti-spam programs, simple questions (e.g. "what is one plus one") can be used instead of graphical puzzles.

The concept of HIP has been implemented in many security related applications. For example, certain web-based email systems use HIP to verify that password cracking software is not systematically brute-forcing to guess a correct password for email accounts. When a user types his password wrong three times, a distorted image is presented that contains a word or numbers and the user must verify before being allowed to continue. A human can easily convert the image to text, but the same task is extremely difficult for a computer. Some email client programs have anti-spam filtering heuristics using HIP implemented. When such programs receive an email that is not in the white-list of the user, they send the sender a password. A human sender can then resend the email containing the received password. This system can effectively defeat spammers because spam is *bulk*, meaning that the spammers do not bother to check replies manually or commonly use a forged source email address. The cost of creating and verifying the proofs is small, but they can be computationally impossible for automated mass-mailing tools to analyze. Though spammers can still use human labor to manually read and provide the proofs and finally have their spam message sent. HIP actually restricts the number of unsolicited messages that the spammer can send for a certain period of time due to the inability to use cheap automated tools (Carreras & Marquez, 2001). This barrier for spammers effectively introduces additional cost to sending spam messages.

In this chapter, spam recognition experiments are conducted in a cost-sensitive manner. As emphasized previously, misclassifying a legitimate message as spam is generally more severe than mistakenly recognizing a spam message as legitimate. Let $L \to S$ (legitimate classified as spam) and $S \to L$ (spam classified as legitimate) denote the two types of error, respectively. We invoke a decision-theoretic notion of cost, and assume that $L \to S$ is λ times more costly than $S \to L$. A mail is classified as spam if the following criterion is met (Androutsopoulos et al., 2000):

$$\frac{Prob(C = spam|\vec{X} = \vec{x})}{Prob(C = legitimate|\vec{X} = \vec{x})} > \lambda$$

In the case of anti-spam filtering:

$$Prob\big(C = spam\big|\vec{X} = \vec{x}\big) = 1 - Prob\big(C = legitimate\big|\vec{X} = \vec{x}\big)$$

The above criterion becomes:

$Prob\big(C = spam\big|\vec{X} = \vec{x}\big) > t$,with $t = \frac{\lambda}{1+\lambda}, \lambda = \frac{t}{1-t}$

Depending on which cost scenarios are considered, the value of λ is adjusted accordingly.

- No-cost scenario (e.g. flagging spam messages): $\lambda = 1$
- Moderate-cost scenario (e.g. semi-automatic filter which notifies senders about blocked messages): $\lambda = 9$
- High-cost scenario (e.g. automatically removing blocked messages): $\lambda = 999$

b) Total Cost Ratio

Accuracy and error rates assign equal weights to the two error types ($L \rightarrow S$, $S \rightarrow L$) and are defined:

$Acc = \frac{n_{L\to L}+n_{S\to S}}{N_L+N_S}$ $\qquad Err = \frac{n_{L\to S}+n_{S\to L}}{N_L+N_S}$

However, in the cost-sensitive contexts, the accuracy and error rates should be made sensitive to the cost difference, i.e. each legitimate message is counted for λ times. That is, when a legitimate message is misclassified, this counts as λ errors; and when it passes the filter, this counts as λ successes. This leads to the definition of *weighted accuracy* and *weighted error* (*WAcc and WErr*):

$$WAcc = \frac{\lambda.n_{L\to L}+n_{S\to S}}{\lambda.N_L+N_S} \qquad WErr = \frac{\lambda.n_{L\to S}+n_{S\to L}}{\lambda.N_L+N_S}$$

The values of performance measures (weighted or not) are misleadingly high. To get a true picture of the performance of a spam filter, its performance measures should be compared against those of a "baseline" approach where no filter is used. Such a baseline filter never blocks legitimate messages while spam emails always pass through the filter. The weighted accuracy and error rates for baseline are:

$$WAcc^b = \frac{\lambda.N_L}{\lambda.N_L+N_S} \qquad WErr^b = \frac{N_S}{\lambda.N_L+N_S}$$

Total cost ratio (*TCR*) is another measure which evaluates performance of spam filter to that of a baseline.

$$TCR = \frac{WErr^b}{WErr} = \frac{N_S}{\lambda.n_{L\to S}+n_{S\to L}}$$

Greater *TCR* values indicate better performance. For *TCR* < 1, the baseline is better. If cost is proportional to wasted time, a *TCR* is intuitively equivalent to measuring how much time is wasted to manually delete all spam messages when the filter is used (N_S) compared to the time wasted to manually delete any spam messages that passed the filter ($n_{S\to L}$) plus the time needed to recover from mistakenly blocked legitimate messages ($\lambda.n_{L\to S}$)

5.Experiment Results

5.1. Experiment Design

The proposed spam recognition framework is tested on the Ling-Spam corpus to compare with other existing learning methods including Naïve Bayes (NB), Weighted Memory Based Learning (WMBL), Boosted Trees (BT), Support Vector Machine (SVM) and Neural Network models (Multilayer Perceptron – MLP. Unlike other text categorization tasks, filtering spam messages is cost sensitive (Cohen, 1996), hence evaluation measures that account for misclassification costs are used. In particular, we define a cost factor λ with different values corresponding to three cost scenarios: first, no cost considered ($\lambda = 0$) e.g. marking messages as spam; second, semi-automatic filtering ($\lambda = 9$) e.g. issuing a notification about spam; and fully automatic filtering ($\lambda = 999$), e.g. discarding the spam messages.

The rate at which a legitimate mail is misclassified as spam is calculated by False Alarm Rate (*FAR*) and it should be low for a filter to be useful. Spam Recall (*SR*) measures the *effectiveness* of the filter, i.e. the percentage of messages correctly classified as spam, while Spam Precision (*SP*) indicates the filter's *safety*, i.e. the degree to which the blocked messages are truly spam. Because *SR* can be derived from *FAR* (e.g. *FAR = 1 – SR*), we will use *SR*, *SP*, and Total Cost Ratio (TCR) for evaluation. Besides comparing how accurately the filters perform, their computation is also measured using the computation time (in seconds) required for each classifier. Particularly, the total computation time is a summation of the time that a classifier needs to perform cross validation, testing on data and to calculate the relevant performance metrics (e.g. misclassification rate, accuracy …).

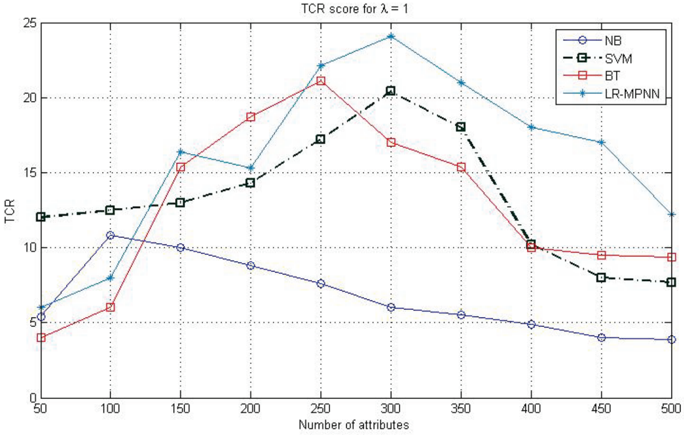
Stratified tenfold cross validation is employed for all experiments. That is, the corpus is partitioned into 10 stratified parts and each experiment was repeated 10 times, each time reserving a different part as the testing set and using the remaining 9 parts as the training set. Performance scores are then averaged over the 10 iterations.

In addition to the studies conducted by other researchers on the same Ling-Spam corpus (NB (Androutsopoulos et al., 2000), WMBL (Sakkis et al., 2003), SVM (Hsu et al., 2003), BT (Carreras & Marquez, 2001)), we also reproduced their experiments (based on the average value of *TCR* of three cost scenarios) to confirm and determine the parameters' values that give best performance for different learning methods. The optimal attribute size of these methods can be found in Figure 2. An MLP wth 15 neurons in hidden layer is deployed using the Matlab Neural Network toolbox.

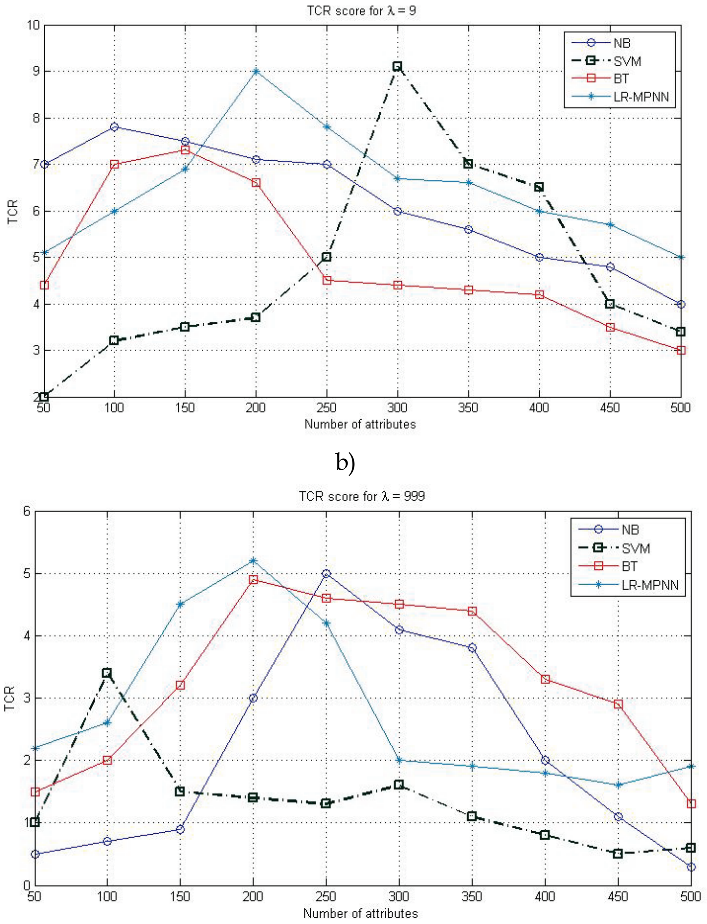
5.2. Experiment Result

5.2.1. TCR and Attribute Selection

From Figure 2, for $\lambda = 1$ and $\lambda = 9$, most of filters demonstrate a stable performance, with TCR constantly greater than 1. These filters differ from one another in terms of their sensitivity on attribute selection and the number of attributes which give maximum TCR. Our LR-MPNN is found to be moderately sensitive to attribute selection and it obtains the highest TCR for $\lambda = 1$ with 300 attributes selected. When $\lambda = 9$, LR-MPNN achieves very competitive TCR compared to SVM but with less number of attributes (200 attributes) and hence involves less computation overheads.



a)



b)

c)

Fig. 2. TCR score of spam recognition methods

For $\lambda = 999$, all classifiers have their TCR reduced significantly for the effect of very high misclassification cost. The difference between low and high values of misclassification cost λ is the increased performance of the baseline filter when λ increases. That is, without a filter in use (baseline), all legitimate mails are retained, preventing the baseline from misclassifying those legitimate mails as spam. Therefor, large λ benefits the baseline and make it hard to be defeated by other filters. Recall that TCR is the measure of performance that a filter improves on the baseline case. As a result, TCR generally reduces when λ increases. Another important observation is that, the performance of most classifiers, except for BT and LR-MPNN, fall below the base case (TCR<1) for some numbers of selected attributes. This is due to the relative insensitivity of BT and LR-MPNN to attribute selection. In this case, the LR-MPNN is considered to be the best performing filter with the highest TCR.

5.2.2. Spam Precision and Spam Recall

In this experiment, the classifiers are run iteratively by a tenfold cross-validation process. The SP ans SR rates are recorded in Table 1. We observe that, for the no-cost scenario (λ = 1), our method, LR-MPNN, is found to have best SP while its SR (0.958) is very similar to the highest SR of NB (0.959). For λ = 9, LR-MPNN obtains the highest SR (0.869) and second highest SP (0.991) after BT algorithm. Finally, in the case of extremely high misclassification cost (λ = 999), LR-MPNN significantly outperforms other methods with all evaluation metrics are of highest values.

Method	λ =1		λ =9		λ =999	
	SR	**SP**	**SR**	**SP**	**SR**	**SP**
NB	**0.959**	0.973	0.861	0.975	0.790	0.984
WMBL	0.860	0.917	0.790	0.982	0.601	0.857
SVM	0.954	0.981	0.847	0.983	0.671	0.995
BT	0.957	0.980	0.864	**0.993**	0.768	0.996
MLP	0.852	0.975	0.782	0.977	0.623	0.979
LR-MPNN	*0.958*	***0.986***	***0.869***	*0.991*	***0.793***	***0.998***

Table 1. Precision/Recall evaluation on Ling-Spam data

5.2.3. Computational Efficiency

Apart from comparing precision, recall and TCR scores between classifiers, we also measure their computational efficiency. Table 2 shows that WMBL had the minimum computation time (2.5 mins), followed by NB, LR-MPNN, SVM, MLP, BT respectively. LR-MPNN can achieve comparative spam precision and recall with a shorter computation time (3.5 mins) compared with BT (11.5 mins) and SVM (5 mins). Moreover, considering TCR scores, the models that require less time (WMBL, NB) than LR-MPNN do not perform as accurately as LR-MPNN.

Method	Computation Time (mins)	λ =1	λ =9	λ =999
		TCR	TCR	TCR
NB	3	10.80	7.80	5.02
WMBL	**2.5**	7.11	5.62	1.33
SVM	5	20.47	9.11	3.42
BT	11.5	21.18	7.35	4.91
MLP	7	12.20	4.50	0.25
LR-MPNN	3.5	24.17	8.99	5.25

Table 2. Computation Time, Memory size evaluation on Ling-Spam data

In summary, the most important finding in our experiment is that the proposed LR-MPNN model can achieve very accurate classification (high TCR, SP, SR) compared to other conventional learning methods. Such superior performance of LR-MPNN was observed most clearly for λ = 999 though it always obtains the highest TCR and very competitive SP, SR rates for other cases of λ. Our algorithm also requires relatively small computation time to obtain comparable or even higher predictive accuracy to other methods.

6. Conclusions and Future Work

In this chapter, we proposed a novel anti-spam filtering framework in which appropriate dimension reduction schemes and powerful classification models are employed. Particularly, Principal Component Analysis transforms data to a lower dimensional space. At the classification stage, we combine a simple linear regression model with a lightweight nonlinear neural network in an adjustable way. This learning method, referred to as *Linear Regression Modified Probabilistic Neural Network* (LR-MPNN), can take advantage of the virtues of both. That is, the linear model provides reliable generalization capability while the nonlinear can compensate for higher order complexities of the data. A cost-sensitive evaluation using a publicly available corpus, Ling-Spam, has shown that our LR-MPNN classifier compares favorably to other state-of-the-art methods with superior accuracy, affordable computation and high system robustness. Especially for extremely high misclassification cost, while other methods' performance deteriorates as λ increases, the LR-MPNN demonstrates an absolutely superior outcome but retains low computation cost. LR-MPNN also has significant low computational requirement, i.e. its training time is shorter than other algorithms with similar accuracy and cost. Though our proposed model achieves good results in the conducted experiments, it is not necessarily the best solution for all problems. However, comparatively high predictive accuracy along with low computational complexity distinguish it from other state-of-the-art learning algorithms, and particularly suitable for cost-sensitive spam detection applications.

7. References

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G., & Spyropoulos, C. D. (2000). An Evaluation of Naive Bayesian Anti-Spam Filtering. Paper presented at the Proc. of the ECML.

Bayler, G. (2008). Penetrating Bayesian Spam Filters: VDM Verlag Dr. Mueller e.K.

Carreras, X., & Marquez, L. (2001). Boosting Trees for Anti-Spam Email Filtering. Paper presented at the RANLP, Tzigov Chark, Bulgaria.

Cohen, W. (1996). Learning rules that classify email. AAAI Sump. On Machine Learning in Inf. Access, 18-25.

Cranor, L. F., & LaMacchia, B. A. (1998). Spam! Paper presented at the Communications of ACM.

Cranor, L. F., LaMacchia, B.A. . (1998). Spam! Paper presented at the Communications of ACM.

Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support Vector Machines for Spam Categorization. IEEE Transactions On Neural Networks, 1048-1054.

Hayes, M. H. (1996). Statistical Digital Signal Processing and Modeling: John Wiley & Sons, Inc.

Hsu, C. W., Chang, C. C., & Lin, J. C. (2003). LIBSVM: a library for support vector machines. from http://ww.csie.ntu.edu.tw/~cjlin/libsv

Jolliffe, I. T. (2002). Principle Component Analysis (2 ed.). New York, USA: Springer.

Rumelhart, D. E., & McClelland, J. L. (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition (Vol. 2): The MIT Press. .

Sahami, M., S. Dumais, D. Heckerman, and E. Horvitz. (1998). A Bayesian Approach to Filtering Junk E-Mail. Paper presented at the Learning for Text Categorization – AAAI Technical Report WS-98-05.

Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., & Stamatopoulos, P. (2003). A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists. Paper presented at the Information Retrieval.

Schapire, R. E., & Singer, Y. (2000). BoosTexter: a Boosting-Based System for Text Categorization. .Machine Learning, 39(2), 135–168.

Schryen, G. (2007). Anti-Spam Measures: Analysis and Design: Springer.

Surepayroll. (2007). More than 80 Percent of Small Business Owners Consider E-mail Essential to Success, SurePayroll Insights Survey.

Swartz, N. (2003). Spam costs businesses $13 billion annually. Information Management Journal, 37(2).

Wolfe, P., Scott, C., & Erwin, M. (2004). Anti-Spam Tool Kit: McGraw-Hill Osborne Media.

Zaknich, A. (1998). Introduction to the modified probabilistic neural network for general signal processing applications. IEEE Transactions on Signal Processing, 46(7), 1980-1990.

Zaknich, A. (2003). Neural Networks for Intelligent Signal Processing. Sydney: World Scientific Publishing.

29

Designing and Training Feed-Forward Artificial Neural Networks For Secure Access Authorization

Fadi N. Sibai, Aaesha Shehhi, Sheikha Shehhi,
Buthaina Shehhi and Najlaa Salami
UAE University
United Arab Emirates

1. Introduction

Password-based authorization is a key security feature to gain access to accounts, files, and like PINs, can be used to control access to secure rooms, cabinets, electronic equipments, control panels, and other valuables. It relieves employees from carrying physical keys or smart cards and does not require the use of more expensive biometric devices. Memorizing a username or user number of ID and a matching password is sufficient to get access to the controlled computer account, file, building area or equipment. Typically, a key pad is required to key in the user ID followed by the password, non-volatile storage to record and save the matching (user ID, password) combination, and some controller for controlling the password detection operation. The user ID and passwords entered by the user via the key pad are latched (L) and the memorized (user ID, password) pairs on record are retrieved from storage, and then compared with the (user ID, password) combination entered by the user via the key pad. If there is a match the user is given access, otherwise the reading of saved (user ID, password) combinations on record in the storage are read one by one until a match is found, or the list of pairs are exhausted in the storage and the user is denied access. Other alternative implementations are possible. The major weakness of such digital system implementations is the possibility to steal saved (user ID, password) pairs by probing the bus connecting the storage to the data path containing the comparator as depicted in Fig. 1. Such insecurity can be avoided by encrypting the (user ID, password) pairs prior to saving them in storage, and then decrypting them after reading them from storage and prior to performing the comparison operation. However this requires either more programmable and fast hardware which executes encryption and decryption in software or a dedicated encrypt/decrypt hardware engine. If the encryption scheme’s keys are compromised, then this probing technique can still reveal the matching (user ID, password) combinations. Another security vulnerability of storing the user ID and passwords combinations in a table saved in storage is that even though the user IDs and passwords may be encrypted, the hacker may modify table entries or create new table entries with fake user ID and password combinations.

An alternative is to employ an artificial neural network circuit which “memorizes” the matching (user ID, password) combinations and generates the matching signal as a result of presenting the user-entered (user ID, password) pair via the key pad. Such an artificial neural network achieves this operation by being trained with a set of (user ID, password, output) triplets where the output is 1 when the password matches the user ID, and 0 otherwise. In the training phase, the artificial neural network refines its link weights to more closely match the outputs of each training line. In the training set, the majority of the triplets will have an output of 0 (non-matching) and only the matching combination of user ID and password will have an output of 1. Such an artificial neural network has the advantage that it can be implemented on a single chip and does not require the matching (user ID, password) combinations to be stored on external storage, but requires that the network weights be stored. While these can be stored on-chip making the information stealing attempt very difficult, the weights can be still be stored externally and retrieved from external storage. If the weights are stolen while being read from external storage by probing, this stolen information is of little value as it makes the process of deriving the matching (user ID, password) pair very difficult. Indeed, the matching (user ID, password) pair is “memorized” and integrated in to the neural network’s weights and without knowing the internal organization of the network (e.g. whether feedforward, presence of feedback loops, number of network layers and number of neurons in each layer) and the exact order of the weights and which neural link they correspond to, the process of extracting the matching (user ID, password) pair is virtually exhaustive.

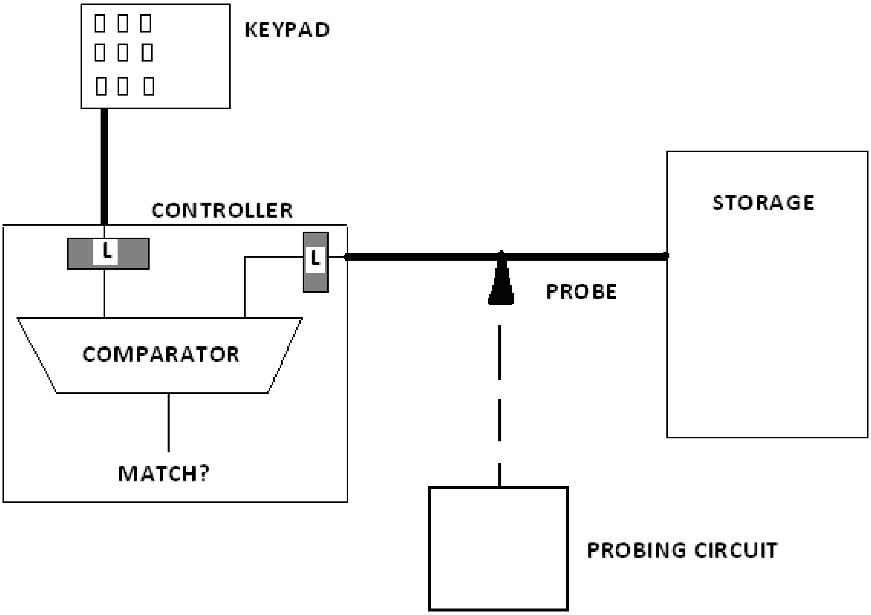


Fig. 1. Insecure password detection digital system

In this Chapter, we present our research work on feed-forward artificial neural networks for secure access authorization. In Section 2, we briefly review artificial neural networks. In

Sections 3 and 4, we review previous work on this subject and present our artificial neural network organization, respectively. In Section 5, we present our data coding for binary and analog inputs. Our network's training, simulation and testing is then described in Section 6. The accuracy results of all the variation of trained and tested networks are then presented in Section 7. Section 8 concludes the Chapter.

2. Artificial Neural Networks

Artificial neural networks model biological neural networks in the brain and have proven their effectiveness in a number of applications such as classification and categorization, prediction, pattern recognition and control. An artificial neural network consists of an interconnected group of artificial neurons. Such a network performs computation and manipulates information based on the connectionist approach in a similar but simpler fashion than the brain would perform. Many types of artificial neural networks (Faussett, 1994) exist including feed-forward neural networks, radial basis function (RBF) networks, Kohonen self-organizing networks, recurrent networks, stochastic neural networks, modular neural networks, dynamic neural networks, cascading neural networks, and neuro-fuzzy networks. Multi-Layer Perceptron (Rumelhart & Mclellan, 1986; Haykin, 1998) (MLP) are perhaps the most popular, where neurons in a feed-forward type network perform a biased weighted averaging of their inputs and this sum is then subjected to a transfer function, in order to limit the output value.

As depicted in Fig. 2, a neuron is made of a cell body bordered by a membrane, inside of which is a nucleus, across which incoming electric (or chemical) signals composed of polarised ions arrive via neuron inputs known as dendrites. The neuron output or outgoing signal travels over a connector or terminal known as axon --where the neurotransmitters reside – and which connect to other neuron dendrites via synapses. Ten thousand of neuron types are known to exist in the brain of different shapes and terminal densities. A neuron may have thousands of dendrites but only one axon. A neuron output –axon—connects to another neuron input –dendrite—in an area called a synapse where the axons terminate. When enough positive ions gather inside the neuron's membrane the neuron fires, i.e. a large electric signal is generated and travels out over the axon to reach the axon terminal. At electric synapses, the output is the electrical signal transmitted over the axon while at chemical synapses, the output is a neurotransmitter. Neurons are either sensory, motor or inter-neuron. The first type conveys sensory information. The second type conveys motor information. The third type conveys information between neurons.

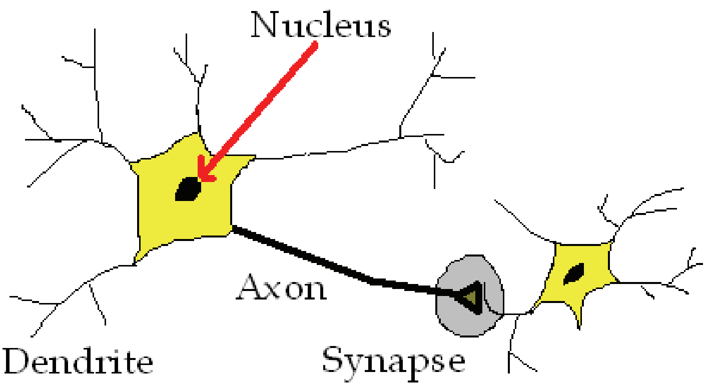


Fig. 2. Structure of a neuron network in the brain

An artificial neuron models a real neuron as depicted in Fig. 3. First, electric signals from other neurons are multiplied by weights (represented by the rectangles in Fig. 3) and then are input into the artificial neuron. The weighted signal values are them summed by an adder ("Σ" in Fig. 3) and the sum is subjected to a transfer function ("T" in Fig. 3) which is one of: i. linear, where the output is proportional to the weighted sum of inputs; ii. threshold, where the output is one of two values based on whether the weighted sum is greater or smaller than the threshold value; and iii. sigmoid, a non-linear function which most closely mimicks real neurons. Artificial neural networks are composed of several artificial neurons as a real neuron network is composed of many real neurons. Artificial neural networks come in different forms and shapes.

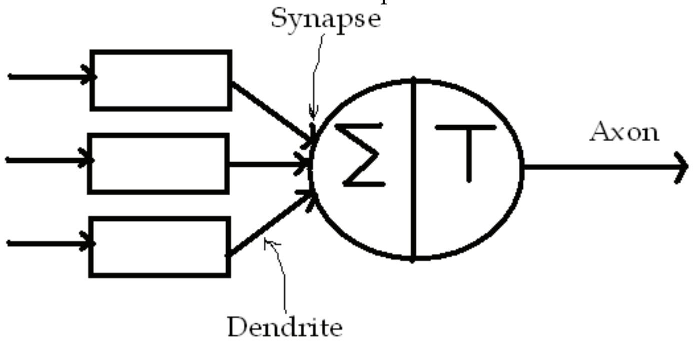


Fig. 3. Artificial neuron model

Artificial neural network organizations are either feedforward, or recurrent. Feedforward networks do not have cycles and signals flow from input to output. Recurrent neural networks have loops, i.e. links to neurons in the same or previous layers. The MLP neural network organization, shown in Fig. 4, is an example of feedforward network. Hebbian networks are example of recurrent networks. Recurrent networks because of their feedback cycles go through dynamic state changes until the network stabilizes. Neural networks can be either fixed or adaptive. Fixed networks have unchanging weights usually set at the start and need not change as the network is expected to keep operating in the same way. Adaptive neural networks as those which allow their weights to change (i.e. allow the network to adapt and learn). Adaptive neural networks can therefore learn and their learning usually falls under two large classes: supervised or unsupervised. Supervised learning involves a supervisor who tells the network how to learn, i.e. what the network output should be for each input combination. In the supervised learning, the inputs and corresponding outputs are known, so the neural network learns by applying a "cost" function to generate the desired outputs for each input combination. Popular cost functions include the mean squared error function, and random functions. The mean squared error function attempts to minimize the error between the actual output value computed by the network and the desired output value. In unsupervised learning, the network is not told what the generated output should be for each input combination but the neural network learns by itself by self-organizing the data and identifying the data's characteristics and properties. Yet another class of learning is reinforcement learning. Reinforcement learning differs from the supervised learning problem in that correct data inputs and matching output are never presented to the network. In addition, sub-optimal actions are not explicitly corrected. Instead, in reinforcement learning, agents interact with the environment and supply the data to the network which attempts to formulate a policy for agent actions which optimizes some cost function. Evolutionary algorithms and simulated annealing are examples of other types of neural network training methods.

The MLP is an example of feedforward artificial neural network with multiple layers and where each neuron output in one layer feeds as input to the neurons in the next layer as shown in Fig. 4. A radial basis function (RBF) neural network, pictured in Fig. 5, is a 3-layer network where the output is the weighted basis function (usually Gaussian function) of the Euclidian distance of the input vector and the neuron's center vector.

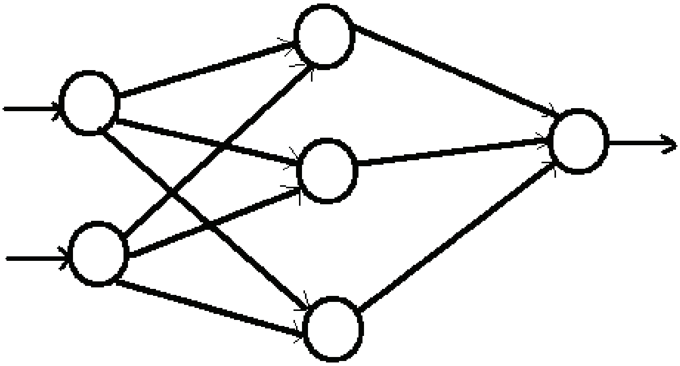


Fig. 4. MLP neural network

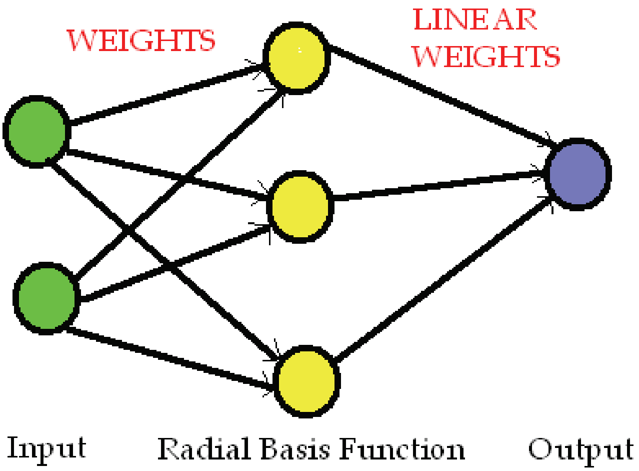


Fig. 5. RBF neural network

We chose our artificial neural network for secure password detection of the feed-forward type due to its simplicity and its suitability for this application.

We also employ the backpropagation algorithm for supervised training our network, a well known and widely used algorithm. The training algorithm minimizes the error between the obtained output and the required target output by finding the lowest point or minimum in the error surface. Starting with initial weights and thresholds, the training algorithms looks for the global minimum of the error surface. Usually the slope of the error surface at the present location and guides the next move down. It is the goal of the training algorithm to reach a low point on the surface which happens to be the local minimum, and in some unlucky situations, the algorithm stops at a local minimum. In the backpropagation algorithm, training moves the state along the steepest descent, reducing in each training epoch the error. The size of the move during each epoch is a function of the learning rate and the gradient (slope). At the end of each epoch, the weights are adjusted as a function of the error and the error surface gradient. The derivatives of the weights are computed to determine how the error changes as a result of increasing or decreasing the weights. The number of epochs taken to conduct training is determined by the error reaching a satisfactory value, or when the number of epochs reaches a predetermined number. Alternatively, training ends when the error stops shrinking.

Other training algorithms exist such as conjugate gradient descent --where after minimizing along one direction on the error surface, next moves assume that the second derivative along the first direction is held at zero-- and Quasi-Newton training (Bishop, 1995).

3. Previous work

Previous work on the subject include the work of (Reyhani & Mahdavi, 2007) who used a Radial Basis Function neural network and hash the (user ID, passwords) with a one-way hash function and then encrypt them. The authors claim that RBF trains much faster than with backpropagation learning (30x for a 200-sample training set and user ID and password hashed to 13 characters) as RBF networks allow the selection of parameters for the hidden layer without the need for their optimization. Other recent work on the subject can be found in (Ciaramella et al, 2006) and (Wang & Wang, 2008) who used a Hopfield network, and (Li, Lin, & Hwang, 2001) who used a multilayer neural network with back-propagation training. (Reyhani & Mahdavi, 2007) and (Wang & Wang, 2008) both proposed authentication based on neural networks. Both (Reyhani & Mahdavi, 2007) and (Wang & Wang, 2008)'s solution are composed of two phases: a registration phase and a authorization phase. (Wang & Wang, 2008) allows a 3rd phase to allow subsequent password change.

In the registration phase, the neural network is trained to recognize a (user ID, password) combination. In both (Reyhani & Mahdavi, 2007) and (Wang & Wang, 2008), the user ID and passwords supplied by the user are encrypted before been fed to the neural network. (Wang & Wang, 2008) additionally sparsely-encodes the encrypted data with Reed Solomon code.

In the authorization phase, the user supplies the user ID and a password which are then encrypted and subsequently fed as input to the neural network. The network then processes this data and generates an output consisting of the encrypted password. If the two encrypted passwords --one generated by the neural network and the other supplied by the user in unecrypted form—match, then the password entered by the user matches the user ID supplied by the user and the access is authorized. Otherwise, there is no match between the user-supplied ID and password and the access is denied.

In both (Reyhani & Mahdavi, 2007) and (Wang & Wang, 2008), after neural network processing completes, the system must compare the ANN-generated password to the one provided by the user and determine if there is a match or not. This increases system vulnerability as the password generated by the neural network must be transferred to hardware which performs comparison. Although the password is encrypted, the password can be decrypted if the key is compromised, or the encrypted password can be saved on external storage and injected in the circuit (e.g. input of comparison hardware) in the future to gain illegal access.

A better approach which we employ is to let the neural network perform the comparison itself without the need for the neural network to generate a password and then transfer it to a comparator. In this approach, the neural network "memorizes" the combination of user ID and password during training and indirectly "codes" that information in its weights by enforcements and inhibitions and directly and internally performs the comparison with the user-supplied password without the need to transfer the 2 passwords for comparison by an entity external to the neural network. In that case, the output neuron generates a 1 for a (user ID, password) match or access authorization, or a 0 for access denial. In this neural network-based approach, the table of weights has to store the weights of the neural network and store new ones when the password is changed or new users are given permission to access the protected and secured valuables. Yet, the weights may not mean much to any one if the topology of the neural network (type of network, number of layers, number of neurons per layer, …) and weight format (order or weights and exact mapping of stored values into each weight variable) is kept secret. In addition as in (Reyhani & Mahdavi, 2007) and (Wang & Wang, 2008), the user ID and password combination can be immediately encrypted before being fed to the neural network and the neural network can be trained with those encrypted values to match the encrypted password. This is valuable if the distance between the keypad and the neural network is long necessitating transfer by Ethernet cables or other communication cables. Yet in either case whether the keypad-to-neural network distance is long or short (both housed in same box), the password can be compromised if the box is bugged with a hardware sniffer to record and transmit keystrokes. One effective solution to this problem is to combine passwords or PINs with biometric data for a larger solution cost. Biometric techniques can render useless sniffing techniques employing vibration sensing (resulting from keystroke) or monitoring ground lines but also have their weaknesses. In the remainder of this section, we will assume that the keypad is safe and that keystrokes are neither transmitted nor recorded.

(Reyhani & Mahdavi, 2007) and (Wang & Wang, 2008) report that the training time of multi-layer networks is long. When a new user ID and password combination requires recognition from the neural network, the neural network must be retrained again. However although RBF and Hopfield neural networks can be trained faster than multi-layer feedforward networks, the training time of multi-layer feedforward networks is often acceptable.

4. Artificial Neural Network Organization

In our artificial neural network for secure password detection, we chose the number of neuron layers to be 3, one for the input layer neurons which hold the input values, one for the hidden layer neurons which hold some of the key network's functionality, and one output layer neuron to generate the match/no match result. We later experimented with the number of hidden layer to determine which number of hidden layers results in the best match accuracy.

The number of neurons in the input layer is determined by i. the sizes (in characters) of the user ID and password; ii. the coding type of the user ID and password information, whether with binary values or with analog ones.

The number of neurons in the middle layer is also affected by the above network attributes as well as the connections between the input layer neurons and the middle layer numbers, their number and their origins and destinations.

In our experiment, we have chosen to limit our user ID and password each to 5 characters each, where each character is the A-H letter range, where the case is ignored, meaning that upper and lower case versions of the same letter are identical for all practical purposes. Thus each user ID has 1 matching password out of 85=32,768 possible password combinations of 8 letters.

Fig. 6 shows our artificial neuron organization assuming that each input user ID and password character is digitally coded by three ($\log_2 8$) binary values. Note that the top row represents the 15 input neurons each representing one of the 3x5=15 input user ID binary values (3 neurons per character), and the 15 input neurons below the top row represent the 15 input password binary values in the same input layer but drawn below for better presentation. The hidden layer shows 30 neurons, as many as input layer neuron, with each input layer neuron feeding into each of the hidden layer neurons. The bottom row shows the single output layer neuron fed from each neuron in the hidden layer.

When the user ID and password characters are represented by analog values, each analog value can be input to a single neuron, so the version of Fig. 6 with analog data values contains 10 neurons in the input layer, 10 in the hidden layer and 1 in the output layer. Again, all input layer neurons feed into each of the 10 hidden layer neurons which feed in turn into the single output layer neuron.

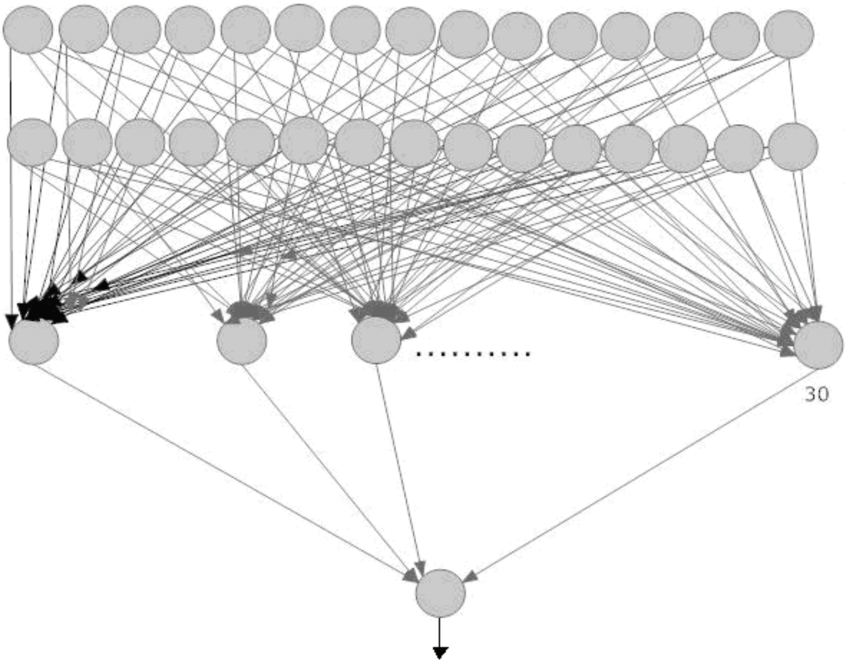


Fig. 6. Artificial neural network organization.

5. Data Coding

In the analog coding of the data, each input user ID and password character takes on a real number according to Table 1. Thus character "A" or "a" takes any value between 0 and 0.4. "B" is represented by all values in the range 0.5-0.9 and so on. Numbers in the boundary gaps, e.g. 0.4-0.5, are not intended to represent any useful characters.

Character	Ranges
A	0 - 0.4
B	0.5 - 0.9
C	1.0 – 1.4
D	1.5 – 1.9
E	2.0 – 2.4
F	2.5 – 2.9
G	3.0 – 3.4
H	3.5 – 3.9

Table 1. Real number coding

In the binary data coding version, each input user ID and password character in the range A-H is represented by 3 bits according to the coding convention of Table 2.

One major strength of real number (analog) coding is its cost saving as it represents each input character by a single neuron. Its major weakness is that characters at the bottom such as “G′ and “H” are assigned larger values than characters on the top such as “A” and “B”. This is problematic because input values are multiplied by link weight values before being summed by the 2nd and 3rd layer neurons. Thus inputs with larger values result in larger products (than inputs with smaller values) which could be wrongly misinterpreted by the circuit as a strong match, and must be inhibited by the next layers in order to reduce their message strengths. Also smaller products are diluted along large product values when summed by a neuron.

This weakness is avoided in the binary coding where the character is represented by three independent and separate bits. The binary coding also removes the fuzziness between analog code boundaries such as 0-0.4 and 0.5-0.9 above, as values which fall into the boundary areas 0.4-0.5 could be either interpreted as an “A” or a “B.” With 0 and 1 digits, the likelihood of getting into these kind of debates is greatly reduced.

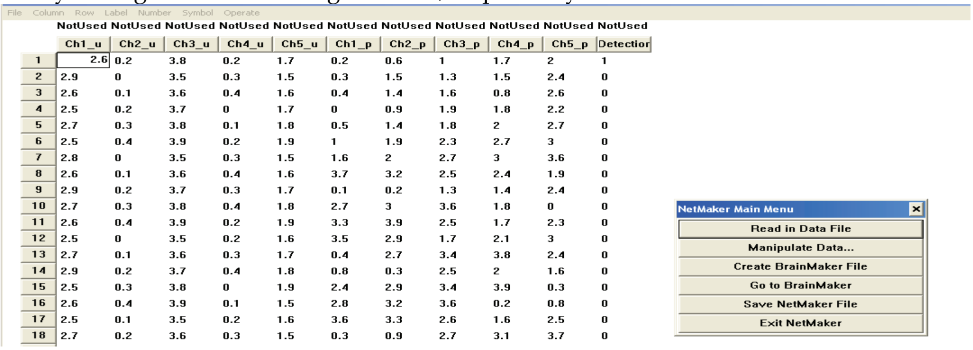
Character	**Binary value**
A	"000"
B	"001"
C	"010"
D	"011"
E	"100"
F	"101"
G	"110"
H	"111"

Table 2. Binary coding

Compared to real number coding, digital coding requires more neurons. Compared to other digitl codigns, one strength of the selected binary coding of Table 2 is that it simplies the representation of each character by the least number of bits, rather than, say, to enhance its fault tolerance snd/or sparseness as in 1-hot assignment and other encoding schemes. Other coding techniques based on Hamming, Reed Solomon, and others could be also used. We just chose the simplest coding.

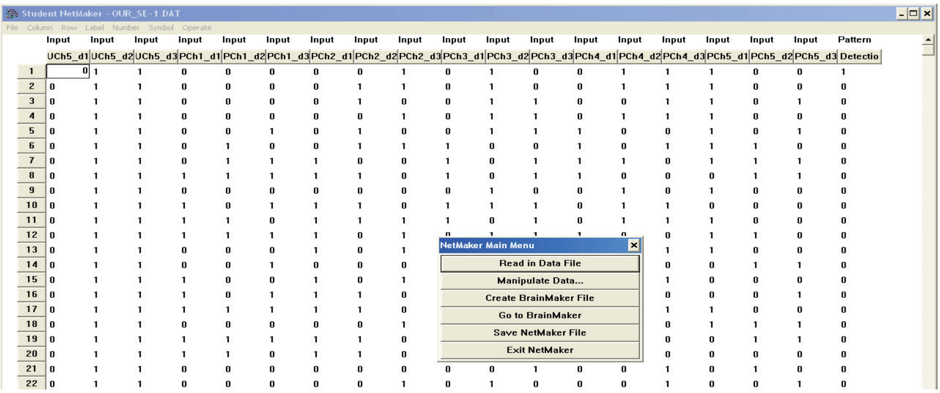
6. Training, Simulation and Test

We used the BrainMaker (IEEE, 1992) application for training and simulating our network and retrieving match accuracy results. BrainMaker is a popular neural network simulation package and has been used extensively by researchers such as (Dombi & Lawrence, 1994). We created a list of 1000 (user ID, password, output) triplets where only one triplet had an output of 1 (means match) for the matching (user ID, password combination). We focused on a single user ID so all the user ID values in all 100 triplets were identical. By providing the output match/no-match value during training, we avoided the need to compare the memorized password from the entered password as in (Reyhani & Mahdavi, 2007) and (Wang & Wang, 2008). The password values varied slightly to a great extent from the matching password value. In the real number coding experiment we assigned values to the user ID and password characters as in Table 1 and we simulated a network of 10 input neurons and 1 hidden layer neuron and 1 output neuron. In the binary coding experiment, we assigned values to the user ID and password characters as in Table 2 and we simulated a network of 30 input neurons and 30 hidden layer neurons and 1 output neuron. The created training set of 1000 triplets or lines was first written into an Excel spreadsheet and then we changed that file's extension to ".dat." We then started the NetMaker application to read the .dat file and to create from it the BrainMaker files including the definition file, training fact file and the testing fact file. The resulting training set in NetMaker format for analog and binary codings are shown in Figs. 7 and 8, respectively.



	NotUsed	NotUsed	NotUsed	NotUsed	NotUsed	NotUsed	NotUsed	NotUsed	NotUsed	NotUsed	NotUsed
	Ch1_u	Ch2_u	Ch3_u	Ch4_u	Ch5_u	Ch1_p	Ch2_p	Ch3_p	Ch4_p	Ch5_p	Detectior
1	2.6	0.2	3.8	0.2	1.7	0.2	0.6	1	1.7	2	1
2	2.9	0	3.5	0.3	1.5	0.3	1.5	1.3	1.5	2.4	0
3	2.6	0.1	3.6	0.4	1.6	0.4	1.4	1.6	0.8	2.6	0
4	2.5	0.2	3.7	0	1.7	0	0.9	1.9	1.8	2.2	0
5	2.7	0.3	3.8	0.1	1.8	0.5	1.4	1.8	2	2.7	0
6	2.5	0.4	3.9	0.2	1.9	1	1.9	2.3	2.7	3	0
7	2.8	0	3.5	0.3	1.5	1.6	2	2.7	3	3.6	0
8	2.6	0.1	3.6	0.4	1.6	3.7	3.2	2.5	2.4	1.9	0
9	2.9	0.2	3.7	0.3	1.7	0.1	0.2	1.3	1.4	2.4	0
10	2.7	0.3	3.8	0.4	1.8	2.7	3	3.6	1.8	0	0
11	2.6	0.4	3.9	0.2	1.9	3.3	3.9	2.5	1.7	2.3	0
12	2.5	0	3.5	0.2	1.6	3.5	2.9	1.7	2.1	3	0
13	2.7	0.1	3.6	0.3	1.7	0.4	2.7	3.4	3.8	2.4	0
14	2.9	0.2	3.7	0.4	1.8	0.8	0.3	2.5	2	1.6	0
15	2.5	0.3	3.8	0	1.9	2.4	2.9	3.4	3.9	0.3	0
16	2.6	0.4	3.9	0.1	1.5	2.8	3.2	3.6	0.2	0.8	0
17	2.5	0.1	3.5	0.2	1.6	3.6	3.3	2.6	1.6	2.5	0
18	2.7	0.2	3.6	0.3	1.5	0.3	0.9	2.7	3.1	3.7	0

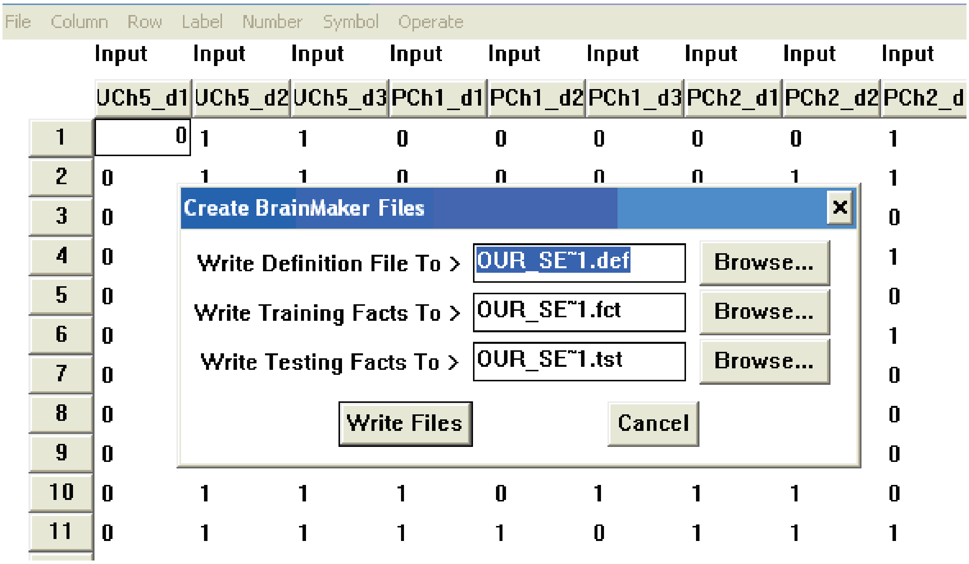
Fig. 7. Sample real number-coded training data in NetMaker format



	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Pattern
	UCh5_d1	UCh5_d2	UCh5_d3	PCh1_d1	PCh1_d2	PCh1_d3	PCh2_d1	PCh2_d2	PCh2_d3	PCh3_d1	PCh3_d2	PCh3_d3	PCh4_d1	PCh4_d2	PCh4_d3	PCh5_d1	PCh5_d2	PCh5_d3	Detectio
1	0	1	1	0	0	0	0	0	1	0	1	0	0	1	1	1	0	0	1
2	0	1	1	0	0	0	0	1	1	0	1	0	0	1	1	1	0	0	0
3	0	1	1	0	0	0	0	1	0	0	1	1	0	0	1	1	0	1	0
4	0	1	1	0	0	0	0	0	1	0	1	1	0	1	1	1	0	0	0
5	0	1	1	0	0	1	0	1	0	0	1	1	1	0	0	1	0	1	0
6	0	1	1	0	1	0	0	1	1	1	0	0	1	0	1	1	1	0	0
7	0	1	1	0	1	1	1	0	0	1	0	1	1	1	0	1	1	1	0
8	0	1	1	1	1	1	1	1	0	1	0	1	1	0	0	0	1	1	0
9	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
10	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	0	0	0	0
11	0	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1	0	0	0
12	0	1	1	1	1	1	1	0	1	0	1	1	1	0	0	1	1	0	0
13	0	1	1	0	0	0	1	0	1						1	1	0	0	0
14	0	1	1	0	0	1	0	0	0						0	0	1	1	0
15	0	1	1	1	0	0	1	0	1						1	0	0	0	0
16	0	1	1	1	0	1	1	1	0						0	0	0	1	0
17	0	1	1	1	1	1	1	1	0						1	1	0	0	0
18	0	1	1	0	0	0	0	0	1						0	1	1	1	0
19	0	1	1	1	1	1	1	1	0						0	0	1	1	0
20	0	1	1	1	1	0	1	1	0						0	0	0	0	0
21	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0
22	0	1	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0

Fig. 8. Sample binary-coded training data in NetMaker format

A snapshot of the BrainMaker files and their extensions is shown in Fig. 9.



	Input	Input	Input	Input	Input	Input	Input	Input	Input
	UCh5_d1	UCh5_d2	UCh5_d3	PCh1_d1	PCh1_d2	PCh1_d3	PCh2_d1	PCh2_d2	PCh2_d
1	0	1	1	0	0	0	0	0	1
2	0	1	1	0	0	0	0	1	1
3	0								0
4	0								1
5	0								0
6	0								1
7	0								0
8	0								0
9	0								0
10	0	1	1	1	0	1	1	1	0
11	0	1	1	1	1	0	1	1	1

Create BrainMaker Files

Write Definition File To > OUR_SE~1.def

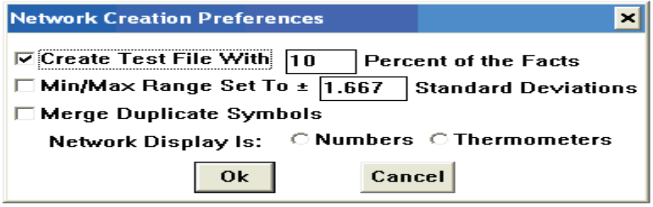
Write Training Facts To > OUR_SE~1.fct

Write Testing Facts To > OUR_SE~1.tst

Fig. 9. Associated BrainMaker files

The BrainMaker application was then launched to train, test, manipulate data, and run network simulations. During the training phase, we could see the changes that occurred to the output in graph, numeric or histogram formats, allowing us to immediately discover how big a training set was necessary to reach the output neuron's password match accuracy, or if the simulated network with its tested parameters will satisfy our match accuracy requirements.

Out of the 1000 lines in the training set, we reserved 90% or 900 for training the network and 10% or 100 for testing the network after it has been trained for password detection accuracy, as illustrated in Fig. 10. The 10% reserved for testing are not used at all for training the network and are exclusively used for testing the network, i.e. checking that it generates a correct output, 0 when the password does not match the user ID, and 1 otherwise.


Network Creation Preferences

☑ Create Test File With 10 Percent of the Facts

□ Min/Max Range Set To ± 1.667 Standard Deviations

□ Merge Duplicate Symbols

Network Display Is: ○ Numbers ○ Thermometers

Fig. 10. Creating test set

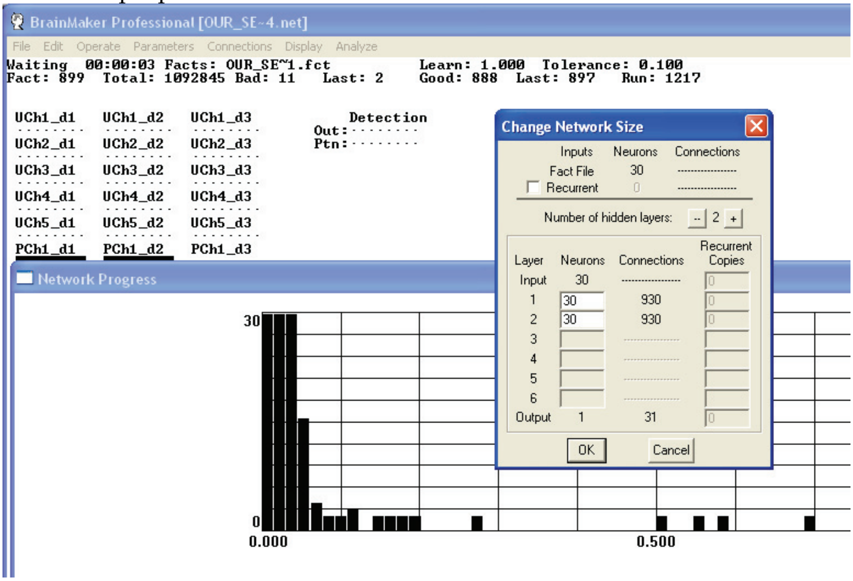
7. Results

After training the network with real number coded inputs with the BrainMaker simulator and with a training set of 900 lines, we obtained 1.44% incorrect detections and 98.5% good password detections. The learn and tolerance values were 0.085 and 0.1, respectively. Next, we switched to binary coded inputs, and at the end of training, the accuracy stood at 99.7%

good detections and 0.0022% bad or incorrect detections. In the testing phase, we tested the network with binary-coded inputs with the 100 lines and obtained a perfect 100% accuracy. Attempting to validate these accuracy results and discovering any correlation to the size of the testing set, we increased the testing file to 50% (500 lines or test cases) instead of 10%. The accuracy results did not change which means that our artificial neural network has learnt well and has been sufficiently trained.

7.1 Experimenting with number of hidden layers and number of neurons in the hidden layer

Afterwards, we experimented with the number of hidden layers and adding noise as illustrated in Fig. 11. By doubling the number of hidden layers to 2, we got 11 incorrect detections, and thus more hidden layers reduced the accuracy. By increasing the number of neurons in the hidden layer from 30 to 50 neurons, the password detection accuracy also decreased. We got 2 bad values (false matches) only when the hidden layer had 30 neurons, but after increasing the number of neurons in the hidden layer we got 12 bad values. These last two experiments caused our network to overlearn. Extra added neurons or hidden layers of neurons falsely twisted the results. This is similar to asking the opinions of two or more persons on some subject when they have unidentical opinions, and then somewhat averaging their answers in order to make a judgment, rather than taking the advice of a fewer number of people but who are in accord.



```
Waiting   00:00:03 Facts: OUR_SE~1.fct          Learn: 1.000  Tolerance: 0.100
Fact: 899  Total: 1092845 Bad: 11   Last: 2     Good: 888  Last: 897   Run: 1217

                                          Detection
                                     Out:........
                                     Ptn:........
```

Change Network Size

	Inputs	Neurons	Connections
	Fact File	30	----------------
□	Recurrent	0	----------------

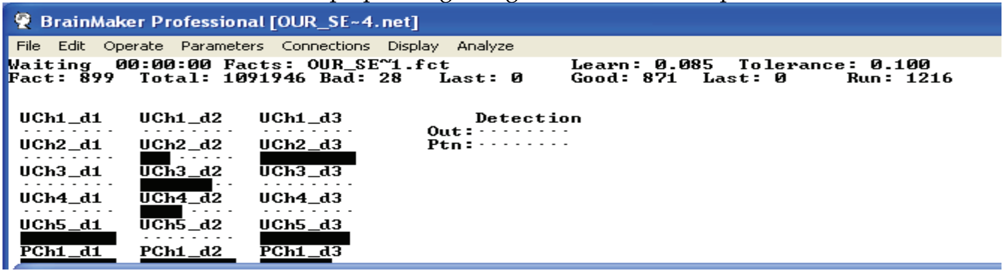
Number of hidden layers: -- 2 +

Layer	Neurons	Connections	Recurrent Copies
Input	30	----------------	0
1	30	930	0
2	30	930	0
3		----------------	
4		----------------	
5		----------------	
6		----------------	
Output	1	31	0

Fig. 11. Changing the artificial neural network size (number of hidden layers, and number of neurons per hidden layer)

7.2 Adding noise to the network

Moreover, we added noise to the network (see Fig. 12) and observed that as expected, the more noise was added the more the results worsened. In fact, as a result of noise addition, the accuracy dropped to 96% and the number of bad result increased to 28 bad values which mean 3.11% incorrect detections. Noise causes the data values to increase or decrease from their original values and throws the detection accuracy off but it is important for the artificial neural network to keep operating at high accuracies in the presence of noise.



```
Waiting   00:00:00 Facts: OUR_SE~1.fct          Learn: 0.085   Tolerance: 0.100
Fact: 899   Total: 1091946 Bad: 28    Last: 0    Good: 871   Last: 0       Run: 1216

 UCh1_d1    UCh1_d2    UCh1_d3          Detection
                                   Out:
 UCh2_d1    UCh2_d2    UCh2_d3     Ptn:

 UCh3_d1    UCh3_d2    UCh3_d3

 UCh4_d1    UCh4_d2    UCh4_d3

 UCh5_d1    UCh5_d2    UCh5_d3

 PCh1_d1    PCh1_d2    PCh1_d3
```

Fig. 12. Adding noise

The results of all our experiments with 1000 (user ID, password) combinations in total, of which 900 are used for training and 100 for testing, are summarized in Table 3.

ANN EXPERIMENT	Accuracy
Default (with binary coded inputs, one hidden layer & one input layer of 30 neurons each)	100% (after testing phase)
Binary coding with 2 hidden layers	97.6 %
Binary coding with added noise	96 %
Binary Coding, with number of neurons in the hidden layer increased to 50 neurons	98.2 %
Binary Coding, with number of neurons in the hidden layer reduced to 6 neurons	95.6 %
With real number coded inputs	98.4 %

Table 3. Accuracy results

8. Conclusion

Artificial neural networks are used in many applications ranging from control, to pattern recognition, to classification. In this Chapter, we have described our experience in designing artificial neural networks for secure access authorization. We designed our feedforward network with 3 layers, created the training set and trained our network with the

backpropagation algorithm. We then simulated and tested our design with BrainMaker and collected accuracy results. We were able to achieve 100% result accuracy with our network. Increasing the number of hidden layers or the number of neurons in the hidden layer or adding noise all individually hurt our network's password detection accuracy.
Potential future work is to extend the network to accept alphanumeric inputs and extend the input range to A-Z while keeping the number of neurons under control.

9. Acknowledgments

We acknowledge Dr. Azam Beg of the College of Information Technology, UAE University for useful discussions.

10. References

Bishop, C. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press, UK.

Ciaramella, A.; D'Arco, P., De Santis, A., Galdi, G. & Tagliaferri, R. (2006). Neural Network Techniques for Proactive Password Checking. *IEEE Trans. on Dependable and Secure Computing*, Vol. 3, No. 4, pp. 327-339.

Dombi, G. & Lawrence, J. (1994). Analysis of Protein Transmembrane Helical Regions by a Neural Network. *Protein Science*, Vol. 3, (1994), pp. 557-566. Cambridge University Press.

Fausett, L. (1994). *Fundamentals of Neural Networks*, Prentice Hall, New York.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*, 2nd Ed, Prentice Hall, NY.

IEEE Expert Staff. (1992). Brainmaker Professional. *IEEE Expert: Intelligent Systems and Their Applications*, Vol. 7, No. 2, (April 1992), pp. 70-71.

Li, L.; Lin, I. & Hwang, M. (2001). A Remote password authentication scheme for multiserver architecture using neural networks. *IEEE Transactions on Neural Networks*, Vol. 12, No. 6, (November 2001), pp. 1498-1504.

Lin, I., Ou, H. & Hwang, M. (2005). A User authentication system using back-propagation network. *Neural Computing and Applications*, 14, (2005), pp. 243-249.

Reyhani, S. & Mahdavi, M. (2007). User Authentication Using Neural Network in Smart Home Networks. *International Journal of Smart Home*, Vol. 1, No. 2, (2007), pp. 147-154.

Rumelhart, D; McClelland, J. (Eds, 1986). *Parallel Distributed Processing: Explorations in the MicroStructure of Cognition*, MIT Press, Cambridge.

Wang, S. & Wang, H. (2008). Password Authentication Using Hopfield Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 38, No. 2, (March 2008), pp. 265-268.

Complementary Relevance Feedback Methods for Content-Based Image Retrieval

Peng-Yeng Yin and Chia-Mao Chen
Department of Information Management, National Chi Nan University
Taiwan

1. Introduction

Traditional keyword-based image retrieval (KBIR) is considered a crippled process that suffers the following drawbacks. (1) It is laborious to annotate manually every stored image by keywords selected from a predefined set because modern databases easily contain thousands of images. (2) A real-world image usually involves many concepts, it is difficult to annotate such an image by a small number of keywords. (3) Ambiguities exist between different people's subjects for a given image and would deteriorate retrieval precision when matching the query to the stored images.

Recently, content-based image retrieval (CBIR) (Yoshitaka & Ichikawa, 1999; Smeulders et al., 2000) has emerged as one of the solutions to overcome the limitations entailed by KBIR. Users access CBIR systems (Flickner & Sawhney, 1995; Pentland et al., 1996; Rui et al., 1997; Nastar et al., 1998; Mokhtarian et al., 1996) by directly submitting image examples, object sketches, or other visual information (e.g., color, shape, texture, etc.) The retrievals are ranked based on image processing and similarity matching techniques, alleviating the burden for manual annotations. However, as the ranking of retrievals is calculated according to machine's subject (selected image features), the precision may be unsatisfactory due to the gap between the visual and semantic concepts.

To this end, relevance feedback (RF) treats the retrieval session as repetitive query reformulation operations. Through successive human-computer interactions, the query descriptive information (features, matching models, metrics or any meta-knowledge) is repeatedly modified as a response to the user's feedback on retrieved results. Therefore, the query close to the optimal is eventually produced and the retrieval precision is improved.

Most of the RF approaches for CBIR applications can be classified into three categories. The query vector modification method reformulates the query through user's feedback, so the query is moved towards a region containing more relevant images. The feature relevance estimation approach learns the relevance weight for each image feature and uses the weight to bias the matching. The classification-based method trains a classifier from historic feedbacks for classifying the database images as relevant or irrelevant. Each category of RF methods has its own strengths and weaknesses to be noted in the next section. So it is difficult to design a new RF method which performs best for all kinds of image content. A more practical way is to identify the shared and contrasting features between different RF methods and design a complementary strategy which maximizes the synergism between multiple RF methods and alleviates the individual weaknesses. In the light of this, we propose a general complementary RF framework which identifies three new RF methods that have proven to be more effective than traditional methods on a real-world image database.

The remainder of this paper is organized as follows. Section 2 reviews the major RF models. Section 3 describes the proposed complementary RF framework. Section 4 presents the experimental results and comparative performances. Finally, conclusions are made in Section 5.

2. Related Work

We begin by describing three major RF models. Then, the comparative strengths and weaknesses between these models are analyzed.

2.1 RF Models

Assume that there are n image items stored in a CBIR database provided for access by many users using the Query_by_example interface. Let Q be the image example submitted by current user and D a database image. Both Q and D are described by r visual features (texture, color, shape, etc.) An option to estimate the visual dissimilarity between Q and D is to compute the Euclidean distance between their visual feature vectors, $\vec{Q} = (q_1, q_2, ..., q_r)$ and $\vec{D} = (d_1, d_2, ..., d_r)$, as follows.

$$dist_{Euclidean} = \left\|\vec{Q} - \vec{D}\right\| = \sqrt{\left(\vec{Q} - \vec{D}\right) \bullet \left(\vec{Q} - \vec{D}\right)} \tag{1}$$

where the operation • denotes the inner product in the Euclidean space. By deriving the visual dissimilarity between Q and every database image, the retrieval system is able to return a set of v database images that are closest to Q in the visual space. However, owing to the imperfection of feature selection and noise in the feature values, not every returned image is considered relevant by the user. The RF models accommodating user's relevance feedback on the retrieved result can determine a new list of top v similar images to increase the degree of user's satisfaction. Three major RF methods operating in the visual space are reviewed in the following.

- *Query vector modification* (QVM) (Rocchio, 1971; Ciocca & Schettini, 1999) Let R and N denote the subsets of the retrieved result that are marked relevant and irrelevant, respectively, by the user in the incumbent feedback round. QVM reproduces a new query vector by a weighting sum of Q and the mean vectors of R and N. In particular, the new query vector is computed by the following formula.

$$Q \leftarrow \alpha Q + \beta \sum_{D_j \in R} \frac{D_j}{|R|} - \gamma \sum_{D_j \in N} \frac{D_j}{|N|} \tag{2}$$

where Dj is a retrieved image that belongs to R or N, α is the inertia weight promoting the query to move in the same direction as in the previous moving trajectory, and β and γ are the weights controlling the relative importance contributed by relevance and irrelevance experience. The newly produced query vector is then used for searching the next retrievals based on Euclidean metric. QVM has the effect for guiding the reformulation of the query towards relevant images and away from irrelevant ones, and the moving velocity is accelerated by an inertia term considering previous trajectory.

QVM suffers at least the following drawbacks. (1) QVM assumes that each feature is equally relevant to the query; however, the importance of some features may be discounted due to the semantic concept the user is seeking. (2) The parameters α, β and γ need to be empirically tuned in order to perform both effectively and efficiently on databases with various content.

- *Feature relevance estimation* (FRE) (Rui et al., 1998; Peng et al., 1999) The FRE approach assumes that each feature can have a various weight in judging the relevance between Q and D. The appropriate weight of a feature can be learned from the user's incumbent feedback information. A simple notion to estimate the relevance weight of individual feature is the feature projection technique that assesses the retrieval ability (in terms of the number of relevant images retrieved) using each feature alone. Firstly, all the database images are projected onto the axis of the tested feature, so the top s ($\geq v$) closest images to the query with respect to the corresponding feature can be derived. The value of s is at least as large as v because there might be no relevant images in the list of the top v retrieved images using a single feature and little knowledge can be learned. Typically, we set s = 2v. Secondly, let Ω_i denote the set of the top s retrieved images using only the ith feature, the relevance weight (w_i) of this feature is apparently related to the number of members of Ω_i that are also in R or N. In general, the relevance weight is estimated by $w_i = f\left(\left|R \cap \Omega_i\right|\right) - g\left(\left|N \cap \Omega_i\right|\right)$ where f and g can be linear, quadratic, or exponential functions, depending on the desired learning ratio. Finally, the relevance weights are normalized such that $\sum_{i=1}^{r} w_i = 1$ and they are incorporated into the dissimilarity metric to express the degree of emphasis on the corresponding feature, *viz.*,

$$dist_{FRE} = \left\|\vec{Q} - \vec{D}\right\|_W = \sqrt{\left(\vec{Q} - \vec{D}\right)W\left(\vec{Q} - \vec{D}\right)^T} \tag{3}$$

where W is the feature weight matrix whose diagonal entries are equal to w_i and off-diagonal entries are zero. So $dist_{Euclidean}$ can be viewed as a special case of $dist_{FRE}$ where W is equal to the identity matrix.

Practical applications of FRE also manifest a few shortcomings. (1) The query vector cannot be moved towards a more desired region in the feature space. It is likely that some relevant images may not be selected in the regional neighborhood of the original query. (2) The estimation of relevance weight using the projection technique can be computationally expensive if the feature space involves large dimension.

- *Classification-Based Method* (CBM) (Meilhac & Nastar, 1999; Cox et al., 2000; Tieu & Viola, 2000; Huang et al., 2000; Tong & Chang, 2001; Su et al., 2003; Yin et al., 2008; Li & Hsu, 2008) The CBM approach realizes the retrieval process as a classification task. The collected feedback information (relevant and irrelevant examples) is used as training data such that the employed classifier can be incrementally trained to obtain an improving capability for classification of database images. The popularly used classifiers for image retrieval applications range from Bayes classifier (Meilhac & Nastar, 1999; Cox et al., 2000; Su et al., 2003), to boosting (Tieu & Viola, 2000), graph matching (Li & Hsu, 2008), virtual feature (Yin et al., 2008), and the support vector machine (SVM) (Huang et al., 2000; Tong & Chang, 2001).

Cox et al. (2000) used a Bayesian framework to estimate the *a posteriori* probability that a database image is relevant to the query given the *a priori* probability densities of feature values contributed by the labeled examples from history of feedbacks. Since the probability density function is updated after each feedback round, the system is able to improve the performance of next retrieval. Tieu and Viola (2000) extracted the 20 most relevant features for a given query from more than 45,000 highly selective ones based on the boosting technique. They assumed that relevant images share some visual causes and the learned classifier can focus on a small set of relevant features for a particular query, so the matching is computationally efficient even for a very large database. Huang et al. (2000) incorporated the SVM to determine the preference weight of each positive example collected from the relevance feedback. The farthest positive example from the optimal separating hyperplane (OSH) learned by the SVM receives the highest weight and vice versa. This mechanism releases the user from manually providing the preference weight for each positive example.

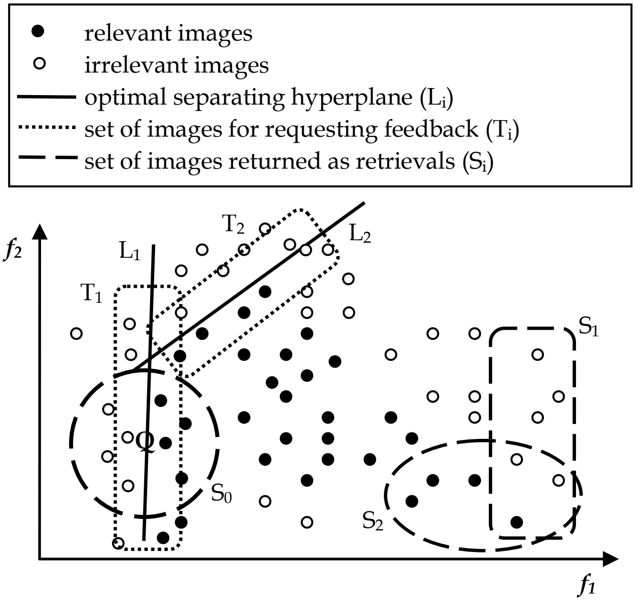

● relevant images
○ irrelevant images
optimal separating hyperplane (L_i)
set of images for requesting feedback (T_i)
set of images returned as retrievals (S_i)
f_2
L_1
T_2
L_2
T_1
S_1
Q
S_0
S_2
f_1

Fig. 1. Illustration of SVM$_{Active}$ method.

Tong and Chang (2001) proposed an SVM active learner (SVM$_{Active}$) for CBIR with relevance feedback. The database images residing in the farthest places at the positive side from the OSH are returned as current retrievals while the selective images closest to the OSH are shown to the user for providing his/her relevance feedback. Those images marked as relevant or irrelevant are subject to the training of the SVM for next retrievals. So the SVM actively learns from the selective images instead of those randomly presented. The idea of SVM$_{Active}$ method is illustrated in Fig. 1 where two selected features f_1 and f_2 are assumed for simplicity. The zero-page retrievals (S_0) before any relevance feedback are the nearest neighbors of the user-submitted query Q according to the Euclidean distance (Eq. (1)). Based on the user's feedback regarding to S_0, an SVM is trained and the OSH (L_1) is obtained. The farthest images from the positive side of L_1 are returned as next retrievals (S_1), while the selective images (T_1) closest to L_1 are shown to the user for providing his/her relevance feedback. The SVM is thus trained again using current feedback information and another OSH (L_2) is obtained. The images farthest to L_1 and L_2 are presented as next retrievals (S_2), and the images (T_2) located nearest to L_2 are subject to those asking user's next feedback. Repeating this process, the SVM classifier can be incrementally trained using T_1, T_2, etc. and improves its retrieval performance.

SVM*Active* method entails the following issues. The SVM treats the retrieval problem as a two-class classification task, when the relevance information is little at the beginning of the query session, the retrieval precision could be very low (see S1 in Fig. 1). The number of false alarms could be great because the number of relevant images is significantly smaller than the number of total images stored in the database. Furthermore, the set of retrieved images is disjoint from the set of images requesting user's feedback. Although this design actively selects the example images that are most useful for the SVM training, it also incurs additional costs for the user to examine two sets of images.

2.2 Comparative Analysis

Our computation experience shows that each RF model imposes individual bias in inferring the image relevance because of its model assumptions. The inference biases of the major RF models are summarized in Table 1. We found that QVM assumes equal relevance weight in similarity matching for each used feature, however, the positive images may be not equally relevant to the query along every feature. The FRE stipulates that the query vector is fixed to the original vector example submitted by the user and will not be reformulated during the query session, so the query vector is incapable of moving to desired regions. The CBM method, on the other hand, treats the retrieval problem as a classification task, the employed classifier could easily impose a great number of false alarms because the number of relevant images is significantly smaller than the number of total images stored in the database.

RF Models	Inference Biases
QVM	● Assume equal relevance weight for each feature, however, the positive images may be not equally relevant to the query along every feature.
FRE	● Query vector is not reformulated, so it cannot be moved towards a more desired region of the feature space.
CBM	● Trained classifier could be severely biased due to insufficiency of training data. The initial performance could be unsatisfactory.

Table 1. Inference biases of major RF models.

Recently, some researchers began studying hybrid methods which provide a chance to improve the performance that can be obtained based on existing RF methods. Yin & Liu (2009) proposed an RF strategy combining QVM and FRE. This strategy moves the query vector to a desired region and simultaneously assigns each feature an appropriate weight of relevance. Yin et al. (2005) proposed a sophisticated framework that automatically chooses the best RF model at a particular feedback round for a given query. They used a reinforcement learning algorithm to maximize the accumulated precision over all submitted queries.

Wang et al. (2003) incorporated the Euclidean search into the SVM active learning in two ways. (1) If an image is classified by the trained SVM as relevant, it is assigned a dissimilarity score equivalent to its Euclidean distance from the known relevant image that is farthest to the OSH. (2) Otherwise, a penalized dissimilarity score is given which is equal to the sum of the distance from the image to the OSH and the maximal distance obtained in (1). Wang's method ensures that an SVM classified negative image would be less preferable than any SVM classified positive image. Moreover, the next included retrievals are the images closest to the known relevant image instead of those farthest to the OSH as employed by the SVM_{Active} method.

3. The Proposed Method

In this section, we propose a general RF framework, named the *complementary method*, which takes advantage of multiple existing RF methods and exploits the synergism between them to improve the performance. We differentiate complementary methods from hybrid methods by the following features. (1) Complementary methods combine two different approaches that are complementary to each other in a hope to eradicate the weaknesses of individual approach, while hybrid methods in general look for a combination scheme of two different methods as long as the overall performance is improved. (2) Complementary methods exploit synergism between two categories of methods, so a general meta-strategy is created and the conception can be implemented in several variations. By contrast, the hybrid methods deal with two carefully selected methods and design all the implementation details instructing how the two methods interweave.

In what follows, we elaborate a complementary RF framework which depicts a general conception for collaborating two types of RF models and identify three effective RF methods.

3.1 Complementary RF Framework

Our complementary RF framework combines two RF models denoted by Θ and Ω where Θ belongs to the classification-based methods (CBM) and Ω could be any alternative method using vector space model or probabilistic model. Fig. 2 shows the conception of the proposed complementary RF framework. Let Q denote the incumbent query submitted by the user. The system fist applies Ω to retrieve a set of v database images that are most similar to Q. These retrievals are presented to the user requesting for relevance feedback. The entered relevant (R) and irrelevant (N) feedback information is exploited in two folds. *First*, the received R and N feedbacks are used to update parameters of the employed RF model Ω, so the retrieval performance is improved. *Second*, the CBM method Θ is incrementally trained by the accumulated set of R and N and the learned classification boundaries become increasingly accurate. All the database images are classified using the trained classifier Θ into two classes: positive (C_1) and negative (C_2) sets. Set C_1 contains those images that are residing at the same side of the classification boundaries as that by the training set R, while C_2 consists of the remaining database images which locate at the other side of the classification boundaries. Set C_2 is filtered out by the system, only set C_1 is used as the image pool for next retrieval performed by the RF model Ω. So Ω is actually working on the set C_1 instead of the whole database. Fig. 3 summarizes the algorithm of the proposed complementary RF framework.

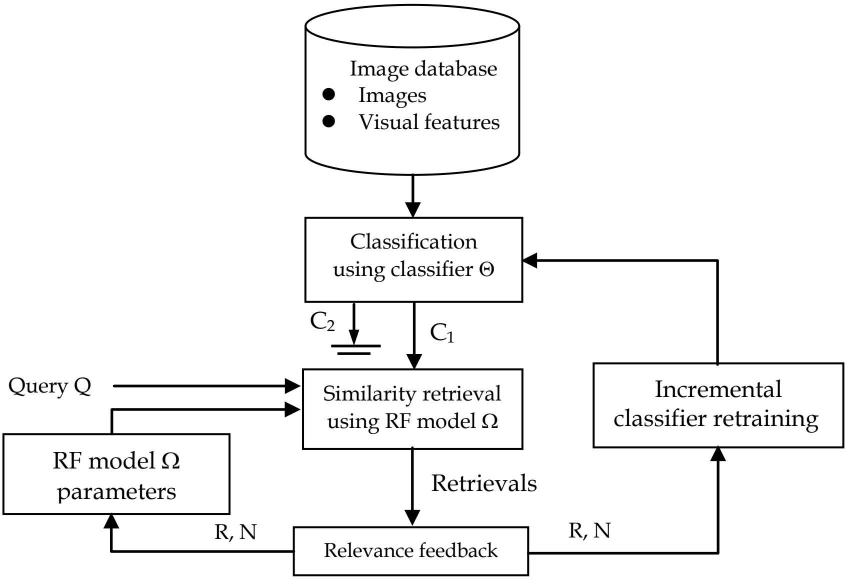


Fig. 2. Conception of the complementary RF framework.

```
1.  Let $Q$ be the current query.
2.  Compute $v$ nearest images to $Q$ using the Euclidean distance metric.
3.  While user is not satisfied with the currently retrieved result Do
    (a) User marks the $v$ images as relevant or irrelevant.
    (b) Denote by $R$ the set of relevant images, and by $N$ the set of irrelevant
        images.
    (c) Update the parameters of the RF model Ω using $R$ and $N$.
    (d) Retrain the classifier Θ using the collective $R$ and $N$.
    (e) Use Θ to classify the database images into two classes, $C_1$ (positive class)
        and $C_2$ (negative class).
    (f) Retrieve from $C_1$ the $v$ most similar images to $Q$ using the RF model Ω.
```

Fig. 3. Algorithmic summary of the complementary RF framework.

The complementary RF framework collaborates Θ and Ω together and maximizes synergism between them. As found in the literature (2003), the CBM method (Θ) usually has low retrieval precision during the first few rounds of relevance feedback. This is due to the fact that the number of relevant images to a query is significantly less than the number of

irrelevant images stored in the database. The classifier Θ easily produces too many false alarms when the training data (user's feedback) are limited at the early feedback period of the query session. But for the same reason, we observed that the negative set produced by Θ are very reliable because the classifier Θ rarely classifies a true relevant image as irrelevant, and this information can be very helpful to another RF model Ω to filter out probable irrelevant images before retrieving. Thus, the proximity for retrieving relevant images using Ω is not constrained to a hyper-sphere or Gaussian neighborhood, but is shaped by the classification boundaries learned by Θ. Moreover, the application of Euclidean-based RF model Ω can avoid producing unsatisfactory precision at the early query session period as encountered by using Θ alone since the retrievals are restrained in the proximity to the reformulated query instead of the farthest positive images to the classification boundaries.
In the following, we identify three implementations of the complementary RF framework that empirically prove to be effective in our experiments.

3.1.1 SVM-complementing QVM

The SVM-complementing QVM (SVM^{c}QVM) uses SVM as the classifier Θ and QVM as the RF model Ω. As previously noted, QVM reformulates the query vector by reference to the feedback information and intends to move the query vector towards a region containing more relevant images. However, QVM does not produce a classification boundary optimally separating the feedback data and, therefore, the next retrievals in the proximity to the reformulated query may include some undesired images that should have been ruled out by carefully exploiting the feedback data. This phenomenon is shown in Fig. 4 where Fig. 4(a) illustrates the retrieving process using QVM and Fig. 4(b) corresponds to the retrieving process using SVM^{c}QVM. It is observed from Fig. 4(a) that QVM reformulates the original query Q_0 based on feedback information contained in S_0 to generate a new query Q_1, so the images in the proximity to Q_1 are returned as new retrievals (S_1). However, some irrelevant images are also contained in S_1 and deteriorate the retrieval precision. On the other hand, as shown in Fig. 4(b), SVM^{c}QVM can improve retrievals by learning a classification boundary (L_1) using the feedback information contained in S_0. The images residing at the same side of L_1 as that by the previously retrieved irrelevant images (denoted by "−") are filtered out by the SVM classifier, the proximity S_1 thus extends to seduce more potential images as retrievals that are not reachable using the traditional QVM. Also, the proximity S_1 of the reformulated query Q_1 is not constrained to a spherical neighborhood, but is shaped by the classification boundary. As the system proceeds with more feedback rounds, the classification boundary learned by the SVM classifier would be more accurate (see Fig. 1), hence, the improving on the retrievals is remarkable.

3.1.2 SVM-complementing FRE

The SVM-complementing FRE (SVM^{c}FRE) uses SVM as the classifier Θ and FRE as the RF model Ω. As shown in Fig. 5(a), the traditional FRE estimates the feature relevance weights by feature projection of the feedbacks and incorporates the weights into the Euclidean metric. So the proximity (S_0) to the query Q is reformed to an elliptical neighborhood (S_1) to search more potentially relevant images. Analogous to the case of using traditional QVM, FRE does not produce a classification boundary from the feedback data and could invite irrelevant retrievals that are likely to be easily screened by the SVM classifier Θ.

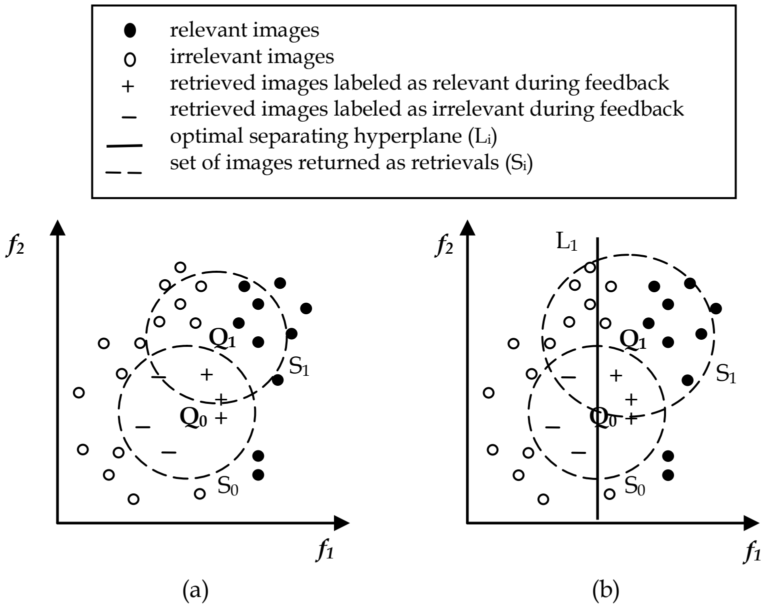

● relevant images
○ irrelevant images
+ retrieved images labeled as relevant during feedback
– retrieved images labeled as irrelevant during feedback
optimal separating hyperplane (Lᵢ)
set of images returned as retrievals (Sᵢ)
f2
f1
Q1
S1
Q0
S0
L1
(a)
(b)

Fig. 4. Comparison between QVM and SVM^{c}QVM methods.

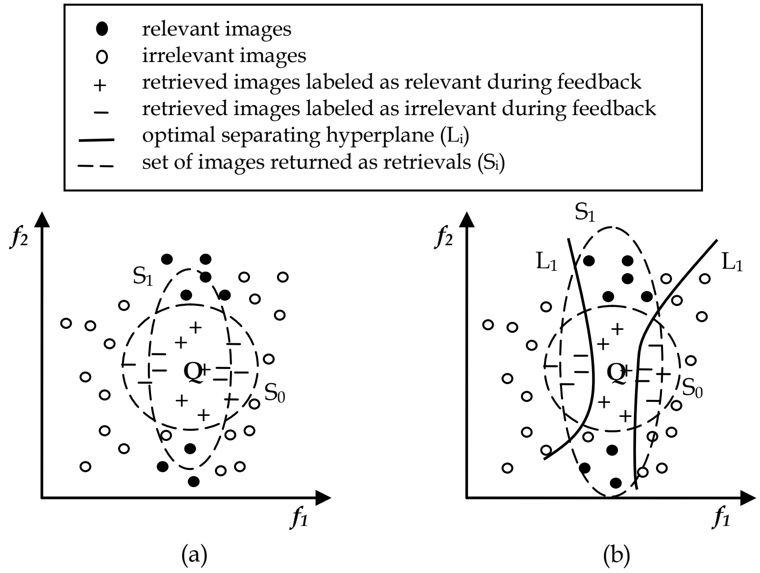
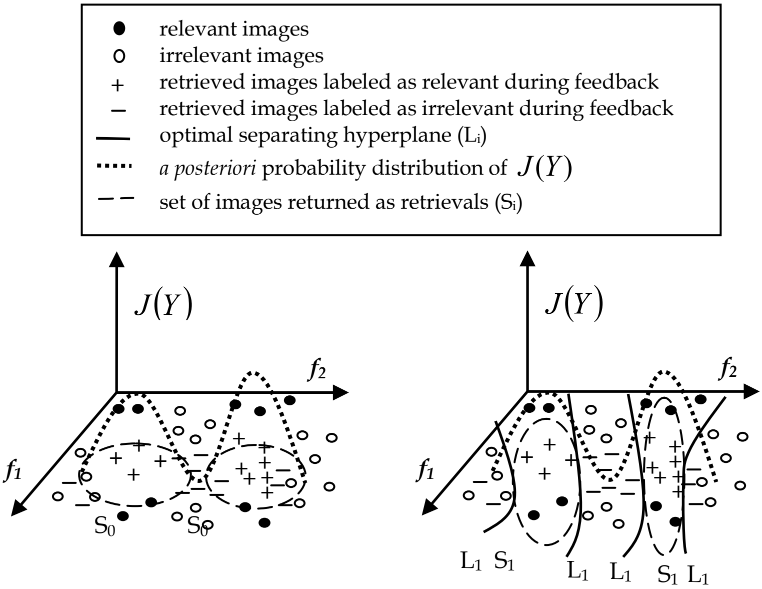

● relevant images
○ irrelevant images
+ retrieved images labeled as relevant during feedback
– retrieved images labeled as irrelevant during feedback
optimal separating hyperplane (Lᵢ)
set of images returned as retrievals (Sᵢ)
f2
f1
S1
Q
S0
L1
L1
(a)
(b)

Fig. 5. Comparison between FRE and SVM^{c}FRE methods.

Using the SVM^{c}FRE approach, it is seen from Fig. 5(b) that SVM^{c}FRE learns the classification boundaries (L_1) separating the feedback data labeled as relevant ("+") and irrelevant ("−"). The boundaries help FRE to filter out the images residing at the same side of L_1 as that by the labeled irrelevant images, so the proximity S_1 can further extend to promising area containing potentially relevant images that are however not reachable by traditional FRE.

3.1.3 SVM-complementing Bayes


● relevant images
o irrelevant images
+ retrieved images labeled as relevant during feedback
– retrieved images labeled as irrelevant during feedback
—— optimal separating hyperplane (Li)
····· a posteriori probability distribution of $J(Y)$
– – set of images returned as retrievals (Si)
$J(Y)$
f_2
f_1
S_0
S_0
$J(Y)$
f_2
f_1
L_1 S_1
L_1 L_1 S_1 L_1

(a) (b)

Fig. 6. Comparison between Bayes and SVM^{c}Bayes methods.

The SVM-complementing Bayes (SVM^{c}Bayes) uses SVM as the classifier Θ and a Bayes framework (Cox et al., 200) as the RF model Ω. The Bayesian framework estimates the *a posteriori* probability that a given image Y is relevant ($p(R|Y)$) or irrelevant ($p(N|Y)$), and they are computed by $p(R|Y) = p(Y|R)p(R)/p(Y)$ and $p(N|Y) = p(Y|N)p(N)/p(Y)$ where conditional probabilities $p(Y|R)$ and $p(Y|N)$ can be approximated by parametric models such as Gaussian kernels using the feedback information R and N, $p(R)/p(N)$ is a small constant for a given query Q since the number of relevant images is much less than that of irrelevant images. Then, the Bayesian framework retrieves the top v images with the highest values of $J(Y) = p(R|Y)/p(N|Y) = p(Y|R)p(R)/p(Y|N)p(N)$. Fig. 6(a) illustrates the distribution of $J(Y)$ using Bayesian framework.

To complement Bayesian framework which relies on probability densities of feedback data, SVM extracts the support vectors from the feedback information and learns the classification boundaries that maximize the margin from the boundaries to the support vectors. Fig. 6(b) gives the retrieving process using SVM^{c}Bayes. It is observed that the classification boundaries (L_1) learned by SVM separate the labeled relevant and irrelevant data without considering their distributions and help the Bayesian framework invite potentially relevant images with relatively low values of $J(Y)$ that are originally not of interest.

4. Experimental Results

We have implemented the major RF models in the literature including QVM (Ciocca & Schettini, 1999), FRE (Peng et al., 1999), Bayes (Cox et al., 2000), and SVM$_{Active}$ (Tong & Chang, 2001), and the proposed complementary RF methods, namely SVM^{c}QVM, SVM^{c}FRE, and SVM^{c}Bayes. The parameter values of the exiting methods follow the suggested values from their original papers. A real-world image database is used for performance evaluation.

4.1 Testing Database and Performance Measure

To testify the robustness and effectiveness of our complementary RF framework, the experiments have been conducted on a real-world image database (UCR database, 2008) containing 2,026 images classified into 19 topics such as ocean, forest, buildings, cars, humans, animals, etc. The sample images from each topic are shown in Fig. 7. To evaluate the retrieval performance of competing methods, the images from the same topic are considered relevant and the images from different topics are deemed irrelevant, so the retrieval precision obtained at different rounds of feedback can be computed automatically. The features used for image matching in our experiments consist of 22 visual features, namely, 16 Gabor features (mean and standard deviation of Gabor images at 4 orientation and 2 scales) and 6 color features (mean and standard deviation from the HSV color domain).

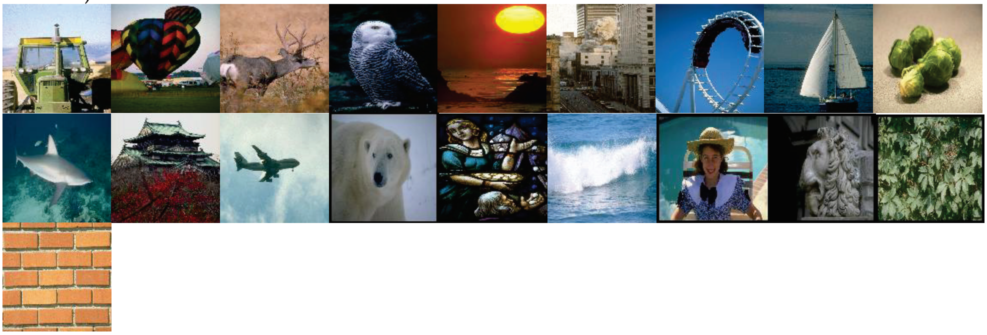


Fig. 7. Sample images from the 19 topics of the testing database.

We use the *Average Precision* (AP) measure defined by NIST TREC video (TRECVID) in our experiment for performance evaluation. Each database image is presented as a query and proceeds with 10 rounds of feedback. The feedback is automatically executed by reference to the 19 topics. The AP value that can be obtained at each round is defined as the average of precision value obtained after each relevant image is retrieved. The precision value is the ratio between the retrieved relevant images and the number of images currently retrieved.

Let $\overline{P}$ be the AP obtained at the current round of feedback and it is computed by $\overline{P} = \sum_{D_i \in R} P_i \Big/ |R|$ where P_i denotes the precision value obtained after the system retrieves i relevant images, D_i is one of the relevant images, R is the set of all relevant images that belong to the same topic as the query, and $|R|$ denotes the cardinality of R. As an example, assume one of the topics consists of six relevant images and the retrieval system ranks these relevant images at the first, second, fourth, seventh, thirteenth, and eighteenth places. Thus, the precision value obtained when each relevant image is retrieved is 1, 1, 0.75, 0.57, 0.38, and 0.33, respectively. The AP computes the average of these precision values and it is 0.67. The AP calculated over all relevant images can avoid precision fluctuation that is usually encountered by the traditional precision measure.

4.2 Comparative Performance Evaluation

In this section, we compare the complementary RF method with its counterparts by submitting each of the 2,026 database images as a query and compute the average AP obtained at various numbers of feedback rounds. Fig. 8 shows the average AP obtained by SVM^{c}QVM and its related methods, the individual SVM$_{Active}$ and QVM. We observe that SVM$_{Active}$ performs relatively worse during the first three rounds because it does not rely on the Euclidean-based proximity to the query. Instead, SVM$_{Active}$ retrieves the farthest images to the optimal separating hyperplane (OSH) and could result in low precision when the training feedback data are limited. When the system experiences more than three rounds of feedback, SVM$_{Active}$ becomes more effective and the average AP increases quickly. On the other hand, QVM is effective at the first few rounds of feedback since it progressively predicts the promising region adjacent to the centroid of known relevant images, and moves the reformulated query to that region. But its performance is constrained by the assumption that the retrievals are located in a spherical-shaped proximity to the reformulated query.

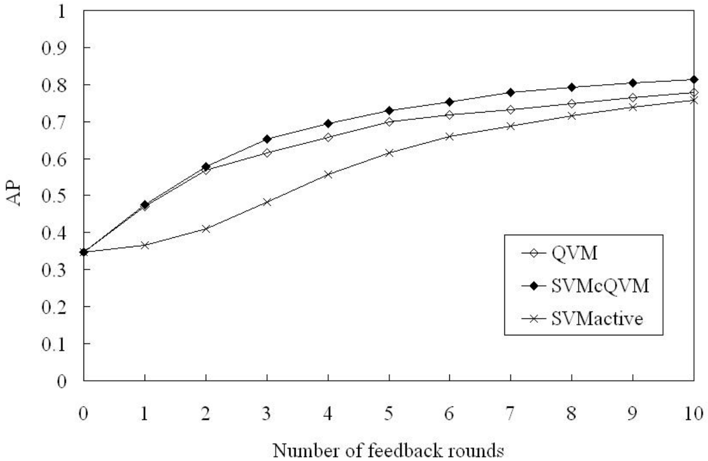


Fig. 8. The average AP obtained by QVM, SVM^{c}QVM, and SVM$_{Active}$ at different numbers of feedback rounds.

By complementing the two methods, SVM^{c}QVM filters out highly possible irrelevant images and shapes the retrieval neighborhood for QVM by the classification boundaries learned by SVM$_{Active}$. In essence, the retrieval neighborhood used for SVM^{c}QVM is no longer constrained by a hyper-sphere and can take any forms of shape. Further, as SVM^{c}QVM applies the query reformulation and locates the most promising region, so it avoids the suffering of inferior performance at early feedback rounds as encountered by SVM$_{Active}$. It is seen from Fig. 8 that SVM^{c}QVM attains the best performance among the compared methods, manifesting the synergistic effect between SVM$_{Active}$ and QVM.

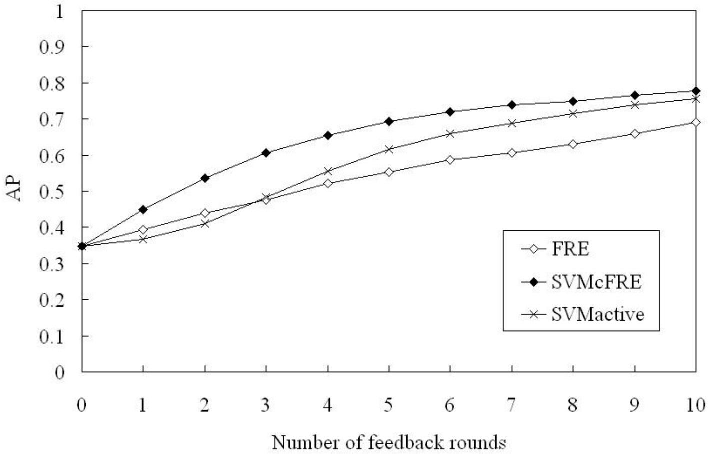


Fig. 9. The average AP obtained by FRE, SVM^{c}FRE, and SVM$_{Active}$ at different numbers of feedback rounds.

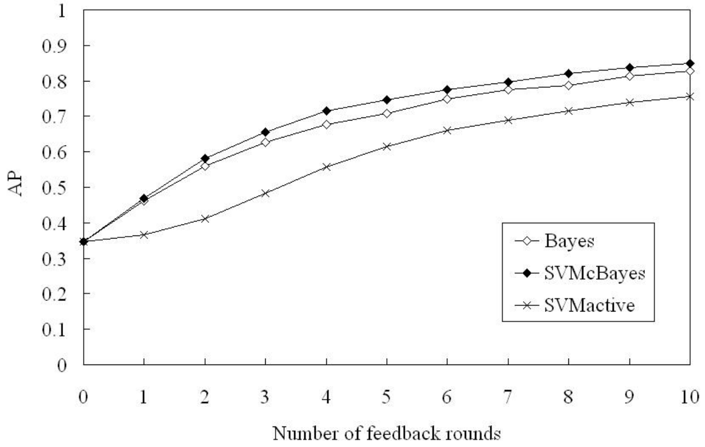


Fig. 10. The average AP obtained by Bayes, SVM^{c}Bayes, and SVM$_{Active}$ at different numbers of feedback rounds.

Figs. 9 and 10 show the comparative performances obtained using SVM^{c}FRE, SVM^{c}Bayes and their counterparts, respectively. Similar phenomena are observed in these figures. The complementary RF methods always outperform their counterparts at every round of feedback. This means that the proposed complementary RF framework is general and can be applied to a broad range of existing RF methods. By contrast, the traditional ad-hoc hybrid method specifically sticks to the combination components and is hard to be applied to other components. The complementary RF framework is in fact a meta-strategy that guides the search of embedded models to maximize their synergism by exploiting the complementarity between them.

4.3 Retrieval Examples

Here, we present visual results of some retrieval examples using competing methods. Fig. 11(a) shows the zero-page retrieval results before performing any relevance feedback. The query example image is shown at the top of the figure. The retrieved images are ranked according to their Euclidean distances to the query and *seven* of them belong to the same topic as the query, obtaining a precision value of 35%. By marking the retrieved images as relevant or irrelevant, we obtain various next retrieval results with different degrees of precision improving. The traditional QVM, FRE, and Bayes are able to assist the system to locate ten, seven, and nine relevant images in the new retrievals, respectively, as shown in Figs. 11(b)-11(d). However, it is seen in Fig. 11(e) that SVM_{Active} only find three relevant images after the first feedback round because SVM_{Active} searches the farthest positive images to the OSH instead of the closest images to the reformulated query, this mechanism is not effective when the feedback information is little, as most of other machine learning algorithms suffer this limitation as well. By contrast, our proposed complementary RF methods, namely, $\mathrm{SVM}^c\mathrm{QVM}$, $\mathrm{SVM}^c\mathrm{FRE}$, and $\mathrm{SVM}^c\mathrm{Bayes}$ are very effective in utilizing the relevance feedback, obtaining 12, 13, and 12 relevant images, respectively, as seen in Figs. 11(f)-11(h). The results exhibit a significant performance improving on their combining components. This remarkable contribution should be dedicated to the filtering of highly-possible irrelevant images, the classification boundaries learned by SVM shape the similarity neighborhood used by QVM, FRE, and Bayes more accurately.

The synergistic effect produced by our complementary methods is more profound after the second round of relevance feedback. As shown in Figs. 12(a)-12(d), QVM, FRE, Bayes, and SVM_{Active} improve the retrievals by finding 13, 10, 12, and 15 relevant images, compared to just locating 10, 7, 9, and 3 relevant images in the results obtained in the first round. It is noteworthy that SVM_{Active}, although less effective in the first round, surpasses QVM, FRE, and Bayes in the second round because the volume of feedback information is increasing and the learned classification boundaries are more accurate. This also enhances the filtering capability of complementary methods. We observe from Figs. 12(e)-12(g) that $\mathrm{SVM}^c\mathrm{QVM}$, $\mathrm{SVM}^c\mathrm{FRE}$, and $\mathrm{SVM}^c\mathrm{Bayes}$ find 17, 16, and 17 relevant images in the second feedback round, revealing the synergism between the combining components is being maximized.

Table 2 summarizes the retrieval results obtained using the competing methods at different rounds of relevance feedback. Clearly, our complementary methods can enhance the retrieval capability of combining methods at all rounds of relevance feedback.

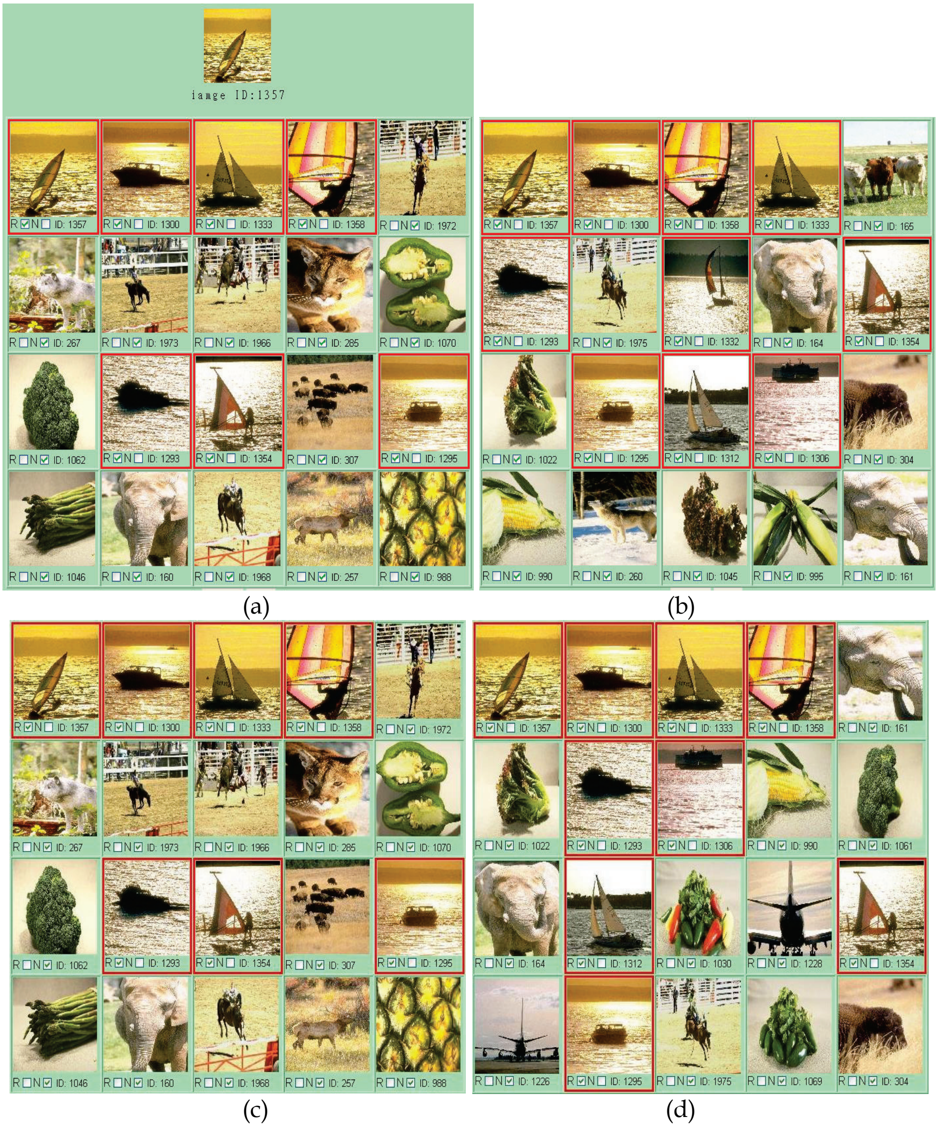
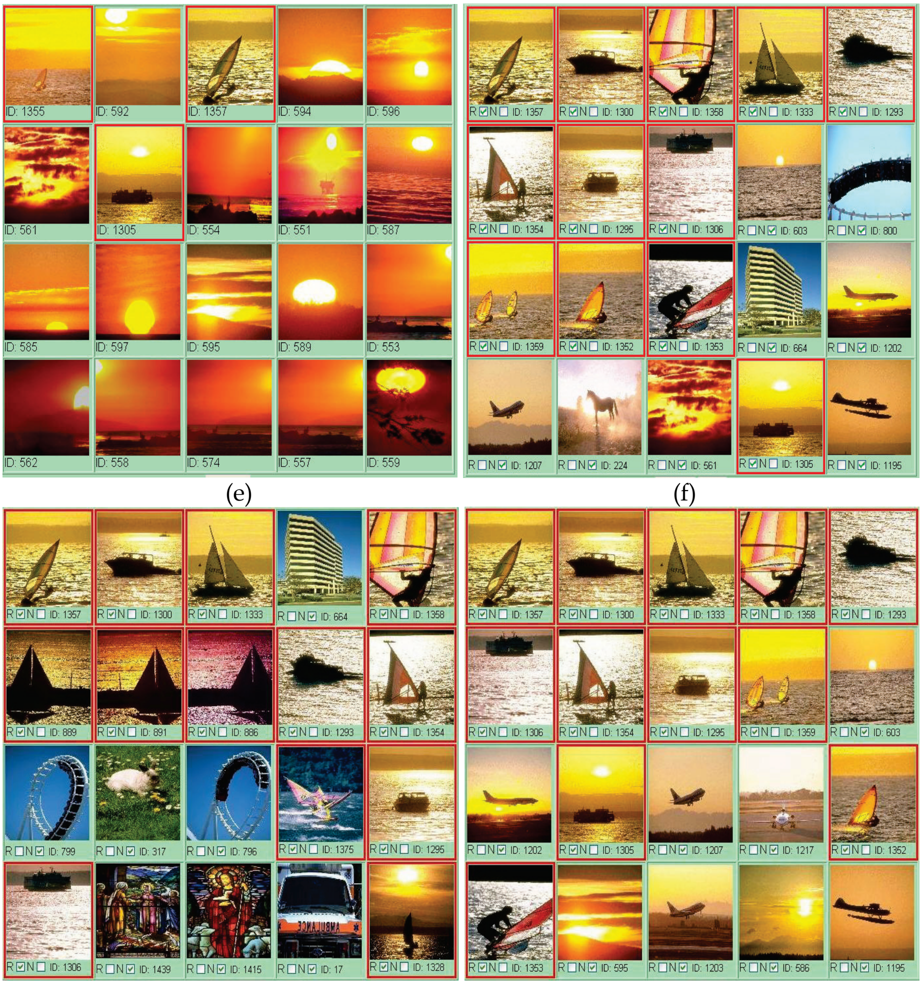


Fig. 11. Visual results of some retrieval examples in the zero-page and the first feedback round using competing methods.



(e) (f)

(g) (h)

Fig. 11. Visual results of some retrieval examples in the zero-page and the first feedback round using competing methods (continued.)

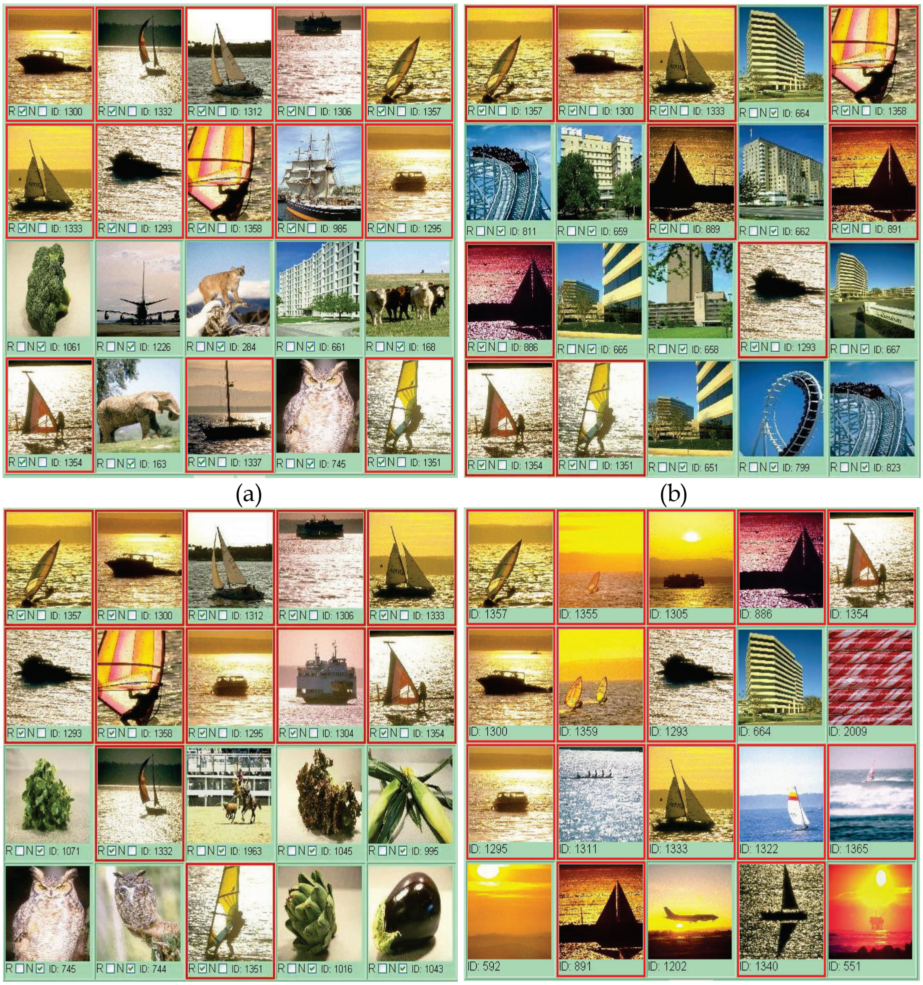
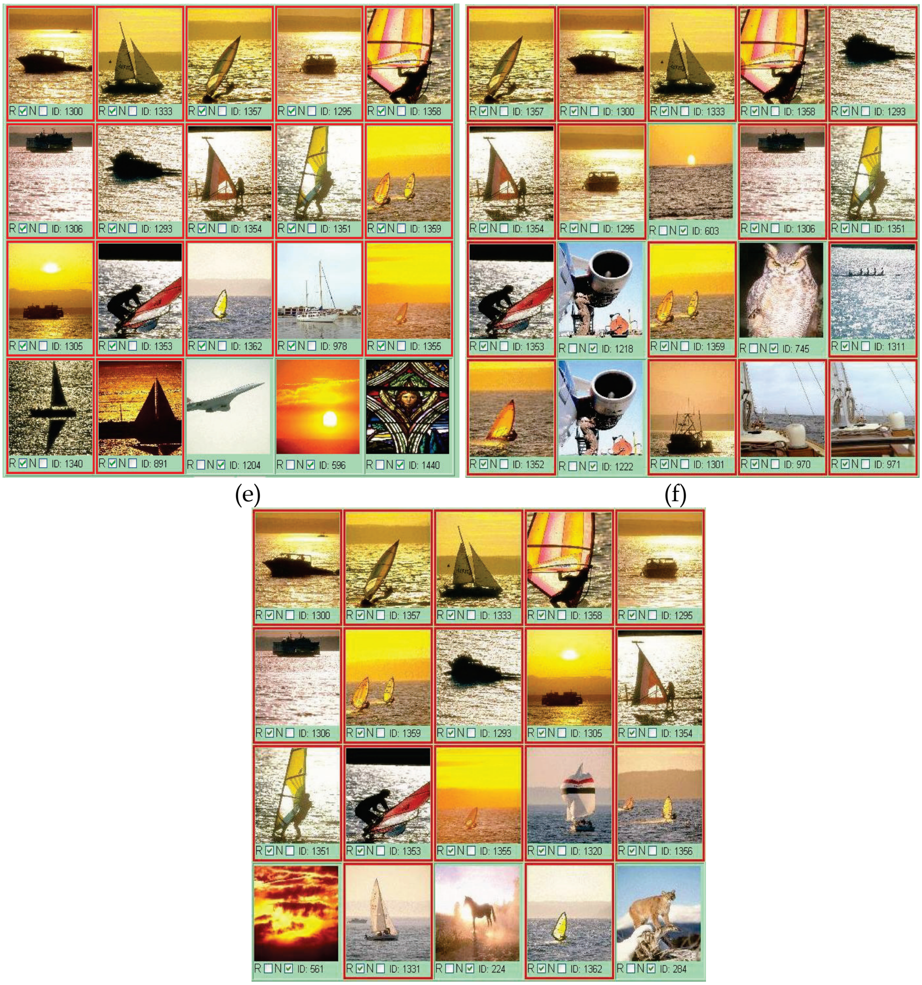


Fig. 12. Visual results of some retrieval examples in the second feedback round using competing methods.



(e) (f)

(g)

Fig. 12. Visual results of some retrieval examples in the second feedback round using competing methods (continued.)

	QVM	FRE	Bayes	SVM$_{Active}$	SVM^{c}QVM	SVM^{c}FRE	SVM^{c}Bayes
1st round	10	7	9	3	12	13	12
2nd round	13	10	12	15	17	16	17

Table 2. Retrieval results obtained using the competing methods at different rounds of relevance feedback.

5. Conclusions

Our recent survey on previous relevance feedback (RF) approaches disclosed that each type of RF methods has its own strengths and weaknesses, and that there is no RF method which performs best for all kinds of image content. We thus have proposed an innovation for the design of complementary methods which exploit the complementarity between different types of RF methods and create a meta-strategy that maximizes the synergism by guiding the collaboration between the combining RF methods. In particular, we have identified three implementations of the complementary methods, namely, the SVM^{c}QVM, SVM^{c}FRE, and SVM^{c}Bayes. These methods not only essentially avoid the inferior performance during early period of query session as suffered by SVM$_{Active}$, but also shape the retrieval proximity to the reformulated query by classification boundaries, relaxing the restriction to hyper-spherical or Gaussian neighborhoods as faced by QVM, FRE, and Bayes. Experimental results obtained by testing on a real-world image database manifest that the proposed complementary methods outperform their original counterparts and the improving on retrievals is significant.

6. References

Ciocca, G. & Schettini, R. (1999) A relevance feedback mechanism for content-based image retrieval, *Information Processing and Management*, Vol. 35, pp. 605-632.

Cox, I.; Miller, M.; Minka, T.; Omohundro, S. & Yianilos, P. (2000) The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments, *IEEE Trans. Image Processing*, Vol. 9, pp. 20-37.

Flickner, M. & Sawhney, H. (1995) Query by image and video content: the QBIC system, *IEEE Comput*. Vol. 28, pp. 23-32.

Huang, T. S.; Hong, P. & Tian, Q. (2000) Incorporate support vector machines to content-based image retrieval with relevant feedback, *Proc. International Conference on Image Processing*, Vancouver, Canada, Vol. 3, pp. 750-753.

Li, C. Y. & Hsu, C. T. (2008) Image retrieval with relevance feedback based on graph-theoretic region correspondence estimation, *IEEE Trans. Multimedia*, Vol. 10, pp. 447-456.

Meilhac, C. & Nastar, C. (1999) Relevance feedback and category search in image database, *Proc. International Conference on Multimedia Computing and Systems*, pp. 512-517.

Mokhtarian, F.; Abbasi, S. & Kittler, J. (1996) Robust and Efficient Shape Indexing through Curvature Scale Space, *Proc. British Machine Vision Conference*, pp. 53-62.

Nastar, C.; Mitschke, M.; Boujemaa, N.; Meilhac, C.; Bernard, H. & Mautref, M. (1998) Retrieving images by content: the Surfimage system, *Proc. 4th International Workshop on Advances in Multimedia Information Systems*, Istanbul, Turkey, pp. 1611-3349.

Peng, J.; Bhanu, B. & Qing, S. (1999) Probabilistic feature relevance learning for content-based image retrieval, *Computer Vision and Image Understanding*, Vol. 75, pp. 150-164.

Pentland, A.; Picard, R. W. & Sclaroff, S. (1996) Photobook: Content-based manipulation of image databases, *International Journal of Computer Vision*, vol. 18, pp. 233-254.

Rocchio, J. J. Jr. (1971) Relevance feedback in information retrieval, *The SMART System*, Salton, G. (Ed.), Prentice-Hall, New Jersey, pp. 313-323.

Rui, Y.; Huang, T. S.; Mehrotra, S. & Ortega, M. (1997) Automatic matching tool selection using relevance feedback in MARS, *Proc. 2nd International Conference on Visual Information Systems*.

Rui, Y.; Huang, T. S.; Ortega, M. & Mehrotra, S. (1998) Relevance feedback: a power tool for interactive content-based image retrieval, *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 8, pp. 644-655.

Smeulders, A. W. M.; Worring, M; Santini, S; Gupta, A & Jain, R. (2000) Content-based image retrieval at the end of the early years, *IEEE Trans. Pattern Analysis & Machine Intelligence*, Vol. 22, pp. 1349-1380.

Su, Z.; Zhang, H.; Li, S. & Ma, S. (2003) Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning, *IEEE Trans. Image Processing*, Vol. 12, pp. 924-937.

Tieu, K. & Viola, P. (2000) Boosting image retrieval, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 228-235.

Tong, S. & Chang, E. Y. (2001) Support vector machine active learning for image retrieval, *Proc. ACM International Conference on Multimedia*, pp. 107-118.

University of California, Riverside (UCR) database, http://www.cris.ucr.edu/Database.html, retrieved on 2008/5/15.

Wang, L.; Chan, K. L. & Tan, Y. P. (2003) Image retrieval with SVM active learning embedding Euclidean search, *Proc. International Conference on Image Processing*, Vol. 1, pp. 725-728.

Yin, P. Y.; Bhanu, B.; Chang, K. C. & Dong, A. (2005) Integrating relevance feedback techniques for image retrieval using reinforcement learning, *IEEE Trans. Pattern Analysis & Machine Intelligence*, Vol. 27, pp. 1536-1551.

Yin, P. Y.; Bhanu, B.; Chang, K. C. & Dong, A. (2008) Long term cross-session relevance feedback using virtual features, *IEEE Trans. on Knowledge and Data Engineering*, Vol. 20, pp. 300-320.

Yin, P. Y. & Liu, Y. M. (2009) Adaptive relevance feedback model selection for content-based image retrieval, to appear in *The Imaging Science Journal*.

Yoshitaka, A. & Ichikawa, T. (1999). A survey on content-based retrieval for multimedia databases, *IEEE Trans. Knowledge and Data Engineering*, Vol. 11, pp. 81-93.