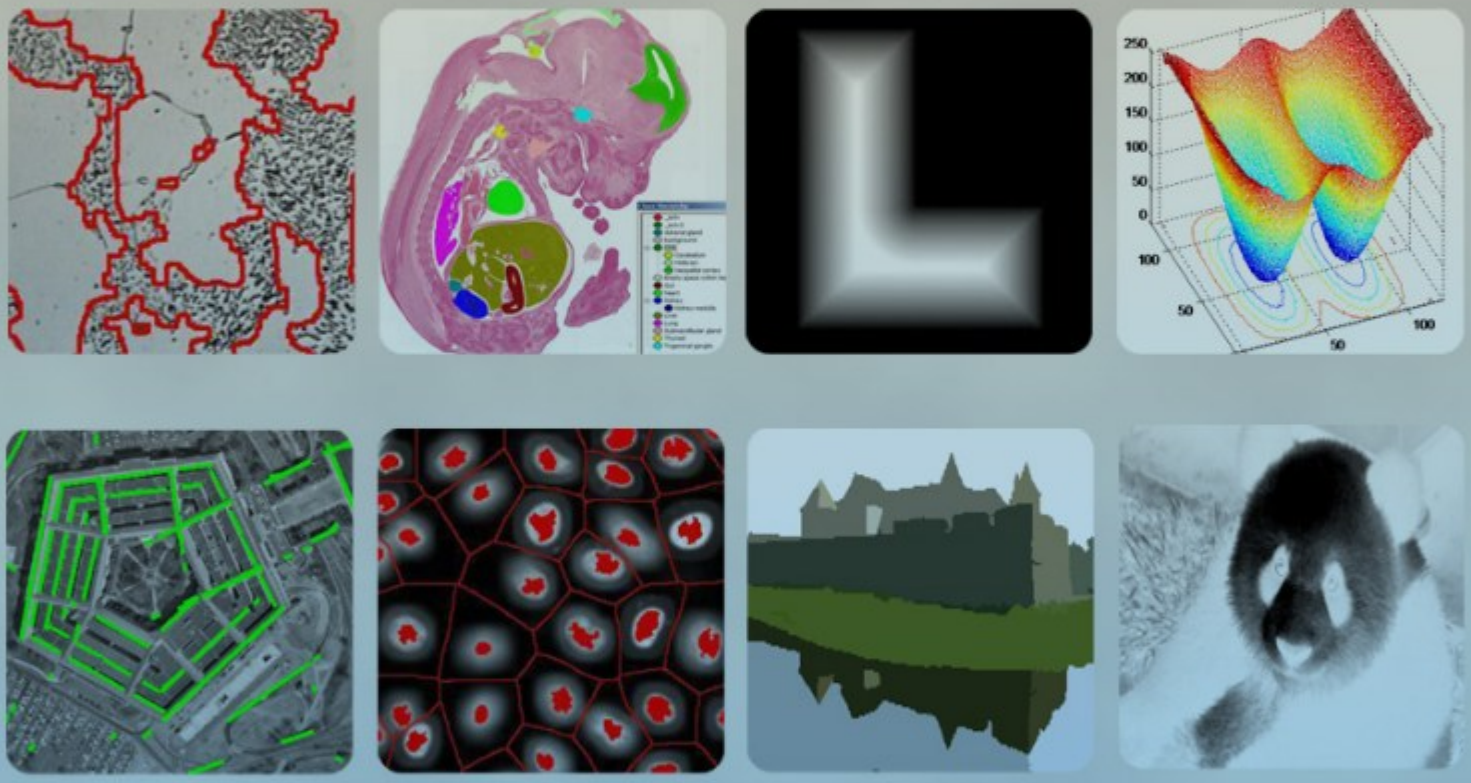
bookboon.com

Digital Image Processing: Part II

Huiyu Zhou; Jiahua Wu; Jianguo Zhang



Download free books at

bookboon.com

Huiyu Zhou, Jiahua Wu & Jianguo Zhang

Digital Image Processing
Part II

Download free ebooks at bookboon.com

Digital Image Processing – Part II
© 2010 Huiyu Zhou, Jiahua Wu, Jianguo Zhang & Ventus Publishing ApS
ISBN 978-87-7681-542-4

Download free ebooks at bookboon.com

Contents

	Prefaces	**7**
1.	**Colour Image Processing**	**8**
1.1	Colour Fundamentals	8
1.2	Colour Space	10
1.3	Colour Image Processing	12
1.4	Smoothing and sharpening	15
1.5	Image segmentation	18
1.6	Colour Image Compression	24
	Summary	26
	References	28
	Problems	28
2.	**Morphological Image Processing**	**30**
2.1	Mathematical morphology	30
2.1.1	Introduction	30
2.1.1	Binary images	30
2.1.2	Operators in set theory	30
2.1.3	Boolean logical operators	32
2.1.4	Structure element	32
2.2	Dilation and Erosion	34
2.2.1	Dilation	34
2.2.2	Erosion	35

Please click the advert

Download free ebooks at bookboon.com

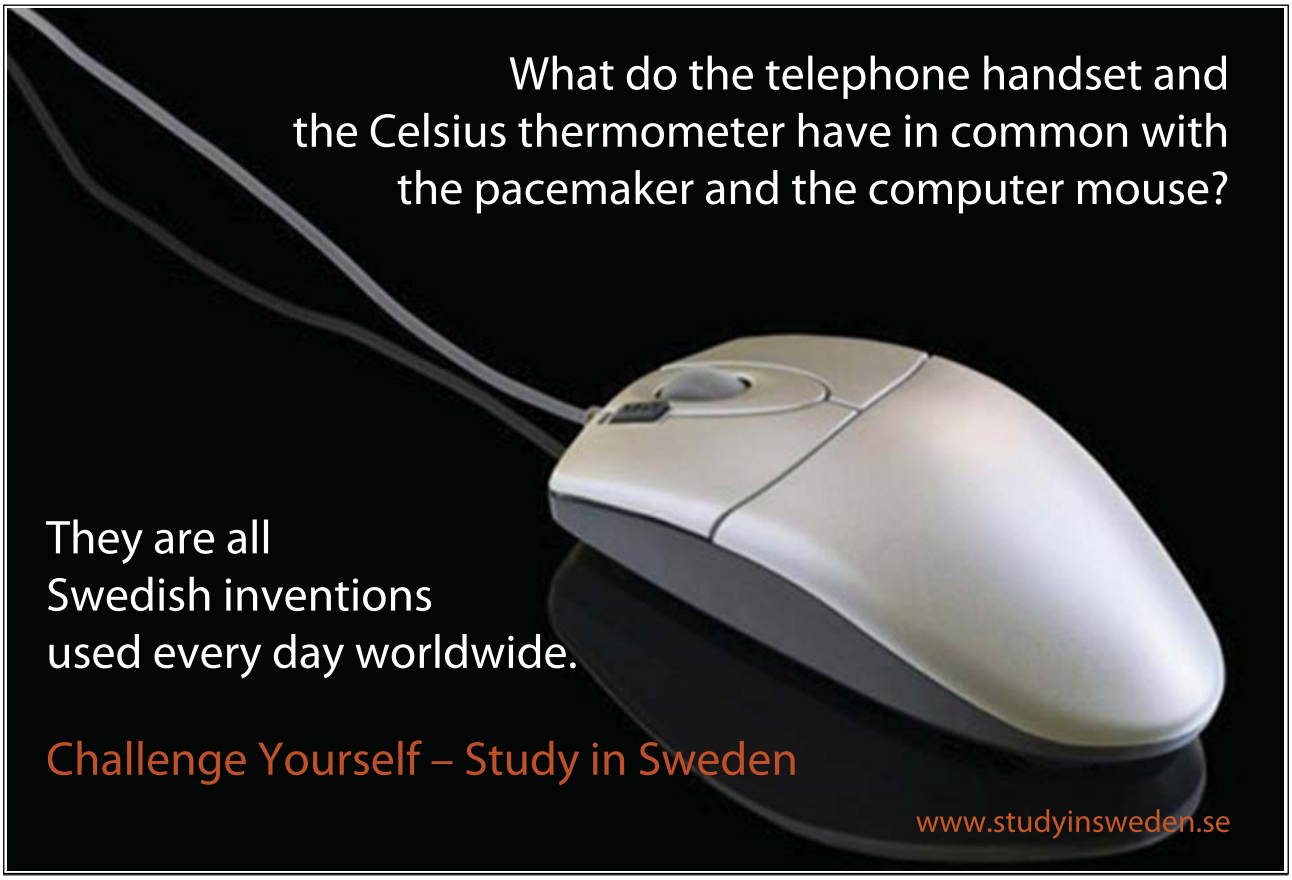
2.2.3	Properties of dilation and erosion	36
2.2.4	Morphological gradient	38
2.3	Opening and closing	40
2.3.1	Opening	40
2.3.2	Closing	41
2.3.3	Properties of opening and closing	44
2.3.4	Top-hat transformation	46
2.4	Hit-or-miss	48
2.5	Thinning and thicken	50
2.6	Skeleton	53
2.7	Pruning	55
2.8	Morphological reconstruction	57
2.8.1	Definition of morphological reconstruction	57
2.8.2	The choice of maker and mask images	59
	Summary	60
	References and further reading	60
	Problems	61
3.	**Image Segmentation**	**62**
3.1	Introduction	62
3.2	Image pre-processing – correcting image defects	62
3.2.1	Image smooth by median filter	63
3.2.2	Background correction by top-hat filter	63
3.2.3	Illumination correction by low-pass filter	65
3.2.4	Protocol of pre-process noisy image	65

Please click the advert

Download free ebooks at bookboon.com

3.3	Thresholding	66
3.3.1	Fundamentals of image thresholding	66
3.3.2	Global optimal thresholding	67
3.3.3	Adaptive local thresholding	69
3.3.4	Multiple thresholding	70
3.4	Line and edge detection	71
3.4.1	Line detection	71
3.4.2	Hough transformation for line detection	72
3.4.3	Edge filter operators	74
3.4.4	Border tracing - detecting edges of predefined operators	76
3.5	Segmentation using morphological watersheds	77
3.5.1	Watershed transformation	77
3.5.2	Distance transform	79
3.5.3	Watershed segmentation using the gradient field	80
3.5.4	Marker-controlled watershed segmentation	81
3.6	Region-based segmentation	83
3.6.1	Seeded region growing	83
3.6.2	Region splitting and merging	84
3.7	Texture-based segmentation	85
3.8	Segmentation by active contour	86
3.9	Object-oriented image segmentation	89
3.10	Colour image segmentation	90
	Summary	91
	References and further reading	91
	Problems	91

Please click the advert



Download free ebooks at bookboon.com

Prefaces

Digital image processing is an important research area. The techniques developed in this area so far require to be summarized in an appropriate way. In this book, the fundamental theories of these techniques will be introduced. Particularly, their applications in the image enhancement are briefly summarized. The entire book consists of three chapters, which will be subsequently introduced.

Chapter 1 reveals the challenges in colour image processing in addition to potential solutions to individual problems. Chapter 2 summarises state of the art techniques for morphological process, and chapter 3 illustrates the established segmentation approach.

Download free ebooks at bookboon.com

1. Colour Image Processing

1.1 Colour Fundamentals

Colour image processing is divided into two main areas: full colour and pseudo-colour processing. In the former group, the images are normally acquired with a full colour sensor such as a CCTV camera. In the second group, a colour is assigned to a specific monochrome intensity or combination of intensities.

People perceive colours that actually correspond to the nature of the light reflected from the object. The electromagnetic spectrum of the chromatic light falls in the range of 400-700 nm. There are three quantities that are used to describe the quality of a chromatic light source: radiance, luminance and brightness.

- Radiance: The total amount of energy that flows from the light source (units: watts);
- Luminance: The amount of energy an observer can perceive from the light source (lumens);
- Brightness: The achromatic notion of image intensity.

To distinguish between two different colours, there are three essential parameters, i.e. brightness, hue and saturation. Hue represents the dominant colour and is mainly associated with the dominant wavelength in a range of light waves. Saturation indicates the degree of white light mixed with a hue. For example, pink and lavender are relatively less saturated than the pure colours e.g. red and green.

A colour can be divided into brightness and chromaticity, where the latter consists of hue and saturation. One of the methods to specify the colours is to use the CIE chromaticity diagram. This diagram shows colour composition that is the function of x (red) and y (green). **Figure 1** shows the diagram, where the boundary of the chromaticity diagram is fully saturated, while the points away from the boundary become less saturated. **Figure 1** illustrates the colour gamut.

The chromaticity diagram is used to demonstrate the mixed colours where a straight line segment connecting two points in the chart defines different colour variations. If there is more blue light than red light, the point indicating the new colour will be on the line segment but closer to the blue side than the green side. Another representation of colours is to use the colour gamut, where the triangle outlines a range of commonly used colours in TV sets and the irregular region inside the triangle reflects the results of the other devices.

Download free ebooks at bookboon.com

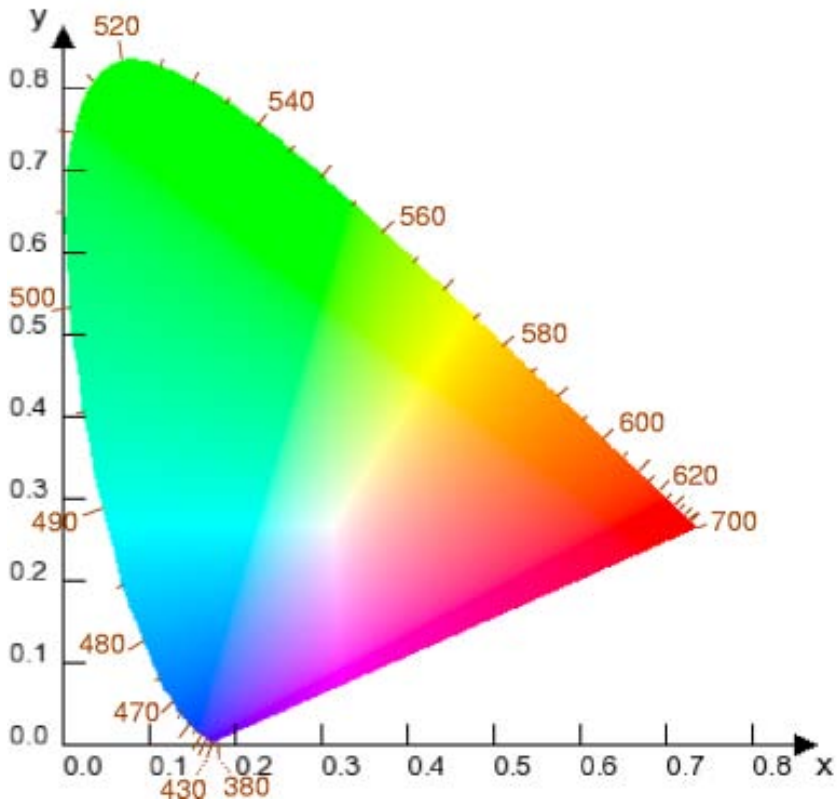


Figure 1 Illustration of the CIE chromaticity diagram ([8]).

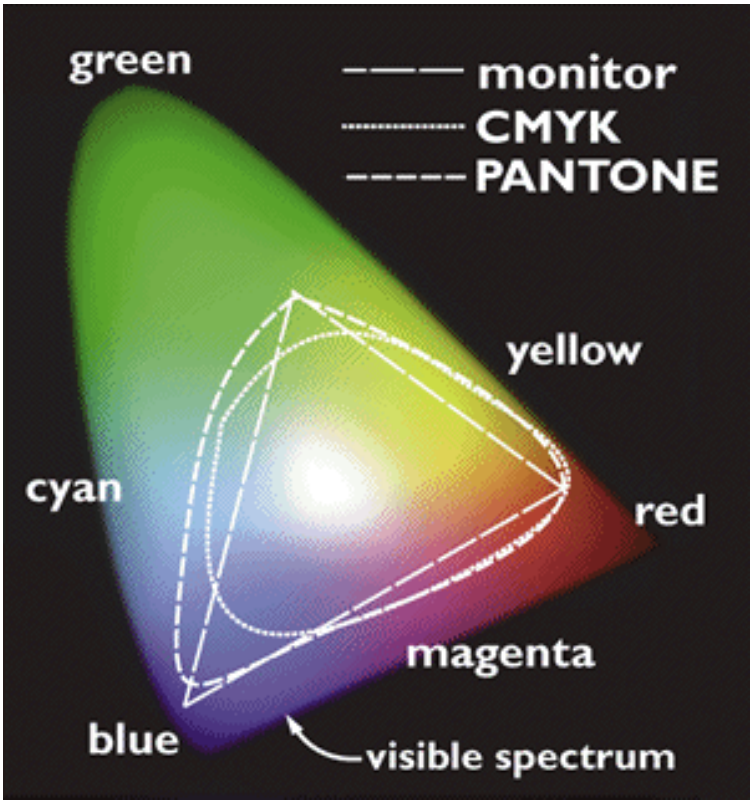


Figure 2 Illustration of the colour gamut ([9]).

1.2 Colour Space

Colour space or coulour model refers to a coordinate system where each colour stands for a point. The often used colour models consist of the RGB (red, green abd blue) model, CMY (cyan, magentia and yellow) model, CMYK (cyan, magenta, yellow and black) model and HIS (hue, saturation and intensity) model.

RGB model: Images consist of three components. These three components are combined together to produce composite colourful images. Each image pixel is formed by a number of bits. The number of these bits is namely pixel depth. A full colour image is normally 24 bits, and therefore the totoal number of the colours in a 24-bit RGB image is 16,777,216. **Figure 3** illustrates the 24-bit RGB colour cube that describes such a colour cube.

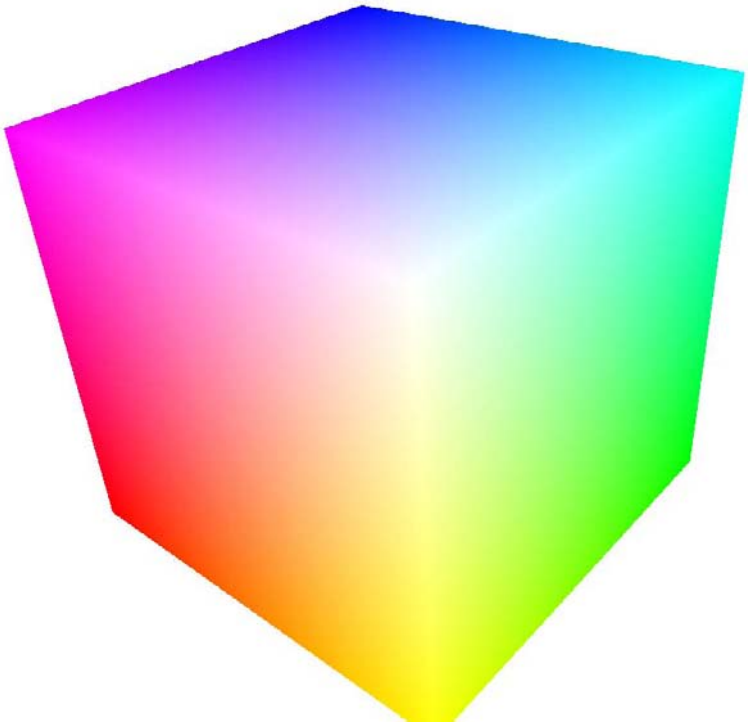


Figure 3 A colour cube ([10]).

CMY/CMYK colour models: These models contain cyan, magenta and yellow components, and can be formed from RGB using the following equation:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.2.1}$$

HSI colour models: These models work as follows:

$$H = \begin{cases} \theta \\ 360 - \theta \end{cases} \tag{1.2.2}$$

Where the upper case is the result of $B \leq G$, and the lower case results from $B \geq G$. In the meantime,

$$\theta = \cos^{-1}\left\{\frac{0.5[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\right\} \tag{1.2.3}$$

The saturation is

$$S = 1 - \frac{3}{R+G+B}[\min(R,G,B)] \tag{1.2.4}$$

The intensity is given by

$$I = 1/3(R+G+B) \tag{1.2.5}$$

Figure 4 shows the separation of hue, sauration and intensity of a color image.

(a)

(b)

(c)

Figure 4 Illustration of Hue (a), Saturation (b) and Intensity (c) of a colour image.

1.3 Colour Image Processing

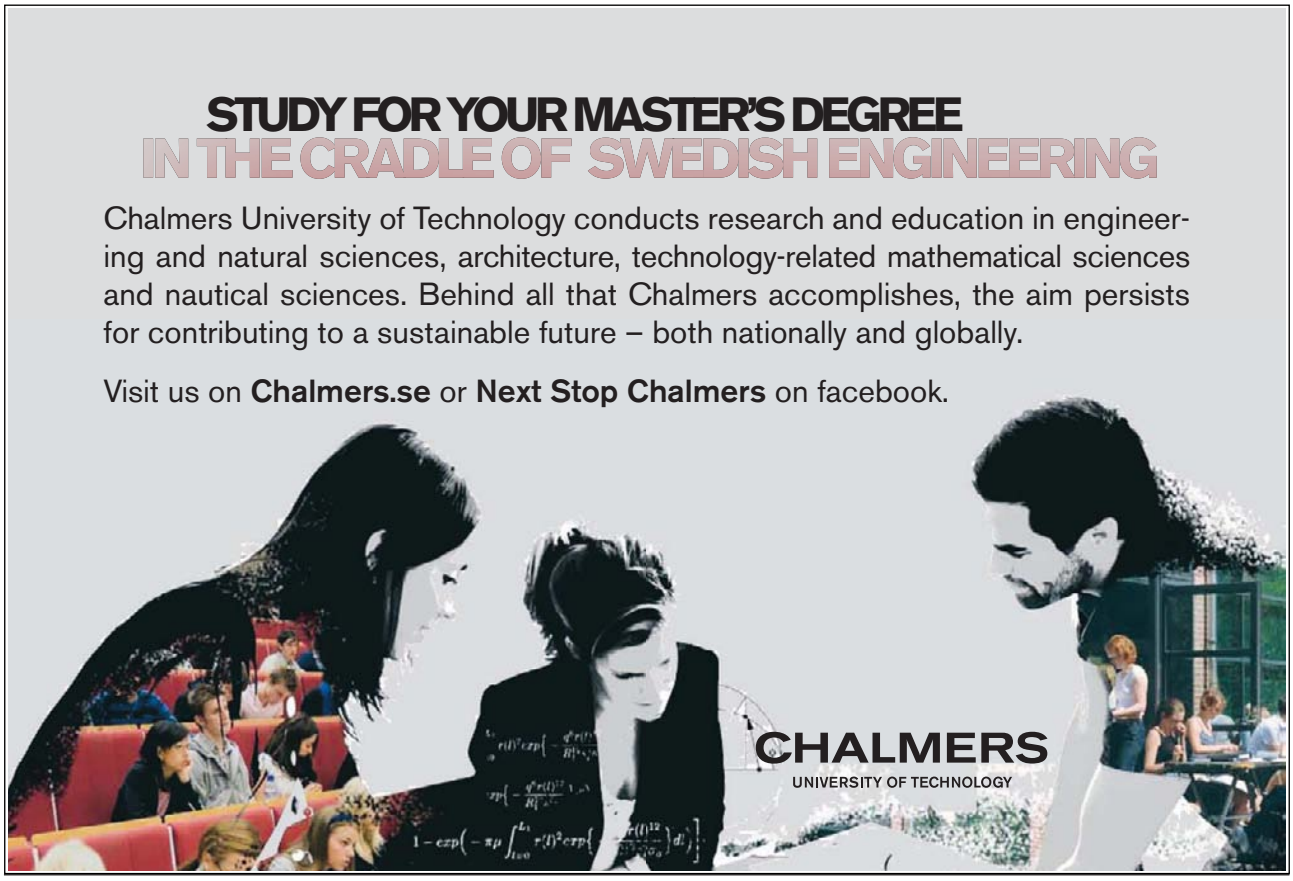
Colour image processing consists of pseudo- and full-colour image processing. Pseudo-colour image processing is about the assignment of colours to gray levels according to certain evidence. To do so, one of the options is to use a technique called intensity slicing. This is a simple but effective approach. In an image domain of intensity and spatial coordinates, the intensity amplitudes are used to assign the corresponding colours: The pixels with gray levels larger than the pre-defined threshold will be assigned to one colour, and the remainder will be assigned to another colour. One of the examples using the intensity slicing technique is shown in **Figure 5**, where 10 colours have been assigned to the various slices.

(a)

(b)

Figure 5 Illustration of intensity slicing and colour assignment.

Please click the advert



Download free ebooks at bookboon.com

Full-colour image processing is more complex than the pseudo-colour case due to the three colour vectors. First of all, one basic manipulation of colour images is namely colour transformation. For example, RGB is changed to HSI and vice versa.

If a colour transformation can be expressed as follows:

$$\chi_i = T_i(\tau_1, \tau_2, ..., \tau_n) \tag{1.3.1}$$

where $i = 1, 2, \ldots, n$, χ is target colour image, τ is the original colour image and T is the transformation function. In a very simple case, the three components in the RGB colour space can be

$$\chi_i = k\tau_i \tag{1.3.2}$$

where $i = 1, 2, 3$ and k is a constant. Similarly, the CMY space has the following linear transformation:

$$\chi_i = k\tau_i + (1-k) \tag{1.3.3}$$

Figure 6 demonstrates the colour transformation using three common techniques.

(a)

Hue Saturation Intensity

Figure 6 Examples of grouping colour components.

On the other hand, like intensity slicing, colour slicing is such a technique that

$$\chi_i = \begin{cases} 0.5, \\ \tau_i \end{cases} \tag{1.3.4}$$

where the former condition is $[|\tau_j - a_j|] > d/2$ (a colour cube with a width d).

Now the main attention is shifted to histogram analysis which has played a key role in image transformation. Particularly, histogram equalization is an example. To produce an image with an uniform histogram of colour values, one of the possible ways is to spread the colour intensities uniformly while leaving the colour values unvaried. See the outcome of the histogram equalization, shown in **Figure 7**.

Figure 7 Colour histogram equalisation.

1.4 Smoothing and sharpening

Smoothing and sharpening are two basic manipulation tools on colour images. They are two reverse processes, where the latter is a procedure of reproducing image intensities by adding more details and the former refers to an averaging process within a window.

Download free ebooks at bookboon.com

The smoothing process can lead to the mean colour intensity as follows:

$$\bar{I}(x,y) = \begin{bmatrix} \frac{1}{\lambda} \sum_{(x,y)\in w} R(x,y) \\ \frac{1}{\lambda} \sum_{(x,y)\in w} G(x,y) \\ \frac{1}{\lambda} \sum_{(x,y)\in w} B(x,y) \end{bmatrix} \tag{1.4.1}$$

This smoothing can be illustrated in **Figure 8**, where RGB images of the original image are shown accompanying the mean and difference images. The strategy used in the averaging procedure is to apply a Gaussian mask (width = 3) to the original image.

Please click the advert

GCHQ
it's an interesting world

Get under the skin of it.

Graduate opportunities
Cheltenham | £24,945 + benefits

One of the UK's intelligence services, GCHQ's role is two-fold: to gather and analyse intelligence which helps shape Britain's response to global events, and, to provide technical advice for the protection of Government communication and information systems. In doing so, our specialists – in IT, internet, engineering, languages, information assurance, mathematics and intelligence – get well beneath the surface of global affairs. If you thought the world was an interesting place, you really ought to explore our world of work.

TOP 100 | TOP GOVERNMENT EMPLOYER | INVESTORS IN PEOPLE

www.careersinbritishintelligence.co.uk

Applicants must be British citizens. GCHQ values diversity and welcomes applicants from all sections of the community. We want our workforce to reflect the diversity of our work.

Download free ebooks at bookboon.com

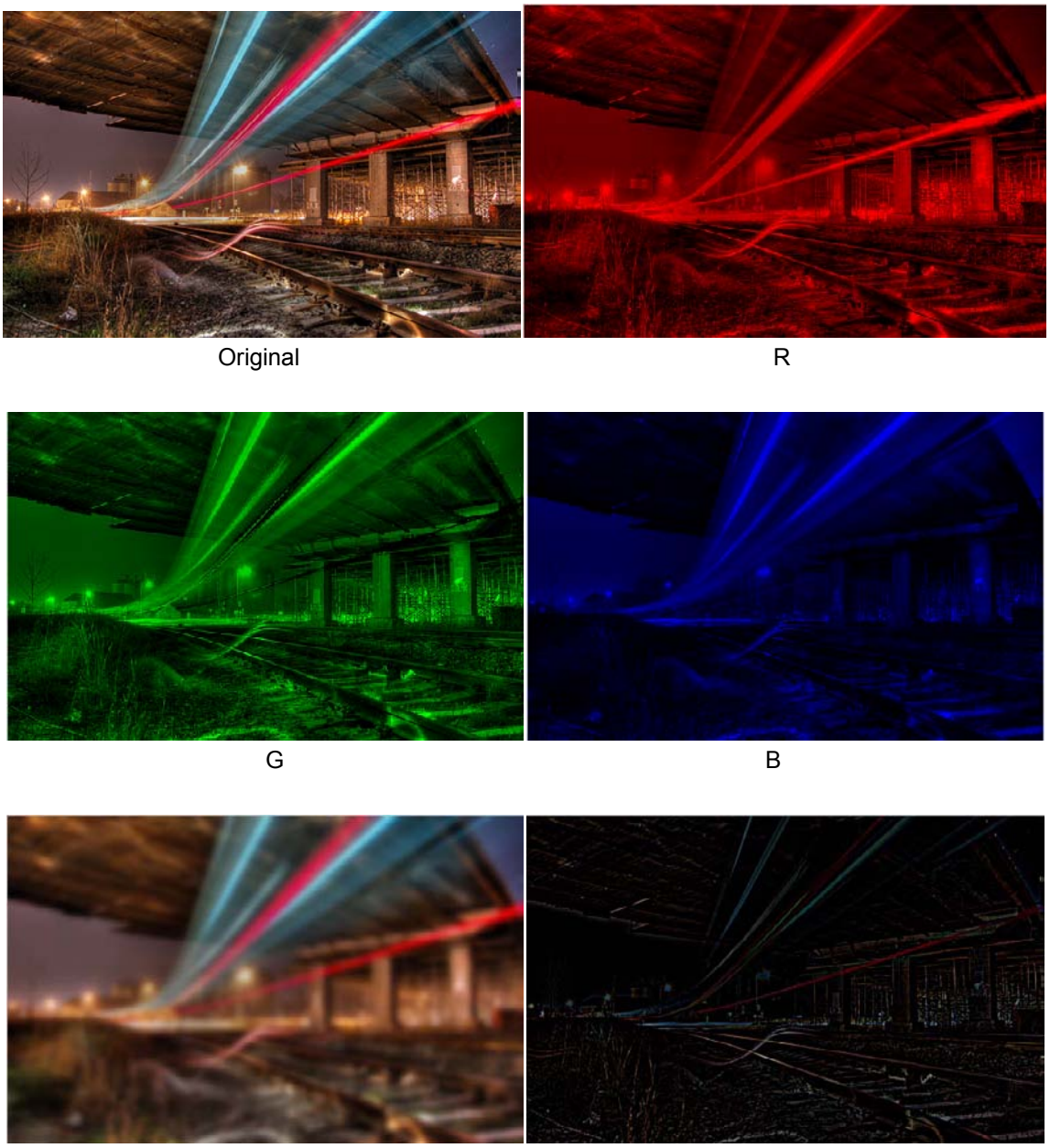


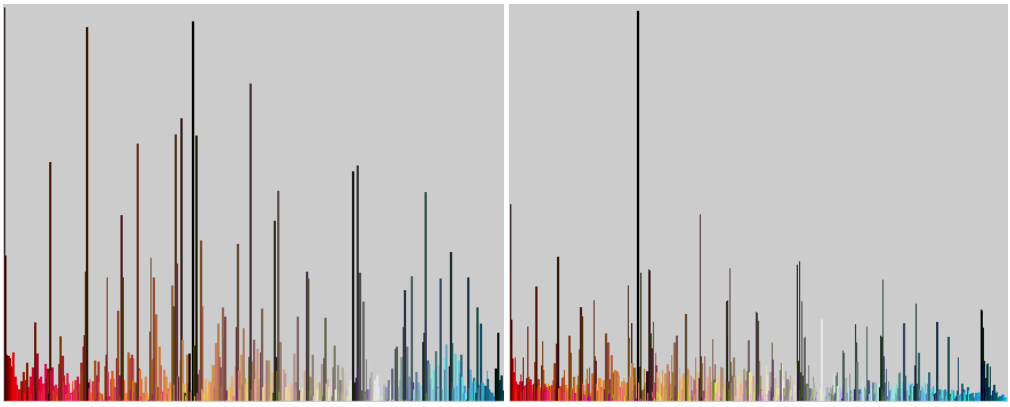
Figure 8 Image smoothing and the individual components.

A simple sharpening stage is provided as an example. This process involves the Laplacian transformation of an image. In a RGB domain, the sharpening outcome is:

$$\nabla^2[I(x,y)] = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix} \tag{1.4.2}$$

Figure 9 illustrates the sharpened image and two colour distributions before and after the sharpening. It is observed that the sharpening process has changed the colour distribution of the intensities.

(a)



(b) (c)

Figure 9 Image sharpening and colour bars: (a) is the sharpened image, (b) and (c) are the histograms before and after the sharpening.

1.5 Image segmentation

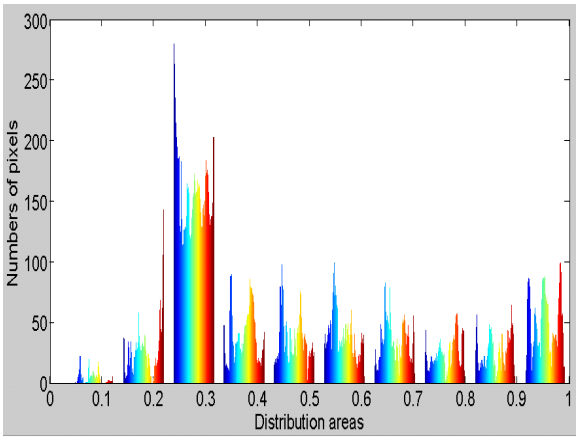
In this subsection, image segmentation is mainly conducted based on the colour differentiation. It is a grouping process that enables image pixels to be separated according to their colour intensities. One of the segmentation schemes is hard thresholding (or namely binarisation), where a threshold is determined manually or empirically. For example, a colour image can be segmented according to its histogram of intensity values (**Figure 10**). However, this segmentation easily leads to mistaken grouping outcomes if the image pixels are cluttered. In addition, it mainly relies on the experience of a professional user. To reduce erroneous segmentations, soft thresholding techniques are hence developed. These approaches perform automatic and adaptive determination of the thresholds. In this section, only a couple of examples of the "soft" thresholding approaches will be presented, besides the classical neural networks, genetic and evolutionary algorithms.

(a)

(b)

(c)

(d)



(e)

Figure 10 Illustration of a colour image and HSV decomposition: (a) original image, (b) hue, (c) saturation, (d) intensity value and (e) histogram.

K-means segmentation

K-means segmentation is a technique that aims to partition observations into a number of clusters where each observation belongs to the cluster with the nearest mean. The observations closer to a specific cluster will be assigned a higher weight and this helps remove the effects of some outliers. Suppose that there is a set of observations ($x_1, x_2, \ldots, x_n$), where each observation can be a multi-dimensional vector. Therefore, these observations will be grouped into k sets $S = (S_1, S_2, \ldots, S_k)$ which must satisfy the following minimization of sum of squares [11]:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - v_i \|^2 \tag{1.5.1}$$

where v_i is the mean of S_j.

The standard algorithm to achieve this K-means segmentation is executed in an iterative style. Given an initial state of K means $m_1^1, \ldots, m_k^1$, which can be obtained through empirical study or random guess, we then conduct the following steps. Then, the entire scheme operates as follows:

Initialization step: Each observation is assigned to the cluster with the closest mean.

$$S_i^t = \{x_j : \| x_j - m_i^t \| \leq \| x_j - m_{i^+}^t \| \; for_all_i^+ = 1, \ldots, k\} \tag{1.5.2}$$

Please click the advert

HORIZONS UNIVERSITY

In Paris or Online
International programs taught by professors and professionals from all over the world

BBA in Global Business
MBA in International Management / International Marketing
DBA in International Business / International Management
MA in International Education
MA in Cross-Cultural Communication
MA in Foreign Languages

Innovative – Practical – Flexible – Affordable

Visit: www.HorizonsUniversity.org
Write: Admissions@horizonsuniversity.org
Call: 01.42.77.20.66

www.HorizonsUniversity.org

Download free ebooks at bookboon.com

Update step: Calculate the new means to be the centroid of the observations in the cluster.

$$m_i^{t+1} = \frac{1}{|S_i^t|} \sum_{x_j \in S_i^t} x_j \tag{1.5.3}$$

These two steps will be iterated until a pre-defined threshold is met. This algorithm is illustrated in **Figure 10**.

As an extension and variant of K-means, fuzzy c-means recently has been well investigated. This algorithm works without a need to assign the initial locations of the cluster centres. Due to the limit of the pagination only its performance is demonstrated in this section (**Figure 10**).

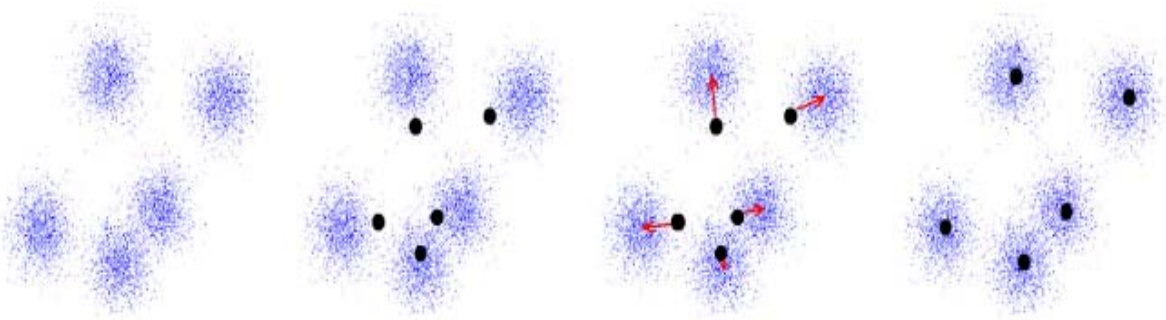


Figure 11 Illustration of K-means segmentation algorithm, where dots are the centres and red arrows refer to the moving direction.

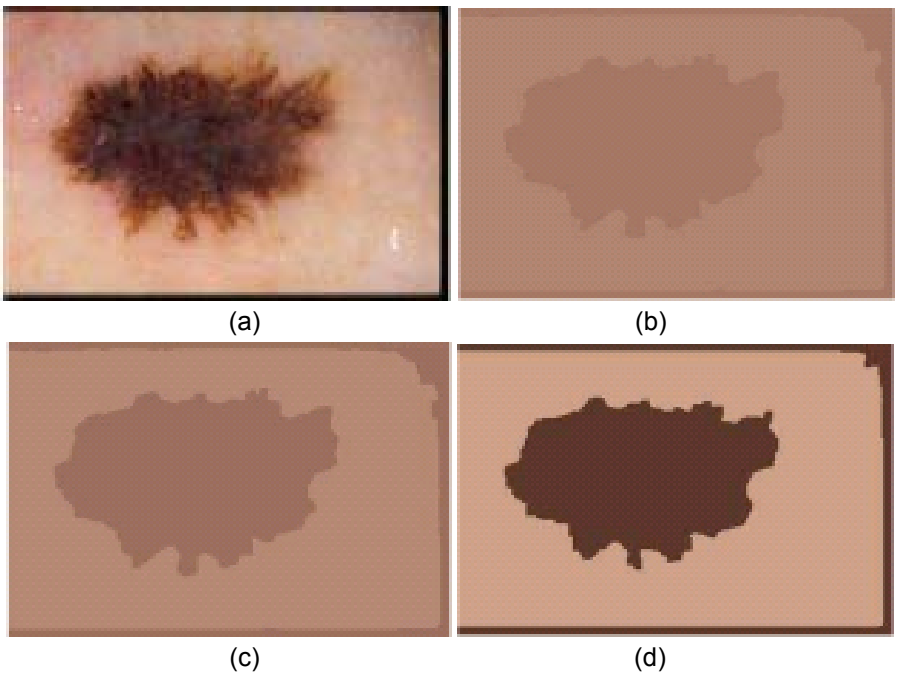


Figure 12 An evolving fuzzy C-means segmentation process.

Mean shift segmentation

Mena shift segmentation is a segmentation/clustering algorithm recently developed. There is no assumption made for the probability distributions. The aim of this algorithm is to find the local maxima of the probability density given by the observations. The algorithm of the mean shift segmentation is followed:

Download free ebooks at bookboon.com

- Start from a random region;
- Determine a centroid of the estimates;
- Continuously move the region towards the location of the new centroid;
- Repeat the iteration until convergence.

Given a set of observations x, a kernel function k and a constant c_k, then the probability distribution function can be expressed as follows:

$$K(x) = c_k k(\| x \|^2) \tag{1.5.4}$$

The kernel function can be an Epanechnikov kernel which has the form like this:

$$k(g) = \begin{cases} 1-g \\ 0 \end{cases} \tag{1.5.5}$$

where $g = \|x\|^2$. The upper case is true when $g \leq 1$; otherwise the lower case stands. The kernel density of the estimated states of the data is described by the following equation:

$$\tilde{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1.5.6}$$

where d is the dimension of the data. When the algorithm reaches a maxima or minima in the iteration, this equation must be satisfied:

$$\nabla \tilde{f}(x) = 0 \tag{1.5.7}$$

Hence,

$$\nabla \tilde{f}(x) = \frac{2c_k}{nh^{d+2}} \sum_{i=1}^{n} \hat{K}_i \left(\frac{\sum_{i=1}^{n} x_i \hat{K}_i}{\sum_{i=1}^{n} \hat{K}_i} - x \right) = 0 \tag{1.5.8}$$

where the intermediate functions

$$\begin{cases} \hat{K}(g) = k'(g) \\ \hat{K}_i = K(\| x - x_i) / h \|^2) \end{cases} \tag{1.5.9}$$

Finally, the mean shift vector is obtained in the computation loop:

$$m(x) = \frac{\sum_{i=1}^{n} x_i \hat{K}_i}{\sum_{i=1}^{n} \hat{K}_i} - x \tag{1.5.10}$$

To demonstrate the performance of the mean shift scheme, **Figure 13** shows some examples of mean shift segmentation. In general, the segmentation results reflect the embedded clusters in the images and therefore the mean shift algorithm works successfully.

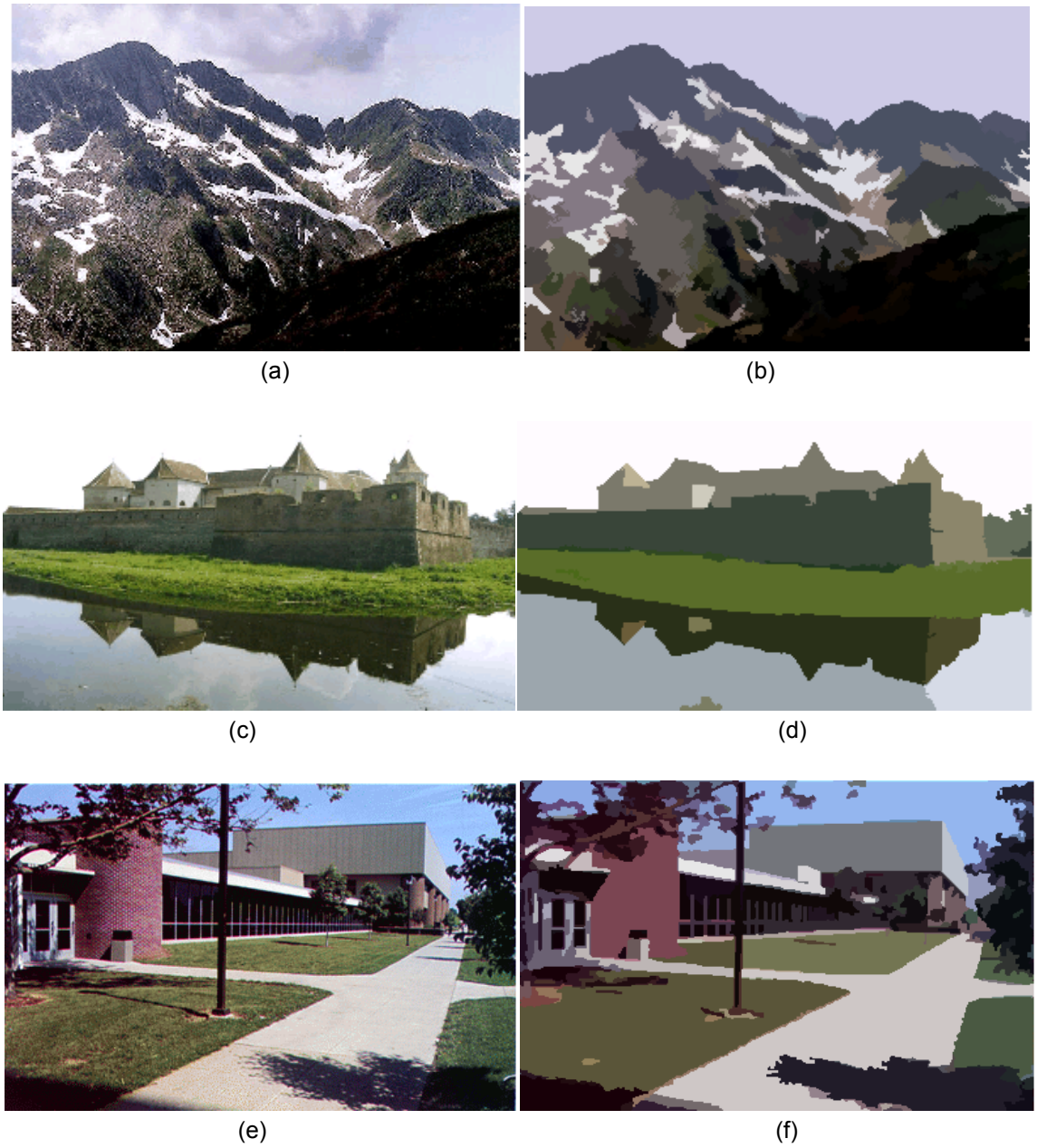


Figure 13 Examples of mean shift segmentation (image courtesy of [12]).

1.6 Colour Image Compression

In this subsection, image compression is discussed. The reason why this issue is important to talk about is the fact that the number of bits of a colour image is three or four times greater than its counterpart in gray level style. Storage and transmission of this colour image takes tremendous time with a more complicated process, e.g. encoding and decoding. If this colour image can be reduced in terms of its bits, the relevant process will be much simplified.

A comprehensive introduction to the colour image compression is non-trivial and this will be detailed in a later study and other references. In this section, some recently developed techniques are briefly introduced. These techniques are mainly comprised of two types, “lossless” and “lossy” compression. Digital Video Interface (DVI), Joint Photographic Experts Group (JPEG) and Motion Pictures Experts (MPEG) are the widely used techniques. No doubt, the lossy techniques normally provide greater compression ratio than the lossless ones.

Please click the advert

Download free ebooks at bookboon.com

Lossless compression: These methods aim to retain lower compression ratios but preserve all the pixels in the original image. The bits of the resulting image are larger than the lossy compression. The common methods are Run-Length Encoding (RLE), Huffman encoding, and entropy coding. RLE checks the image stream and inserts a special token each time a chain of more than two equal input tokens is found. Huffman encoding assigns a longer code word to a less common element, while a weighted binary tree is built up according to their rate of occurrence. In the entropy coding approaches, if a sequence is repeated after a symbol is found, then only the symbol is part of the coded data and the sequence of tokens referred to the symbol can be decoded later on.

Lossy compression: These approaches retain higher compression rates but sacrifice with a less resolution in the final compressed image. JPEG is the best known lossy compression standard and widely used to compress still images. The concept behind JPEG is to segregate the information in the image by levels of their importance, and discard the less important information to reduce the overall quantity of data. Another commonly used coding scheme is namely "transform coding" that subdivides an N-by-N image into smaller n-by-n blocks and then performs an unitary transform on each block. The objectives of the transform are to de-correlate the original image, which results in the image energy being distributed over a small amount of transform coefficients. Typical schemes consist of discrete cosine transform, wavelet and Gabor transforms. **Figure 14** demonstrates the performance of a wavelet analysis in the image compression and reconstruction of the compressed image.

(a)

(b)

(c)

Figure 14 Colour image compression using wavelet analysis: (a) original, (b) compressed image and (c) reconstructed image.

The algorithm of the transform coding can be summarized as follows:

- Image subdivision
- Image transformation
- Coefficient quantization
- Huffman encoding

Another commonly used compression scheme is vector quantization. This is a transform from a higher dimensional Euclidean space to a finite subset. The subset can be the vector codebook. One of the best vector quantization algorithms is described as follows:

- Subdivide the training set into N groups, which are associated with the N codebook letters.
- The centroids of the partitioned regions become the updated codebook vectors.
- Compute the average distortion. If the percent reduction in the distortion is less than a pre-defined threshold, then stop.

In addition, segmented image coding and fractal coding schemes can be used to handle different circumstances. For example, segmented image coding considers images to be composed of slowly varying image intensity. These slowly moving regions will be identified and then used as the main structure of the encoded image.

Summary

In this chapter, the concepts of radiance, luminance and brightness have been introduced. The chromaticity diagram was used to illustrate the complexity of colours. In the colour space, RGB, CMY and HSI colour models have been summarised. Afterwards, intensity slicing and colour assignment were also introduced. To further improve the quality of a colour image colour equalisation was presented to generate uniformly distributed colour intensities.

Download free ebooks at bookboon.com

Colour smoothing and sharpening are two important methods that can be used to enhance the quality of an image. One example of smoothing by using a Gaussian mask is denoted. The image sharpening is demonstrated using a Laplacian operator. In the following sections, image segmentation and compression have been respectively discussed. The former include two examples, k-means and mean shift. The latter has looseless and lossy compression techniques. In particular, the application of a wavelet analysis based compression is shown.

In general, image smoothing/sharpening, segmentation and compression are the key contents in this section. In spite of their brief introduction, these descriptions demonstrate the necessity of these algorithms in real life. In addition, it has been observed that further investigation for a better image quality must be taken into account. These issues will be addressed on the later sections.

Please click the advert

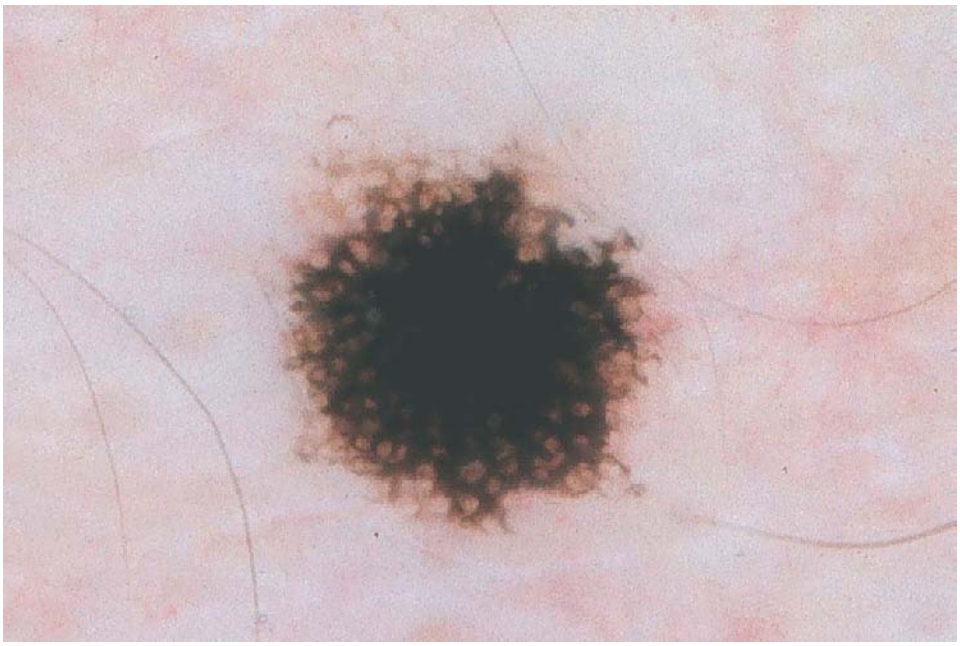
Download free ebooks at bookboon.com

References

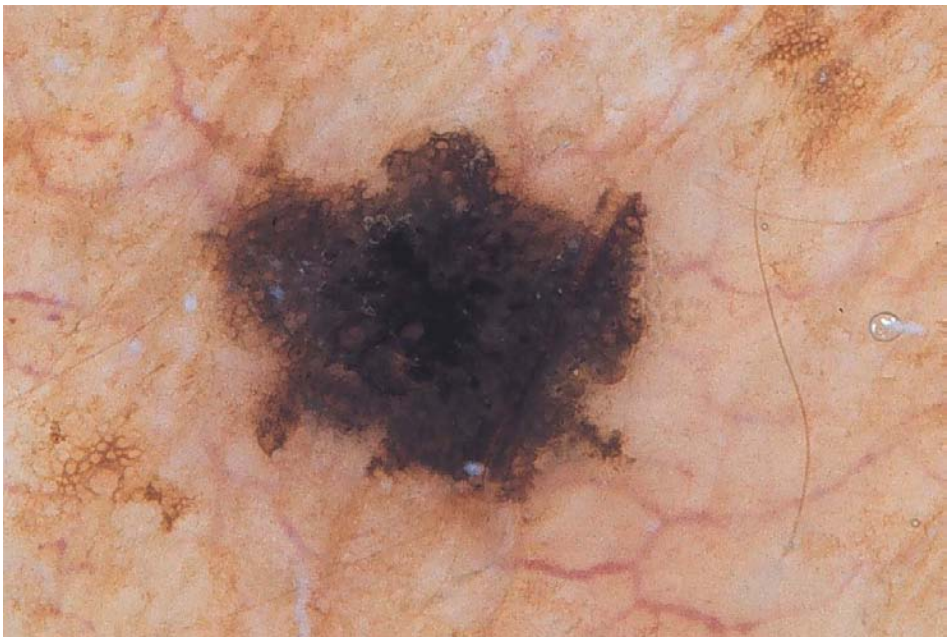
[10] www.knowledgerush.com/kr/encyclopedia/Colour/, accessed on 30 September, 2009.
[11] http://dx.sheridan.com/advisor/cmyk_color.html, accessed on 30 September, 2009.
[12] http://luminous-landscape.com/forum/index.php?s=75b4ab4d497a1cc7cca77bfe2ade7d7d&showtopic=37695&st=0&p=311080&#entry311080, accessed on 30 September, 2009.
[13] http://en.wikipedia.org/wiki/K-means_clustering, accessed on 4 October, 2009.
[14] D. Comaniciu, P. Meer: *Mean Shift: A Robust Approach toward Feature Space Analysis*, IEEE Trans. Pattern Analysis Machine Intell., Vol. 24, No. 5, 603-619, 2002.

Problems

(1) What is a colour model?
(2) What is image smoothing and sharpening? Try to apply Gaussian smoothing and edge sharpening respectively to following image:



(3) How to perform image segmentation? Hints: One example can be used to explain the procedure.
(4) Try to apply mean shift algorithms for image segmentation of the following image:



(5) Is this a true statement? Image compression is a process of reducing image size.

(6) Can you summarise the algorithm of RLE compression?

Please click the advert

Download free ebooks at bookboon.com

2. Morphological Image Processing

2.1 Mathematical morphology

2.1.1 Introduction

Mathematical morphology is a tool for extracting geometric information from binary and gray scale images. A shape probe, known as a *structure element (SE)*, is used to build an image operator whose output depends on whether or not this probe fits inside a given image. Clearly, the nature of the extracted information depends on the shape and size of the structure element. Set operators such as union and intersection can be directly generalized to gray-scale images of any dimension by considering the point-wise maximum and minimum operators.

Morphological operators are best suited to the selective extraction or suppression of image structures. The selection is based on their shape, size, and orientation. By combining elementary operators, important image processing tasks can also be achieved. For example, there exist combinations leading to the definition of morphological edge sharpening, contrast enhancement, and gradient operators.

2.1.1 Binary images

Morphological image transformations are image-to-image transformations, that is, the transformed image has the same definition domain as the input image and it is still a mapping of this definition domain into the set of nonnegative integers.

A widely used image-to-image transformation is the threshold operator T, which sets all pixels x of the input image f whose values lie in the range $[T_i, T_j]$ to 1 and the other ones to 0:

$$[T_{[t_i,t_j]}(f)](x) = \begin{cases} 1 & \text{if } t_i \le f(x) \le t_j \\ 0 & \text{otherwise} \end{cases} \tag{2.1.1}$$

It follows that the threshold operator maps any gray-tone image into a binary image.

2.1.2 Operators in set theory

The field of mathematical morphology contributes a wide range of operators to image processing, all based around a few simple mathematical concepts from *set theory*. Let A be a set, the elements of which are pixel coordinates (x, y), If $w = (x, y)$ is an element of A, then we write

$$w \in A \tag{2.1.2}$$

Similarly, if w is not an element of A, we write

$$w \notin A \tag{2.1.3}$$

Download free ebooks at bookboon.com

The set B of pixel coordinates that satisfy a particular condition is written as

$$B = \{w \mid condition\} \tag{2.1.4}$$

The basic set operators are ***union*** $\cup$ and ***intersection*** $\cap$. For *binary image*, they are denoted by

$$\begin{aligned} C &= A \bigcup B \\ C &= A \bigcap B \end{aligned} \tag{2.1.5}$$

For *gray level images*, the union becomes the point-wise maximum operator $\vee$ and the intersection is replaced by the point-wise minimum operator $\wedge$:

$$\begin{aligned} &\text{union}: && (f \vee g)(x) = \max[f(x), g(x)] \\ &\text{intersection}: && (f \wedge g)(x) = \min[f(x), g(x)] \end{aligned} \tag{2.1.6}$$

Another basic set operator is ***complementation***. For *binary images*, the set of all pixel coordinates that do not belong to set A, denote A^c, is given by

$$A^c = \{w \mid w \notin A\} \tag{2.1.7}$$

For *gray level images*, the complement of an image f, denoted by f^c, is defined for each pixel x as the maximum value of the data type used for storing the image minus the value of the image f at position x:

$$f^c(x) = t_{\max} - f(x) \tag{2.1.8}$$

The complementation operator is denoted by C: $C(f) = f^c$.

For *binary images*, set ***difference*** between two sets A and B, denoted by

$$A - B \tag{2.1.9}$$

For *gray level images*, the set difference between two sets X and Y, denoted by $X \setminus Y$, is defined as the intersection between X and the complement of Y

$$X \setminus Y = X \bigcap Y^c \tag{2.1.10}$$

The ***reflection*** of set A, denoted $\hat{A}$, is define as

$$\hat{A} = \{w \mid w = -a, \; for \; a \in A\} \tag{2.1.11}$$

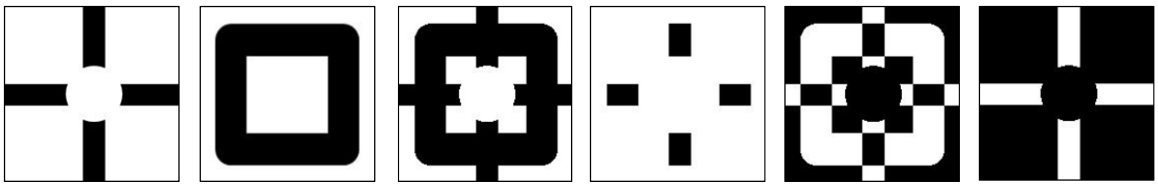
Finally, the ***translation*** of set A by *point* $z = (z_1, z_2)$, denoted $(A)_z$, is defined as

$$(A)_z = \{c \mid c = a + z, \; for \; a \in A\} \tag{2.1.12}$$

Download free ebooks at bookboon.com

2.1.3 Boolean logical operators

In the case of binary images, the set operators become Boolean logical operators, such as "***AND***", "***OR***", "***XOR***" (exclusive "OR") and "***NOT***". The "*union*" operation, $A \cup B$, for example, is equivalent to the "OR" operation for binary image; and the "*intersection*" operator, $A \cap B$, is equivalent to the "AND" operation for binary image. **Figure 15** illustrated each of these basic operations. **Figure 16** shows a few of the possible combinations. All are performed pixel by pixel.



(a) (b) (c) (d) (e) (f)

Figure 15 Basic Boolean logical operators. (a) Binary image A; (b) Binary image B; (c) A AND B; (d) A OR B; (e) A XOR B; (f) NOT A.

(a) (b) (c) (d) (e) (f)

Figure 16 Combined Boolean logical operators. (a) (NOT A) AND B; (b) A AND (NOT B); (c) (NOT A) AND (NOT B); (d) NOT (A AND B); (e) (NOT A) OR B; (f) A OR (NOT B).

2.1.4 Structure element

A *structure element (SE)* [18] is nothing but a small set used to probe the image under study. An origin must also be defined for each SE so as to allow its positioning at a given point or pixel: an SE at point x means that its origin coincides with x. The elementary isotropic SE of an image is defined as a point and its neighbours, the origin being the central point. For instance, it is a centred 3×3 window for a 2-D image defined over an 8-connected grid. In practice, the shape and size of the SE must be adapted to the image patterns that are to be processed. Some frequently used SEs are discussed hereafter (**Figure 17**).

- **Line segments**: often used to remove or extract elongated image structures. There are two parameters associated with line SEs: length and orientation.
- **Disk**: due to their isotropy, disks and spheres are very attractive SEs. Unfortunately, they can only be approximated in a digital grid. The larger the neighbourhood size is, the better the approximation is.
- **Pair of points**: in the case of binary images, erosion with a pair of points can be used to estimate the probability that points separated by a vector v are both object pixels, that is, by measuring the number of object pixels remaining after the erosion. By varying the modulus of v, it is possible to highlight periodicities in the image. This principle applies to gray-scale images.

Download free ebooks at bookboon.com

- **Composite structure elements**: a composite or two-phase SE contains two non-overlapping SEs sharing the same origin. Composite SEs are considered for performing hit-or-miss transforms (see Section 2.4).
- **Elementary structuring elements**: many morphological transformations consist in iterating fundamental operators with the elementary symmetric SE, that is, a pixel and its neighbours in the considered neighbourhood. Elementary triangles are sometimes considered in the hexagonal grid and 2×2 squares in the square grid. In fact, the 2×2 square is the smallest isotropic SE of the square grid but it is not symmetric in the sense that its centre is not a point of the digitization network.

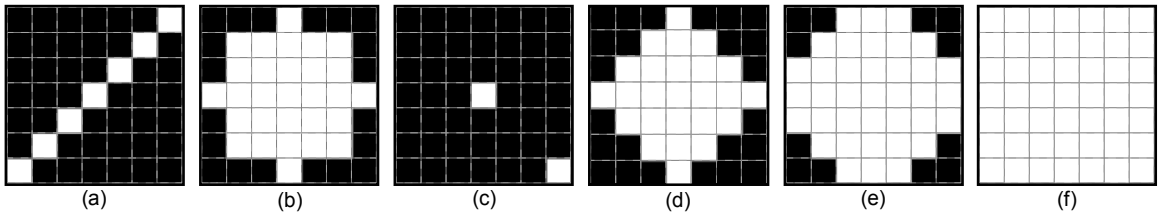


Figure 17 Some typical structure elements. (a) A line segment SE with the length 7 and the angle 45°; (b) A disk SE with the radius 3; (c) A pair of points SE containing two points with the offset 3; (d) A diamond-shaped SE; (e) A octagonal SE; (f) A 7×7 square S.

Please click the advert

Destination MMU

Manchester Metropolitan University

MMU is proud to be one of the most popular universities in the UK. Some 34,000 students from all parts of the globe select from its curricula of over 1,000 courses and qualifications.

We are based in the dynamic yet conveniently compact city of Manchester, located at the heart of a sophisticated transport network including a major international airport on the outskirts. Parts of the campus are acclaimed for their architectural style and date back over 150 years, in direct contrast to our teaching style which is thoroughly modern, innovative and forward-thinking.

MMU offers undergraduate and postgraduate courses in the following subject areas:

- Art, Design & Performance
- Computing, Engineering & Technology
- Business & Management
- Science, Environmental Studies & Geography
- Law, Education & Psychology
- Food, Hospitality, Tourism & Leisure Studies
- Humanities & Social Science

For more details or an application form please contact MMU International.
email: international@mmu.ac.uk
telephone: +44 (0)161 247 1022
www.mmu.ac.uk/international

Download free ebooks at bookboon.com

2.2 Dilation and Erosion

Morphological operators aim at extracting relevant structures of the image. This can be achieved by probing the image with another set of given shape - the structuring element (SE), as described in Section 2.1.5. *Dilation* and e*rosion* are the two fundamental morphological operators because all other operators are based on their combinations [18].

2.2.1 Dilation

Dilation is an operation that "grows" or "thickens" objects in a binary image. The specific manner and extent of this thickening is controlled by a shape referred to as a *structure element (SE)*. It is based on the following question: *"Does the structure element hit the set?"* We will define the operation of dilation mathematically and algorithmically.

First let us consider the mathematical definition. The dilation of A by B, denoted $A \oplus B$, is defined as

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \Phi\} \tag{2.2.1}$$

where Φ is the empty set and B is the structure element. In words, the dilation of A by B is the set consisting of all the structure element origin locations where the reflected and translated B overlaps at least some portion of A.

Algorithmically we would define this operation as: we consider the structure element as a *mask*. The reference point of the structure element is placed on all those pixels in the image that have value *1*. All of the image pixels that correspond to black pixels in the structure element are given the value *1* in $A \oplus B$. Note the similarity to convolving or cross-correlating A with a mask B. Here for every position of the mask, instead of forming a weighted sum of products, we place the elements of B into the output image. **Figure 18** illustrates how dilation works and **Figure 19** gives an example of applying dilation on a binary image.

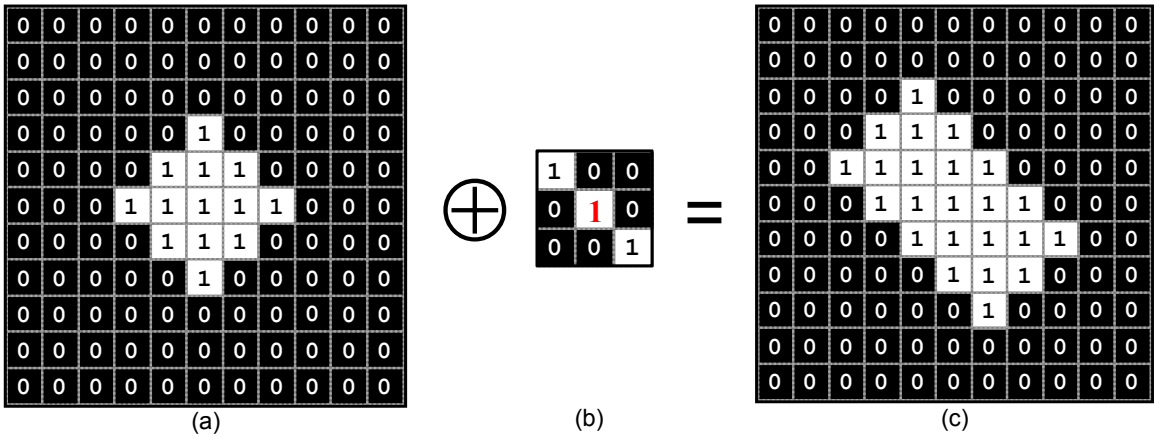


Figure 18 Illustration of morphological dilation. (a) Original binary image with a diamond object; (b) Structure element with three pixels arranged in a diagonal line at angle of 135°, the origin of the structure element is clearly identified by a red 1; (c) Dilated image, 1 at each location of the origin such that the structure element overlaps at least one 1-valued pixel in the input image (a).

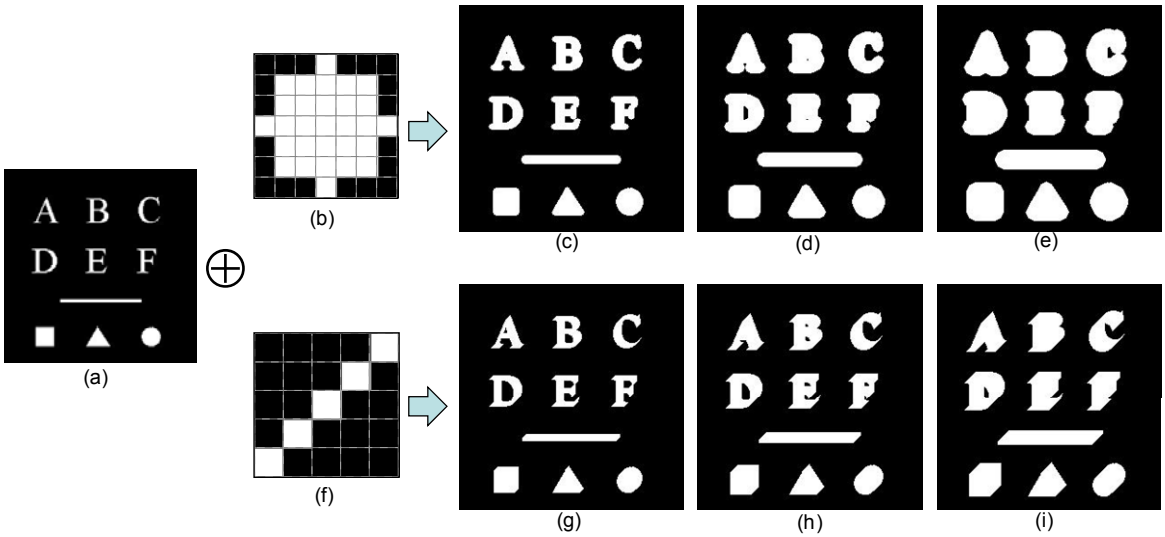


Figure 19 Example of morphological dilation. (a) A binary input image; (b) A disk structure element; (c) Dilated image of (a) by SE (b); (d) After twice dilation by SE (b); (e) After three times dilation by SE (b); (f) A line structure element; (g) Dilated image of (a) by SE (f); (h) After twice dilation by SE (f); (i) After three times dilation by SE (f).

2.2.2 Erosion

Erosion "shrinks" or "thins" objects in an image. The question that may arise when we probe a set with a *structure element (SE)* is *"Does the structure element fit the set?"*

The mathematical definition of erosion is similar to that of dilation. The erosion of A by B, denoted $A\Theta B$, is defined as

$$A\Theta B = \{z \mid (B)_z \cap A^c \neq \Phi\} \tag{2.2.2}$$

In other words, erosion of A by B is the set of all structure element origin locations where the translated B has no overlap with the background of A.

Algorithmically we can define erosion as: the output image $A\Theta B$ is set to zero. B is place at every black point in A. If A contains B (that is, if A *AND* B is not equal to zero) then B is placed in the output image. The output image is the set of all elements for which B translated to every point in A is contained in A. **Figure 20** illustrates how erosion works. **Figure 21** gives an example of applying dilation on a binary image.

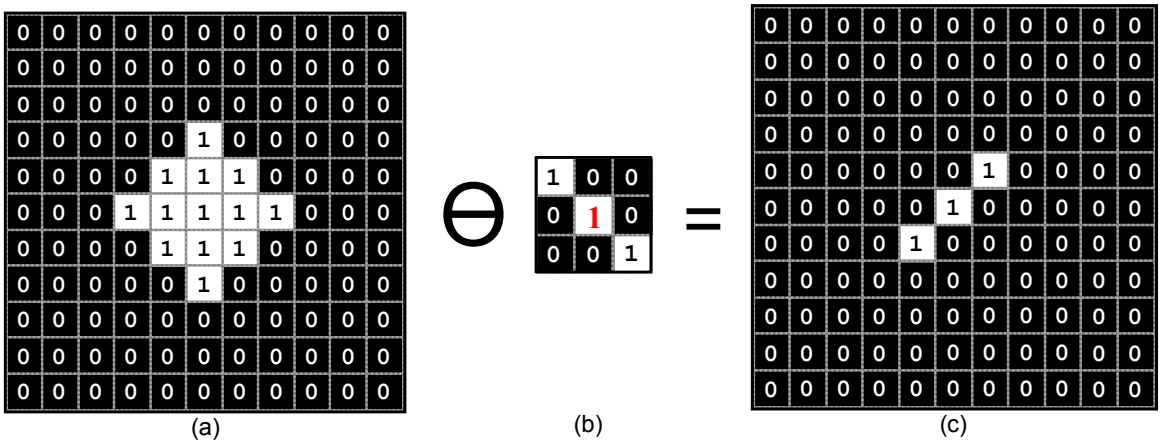


Figure 20 Illustration of morphological erosion. (a) Original binary image with a diamond object; (b) Structure element with three pixels arranged in a diagonal line at angle of 1350, the origin of the structure element is clearly identified by a red 1; (c) Eroded image, a value of 1 at each location of the origin of the structure element, such that the element overlaps only 1-valued pixels of the input image (i.e., it does not overlap any of the image background).

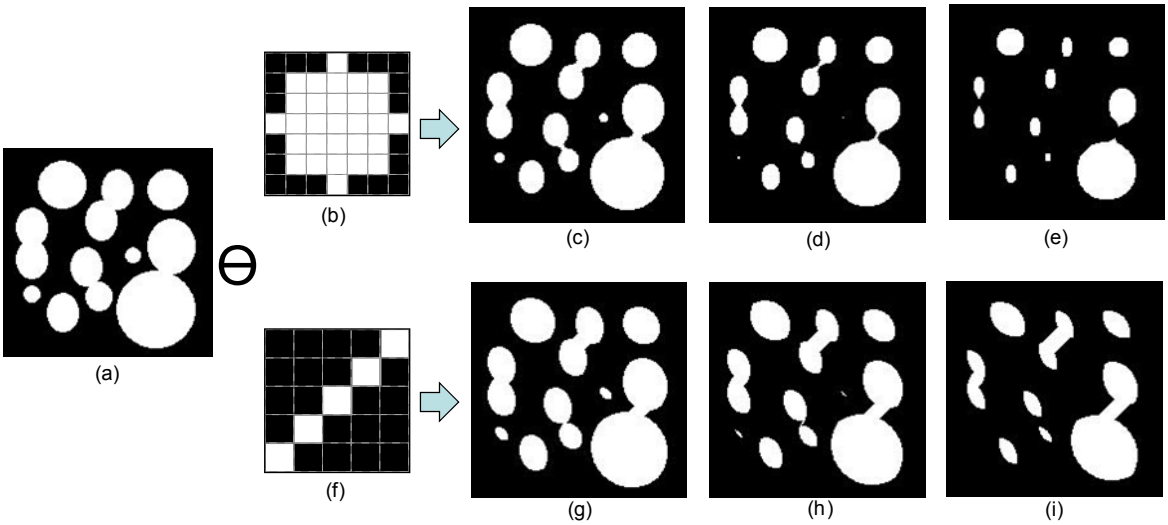


Figure 21 Example of morphological erosion. (a) A binary input image; (b) A disk structure element; (c) Eroded image of (a) by SE (b); (d) After twice erosion by SE (b); (e) After three times erosion by SE (b); (f) A line structure element; (g) Eroded image of (a) by SE (f); (h) After twice erosion by SE (f); (i) After three times erosion by SE (f).

2.2.3 Properties of dilation and erosion

- ***Distributive***: this property says that in an expression where we need to dilate an image with the union of two images, we can dilate first and then take the union. On other words, the dilation can be distributed over all the terms inside the parentheses.

$$A \oplus (B \oplus C) = (A \oplus B) \bigcup (A \oplus C) \tag{2.2.3}$$

- ***Duality***: the dilation and the erosion are dual transformations with respect to complementation. This means that any erosion of an image is equivalent to a complementation of the dilation of the complemented image with the same structuring element (and vice versa). This duality property illustrates the fact that erosions and dilations do not process the objects and their background symmetrically: the erosion shrinks the objects but expands their background (and vice versa for the dilation).

$$\begin{aligned}(A\Theta B)^c &= A^c \oplus \hat{B} \\ (A\oplus B)^c &= A^c \Theta \hat{B}\end{aligned} \tag{2.2.4}$$

- ***Translation***: erosions and dilations are invariant to translations and preserve the order relationships between images, that is, they are increasing transformations, e.g.

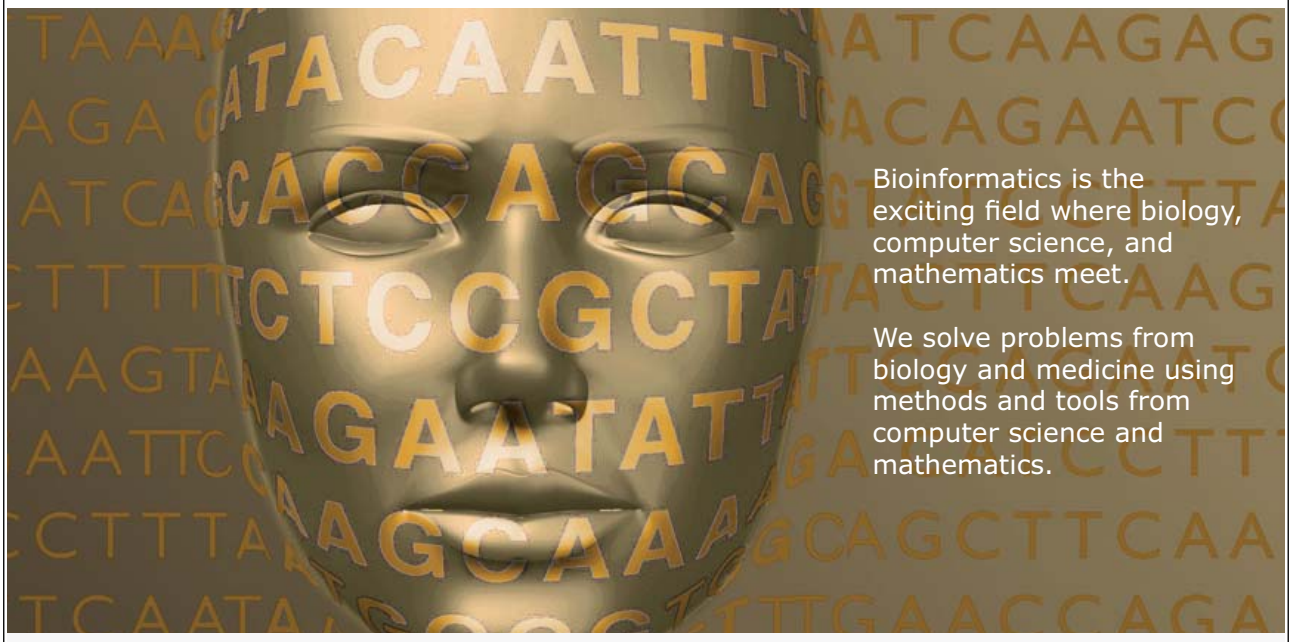
$$\begin{aligned}(A+h)\Theta B &= (A\Theta B)+h \\ (A+h)\oplus B &= (A\oplus B)+h\end{aligned} \tag{2.2.5}$$

The dilation distributes the point-wise maximum operator $\oplus$ and the erosion distributes the point-wise minimum operator Θ. For example, the point-wise maximum of two images dilated with an identical structuring element can be obtained by a unique dilation of the point-wise maximum of the images. This results in a gain of speed.

Please click the advert

UPPSALA UNIVERSITET

Develop the tools we need for Life Science
Masters Degree in Bioinformatics



Read more about this and our other international masters degree programmes at www.uu.se/master

Download free ebooks at bookboon.com

- ***Decomposition***: the following two equations concern the composition of dilations and erosions:

$$\begin{aligned}(A\Theta B_1)\Theta B_2 &= A\Theta(B_1\Theta B_2)\\ (A\oplus B_1)\oplus B_2 &= A\oplus(B_1\oplus B_2)\end{aligned} \tag{2.2.6}$$

These two properties are very useful in practice as they allow us to decompose a morphological operation with a large SE into a sequence of operations with smaller SEs. For example, an erosion with a square SE of side n in pixels is equivalent to an erosion with a horizontal line of n pixels followed by an erosion with a vertical line of the same size. It follows that there are $2(n-1)$ min comparisons per pixel with decomposition and n^2-1 without decomposition. An example of decomposition of structure element is illustrated below (where $n = 3$):

$$\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} = \begin{bmatrix}1 & 1 & 1\end{bmatrix}\oplus\begin{bmatrix}1\\ 1\\ 1\end{bmatrix} \tag{2.2.7}$$

Suppose that a structure element B can be represented as a dilation of two structure elements B_1 and B_2:

$$B = B_1\oplus B_2 \tag{2.2.8}$$

$$A\oplus B = A\oplus(B_1\oplus B_2) = (A\oplus B_1)\oplus B_2 \tag{2.2.9}$$

In other words, dilating A with B is the same as first dilating B_1, and then dilating the result with B_2. We say that B can be *decomposed* in to the structure elements B_1 and B_2.

The decomposition property is also important for hardware implementations where the neighbourhood size is fixed (e.g., fast 3×3 neighbourhood operations). By cascading elementary operations, larger neighbourhood size can be obtained. For example, an erosion by a square of width $2n+1$ pixels is equivalent to n successive erosions with a 3×3 square.

2.2.4 Morphological gradient

A common assumption in image analysis consists of considering image objects as regions of rather homogeneous gray levels. It follows that object boundaries or edges are located where there are high gray level variations. Morphological gradients are operators enhancing intensity pixel variations within a neighbourhood. The erosion/dilation outputs for each pixel the minimum/maximum value of the image in the neighbourhood defined by the SE. Variations are therefore enhanced by combining these elementary operators. Three combinations are currently used:

- ***External gradient***: arithmetic difference between the dilation and the original image;
- ***Internal gradient***: arithmetic difference between the original image and its erosion;
- ***Morphological gradient***: arithmetic difference between the dilation and the erosion.

The basic morphological gradient is defined as the arithmetic difference between the dilation and the erosion with the elementary structure element *B* of the considered grid. This *morphological gradient* of image *A* by structure element *B* is denoted by *AΩB*:

$$A\Omega B = (A \oplus B) - (A \Theta B) \tag{2.2.10}$$

It is possible to detect external or internal boundaries. Indeed, the external and internal morphological gradient operators can be defined as $A\Omega^{+}B$ and $A\Omega^{-}B$ respectively:

$$A\Omega^{+}B = (A \oplus B) - A \tag{2.2.11}$$

$$A\Omega^{-}B = A - (A \Theta B) \tag{2.2.12}$$

It can be seen that the morphological gradient outputs the maximum variation of the gray-level intensities within the neighbourhood defined by the SE rather than a local slope. The thickness of a step edge detected by a morphological gradient equals two pixels: one pixel on each side of the edge. Half-gradients can be used to detect either the internal or the external boundary of an edge. These gradients are one-pixel thick for a step edge. Morphological, external, and internal gradients are illustrated in **Figure 22**.

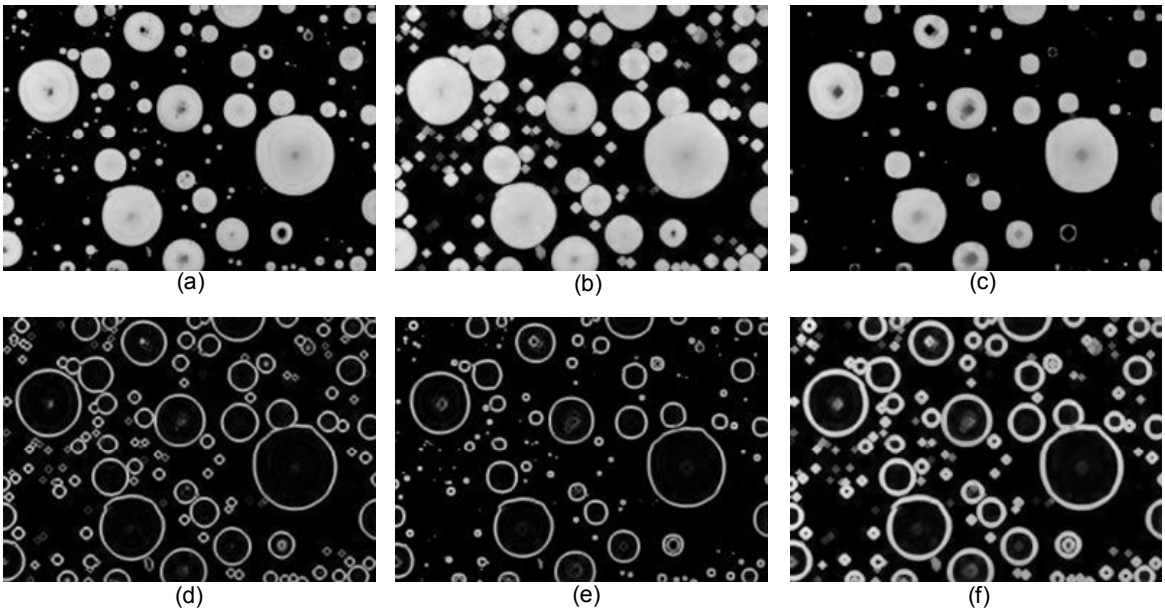


Figure 22 Morphological gradients to enhance the object boundaries. (a) Original image A of enamel particles; (b) Dilated image A by B: A B, note that structure element B is a 5×5 disk; (c) Eroded image A by B: AΘB; (d) External gradient AΩ+B = (A B)-A; (e) Internal gradient AΩ⁻B = A-(AΘB); **(f)** Morphological gradient AΩB = (A⊕B)-(AΘB).

2.3 Opening and closing

In practical image processing application, dilation and erosion are used most often in various combinations. An image will undergo a series of dilations and/or erosions using the same, or sometime different, structure elements. Two of the most important operations in the combination of dilation and erosion are *opening* and *closing*.

2.3.1 Opening

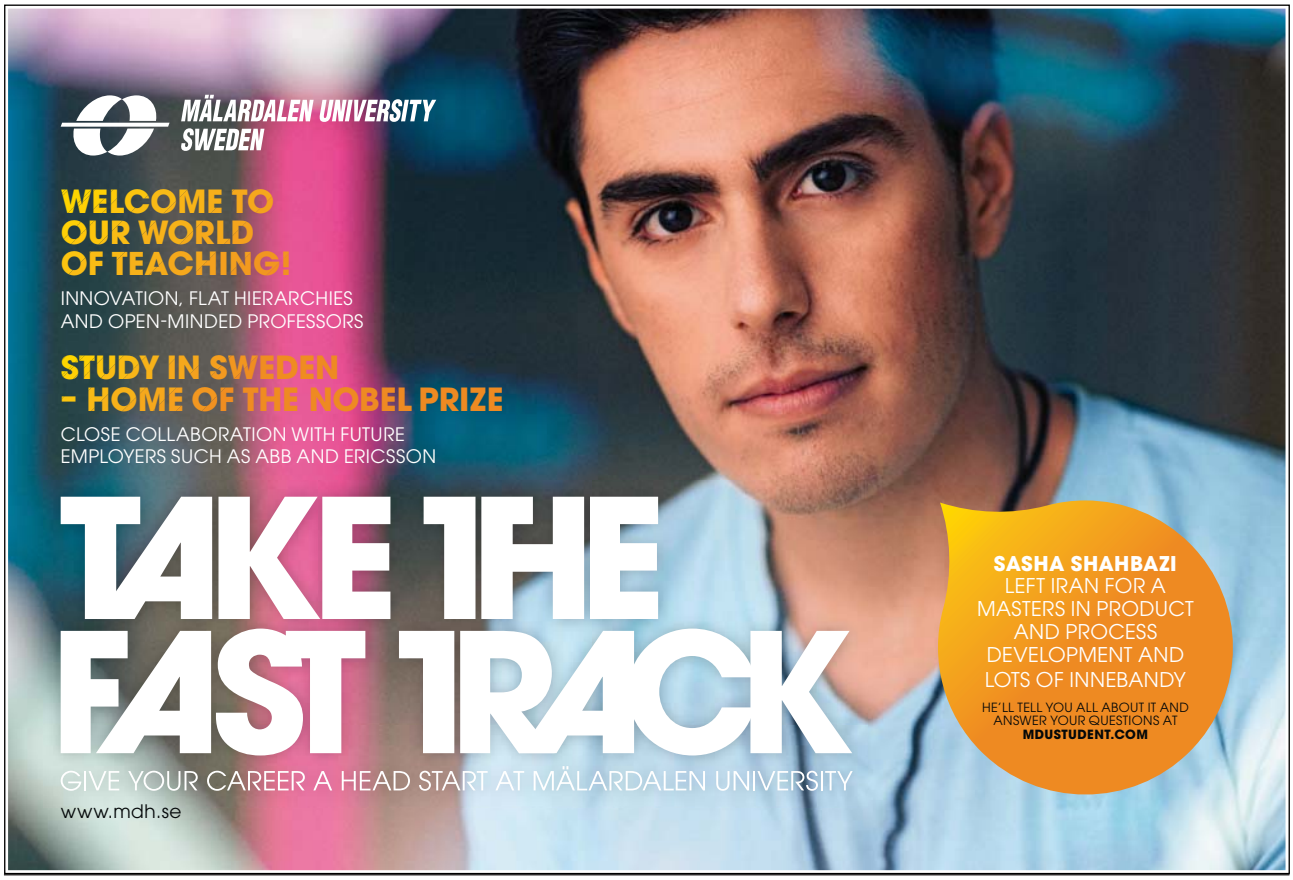
Once an image has been eroded, there exists in general no inverse transformation to get the original image back. The idea behind the morphological opening is to dilate the eroded image to recover as much as possible the original image.

The process of erosion followed by dilation is called *opening*. The opening of A by B, denoted $A \circ B$ is defined as:

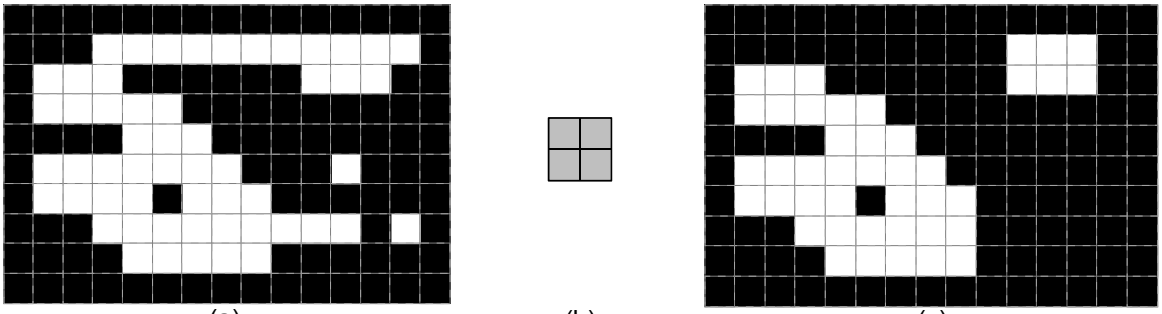
$$A \circ B = (A \Theta B) \oplus B \tag{2.3.1}$$

The geometric interpretation for this formulation is: $A \circ B$ is the union of all translations of B that fit entirely within A. Morphological opening removes completely regions of an object that cannot contain the structure element, generally smoothes the boundaries of larger objects without significantly changing their area, breaks objects at thin points, and eliminates small and thin protrusions. The illustration of opening is shown in **Figure 23**.

Please click the advert



Download free ebooks at bookboon.com



(a) (b) (c)

Figure 23 Illustration of opening. (a) A 10 × 15 discrete binary image (the object pixels are the white pixels); (b) A 2 × 2 structure element; (c) Opening of image (a) by SE (b). All object pixels that cannot be covered by the structure element when it fits the object pixels are removed.

The definition of opening gives an interpretation in terms of *shape matching* – the ability to select from a set or object all those subsets that match the structure element. **Figure 24** shows an example of this property. Note that the radius of the disk structure element must be larger than the widths of the image subsets that are to be eliminated.

(a) (b) (c)

Figure 24 Shape matching by opening. (a) An original binary image A with some disks and lines; (b) Eroded image AΘB by a disk structure element B, where the radius of the disk is 5 pixels; (c) Dilated image of (b) by the same disk structure element B: (AΘB) ⊕B.

2.3.2 Closing

The process of dilation followed by erosion is called *closing*. The closing of A by B, denoted $A \bullet B$, is defined as:

$$A \bullet B = (A \oplus B) \Theta B \tag{2.3.2}$$

Geometrically, the closing $A \bullet B$ is the complement of the union of all translations of B that do not overlap A. It has the effect of filling small and thin holes in objects, connecting nearby objects, and generally smoothing the boundaries of objects without significantly changing their area. The illustration of opening is shown in **Figure 25**.

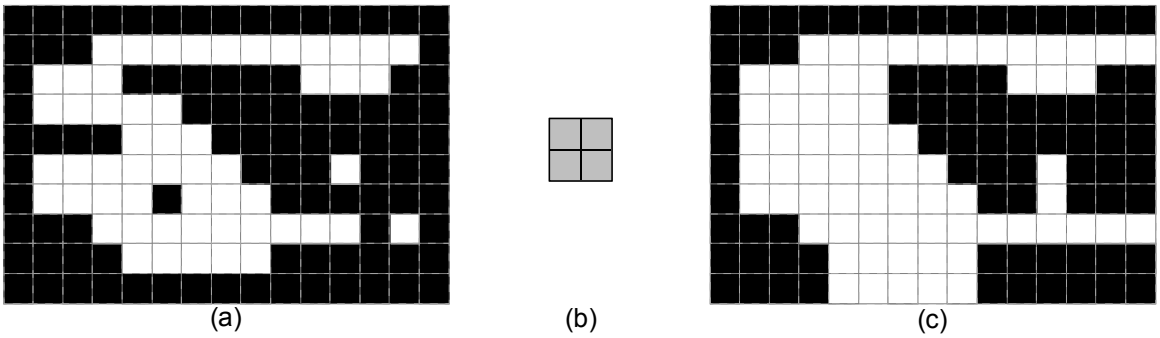


Figure 25 Illustration of closing. (a) A 10 × 15 discrete binary image (the object pixels are the white pixels); (b) A 2 × 2 structure element; (c) Closing of image (a) by SE (b). All background pixels that cannot be covered by the structure element when it fits the background are added to the object pixels.

Note that the opening removes all object pixels that cannot be covered by the structuring element when it fits the object pixels while the closing fills all background structures that cannot contain the structuring element. In **Figure 26**, the closing of a gray-scale image is shown together with its opening.

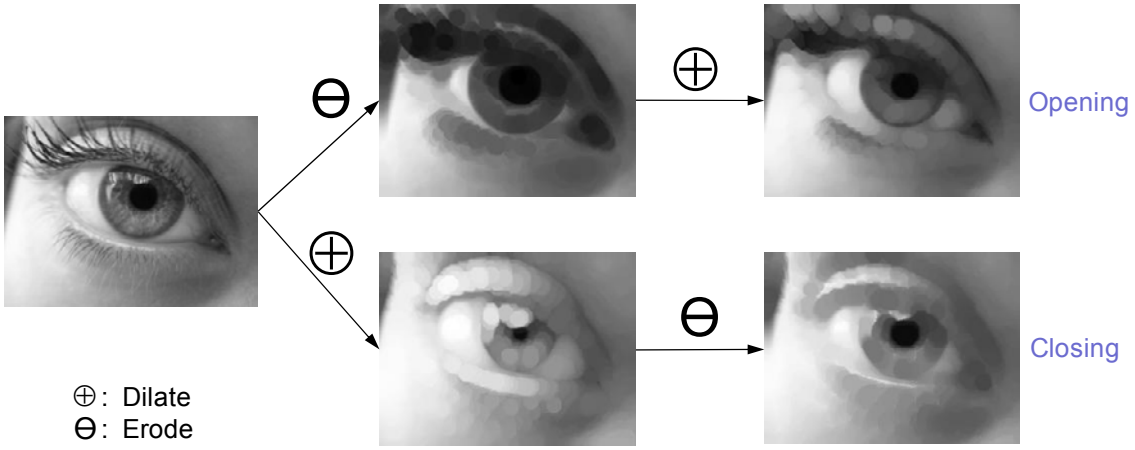
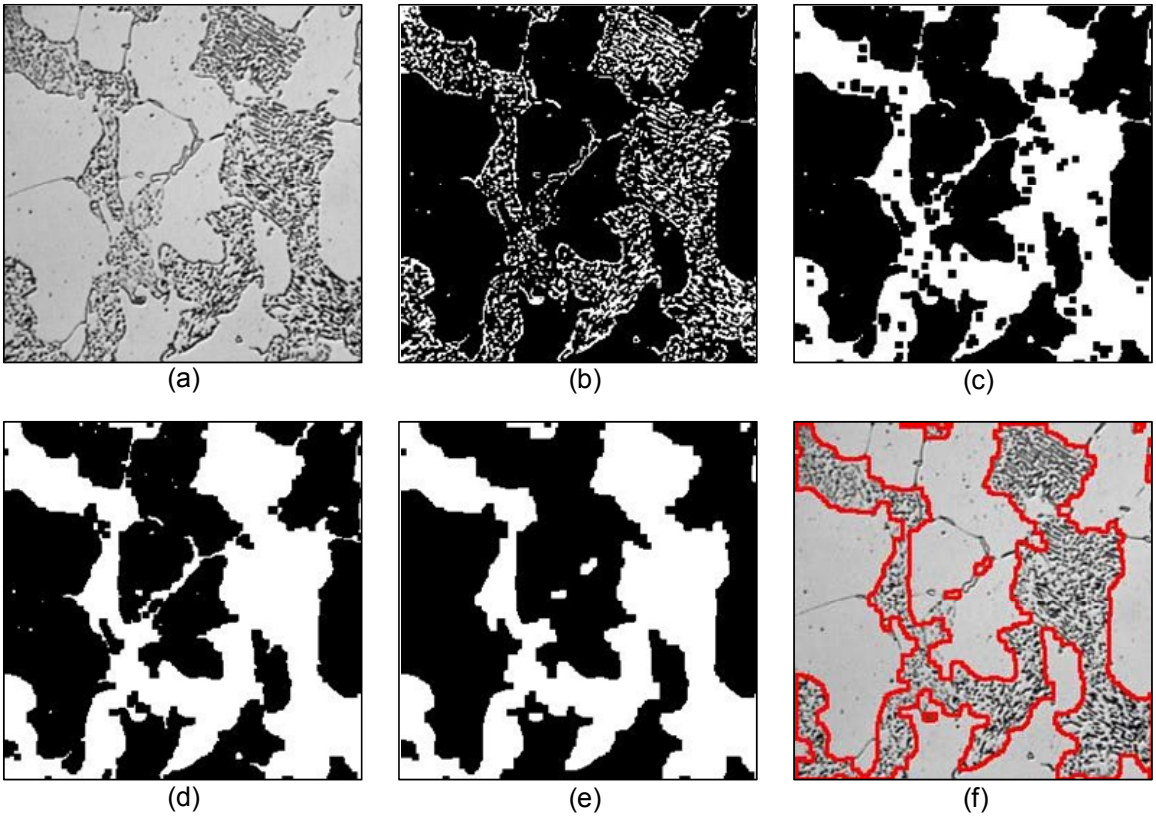
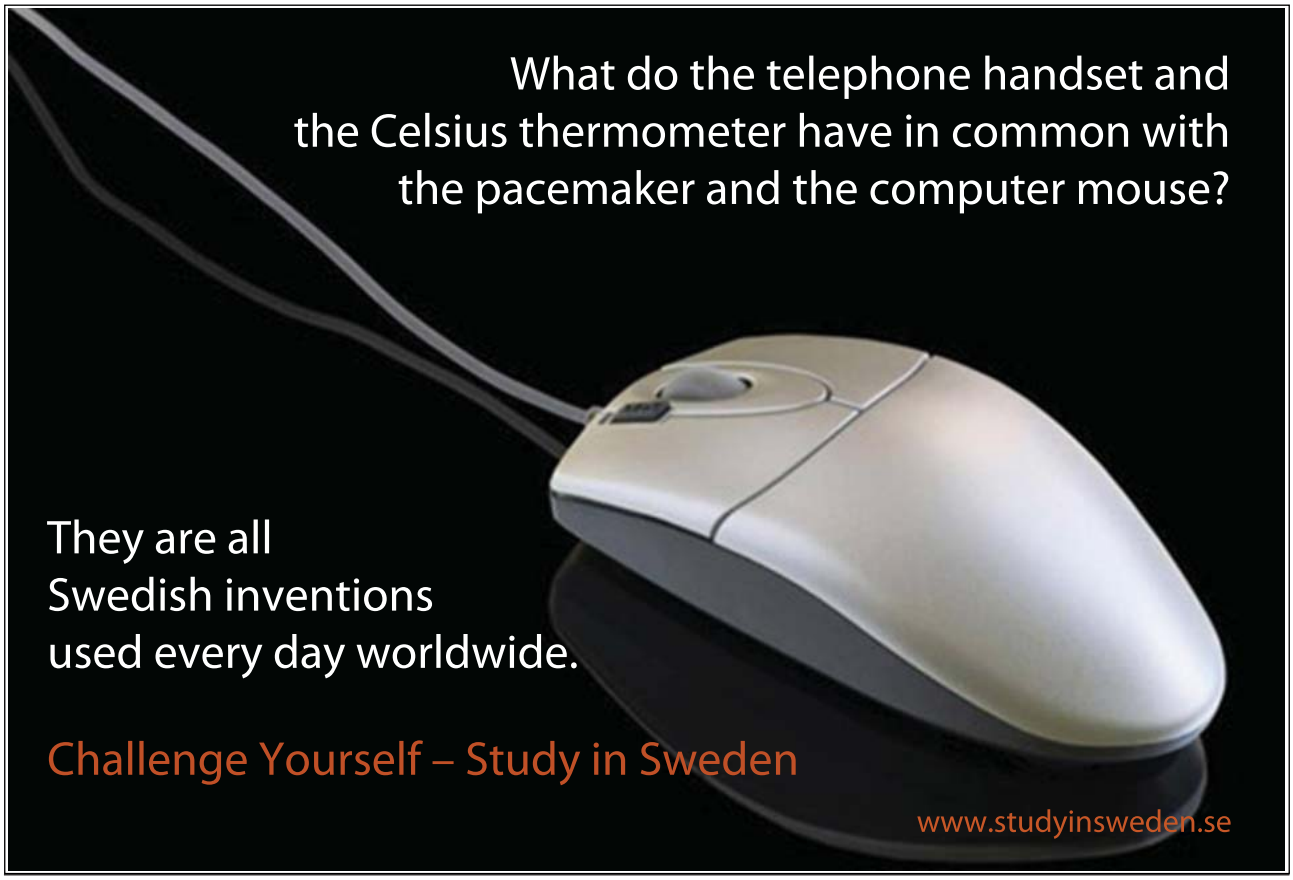


Figure 26 Opening and closing of a gray-scale image with an 8 × 8 disk SE.

Often, when noisy image are segmented by thresholding, the resulting boundaries are quite ragged, the objects have false holes, and the background is peppered with small noise objects. Successive openings or closings can improve the situation markedly. Sometimes several iterations of erosion, followed by the same number of dilations, produce the desired effect. An example of the combination of image opening and closing is illustrated in **Figure 27**.



Please click the advert



Download free ebooks at bookboon.com

Figure 27 An example of image closing and opening. (a) Original gray level image of chemically etched metallographic specimen: dark regions are iron carbide (image courtesy to J. C. Russ); (b) Intensity histogram threshold applied to image (a); (c) Closing image (b) by a disk structure element with radius 3; **(d)** Fill the holes in image (c); (e) Opening the image (d) by a disk structure element with radius 3 in order to remove the debris; (f) Outlines of image (e) superimposed on the original image (a).

2.3.3 Properties of opening and closing

Openings and closings are ***dual transformations*** with respect to set complementation.

$$A \circ B = (A^c \bullet \hat{B})^c \tag{2.3.3}$$

$$A \bullet B = (A^c \circ \hat{B})^c \tag{2.3.4}$$

The fact that they are not self-dual transformations means that one or the other transformation should be used depending on the relative brightness of the image objects we would like to process. The relative brightness of an image region defines whether it is a background or foreground region. Background regions have a low intensity value compared to their surrounding regions and vice versa for the foreground regions. Openings filter the foreground regions from the inside. Closings have the same behaviour on the background regions. For instance, if we want to filter noisy pixels with high intensity values an opening should be considered.

We have already stated that openings are anti-extensive transformations (some pixels are removed) and closings are extensive transformations (some pixels are added). Therefore, the following ***ordering relationship*** always holds:

$$A \circ B \leq A \leq A \bullet B \tag{2.3.5}$$

Morphological openings $A \circ B$ and closings $A \bullet B$ are both ***increasing transformations***. This means that openings and closings preserve order relationships between images.

$$A_1 \subseteq A_2 \Rightarrow A_1 \circ B \subseteq A_2 \circ B \tag{2.3.4}$$

$$A_1 \subseteq A_2 \Rightarrow A_1 \bullet B \subseteq A_2 \bullet B \tag{2.3.5}$$

Moreover, both opening and closing are ***translation invariance***:

$$(A + h) \circ B = (A \circ B) + h \tag{2.3.6}$$

$$(A + h) \bullet B = (A \bullet B) + h \tag{2.3.7}$$

Finally, successive applications of openings or closings do not further modify the image. Indeed, they are both ***idempotent*** transformations:

$$(A \circ B) \circ B = A \circ B \tag{2.3.8}$$

$$(A \bullet B) \bullet B = A \bullet B \tag{2.3.9}$$

The idempotence property is often regarded as an important property for a filter because it ensures that the image will not be further modified by iterating the transformation. This property is exploited when the operations are used repeatedly for decomposition of an object into its constituent parts. A simple example of a series of openings and image subtractions is shown in **Figure 28**.

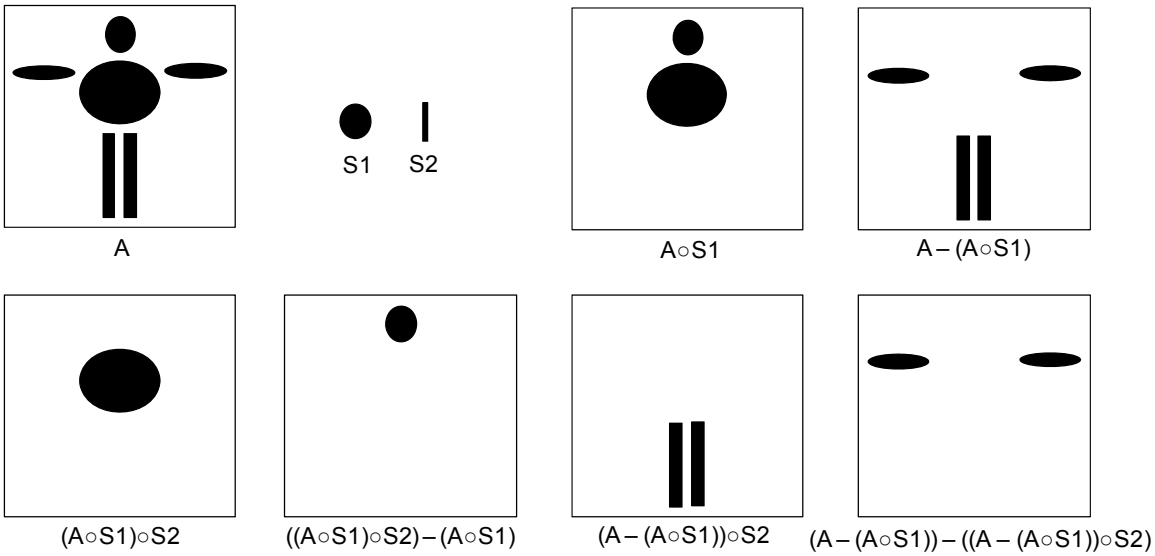


Figure 28 A series of opening and image subtractions in order to decompose an object into its constituent parts.

The ***combination*** of opening and closing is frequently used to clean up artefacts in a segmented image prior to further analysis. The choice of whether to use opening or closing, or a sequence of erosions and dilations, depends on the image and the objective. For example, opening is used when the image has foreground noise or when we want to eliminate long, thin features: it is not used when there is a chance that the initial erosion operation might disconnect regions. Closing is used when a region has become disconnected and we want to restore connectivity: it is not used when different regions are located closely such that the first iteration might connect them. Usually a compromise is determined between noise reduction and feature retention by testing representative images.

2.3.4 Top-hat transformation

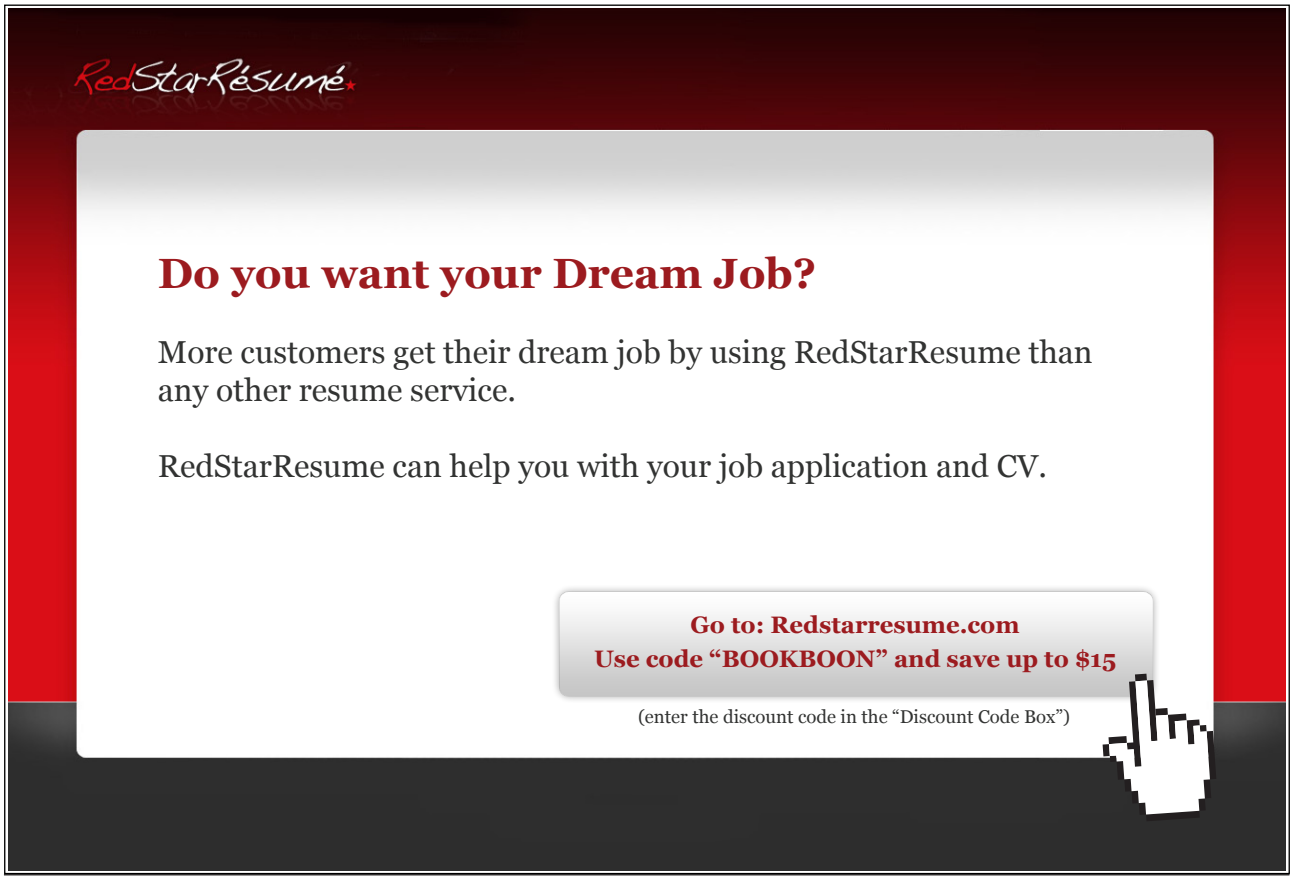
The choice of a given morphological filter is driven by the available knowledge about the shape, size, and orientation of the structures we would like to filter. For example, we may choose an opening by a 2×2 squared SE to remove impulse noise or a larger square to smooth the object boundaries, or openings on *gray scale* image can be used to compensate for non-uniform background illumination **Figure 29** Top-hat transformation. (a) A gray level image of C elegant, where there is non-uniform illumination background; (b) Intensity threshold image of (a), where the object has been over-segmented; (c) Opening image of (a) by a large disk structure element w. Subtracting an opened image from the original is called a *top-hat* transformation, which is denoted as $TH_B(A)$:

$$TH_B(A) = A - (A \circ B) \tag{2.3.10}$$

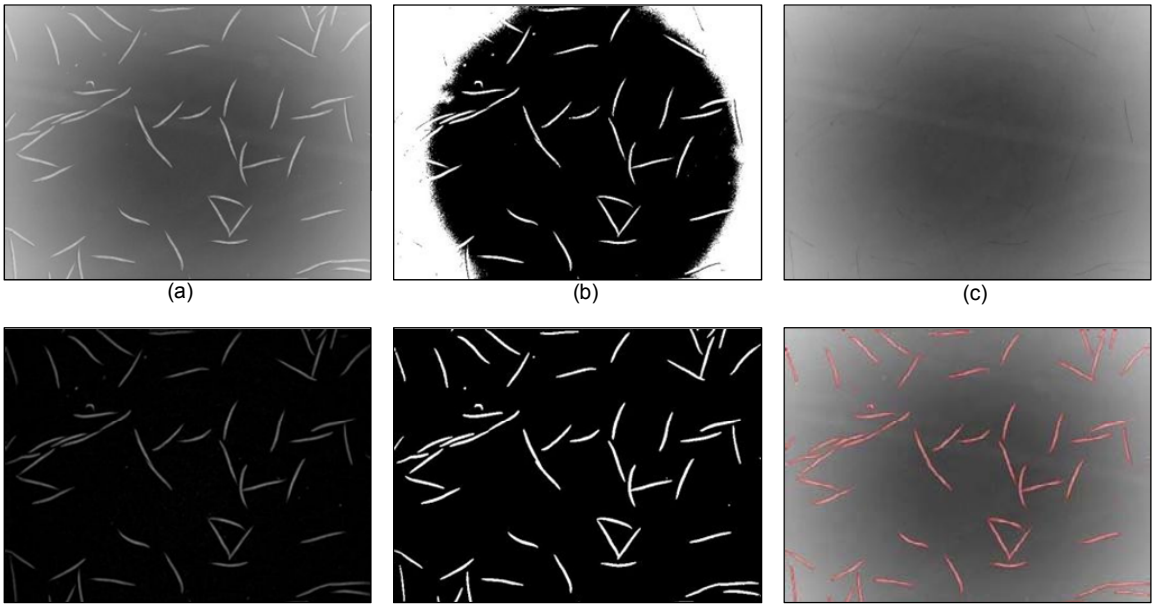
Where A is the original input image, B is a structure element.

Indeed, the approach undertaken with top-hats consists in using knowledge about the shape characteristics that are not shared by the relevant image structures. An opening with an SE that does not fit the relevant image structures is then used to remove them from the image. These structures are recovered through the arithmetic difference between the image and its opening (**Figure 29**). The success of this approach is due to the fact that there is not necessarily a one-to-one correspondence between the knowledge about what an image object is and what it is not. Moreover, it is sometimes easier to remove relevant image objects than to try to suppress the irrelevant ones.

Please click the advert



Download free ebooks at bookboon.com

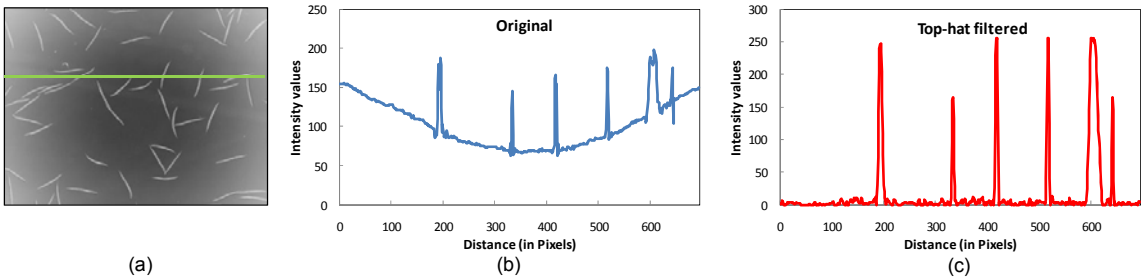


(a) (b) (c)

(d) (e) (f)

Figure 29 Top-hat transformation. (a) A gray level image of C elegant, where there is non-uniform illumination background; (b) Intensity threshold image of (a), where the object has been over-segmented; (c) Opening image of (a) by a large disk structure element with radius 5; (d) Top-hat transformation of image (a) = image (a) – image(c), where the badly illuminated background has been removed; (e) Intensity threshold of top-hat image in (d); (f) Segmentation outline superimposed on the original input image (a).

If the image objects all have the same local contrast, that is, if they are either all darker or brighter than the background, top-hat transforms can be used for mitigating illumination gradients. Indeed, a top-hat with a large isotropic structuring element acts as a high-pass filter. As the illumination gradient lies within the low frequencies of the image, it is removed by the top-hat transformation. For example, an illustration of top-hat transformation on a 1-D line profile is shown in **Figure 30**.



(a) (b) (c)

Figure 30 Illustration of top-hat transformation on a 1-D line profile. (a) A gray level image of C elegant, where there is a non-uniform illumination background, and a line profile is marked in green; (b) The original line profile across the image in (a); (c) Top-hat filtered line profile, note that the signal peaks are extracted independently from their intensity level. It is only a shape criterion that is taken into account.

Contrast to top-hat transformation, a *bottom-hat* transformation is defined as the closing of the image minus the image. Both top-hat and bottom-hat transform can be used together to enhance the image contrast. Subtracting an original image from its closed image is called a *bottom-hat* transformation, which is denoted as $BH_B(A)$:

$$BH_B(A) = (A \bullet B) - A \tag{2.3.11}$$

Where A is the original input image, B is a structure element. It follows that

$$BH_B(A) = (TH_B(A))^c \tag{2.3.12}$$

2.4 Hit-or-miss

The *hit-or-miss* transform is a basic tool for shape detection or pattern recognition. Indeed almost all the other morphological operations, such as *thinning*, *skeleton* and *pruning*, can be derived from it.

Hit-or-miss is an operation that is used to select sub-images that exhibit certain properties. As the name implies it is a combination of two transforms (erosions) and is somewhat similar to template matching, where an input is cross-correlated with a template or mask that contains a sub-image of interest. The hit-or-miss transformation of A by B is denoted $A \otimes B$:

$$A \otimes B = (A \Theta B_1) \bigcap (A^c \Theta B_2) \tag{2.4.1}$$

where A is the object, A^c is the complement (or background) of A, B is a structure pair, $B = (B_1, B_2)$, rather than a single structure element. Thus the operation is performed by *ANDing* together two output images, one formed by eroding the input image with B_1 and the other by eroding the complement of the input image A by B_2. For example, the structure element pair for top left corner detection in **Figure 31** (a) can also be de-composited as:

B =

0	0	X
0	1	1
X	1	X

B1 =

	1	1
	1	

B2 =

1	1	
1		

The structure element B is an extension of those we have used before which contained only *1*s and *0*s: in this case it contains a pattern of *1*s (foreground pixels), *0*s (background pixels) and *X*'s (don't care). An example, used for find right-angle corner point in a binary image, is shown in **Figure 31**.

0	0	X
0	1	1
X	1	X

(a)

X	0	0
1	1	0
X	1	X

(b)

X	1	X
0	1	1
0	0	X

(c)

X	1	X
1	1	0
X	0	0

(d)

Figure 31 Four structure elements used for finding right-angle corner points in a binary image by using hit-or-miss transformation. (a) Top left corner; (b) Top right corner; (c) Bottom left corner; (d) Bottom right corner.

The hit-or-miss is performed by translating the centre of the structure element to all points in the image, and then comparing the pixels of the structure element with the underlying image pixels. If the pixels in the structure element exactly match the pixels in the image, then the image pixel underneath the centre of the structure element is set to the foreground colour, indicating a *"hit"*. If the pixels do not match, then that pixel is set to the background colour, indicating a *"miss"*. The *X*'s (don't care) elements in the structure element match with either *0*s or *1*s.When the structure element overlaps the edge of an image, this would also generally be considered as a "miss". An example of corner detection by using hit-or-miss transform is illustrated in **Figure 32**.

Please click the advert

Download free ebooks at bookboon.com

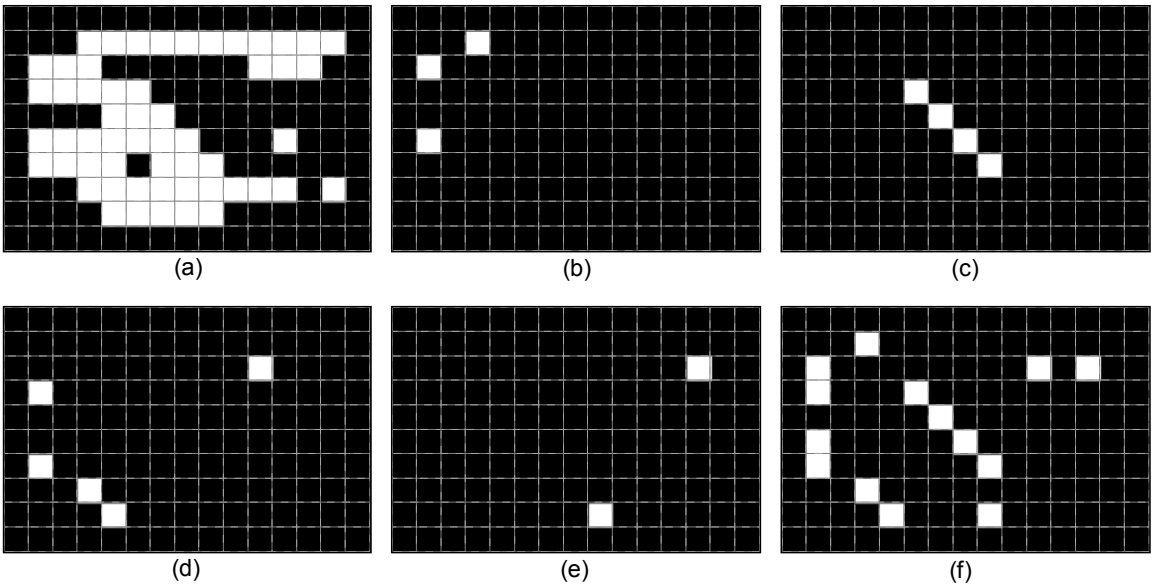


Figure 32 Corner detection by using hit-or-miss transform. (a) A 10 × 15 discrete binary image (the object pixels are the white pixels; (b) Detection for top left corners by hit-or-miss transformation of image(a) by structure element in **Figure 31**; (c) Detection for top left corners; (d) Detection for bottom left corners; (e) Detection for bottom right corners; (f) All right-angle corners by applying "OR" operator on (b), (c), (d) and (e).

2.5 Thinning and thicken

Erosion can be programmed as a two-step process that will not break objects. The first step is a normal erosion, but it is a conditional; that is, pixels are marked as candidates for removal, but are not actually eliminated. In the second pass, those candidates that can be removed without destroying connectivity are eliminated, while those that cannot are retained. This process is called ***thinning***. Thinning consist in removing the object pixels having a given configuration. In other words, the hit-or-miss transform of the image is subtracted from the original image.

The thinning of A by B, denoted $A\emptyset B$, is defined as:

$$A \phi B = A - (A \otimes B) \tag{2.5.1}$$

where B is conveniently defined as composite structure element where:

- *1* means an element belonging to object;
- *0* means an element belonging to background;
- *X* means – can be either.

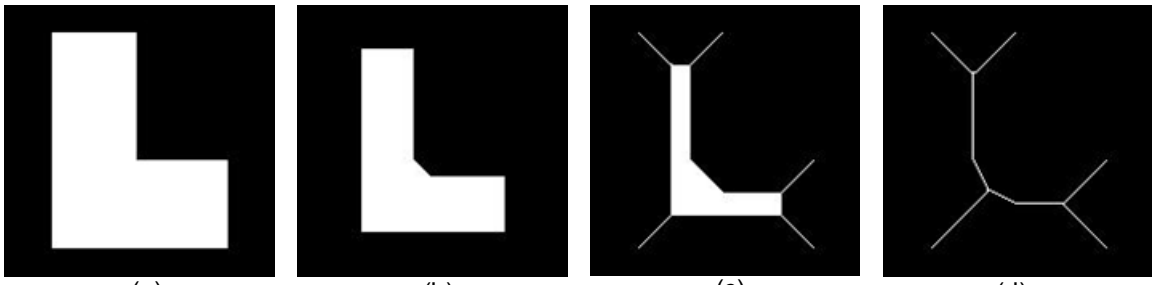
A set of eight structure element that can be used for thinning is:

$$\begin{bmatrix} 0 & 0 & 0 \\ X & 1 & X \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} X & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & X \end{bmatrix} \begin{bmatrix} 1 & X & 0 \\ 1 & 1 & 0 \\ 1 & X & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & X \\ 1 & 1 & 0 \\ X & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ X & 1 & X \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & X \end{bmatrix} \begin{bmatrix} 0 & X & 1 \\ 0 & 1 & 1 \\ 0 & X & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & X \\ 0 & 1 & 1 \\ X & 1 & 1 \end{bmatrix} \tag{2.5.2}$$

where the origin in each case is the centre element.

For example, the first structure element in Equation (2.5.2) and the three rotations of it by *90°* are essentially line detectors. If a hit-or-miss transform is applied to the input image in **Figure 33** using this structure element, a pixel-wide line from the top surface of the object is produced, which is one pixel short at both right and left ends. If the line is subtracted from the original image, the original object is thinned slightly. Repeated thinning produces the image shown in **Figure 33**. If this is continued, together with thinning by the other three rotations of the structuring element, the final thinning is produced.



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

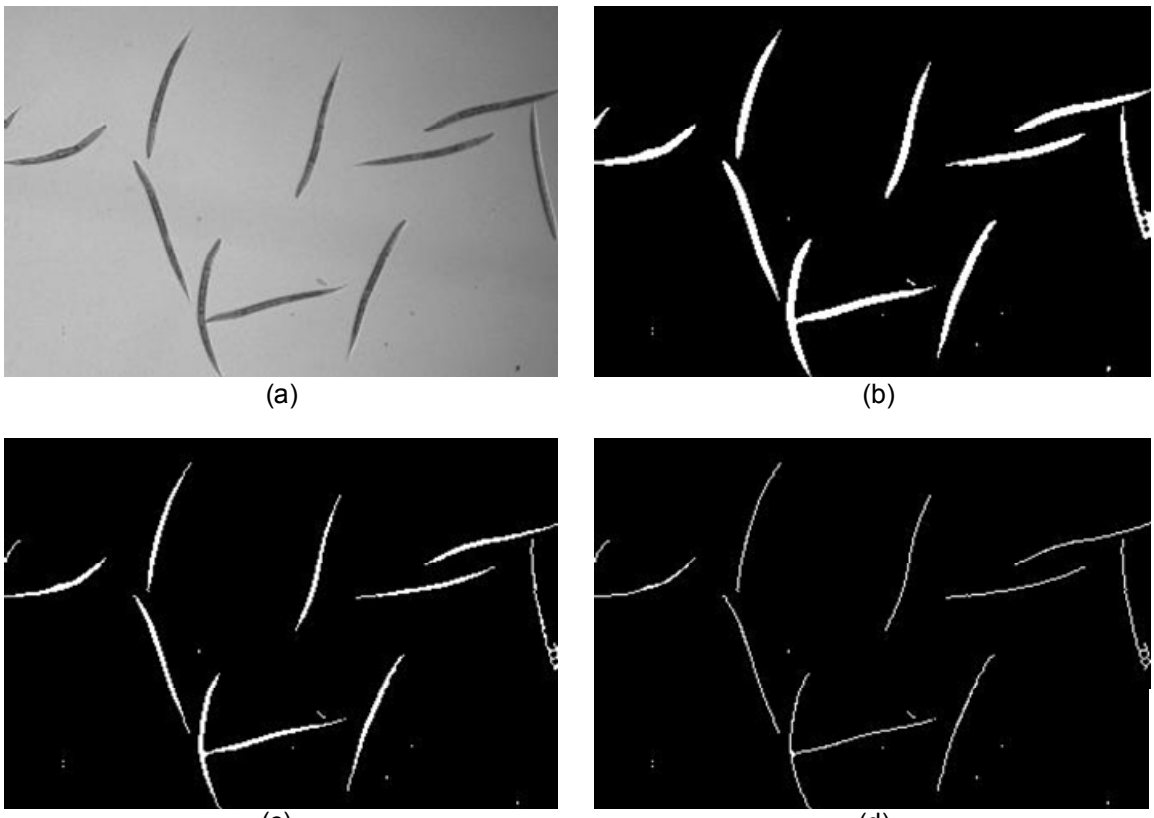
Figure 33 Illustration of thinning for line detection. (a) A binary original image; (b) After 10 iterations of thinning; (c) After 20 iterations of thinning; (d) The thinning process is repeated until no further change occurs, e.g. convergence.

Sequential thinning is defined with respect to a sequence of structure elements *B*:

$$A_{n+1} = ((...((A_n \phi B_1) \phi B_2)...) \phi B_n) \tag{2.5.3}$$

That is, the image *A* is thinned by B_1. The result of this process is thinned by B_2, and so on until all the *n* structure elements have been applied. Then the entire process is repeated until there is no change between two successive images. Thinning reduces a curvilinear object to a single-pixel-wide line. **Figure 34** shows an example of thinning a group of *C elegans*, some of which are touching. This can be used as the basis for a separation algorithm for objects that are in contact.

The thinning is very useful because it provides a simple and compact representation of the shape of an object. Thus, for instance, we can get a rough idea of the length of an object by finding the maximally separated pair of end points on the thinning image. Similarly, we can distinguish many qualitatively different shapes from one another on the basis of how many junction points there are, i.e. points where at least three branches of the thinning meet.



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

Figure 34 Thinning a group of C elegans. (a) An image of a group of C elegant; (b) Top-hat filtering image (a) followed by intensity thresholding; (c) Thinning of image (b) once; (d) Thinning of image (b) to generate a single-pixel-wide line.

Dilation can be implemented so as not to merge nearby objects. This can be done in two passes, similarly to thinning. This conditional two-step dilation is called ***thicken***. An alternative is to complement the image and use the thinning operation on the background. In fact, each of the variants of erosion has a companion dilation-type operation obtained when it is run on a complemented image.

Some segmentation techniques tend to fit rather tight boundaries to objects so as to avoid erroneously merging them. Often, the best boundary for isolating objects is too tight for subsequent measurement. Thickening can correct this by enlarging the boundaries without merging separate objects.

Download free ebooks at bookboon.com

2.6 Skeleton

Thinning and thicken transformations are generally used sequentially. Sequential transformations can be used to derive a "digital skeleton" easily. *Skeleton* is the way to reduce binary image objects to a set of thin strokes that retain important information about the shapes of the original objects. It is also known as *medial axis transform*. The medial axis is the locus of the centres all the circles that are tangent to the boundary of the object at two or more disjoint points. **Figure 35** illustrates the definition of the skeleton of an object.

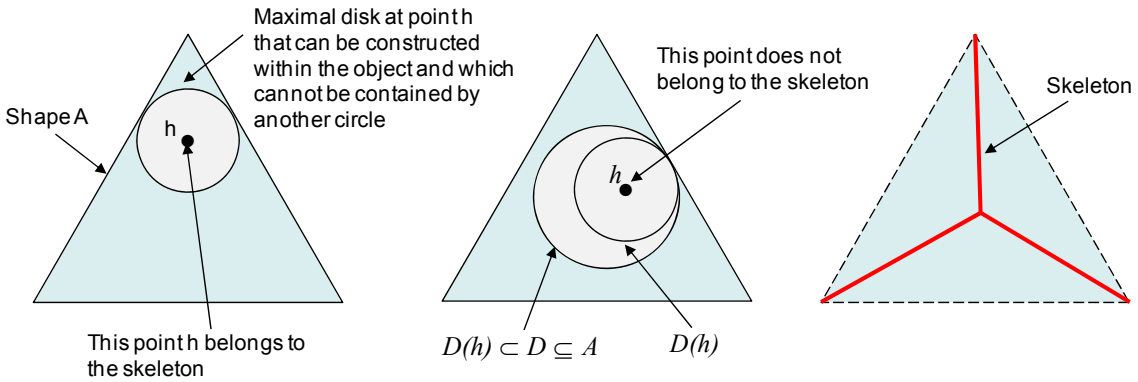


Figure 35 Illustrating the definition of the skeleton of an object: construction of the Euclidean skeleton for a triangular shape.

To define the skeleton of a shape A, for each point $h \in A$, let $D(h)$ denote the largest disk centred at h such that $D(h) \subseteq A$. Then, the point h is a point on the skeleton of A if there does not exist a disk D, such that $D(h) \subset D \subseteq A$. In this case, $D(h)$ is called the maximal disk located at point h. If, in addition to the skeleton, the radii of the *maximal disks* located at all points h on the skeleton of a shape A are known, then A can be uniquely reconstructed from this information as the union of all such maximal disks. Therefore, the skeleton, together with the radius information associated with the maximal disks, contains enough information to uniquely reconstruct the original shape. An example of skeletonization is illustrated in **Figure 36**.

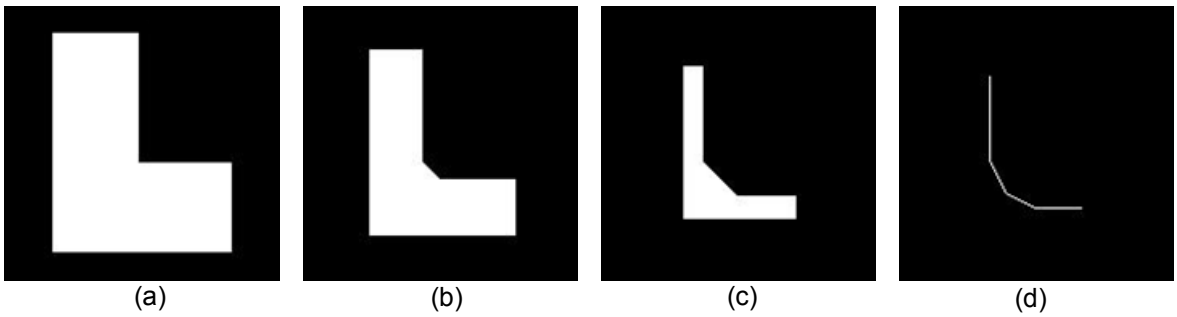
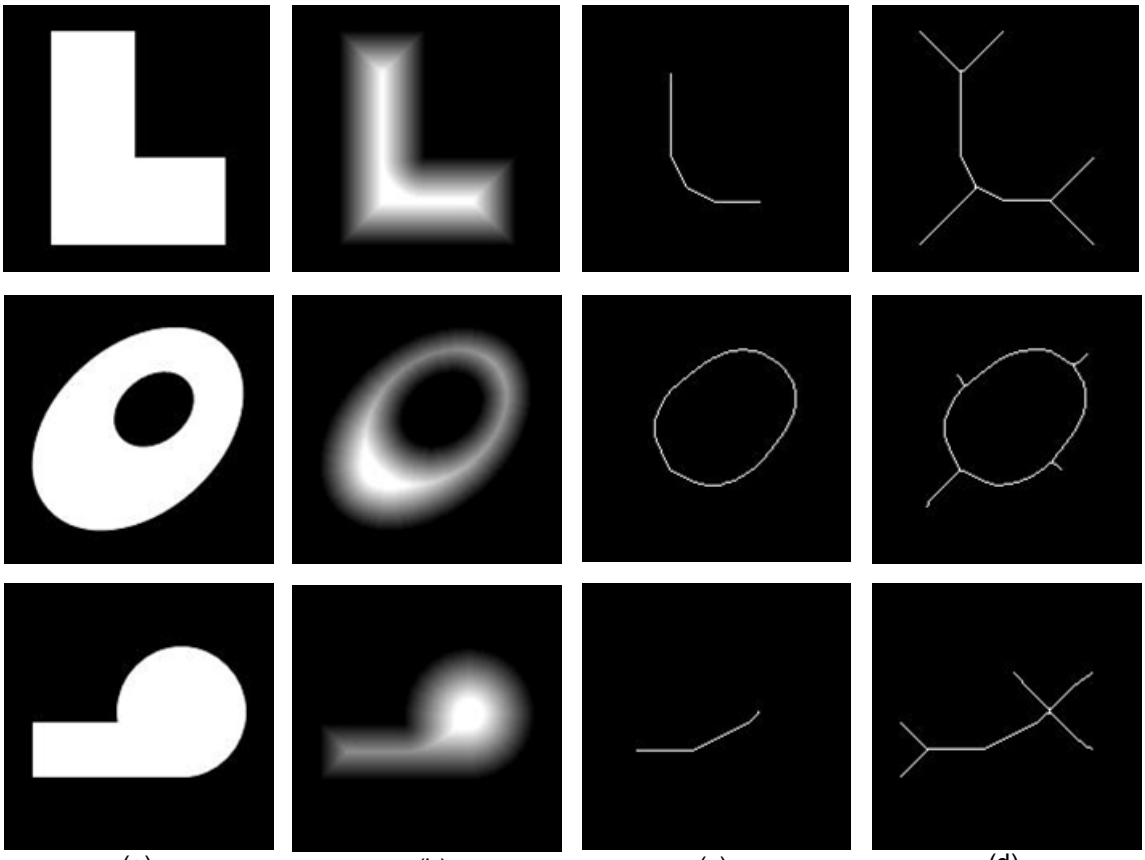


Figure 36 Illustration of skeleton for line detection. (a) A L-shaped original binary image; (b) After 10 iterations of skeletonization; (c) After 20 iterations of skeletonization; (d) The skeletonization process is repeated until no further change occurs, e.g. convergence.

Skeleton can be implemented with a two-pass conditional erosion, as with thinning. The rule for deleting pixels, however, is slightly different. The primary difference is that the medial axis skeleton extends to the boundary at corners, while the skeleton obtained by thinning does not. **Figure 37** shows a comparison between skeleton and thinning.

Download free ebooks at bookboon.com



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

Please click the advert

Trust and responsibility

NNE and Pharmaplan have joined forces to create NNE Pharmaplan, the world's leading engineering and consultancy company focused entirely on the pharma and biotech industries.

Inés Aréizaga Esteva (Spain), 25 years old
Education: Chemical Engineer

– You have to be proactive and open-minded as a newcomer and make it clear to your colleagues what you are able to cope. The pharmaceutical field is new to me. But busy as they are, most of my colleagues find the time to teach me, and they also trust me. Even though it was a bit hard at first, I can feel over time that I am beginning to be taken seriously and that my contribution is appreciated.

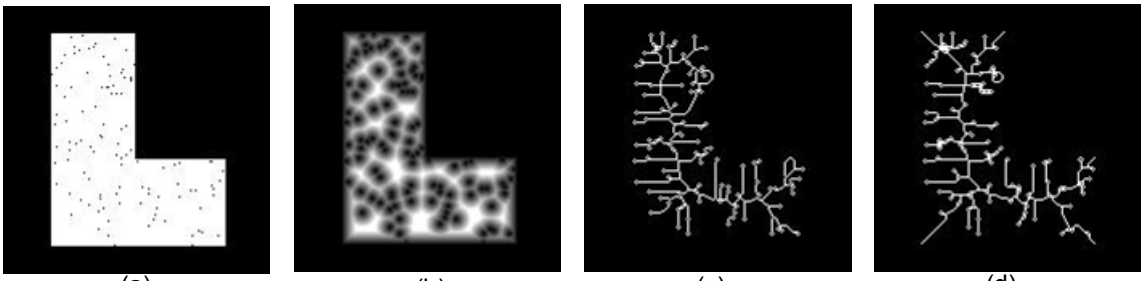
NNE Pharmaplan is the world's leading engineering and consultancy company focused entirely on the pharma and biotech industries. We employ more than 1500 people worldwide and offer global reach and local knowledge along with our all-encompassing list of services. **nnepharmaplan.com**

nne pharmaplan®

Download free ebooks at bookboon.com

Figure 37 Skeleton vs. thinning. (a) Original binary images with various shapes; (b) Distance transformation of image (a); (c) Skeleton images; (d) Thinning images.

Both skeleton and thinning are very sensitive to noise. For example, if some "pepper" noise is added to the image of L shape in **Figure 36** ***(a)***, the resulting skeleton and thinning connects each noise point to the skeleton obtained from the noise free image. This artefact is illustrated in **Figure 38**. Therefore, it is necessary to pre-processing the input image prior to skeletonization.

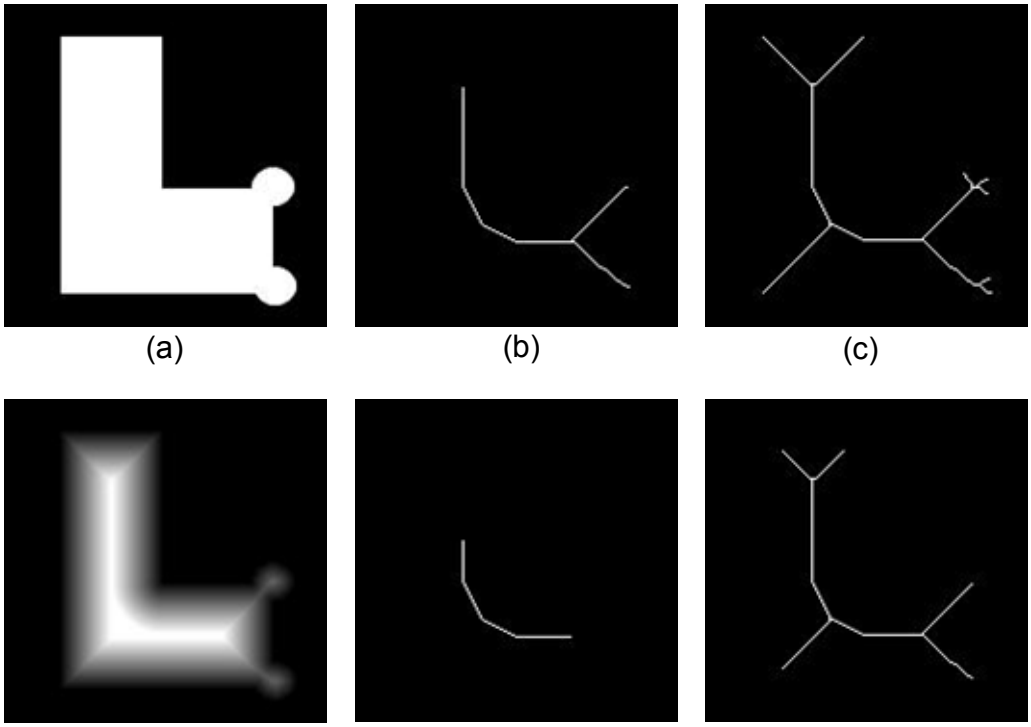


(a) (b) (c) (d)

Figure 38 Both Skeleton and thinning are sensitive to noise. (a) Original L-shaped binary images with added pepper noise; (b) Distance transformation of image (a); (c) Skeleton image; (d) Thinning image.

2.7 Pruning

Often, the thinning or skeleton process will leave *spurs* on the resulting image and they are sensitive to small changes in the boundary of the object, which can produce more artefact skeleton. For example, the skeleton ***(b)*** and thinning ***(c)*** in **Figure 39** compare with those in **Figure 37** respectively.



(a) (b) (c)

(d) (e) (f)

Figure 39 Illustration of pruning. (a) A L-shaped original binary image with some boundary distortion on the bottom right of L; (b) Skeletonization of image (a); (c) Thinning of image (a); (d) Distance transformation of image (a); (e) After 10 iterations of pruning on image (b); (f) After 30 iterations of pruning on image (c).

These are short branches having an endpoint located within three or so pixels of an intersection. Spurs result from single-pixel-sized undulations in the boundary that give rise to a short branch. They can be removed by a series of three-by three operations that remove endpoints (thereby shortening all the branches), followed by reconstruction of the branches that still exist. A three-pixel spur, for example, disappears after three iterations of removing endpoints. Not having an endpoint to grow back from, the spur is not reconstructed. The structure elements for pruning are shown in **Figure 40**.

0	0	0
0	1	0
0	X	X

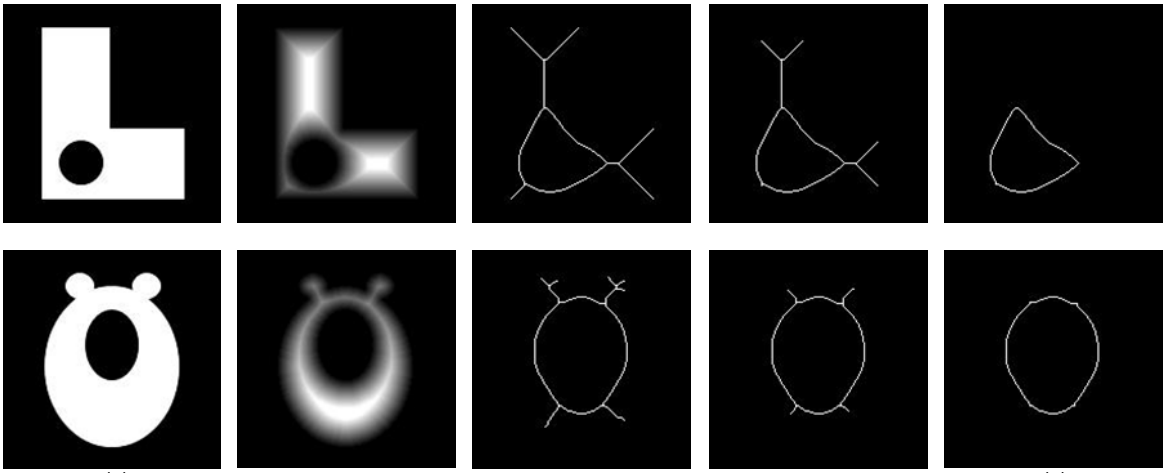
(a)

0	0	0
0	1	0
X	X	0

(b)

Figure 40 Structure elements used for pruning. At each iteration, each element must be used in each of its four 90°.

Pruning is normally carried out only a limited number of iterations to remove short spurs, since pruning until convergence actually removes all pixels except those that form closed loops (see **Figure 41**).



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

Figure 41 Pruning until convergence removes all pixels except those that form a closed loop. (a) Various shaped original binary image; (b) Distance transformation of image (a); (c) Thinning of image (a); (d) After 10 iterations of pruning on image (c); (e) Pruning on image (c) until convergence.

2.8 Morphological reconstruction

2.8.1 Definition of morphological reconstruction

All morphological operators discussed so far involved combinations of *one* input image with specific structuring element. The morphological reconstruction is a transformation involving *two* images and a structure element: the first image is a *marker*, which is a starting point for the transformation; and the second image is a *mask*, which constrains the transformation.

A morphological operator is applied to the first image, *marker*, and it is then forced to remain either greater or lower than the second image, *mask*, (**Figure 43**). Authorized morphological operators are restricted to elementary erosions and dilations. The choice of specific structuring elements is therefore eluded. In practice, these transformations are iterated until stability, making the choice of a size in *marker* image unnecessary. It is the combination of appropriate pairs of input images that produces new morphological primitives. These primitives are at the basis of formal definitions of many important image structures for both binary and gray-scale images.

If G is the mask and F is the marker, the reconstruction of G from F, denoted $R_G(F)$, is defined by the following iterative procedure:

1). Initialize p_1 to be the marker image F.
2). Create a structure element B.
3). Compute the next p_{K+1}

$$p_{k+1} = (p_k \oplus B) \bigcap G \tag{2.8.1}$$

4). Repeat the step *3* until $p_{K+1} = p_K$.

Morphological reconstruction can be thought of conceptually as repeated dilations of the marker image, until the contour of the marker image fits under the mask image. In morphological reconstruction, the peaks in the marker image "spread out," or dilate.

Figure 42 illustrates the morphological reconstruction in 1-D. At each dilation operation, the value of the marker signal at every point takes the maximum value over its neighbourhood. As a result, the values of the dilated marker signal are increased except the local maxima of the marker signal which will stay the same as before. The dilation operation is constrained to lie underneath the mask signal. When further dilations do not change the marker signal any more, the process stops. At this point, the dilated marker signal is exactly the same as the mask signal except the local maxima. By comparing the mask signal and the dilated marker signal, the local maxima of the mask signal can be extracted.

Download free ebooks at bookboon.com

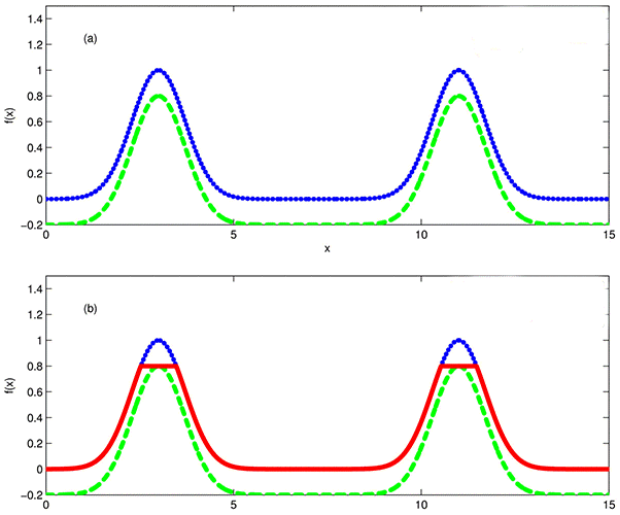
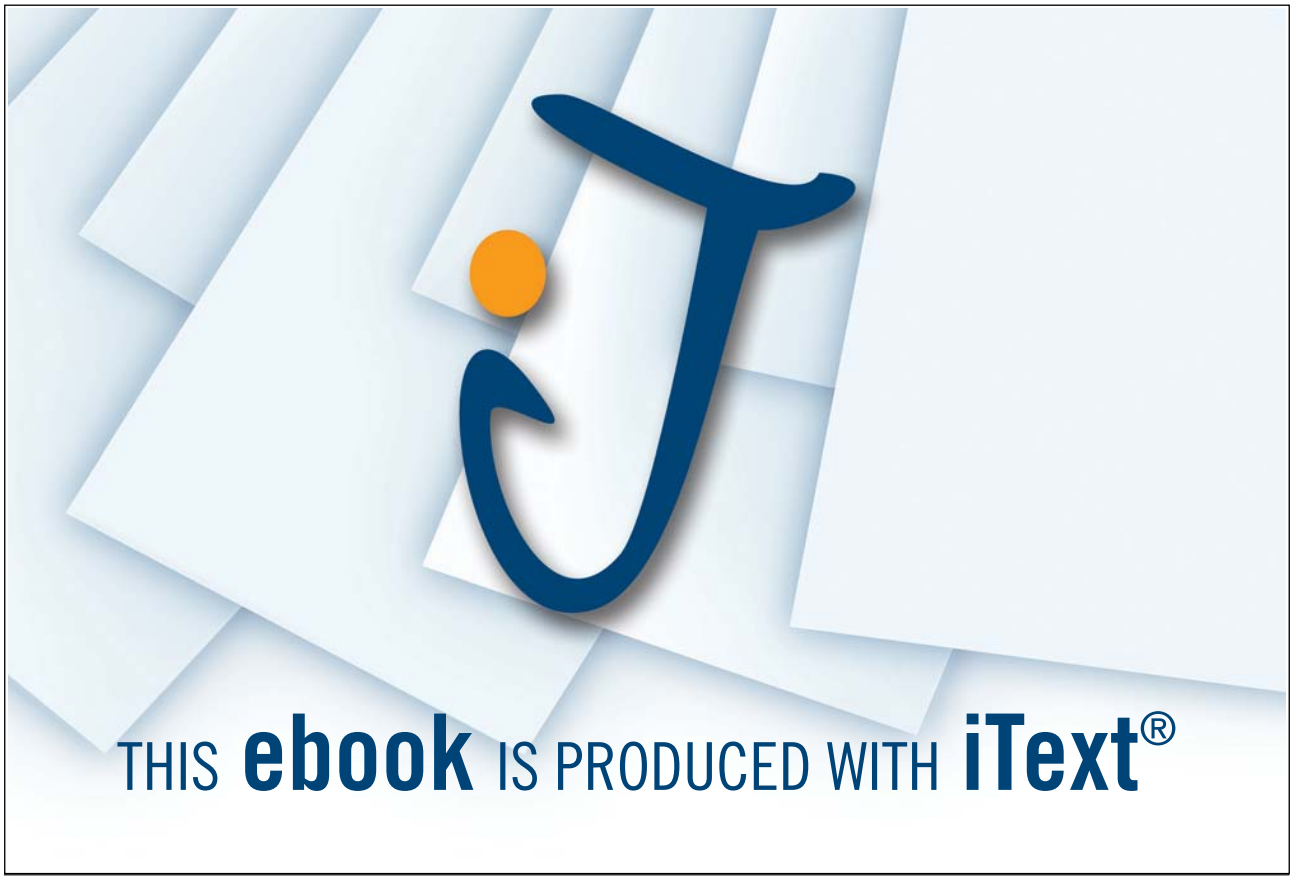


Figure 42 Illustration of morphological reconstruction in 1-D to extract the local maxima. (a) The marker in green is obtained by subtracting a small value of 0.2 from the original signal in blue; (b) Obtain the reconstructed signal in red by using morphological reconstruction, where the difference between the original signal and the reconstructed signal corresponds to the local maxima of the original signal. Note that the marker signal specifies the preserved parts in the reconstructed signal.

Please click the advert



Download free ebooks at bookboon.com

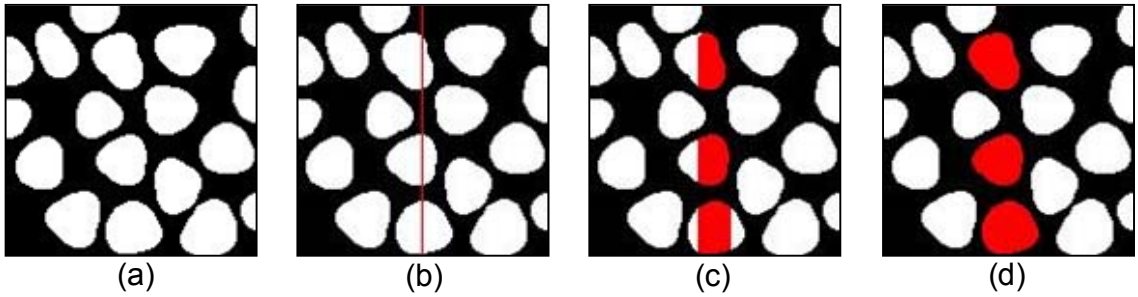


Figure 43 Example of morphological reconstruction.(a) A binary image of the blobs as the mask image; (b) A vertical red line as the marker superimposed on the mask image (a); (c) After 8 successive conditional dilations, the marker gets wider, invading the mask image, mimic the same behaviour of the flood fill effect in the painting. Hereby the structure element used in conditional dilation is an elementary cross, which maintains the connection criteria. (d) The result of morphological reconstruction is stable (in red) and superimposed on the mask image. Note that the red blobs in the image that are connected to the red line (marker). Therefore the reconstruction detects all the pixels that are connected to the markers.

2.8.2 The choice of maker and mask images

Morphological reconstruction algorithms are at the basis of numerous valuable image transformations. These algorithms do not require choosing an SE nor setting its size. The main issue consists of selecting an appropriate pair of mask/marker images. The image under study is usually used as a mask image. A suitable marker image is then determined using:

- Knowledge about the expected result;
- Known facts about the image or the physics of the object it represents;
- Some transformations of the mask image itself;
- Other image data if available (i. e., multispectral and multi-temporal images); and
- Interaction with the user (i.e., markers are manually determined).

One or usually a combination of these approaches is considered. The third one is the most utilized in practice but it is also the most critical: one has to find an adequate transformation or even a sequence of transformations. As the marker image has to be greater (respectively, less) than the mask image, extensive (respectively, anti-extensive) transformations are best suited for generating them.

Some morphological reconstruction-based operations include minima imposition, opening/closing by reconstruction, top-hat by reconstruction, detecting holes and clearing objects connected to the image border.

Download free ebooks at bookboon.com

Summary

The morphological image processing introduced in this chapter is a powerful tool for extracting or modifying features from an image. The basic morphological operators, such as dilation, erosion and reconstruction, are particularly useful to analysis of binary images, although they can be extended for use with gray scale image. Those operators can be used in combination to perform a wide variety tasks, including filtering, background noise reduction, correct uneven illumination, edge detection, feature detection, counting and measuring objects and image segmentation.

References and further reading

[14]. K. R. Castleman, Digital Image Processing, Prentice Hall, 1996.

[15]. E. R. Dougherty and R. A. Lotufo, Hands-on Morphological Image Processing, SPIE Press, 2003.

[16]. R. M. Haralick, S. R. Sternberg and X. Zhang, Image Analysis using Mathematical Morphology, IEEE Transactions on Pattern Analysis and Machine Intelligence, 9:532-550, 1987.

[17]. J. C. Russ, The Image Processing Handbook, 5th edition, CRC Press, 2007.

[18]. P. Soille, Morphological Image Analysis – Principles and Applications, 2nd edition, Springer-Verlag, New York, 2003.

[19]. M. Sonka, and J. M. Fitzpatrick (Editors), Handbook of Medical Imaging: Medical Image Processing and Analysis, SPIE Society of Photo-Optical Instrumentation Engineering, 2000.

[20]. L. Vincent, Morphological grayscale reconstruction in image analysis: efficient algorithms and applications, IEEE Transactions on Image Processing, 2:176-201, 1993.

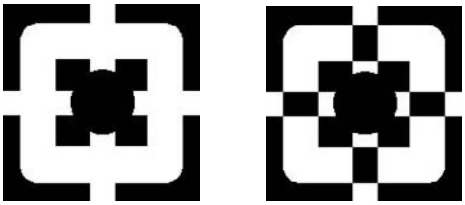
[21]. All illustrations and examples in this chapter have been programmed by Mathworks Matlab, and the source codes can be obtained on request to author.

Please click the advert

Download free ebooks at bookboon.com

Problems

(7) Write down the equations of combined Boolean operations to produce the following binary images by using the images described in this chapter.



(8) Prove that the dilation has the property of duality (see Section 2.2.3).

(9) Prove (or disprove) that binary image A eroded by a structure element B remain invariant under closing by B. That is prove that $A \ominus B = (A \ominus B) \bullet B$.

(10) What is the top-hat transformation and when is it used? Explain how the top-hat transformation can help to segment dark objects on a light, but variable background. Draw a one-dimensional profile through an image to illustrate your explanation.

(11) Sketch the structure elements required for the hit-or-miss transform to locate (i) isolated points in an image; (ii) end points in a binary skeleton and (iii) junction points in a binary skeleton. Several structure elements may be needed in some cases to locate all possible orientations.

(12) What is the major difference between the output of a thinning algorithm and the maxima of the distance transform?

(13) What is the difference between thinning and skeleton?

(14) If an edge detector has produced long lines in its output that are approximately x pixels thick, what is the longest length spurious spur that you could expect to see after thing to a single pixel thickness? Test your estimate on some real images.

Download free ebooks at bookboon.com

3. Image Segmentation

3.1 Introduction

Image segmentation is a fundamental component in many computer vision applications, and can be addressed as a clustering problem [25]. The segmentation of the image(s) presented to an image analysis system is critically dependent on the scene to be sensed, the imaging geometry, configuration, and sensor used to transduce the scene into a digital image, and ultimately the desired output (goal) of the system. **Figure 44** illustrates the typical steps in image analysis, in which the image segmentation is the first step in the workflow.

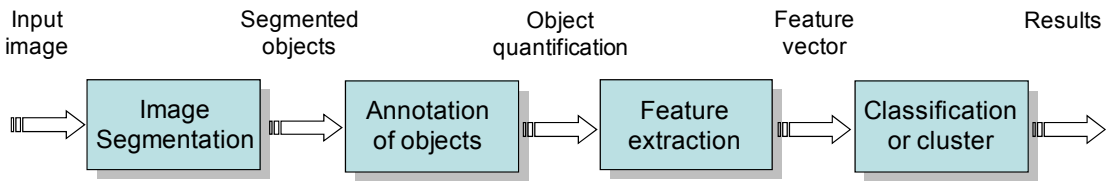


Figure 44 A typical image analysis pipeline.

Image segmentation is typically defined as an exhaustive partitioning of an input image into regions, each of which is considered to be homogeneous with respect to some image property of interest (e.g., intensity, colour, or texture). For example, a segmented image of a slice of mouse embryo is presented in **Figure 45**.

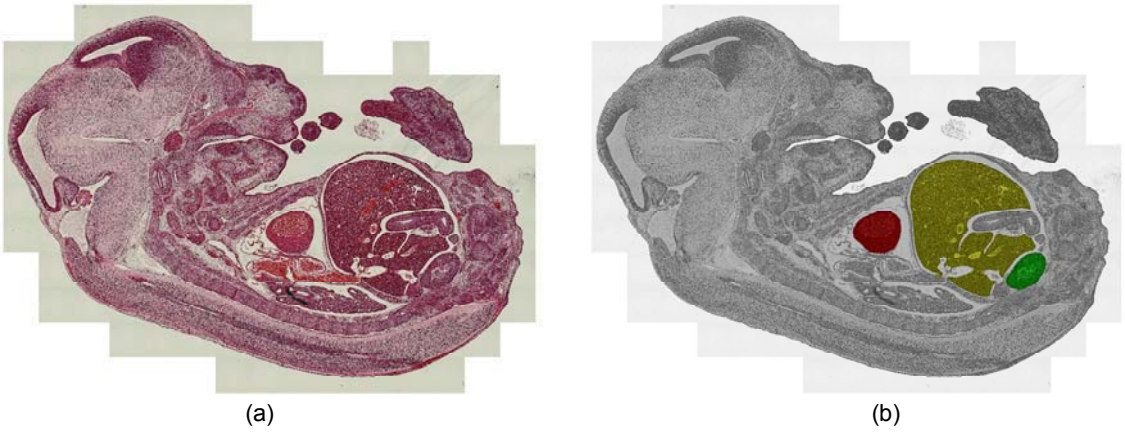


Figure 45 An example of image segmentation. (a) A stained image of a sliced mouse embryo at day 14. (b) Segmented image of (a), where heart, liver and kidney had been identified and marked by colour red, yellow and green.

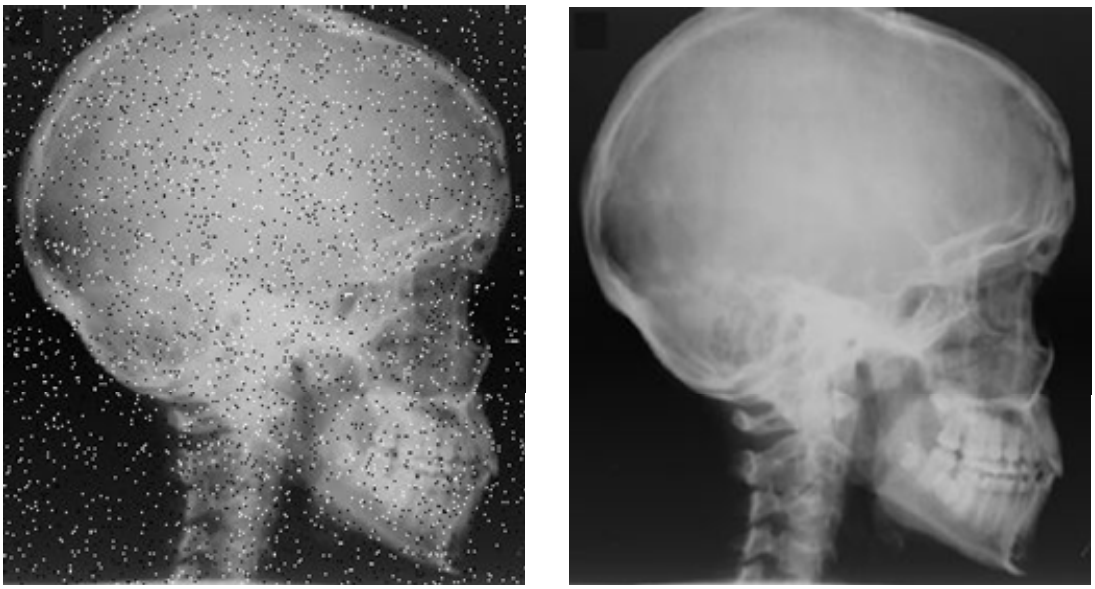
3.2 Image pre-processing – correcting image defects

Many factors can impact the image segmentation procedure. For example effects due to image background, signal-to-noise ratio, feature imaging response and saturation, experimental design and execution all must be taken into account. The first steps in image segmentation are designed to eliminate noise in the image data. This includes both small random specks and trends in background values.

Download free ebooks at bookboon.com

3.2.1 Image smooth by median filter

One of the simplest ways to remove noise is to smooth the image using a non-linear median filter [14]. This simply adds together the pixel brightness values in each small region of the image, divides by the number of pixels in the neighborhood, and then uses the resulting values to construct a new image. **Figure 46** illustrates an X-ray image of skull before and after the application of a 3 × 3 median filter, from which we see that most of the noise has been removed.



(a) (b)

Figure 46 Image smooth by a 3 × 3 median filter. (a) An X-ray image of skull displaying background noise, so called salt and pepper noise; (b) Median filtered image of (a) showing noise removal. For the purposes of printed display the noise in image (a) has been exaggerated.

3.2.2 Background correction by top-hat filter

The top-hat filter (see details in Chapter 2) can be used to remove background articles across a sample. This filter is an example of a suppression operator. It removes pixels from the image if they do not meet some criteria of being "interesting" and leaves alone those pixels that do. It can be used to remove small defects by replacing the pixels with values taken from the surrounding neighbourhood (see **Figure 47** for example).

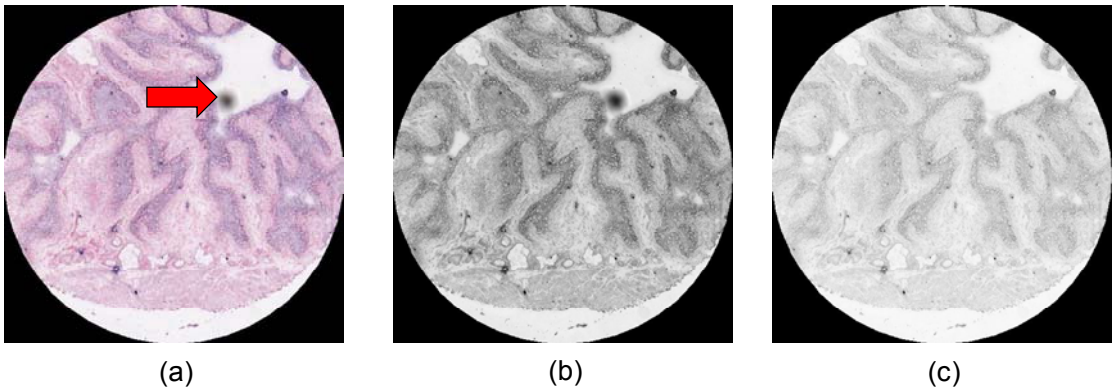
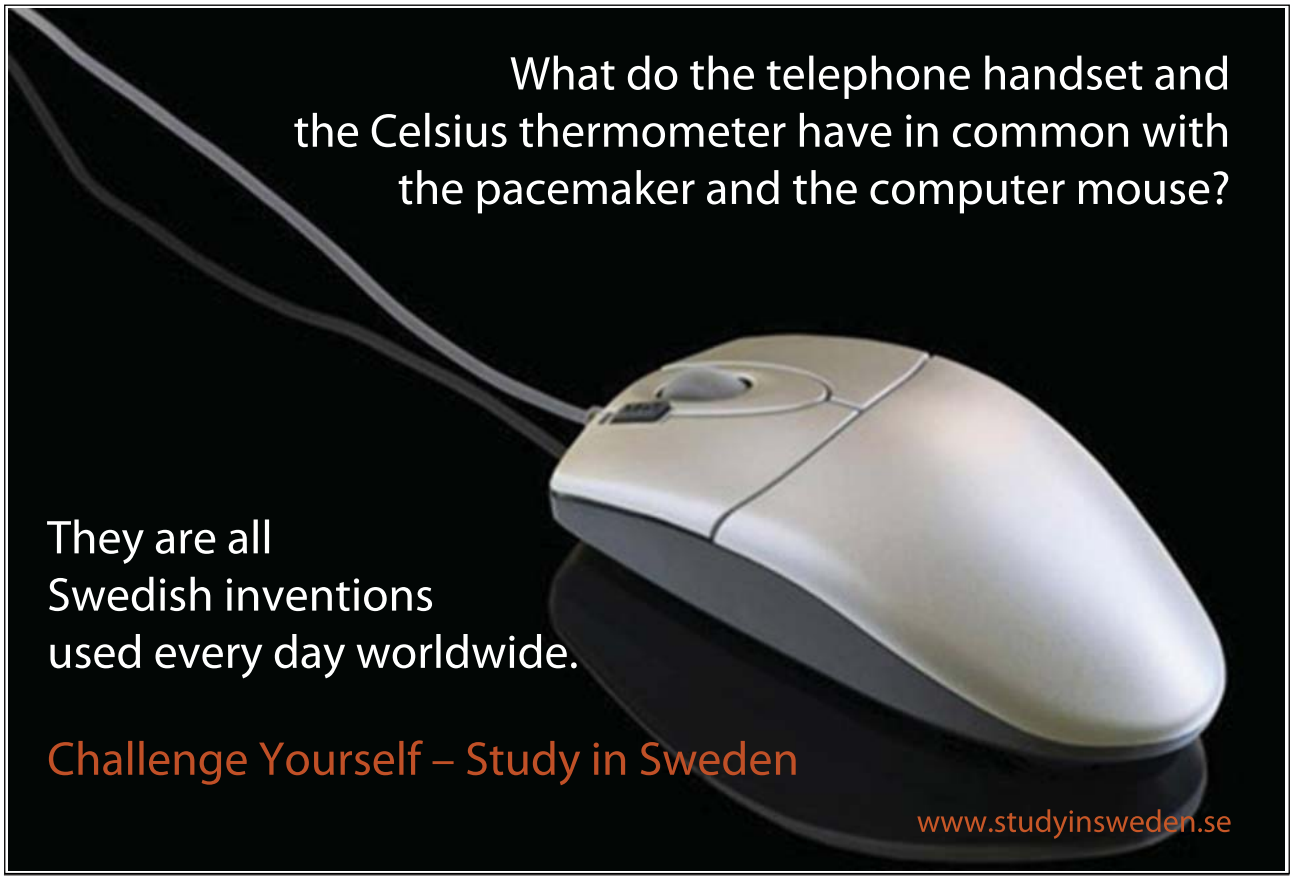


Figure 47 Background correction by top-hat filter. (a) A core image of tissue microarray with background noise, which includes a large spot artefact (red arrow); (b) Image showing colour intensities of the image in (a); (c) Top-hat filtered image showing the removal of the spot artefact.

Please click the advert



Download free ebooks at bookboon.com

3.2.3 Illumination correction by low-pass filter

A low-pass filter can be used to remove large-scale image background variations such as illumination variations across the field of view [14]. Illumination correction exploits the property that illumination variations change at a much lower rate than features do – illumination variation involves only low spatial frequencies. An estimate of the background may be found by removing all but the low spatial frequencies from the image. This can be achieved using a low-pass filter. The background estimate may then be subtracted from the original image to correct for background variation. **Figure 48** shows a low-pass filter correction of an image of nuclei with uneven illumination.

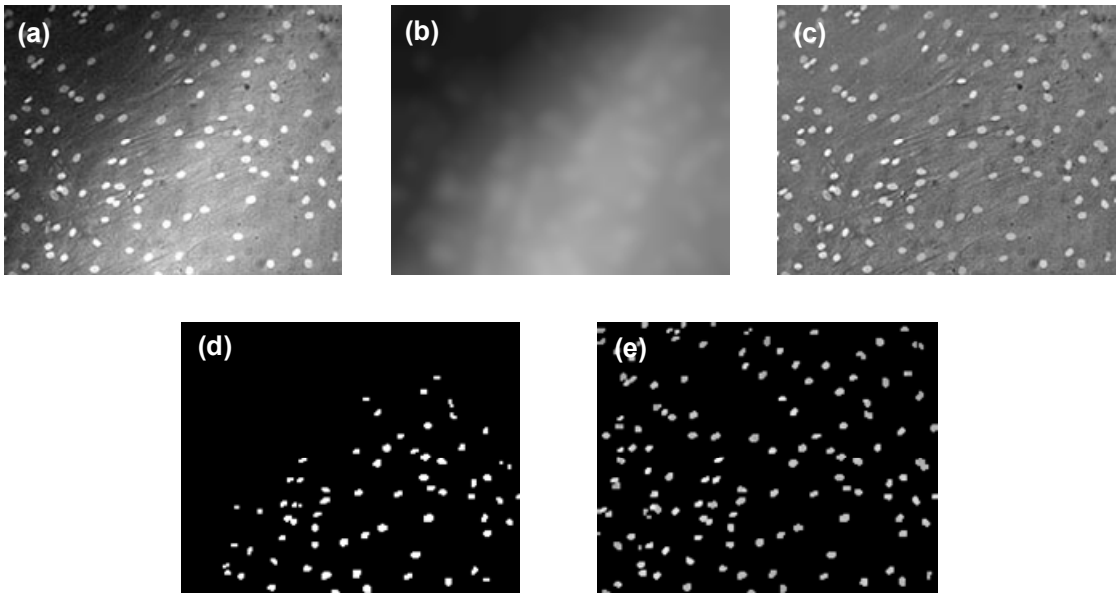


Figure 48 Low-pass filter correction of an image of nuclei with uneven illumination. (a) A field of nuclei with uneven illumination; (b) Estimated background image generated from a low-pass filter on (a); (c) Correction of uneven illumination by subtracting (b) from (a); (d) Segmentation image by thresholding image (a), note that, there are a lot of nuclei missed from the dark background on the left-top corner of the image; (e) Segmentation image by thresholding image (c).

3.2.4 Protocol of pre-process noisy image

We suggest the following methods to deciding how best to pro-process a noisy image.

1). If the noise consists of fine speckle, try a median filter. It should be tried in a small neighbourhood first, and then for larger neighbourhoods [27]. Always evaluate the effectiveness of the noise removal *and* the effect on features of interest.
2). If the speckle is mainly dark or light, try a morphological operation such as a grey level closing or opening with a cylinder or sphere respectively [34].
3). If you have a lot of noise, experiment with wavelet thresholding. The Daubechies-6 and Daubechies-8 wavelets are fairly effective and straightforward to use.
4). If the image has uneven illumination background, try low-pass filter.
5). If noise feature are not all small scale speckle, try alternating sequential morphological filters (see Chapter 2). Note that you have to carefully evaluate a range of filter size.

Download free ebooks at bookboon.com

6). If you intend to process an image with a threshold, try filtering before and after thresholding to determine which is most effective.

7). If none of the above is effective, it may be useful to try morphological operations which filter objects on the basis of size or shape criteria [34]. These need to be used with care to avoid affecting features of interest.

3.3 Thresholding

3.3.1 Fundamentals of image thresholding

Histogram thresholding is one of the popular techniques for monochrome image segmentation. Suppose that the intensity histogram shown in (**Figure 49**) corresponds to an image *f(x, y)*, composed of light object on a dark background. One obvious way to extract the objects from the background is to select a threshold *T* that separates histogram. Then any point *(x, y)* for which $f(x, y) \geq T$ (*e.g.* $T = 124$) is called an object point; otherwise, the point is called a background point. In other words, the thresholded image *g(x, y)* is defined as

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \tag{3.3.1}$$

Please click the advert

FOSS

Sharp Minds - Bright Ideas!

Employees at FOSS Analytical A/S are living proof of the company value - First - using new inventions to make dedicated solutions for our customers. With sharp minds and cross functional teamwork, we constantly strive to develop new unique products - Would you like to join our team?

FOSS works diligently with innovation and development as basis for its growth. It is reflected in the fact that more than 200 of the 1200 employees in FOSS work with Research & Development in Scandinavia and USA. Engineers at FOSS work in production, development and marketing, within a wide range of different fields, i.e. Chemistry, Electronics, Mechanics, Software, Optics, Microbiology, Chemometrics.

We offer
A challenging job in an international and innovative company that is leading in its field. You will get the opportunity to work with the most advanced technology together with highly skilled colleagues.

Read more about FOSS at www.foss.dk - or go directly to our student site www.foss.dk/sharpminds where you can learn more about your possibilities of working together with us on projects, your thesis etc.

The Family owned FOSS group is the world leader as supplier of dedicated, high-tech analytical solutions which measure and control the quality and production of agricultural, food, pharmaceutical and chemical products. Main activities are initiated from Denmark, Sweden and USA with headquarters domiciled in Hillerød, DK. The products are marketed globally by 23 sales companies and an extensive net of distributors. In line with the corevalue to be 'First', the company intends to expand its market position.

Dedicated Analytical Solutions

FOSS
Slangerupgade 69
3400 Hillerød
Tel. +45 70103370

www.foss.dk

Download free ebooks at bookboon.com

Pixels labelled *1* correspond to objects, whereas pixels labelled *0* correspond to the background.

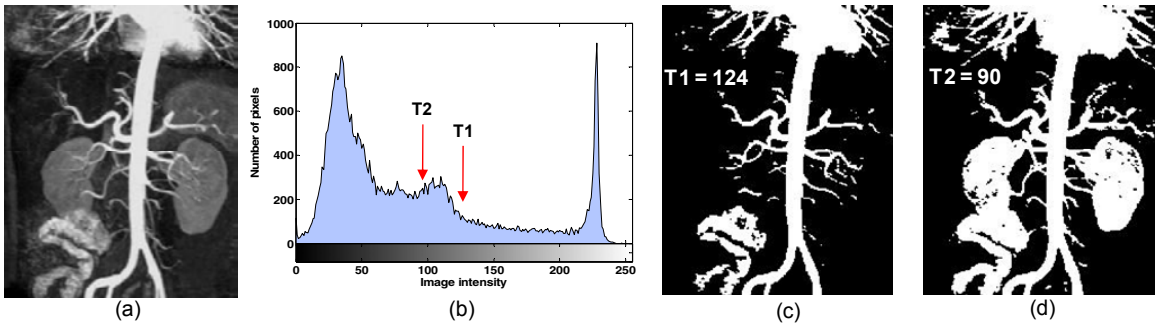


Figure 49 Fundamentals of image histogram thresholding. (a) An MR angiography image showing the aorta and other blood vessels; (b) Intensity histogram of the image in (a); (c) The binary image thresholded with a threshold value T1 = 124, pixels labelled 1 (in white) corresponding to objects; (d) The binary image thresholded with a threshold value T2 = 90.

3.3.2 Global optimal thresholding

Thresholds are either *global* or *local*, i.e., they can be constant throughout the image, or spatially varying. In this section, we discuss the global optimal thresholding. Optimal thresholding methods rely on the maximization (or minimization) of a merit function. The most common non-parametric model is to assume that the histogram is the sum of normal distributions. These methods rely on the definition of a "goodness" criterion for threshold selection. Possibilities include the within-class variance δ^2_w, the between-class variance δ^2_B.

There are a number of approaches to implementing optimal thresholding. The general methodology is to consider the pixels, foreground and background, as belonging to two *classes* or *clusters*. The goal is to pick a threshold such that each pixel on each side of the threshold is closer in value to the mean of the pixels on that side of the threshold than the mean of the pixels on the other side of the threshold. The algorithms proceed automatically, without user intervention, and are said to be *unsupervised*.

A good example of such a technique is the method proposed by Otsu [30] in which the optimum threshold is chosen as the one that maximizes δ^2_B / δ^2_T, with δ^2_T the total variance. **An efficient implementation of optimal thresholding** works as follows [35]:

1). Select two initial thresholds t_1 and t_2; these can be chosen as $G/3$ and $2G/3$, respectively, with G the maximum intensity value of an image.
2). Compute the following:

$$e_1(t_1,t_2) = \frac{m(0,t_1)+m(t_1,t_2)}{2} - t_1$$
$$e_2(t_1,t_2) = \frac{m(t_1,t_2)+m(t_2,G)}{2} - t_2 \qquad (3.3.2)$$

with

$$m(t_i, t_j) = \frac{1}{t_j - t_i + 1} \sum_{g=t_i}^{t_j} g.h(g) \tag{3.3.3}$$

where g is the symbol used for gray level in the histogram $h(g)$ of an image.

3). Update the thresholds t_1 and t_2 as followings:

$$\begin{aligned} t_1 &\leftarrow t_1 + e_1 \\ t_2 &\leftarrow t_2 + e_2 \end{aligned} \tag{3.3.4}$$

4). If e_1 and e_2 are below a preset tolerance, stop; otherwise go to step 2.

An example of Otsu optimal method *vs.* conventional thresholding method is illustrated in **Figure 50**.

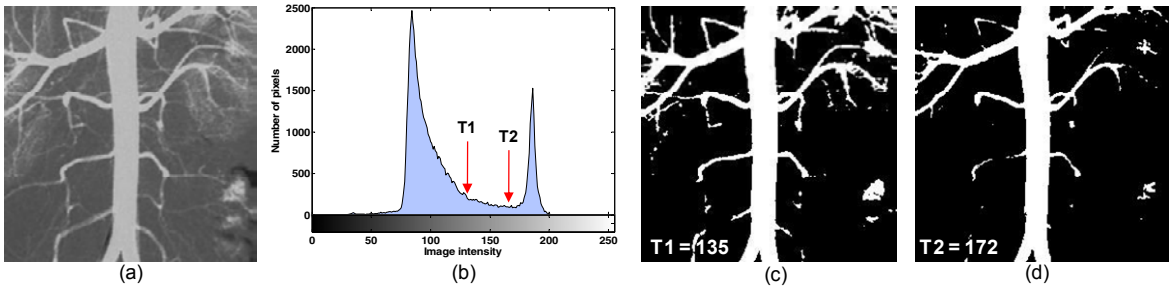


Figure 50 Otsu optimal method vs. conventional thresholding method. (a) An image of blood vessels; (b) Intensity histogram of image in (a); (c) Thresholded image with a threshold value T1 = 135 using Otsu optimal method; (d) Thresholded image with a threshold value T2 = 172 using the conventional thresholding method.

Another **histogram triangle algorithm** is illustrated in **Figure 51**. A line is constructed between the maximum of the histogram at brightness b_{max} and the lowest value $b_{min} = (p=0)\%$ in the image. The distance d between the line and the histogram $h[b]$ is computed for all values of b from $b = b_{min}$ to $b = b_{max}$. The brightness value b_o where the distance between $h[b_o]$ and the line is maximal is the threshold value, that is, $\theta = b_o$. This technique is particularly effective when the object pixels produce a weak peak in the histogram.

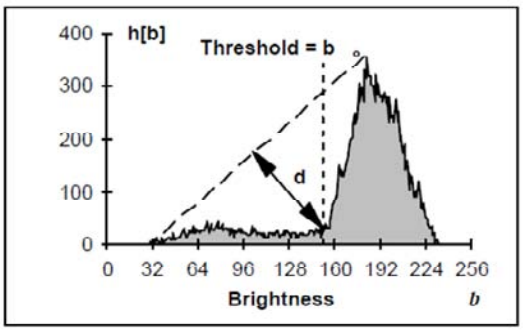


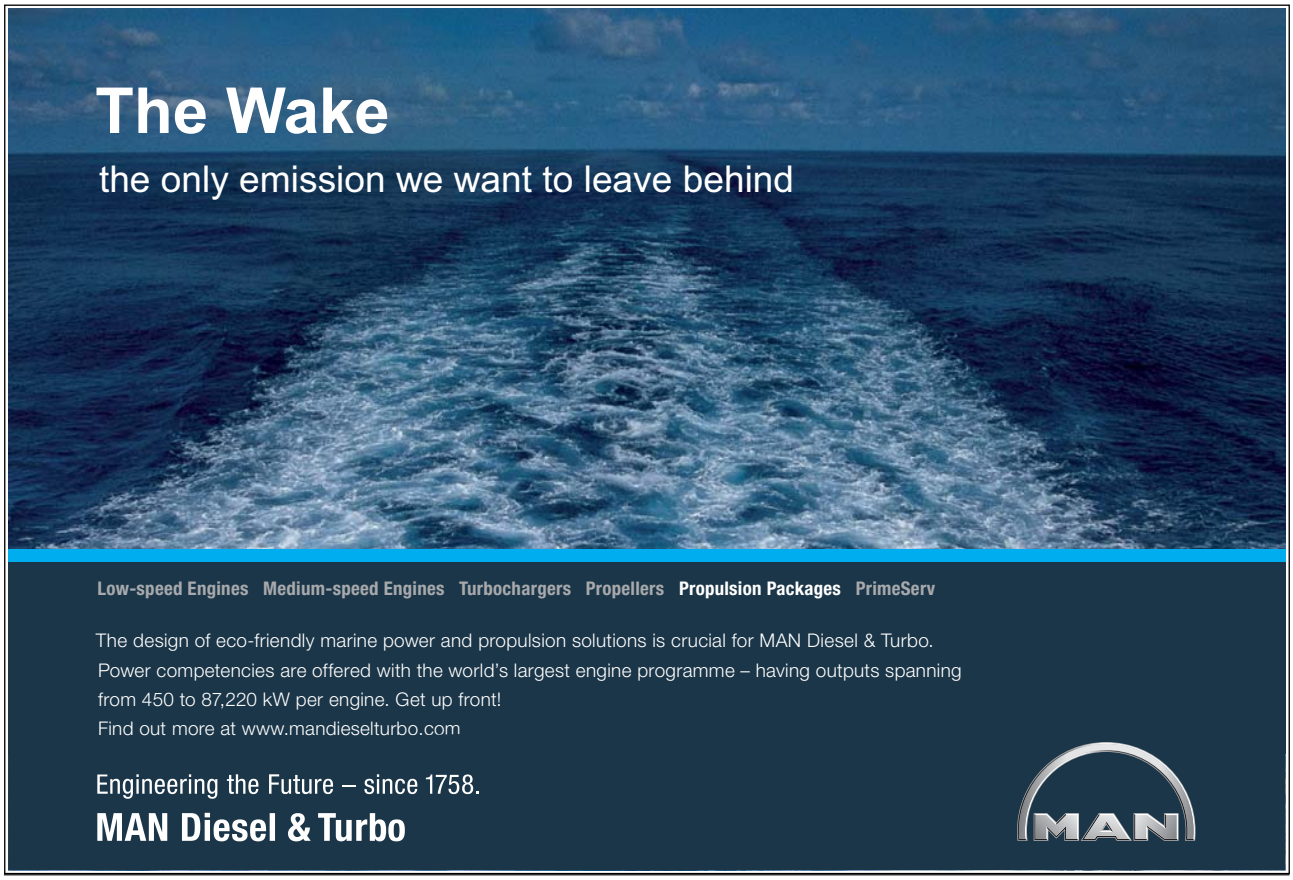
Figure 51 The histogram triangle algorithm is based on finding the value of b that gives the maximum distance d.

3.3.3 Adaptive local thresholding

In order to obtain a satisfactory segmentation result by thresholding, a uniform background is required. Many background correction techniques exist (e.g. see section 3.2.3), but they may not always result in an image that is suitable for thresholding. The transition between object and background may be diffuse, making an optimal threshold level difficult to find. Also, a small change in the threshold level may have a great impact in later analyses.

Adaptive local thresholding can be used to circumvent the problem of varying background, or as a refinement to coarse global thresholding [27]. In adaptive local thresholding, each pixel is considered to have an $n \times n$ neighbourhood around it from which a threshold value is calculated (from the mean or median of these values) and the pixel set to black or white, according to whether it is below or above this local threshold, T_L. The size of the neighbourhood, n, has to be large enough to cover sufficient foreground and background pixels so that the effect of noise is minimal. But not too large that uneven illumination becomes noticeable within the neighbourhood. An example of adaptive local thresholding vs. the Otsu thresholding on an image with a variable background is illustrated in **Figure 52**.

Please click the advert



Download free ebooks at bookboon.com

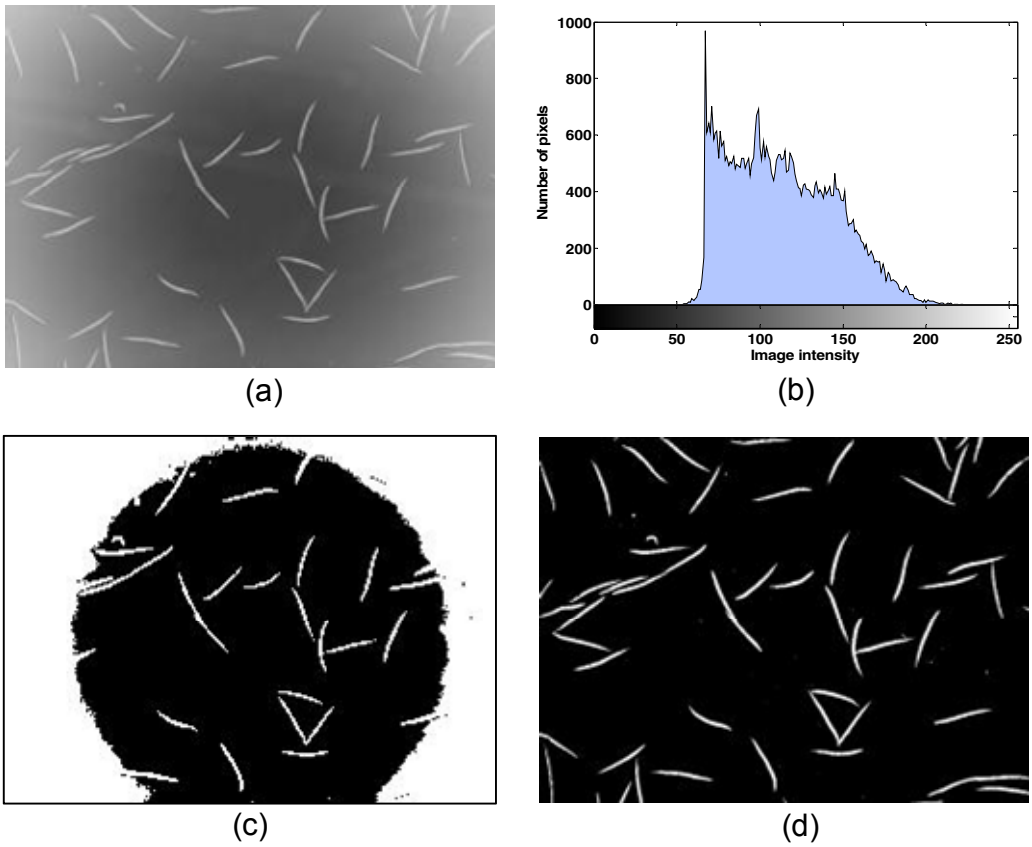


Figure 52 Adaptive local threshold vs. Otsu threshold method. (a) Original image – a microscope image of C. elegans, note that it has uneven illumination background; (b) Intensity histogram of image (a); (c) Segmentation image of (a) using Otsu thresholding, wher threshold value is 117 and it fails to pick up all objects from the surrounding background; (d) Segmentation image of (a) using adaptive local thresholding.

The main drawback in such histogram-based methods is that they do not take shape information into account, and the outcome can be unpredictable especially in cases of low signal-to-noise-ratios. In addition, spike noise or contamination from other spots may be classified into the spot, leading to errors in the estimated intensity values.

3.3.4 Multiple thresholding

A single threshold serves to segment the image into only two regions, a background and a foreground; more commonly however, the objective is to segment the image into multiple regions using multiple thresholds. This multiple thresholding technique considers that an image consist of different regions corresponding to the gray level ranges. The histogram of an image can be separated using peaks (modes) corresponding to the different regions. A threshold value corresponding to the valley between two adjacent peaks can be used to separate these object. The success of thresholding depends critically on the selection of an appropriate threshold.

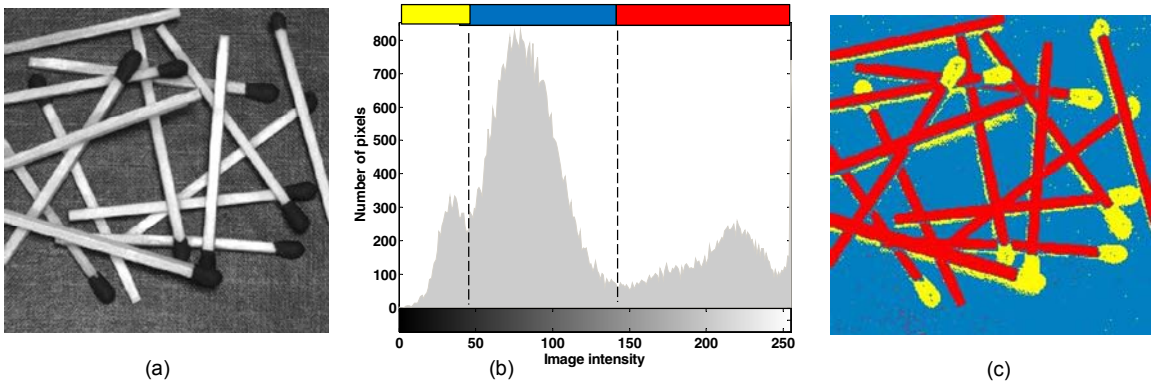


Figure 53 Multiple thresholding. (a) A grey level image of some randomly placed match; (b) Intensity histogram of image in (a); (c) Multiple thresholded image with corresponding threshold value T1 = 45 and T2 = 134.

3.4 Line and edge detection

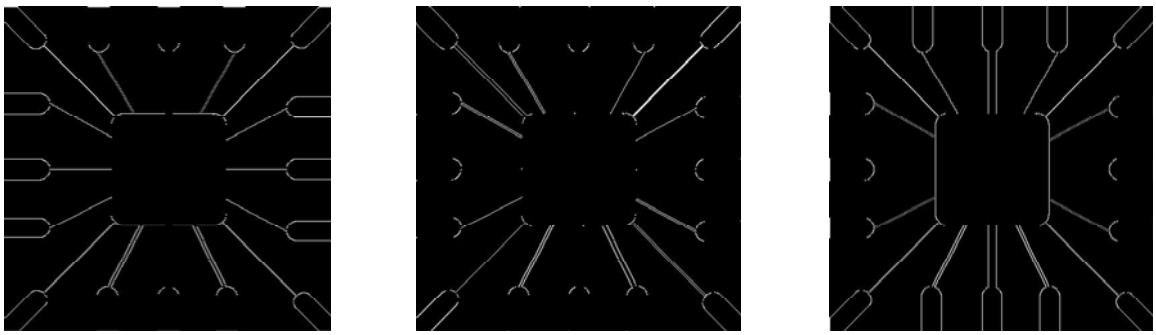
3.4.1 Line detection

Consider the masks in **Figure 54**. If the mask ***b*** were moved around an image, it would respond more strongly to lines (one pixel thick) oriented horizontally. With a constant background, the maximum response would result when the line passed through the middle row of the mask. Similarly, the mask ***c*** in **Figure 54** responds best to lines oriented at *+45°*; and the mask ***d*** to vertical lines. Note that the preferred direction of each mask is weighted with a larger coefficient (i.e. *2*) than other possible directions. The coefficients of each mask sum to zero, indication a zero response from the mask in areas of constant intensity.

(a)

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

(b) (c) (d)



(e) (f) (g)

Figure 54 Line detection. (a) Image of a wire-bond mask; (b) Horizontal line detector mask; (c) +45º line detector mask; (d) Vertical line detector mask; (e) Result of processing with the horizontal line detector mask; (f) Result of processing with the +45º line detector mask; (g) Result of processing with the vertical line detector mask.

3.4.2 Hough transformation for line detection

The Hough transform [26] [35] permits the detection of parametric curves (e.g. straight lines, circles) in a binary image produced by thresholding the output of an edge detector operator. Its major strength is its ability to detect object boundaries even when low-level edge detector operators produce sparse edge maps. The Hough transform is defined for a function *A(x, y)* as:

$$H(\theta,\rho)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}A(x,y)\delta(\rho-x\cos\theta-y\sin\theta)dxdy \qquad (3.4.1)$$

With *A(x, y)*, each point *(x, y)* in the original image, *A*, is transformed into a sinusoid $\rho = x\cos\theta - y\sin\theta$, where ρ is the perpendicular distance from the origin of a line at an angle θ. **Figure 55** indicates the principle of Hough Transform for a line detection and **Figure 56** gives an example of line detection on a satellite image of Pentagon by Hough Transform.

Please click the advert

Download free ebooks at bookboon.com

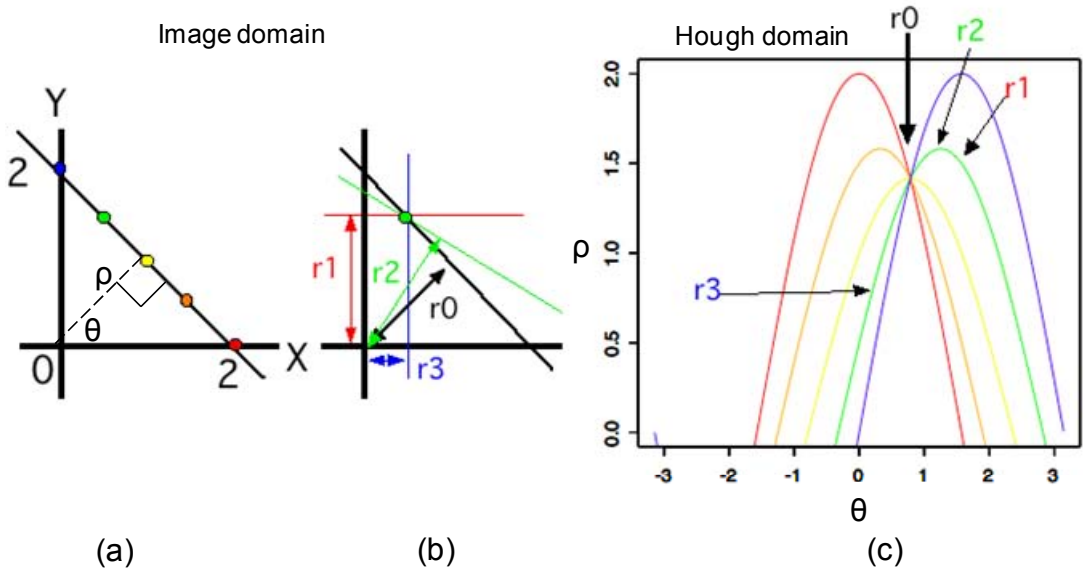


Figure 55 The principle of Hough Transform for line detection. (a). Five points on a straight line in an image domain; (b) To see how to build a sine curve in the Hough domain, we use the green point; For this single point, various lines that pass through it can be drawn (e.g. red, green, black, and blue lines). Then we calculate each line's direction θ and distance ρ from the origin (0, 0). This procedure is accomplished by drawing perpendiculars (lines with arrow heads) from the origin to each line; (c) The Hough domain representations of various lines that lie on each point are shown, and the representing sine curves are coloured correspondingly to the points in the image domain (b). Note that r1, r2, r3, and r4 are points corresponding to perpendiculars in (b), respectively. The r0 point is the crossing point of all sine curves, which represents the line in the image domain (a) that lies on all five points.

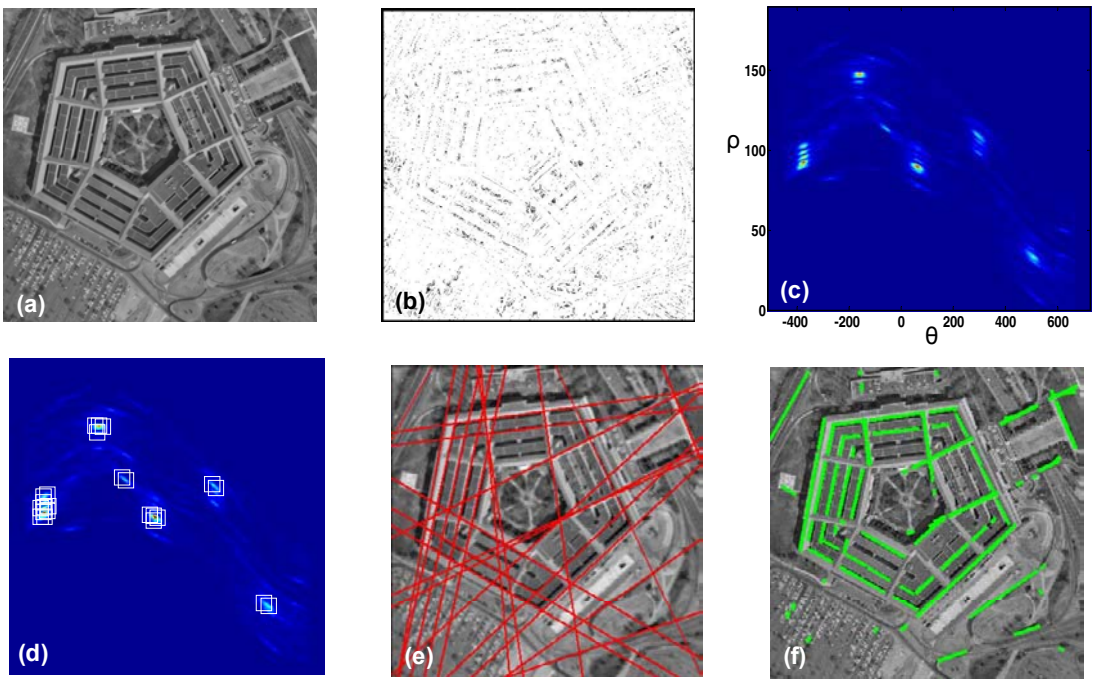


Figure 56 Line detection by Hough Transform. (a) A satellite image of Pentagon; (b) Gradient magnitude image of (a); (c) Hough transformation of the gradient field image (b). Note that a thresholding on the gradient magnitude (b) is performed before the voting process. In other words, pixels with gradient magnitudes smaller than 'threshold' are considered not belong to any line; (d) 24 local maximum peak points in Hough field had been marked by white squares; (e) The straight red line corresponding to the 24 local maximum peak points in Hough field had been projected back to the original image via inverse Hough transformation; (f) Line segments had been automatically extracted.

3.4.3 Edge filter operators

An edge detector finds the boundary of an object. These methods exploit the fact that the pixel intensity values change rapidly at the boundary (edge) of two regions. Examples of edge detectors (**Figure 57**) are *Canny, Laplacian, Prewitt, Roberts and Sobel* filters [14] [28] [32], which generally are named after their inventors.

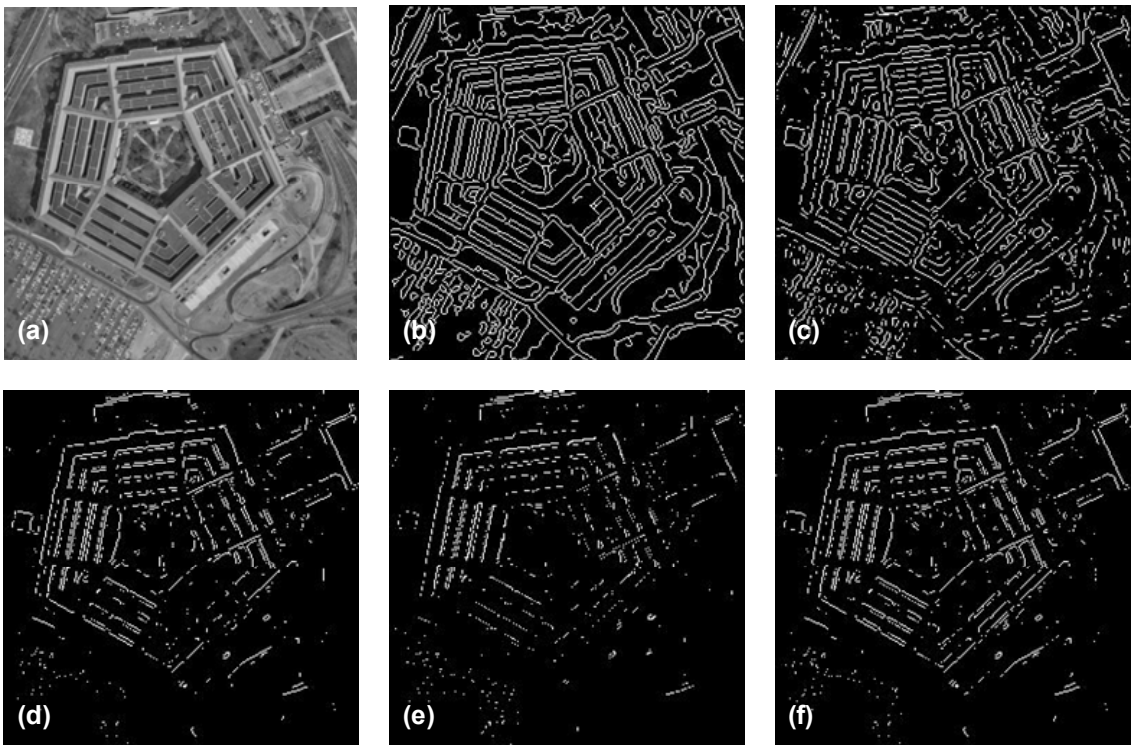


Figure 57 Examples of edge detectors. (a) A satellite image of Pentagon; (b) Canny filter; (c) Laplacian filter; (d) Prewitt filter; (e) Roberts filter; (f) Sobel filter.

The ***Sobel filter*** is one of the most useful and widely available edge filters. It first estimates the intensity gradient in the horizontal and vertical directions from linear filters with coefficients respectively.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.4.2}$$

The maximum intensity gradient is then estimated as the square root of the sum of the squares of the horizontal and vertical gradients. The direction of the maximum gradient is also available from the arctangent of the ratio of the vertical and horizontal gradients.

In addition, ***Canny edge detector*** is also very popular. It takes account of the trade-off between sensitivity of edge detection versus the accuracy of edge localization. The edge detector consists of four stages:

1) Gaussian smoothing to reduce noise and remove small details
2) Gradient magnitude and direction calculation
3) Non-maximal suppression of smaller gradients by larger ones to focus edge localization
4) Gradient magnitude thresholding and linking that uses hysteresis so as to start linking at strong edge positions, gut then also track weaker edges.

It finds edges by looking for local maxima of the gradient of an image. The method uses two thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. Therefore, this method is more likely to detect true weak edges.

The Prewitt filter and Robert filter find edges using the Prewitt and Robert approximation respectively to the derivative. It returns edges at those points where the gradient of input image is at maximum. E.g. ***Prewitt filter*** is specified by:

Please click the advert

Download free ebooks at bookboon.com

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{3.4.3}$$

Roberts filter is specified by:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{3.4.4}$$

The ***Laplacian of Gaussian filter*** finds edges by looking for zero crossings after filtering input image with a Laplacian of Gaussian filter. Consider the Gaussian function

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}} \tag{3.4.5}$$

Where $r^2 = x^2 + y^2$ and is the standard deviation σ. This is a smoothing function which, if convolved with an image, will blur it. The degree of blurring is determined by the value of σ. The Laplacian of this function (the second derivative with respect to r) is

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}} \tag{3.4.6}$$

For obvious reasons, this function is called the Laplacian of a Gaussian (LoG). Because the second derivative is a linear operation, convolving (filtering) an image with $\nabla^2 h(r)$ is the same as convolving the image with the smoothing function first and then computing the Laplacian of the result. This is the key concept underlying the LoG detector. We convolve the image with $\nabla^2 h(r)$, knowing that is has two effects: It smoothes the image (thus reducing noise), and it computes the Laplacian, which yields a double-edge image. Locating edges then consists of finding the zero crossing between the double edges.

3.4.4 Border tracing - detecting edges of predefined operators

The simplest solution to finding the boundary of a structure is to follow the edge detection operation by a border tracing algorithm [35]. Assuming that the edge detector produces both edge magnitude *e(x, y)* and edge orientation *Ø(x, y),* the successor b_{j+1} of a boundary pixel b_j is chosen as the pixel in the neighbourhood (4- or 8-connected) for which the following inequalities hold:

$$\begin{aligned} &\left|e(b_j) - e(b_{j+1})\right| < T_1 \\ &\left|\phi(b_j) - \phi(b_{j+1})\right| \bmod 2\pi < T_2 \\ &\left|e(b_j)\right| > T \\ &\left|e(b_{j+1})\right| > T \end{aligned} \tag{3.4.7}$$

With T_1, T_2, and T predetermined thresholds. If more than one neighbour satisfies these inequalities, then the one that minimizes the differences is chosen. The algorithm is applied recursively, and neighbours can be searched, for instance, starting from the top left and proceeding in a row wise manner.

Once a single point on the boundary has been identified, simply by location a gray level maximum, the analysis proceeds by following or tracking the boundary, and ultimately returning to the starting point before investigating other possible boundaries (**Figure 58**).

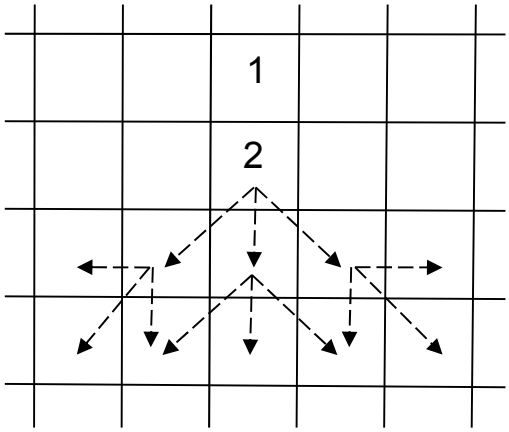


Figure 58 Boundary tracking. In one implementation, find boundary pixel (1); search eight neighbours to find next pixel (2); continues in broadly the same direction as in the previous step, with deviations of one pixel to either side permitted, to accommodate curvature of the boundary, repeat final step until end of boundary.

3.5 Segmentation using morphological watersheds

3.5.1 Watershed transformation

This is best understood by interpreting the intensity image as a landscape in which holes, representing minima in the landscape, are gradually filled in by submerging in water. As the water starts to fill the holes, this creates catchment basins, and, as the water rises, water from neighbouring catchment basins will meet. At every point where two catchment basins meet, as dam, or watershed, is built. These watersheds represent the segmentation of the image. This is illustrated in **Figure 59**.

Download free ebooks at bookboon.com

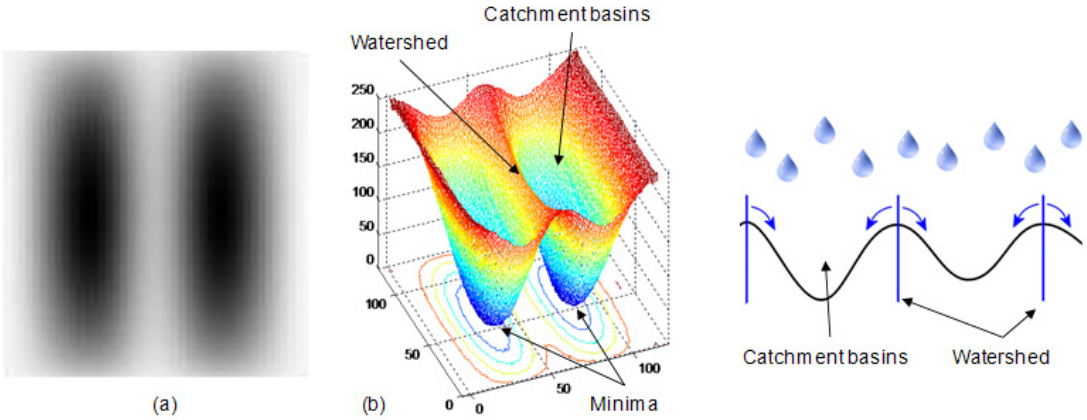


Figure 59 Watershed principle. (a) Synthetically generated grey scale image of two dark blobs; (b) Understanding the watershed transform requires that you think of an image as a topographic surface. If you imagine that bright areas are "high" and dark areas are "low," then it might look like the surface. With surfaces, it is natural to think in terms of catchment basins and watershed lines. If we flood this surface from its minima and, if we prevent the merging of the waters coming from different sources, we partition the image into two different sets: the catchment basins and the watershed lines.

This is a powerful tool for separating touching convex shapes [27] [34]. Indeed, provided that the input image has been transformed so as to output an image whose minima mark relevant image objects and whose crest lines correspond to image object boundaries, the watershed transformation will partition the image into meaningful regions. An example of watershed segmentation is illustrated in **Figure 60**.

Please click the advert

Download free ebooks at bookboon.com

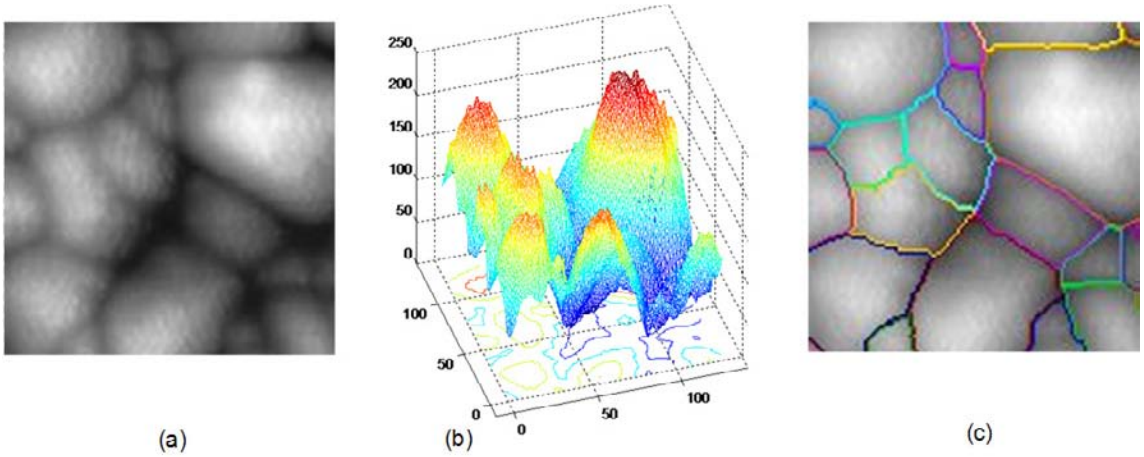


Figure 60 Watershed segmentation. (a) Original gray scale image; (b) Surface representation of the image (a); (c) Watershed segmentation applied and separated feature outlines superimposed on the original image.

3.5.2 Distance transform

A tool commonly used in conjunction with the watershed transform for segmentation is distance transform [27] [32]. The distance transform of a binary image is a relatively simple concept: it is the distance from every pixel to the nearest nonzero-valued pixel. **Figure 61** illustrates the principle of distance transform. **Figure 62** shows an example how the distance transform can be used with watershed transform.

1	1	1	0	0	0
1	1	0	0	0	0
1	1	0	0	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	1	1	1	1	0

(a)

0.00	0.00	0.00	1.00	2.00	3.00
0.00	0.00	1.00	1.41	2.24	2.83
0.00	0.00	1.00	1.00	1.41	2.24
1.00	1.00	1.00	0.00	1.00	2.00
1.41	1.00	1.00	0.00	1.00	1.41
1.00	0.00	0.00	0.00	0.00	1.00

(b)

Figure 61 Principle of distance transformation. (a) A binary image matrix; (b) Distance transform of the binary image. Note that 1-valued pixels have a distance transform value of 0.

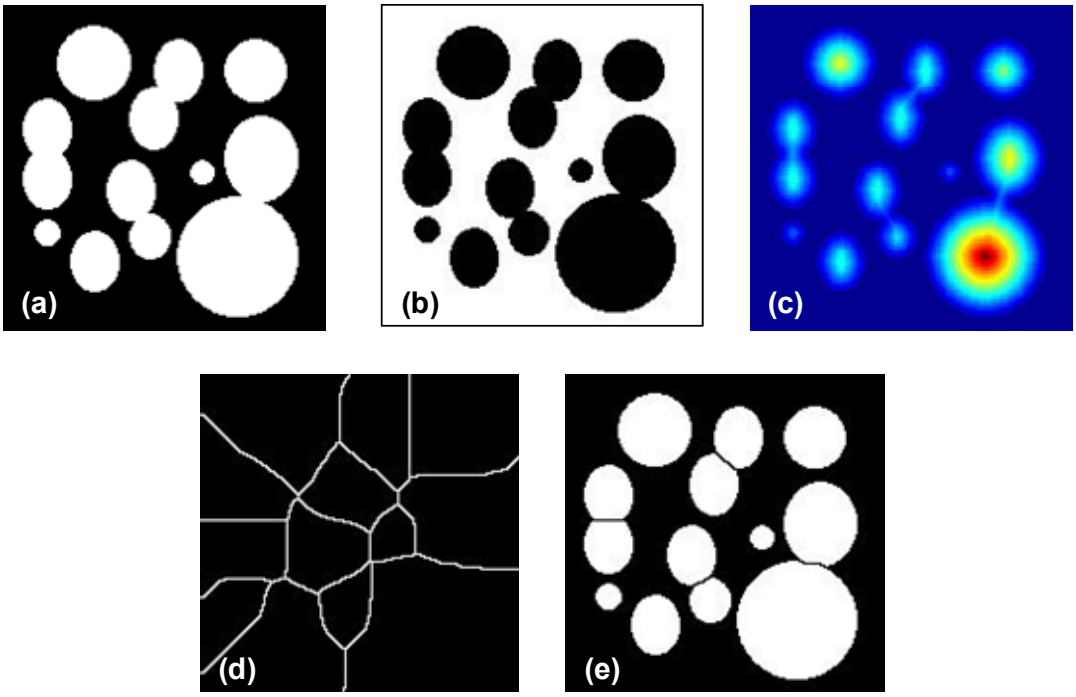


Figure 62 Watershed segmentation using the distance transformation. (a) Original binary image of circular blobs, some of which are touching each other; (b) Complement of image in (a); (c) Distance transform of image in (b); (d) Watershed ridge lines o the negative of the distance transform; (e) Watershed ridge lines superimposed in black over the original binary image.

3.5.3 Watershed segmentation using the gradient field

Often it is preferable to segment the morphological gradient of an image rather than the image itself. The gradient magnitude image has high pixel values along object edges, and low pixel values everywhere else. Ideally, then, the watershed transform would result in watershed ridge lines along object edges. **Figure 63** illustrates this concept.

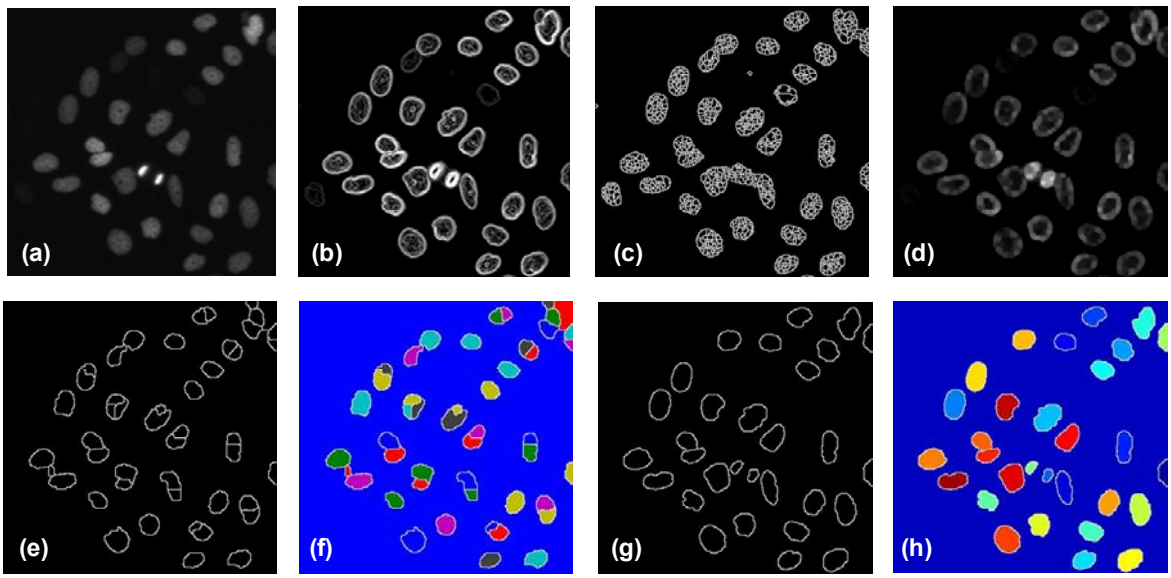
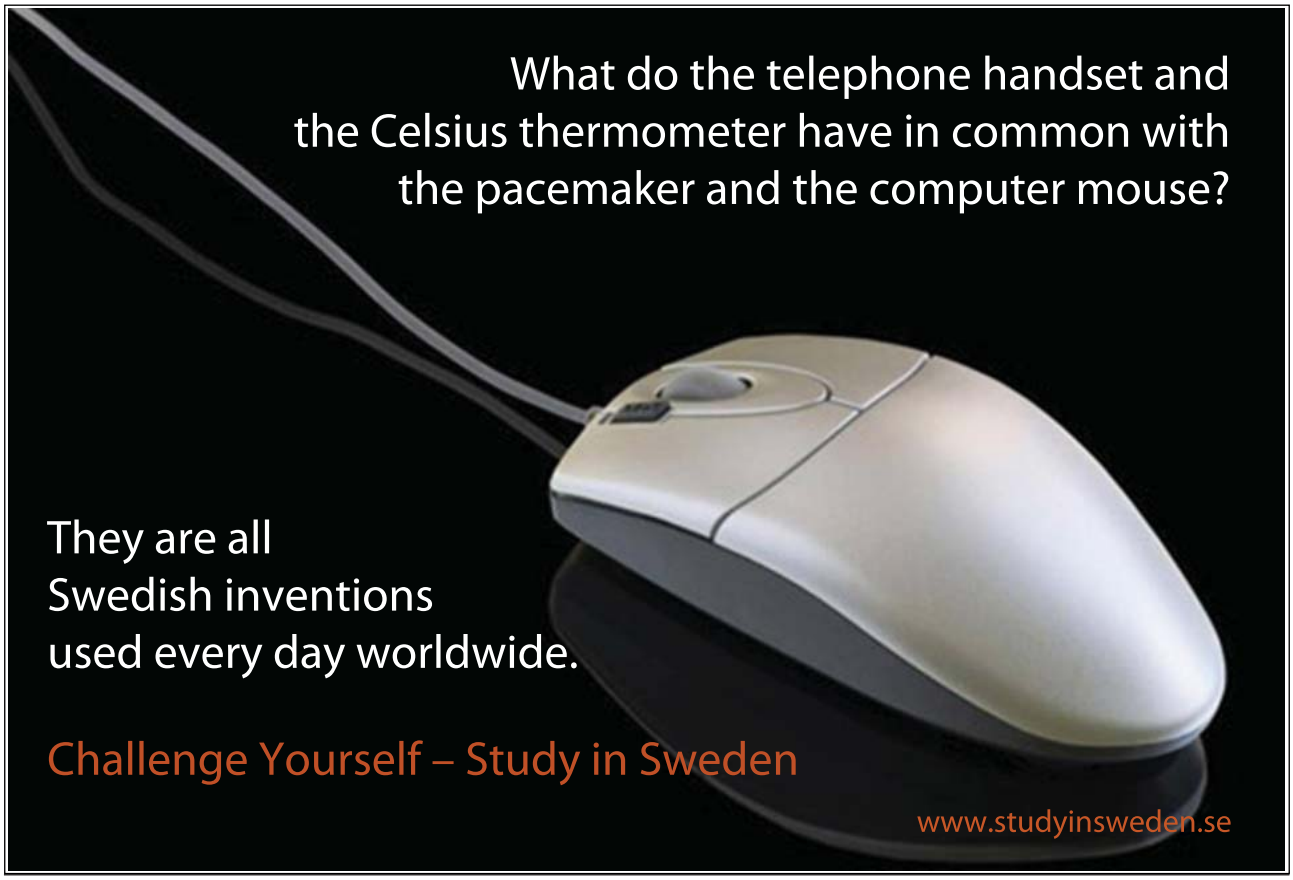


Figure 63 Watershed segmentation using the gradient field. (a) Grey scale image of nuclei from siRNA screening; (b) Gradient magnitude image using Sobel edge filter; (c) Watershed transform of the gradient magnitude image (b), the watershed ridge lines show sever over-segmentation. There are too many watershed ridge lines that do not correspond to the objects in which we are interested; (d) In order to overcome the over-segmentation, a smooth filter (morphological close-opening) is applied on the gradient magnitude image (b); (e) Watershed transform of the smoothed gradient image, but there is still some evident of over-segmentation; (f) Pseudo-colour image of segmented objects in image (e); (g) Improved watershed transform using controlled-markers described in the next section; (h) Pseudo-colour image of improved segmented objects in image (g).

3.5.4 Marker-controlled watershed segmentation

The basic idea behind the marker-controlled segmentation [25] [32] is to transform the input image in such a way that the watersheds of the transformed image correspond to meaningful object boundaries. The transformed image is called the *segmentation function*. In practices, a direct computation of the watersheds of the segmentation function produces an over-segmentation which is due to the presence of spurious minima. Consequently, the segmentation function must be filtered before computing its watersheds so as to remove all irrelevant minima. The minima imposition technique is the most appropriate filter in many applications. This technique requires the determination of a *marker function* marking the relevant image objects and their background. The corresponding markers are then used as the set of minima to impost to the segmentation function (**Figure 64**).

Please click the advert



Download free ebooks at bookboon.com

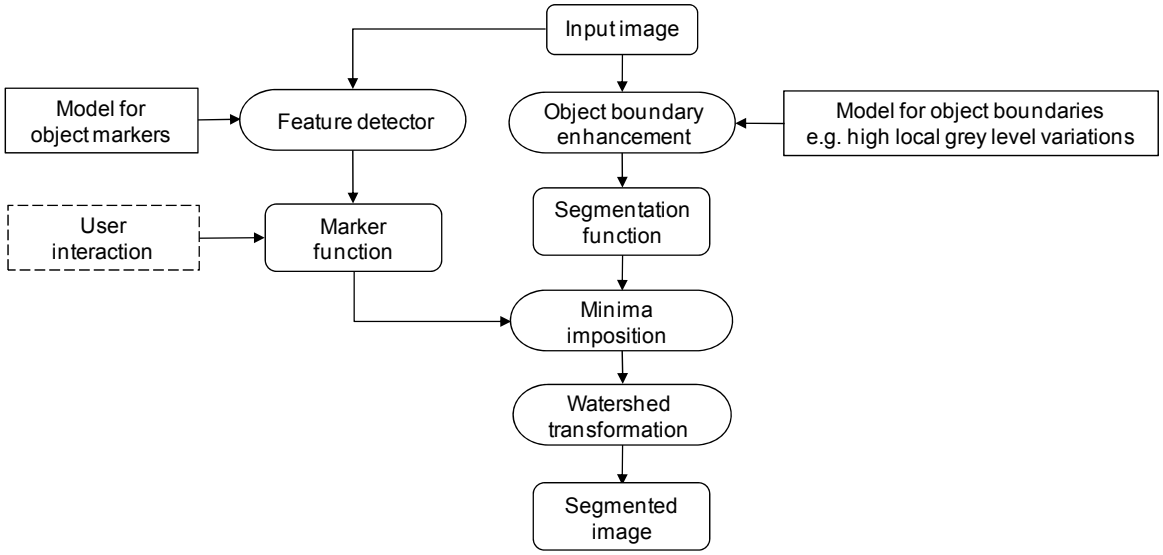


Figure 64 The schematic of marker-controlled watershed segmentation.

In practice, watershed segmentation often produces over-segmentation due to noise or local irregularities in the input image (see image d in **Figure 65**). To reduce this it is common to apply some form of smoothing operation to the input image to reduce the number of local minima. Even so, objects are often segmented into many pieces, which must be merged in a post-processing step based on similarity (e.g. variance of the pixels of both segments together). A major enhancement of the process consists in flooding the topographic surface from a previously defined set of markers, so called marker-controlled watershed segmentation. This prevents over-segmentation taken place (**Figure 65**).

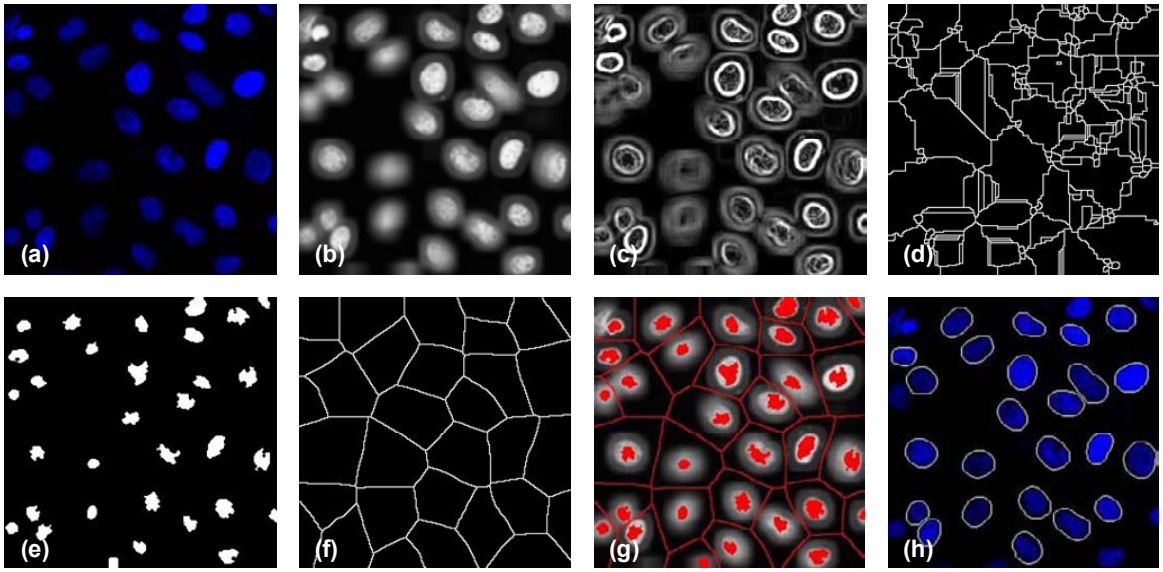


Figure 65 Example of marker-controlled watershed segmentation. (a) An image of nuclei from a siRNA screening; (b) Convert the RGB colour image into a grey level image and also apply a smooth filter in order to remove the noise; (c) Sobel edge detection on image (b); (d) Over-segmentation resulting from applying the watershed transform to the gradient image; (e) Internal markers by computing the local minima of the smoothed image (b); (f) External markers from applying the watershed transform to the internal markers (e); (g) Both internal and external markers superimposed on the segmentation function image; (h) Segmentation results superimposed on the original input image. Note that the objects connected to the border have been removed.

3.6 Region-based segmentation

3.6.1 Seeded region growing

This technique [22] finds the homogeneous regions in an image. The criteria for homogeneity can be based on grey-level, colour, texture, shape model using semantic information, etc. Here, we need to assume a set of seed points initially. The homogeneous regions are formed by attaching to each seed point those neighbouring pixels that have correlated properties. This process is repeated until all the pixels within an image are classified (see **Figure 66**). However, the obscurity with region based approaches is the selection of initial seed points. Moreover, it is superior to the thresholding method, since it considers the spatial association between the pixels.

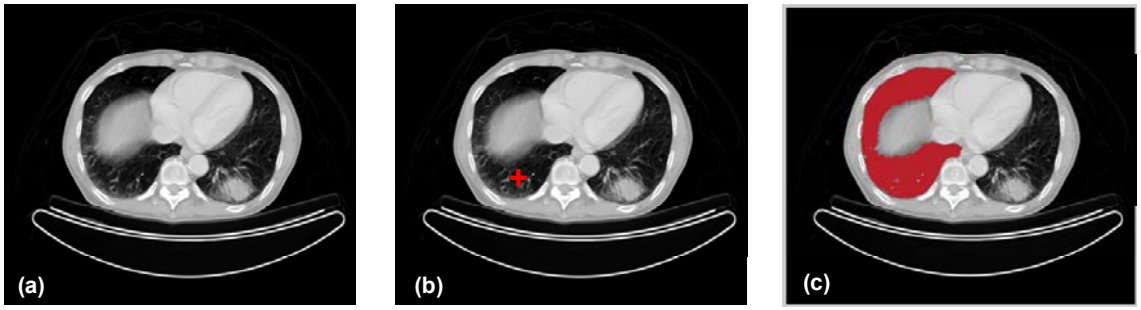


Figure 66 Seeded region growing. (a) Original input CT image; (b) A seed (marked by red "+") had been selected for a region to be segmented; (c) Segmentation mask by seeded region growing.

Seeded region growing algorithm for region segmentation works as followings:

1). Label seed points using a manual or automatic method.
2). Put neighbours of seed point in the sequentially sorted list (SSL).
3). Remove first pixel p from the top of the SSL.
4). Test the neighbours of p.
If all neighbours of p that are already labelled have the same label, assign this label to p, update the statistics of the corresponding region, and add the neighbours of p that not yet labelled the SSL according to their similarity measure $\Delta(p)$ between the pixel and the region.
Else, label p with the boundary label.
5). If the SSL is not empty, go to step 3, otherwise stop.

3.6.2 Region splitting and merging

The opposite approach to region growing is region splitting. It is a top-down approach and it starts with the assumption that the entire image is homogeneous. If this is not true, the image is split into four sub images. This splitting procedure is repeated recursively until we split the image into homogeneous regions. Since the procedure is recursive, it produces an image representation that can be described by a tree whose nodes have four sons each. Such a tree is called a Quad-tree (**Figure 67**).

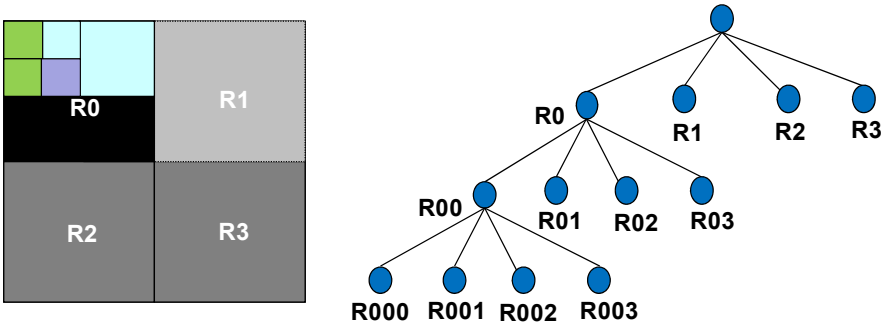
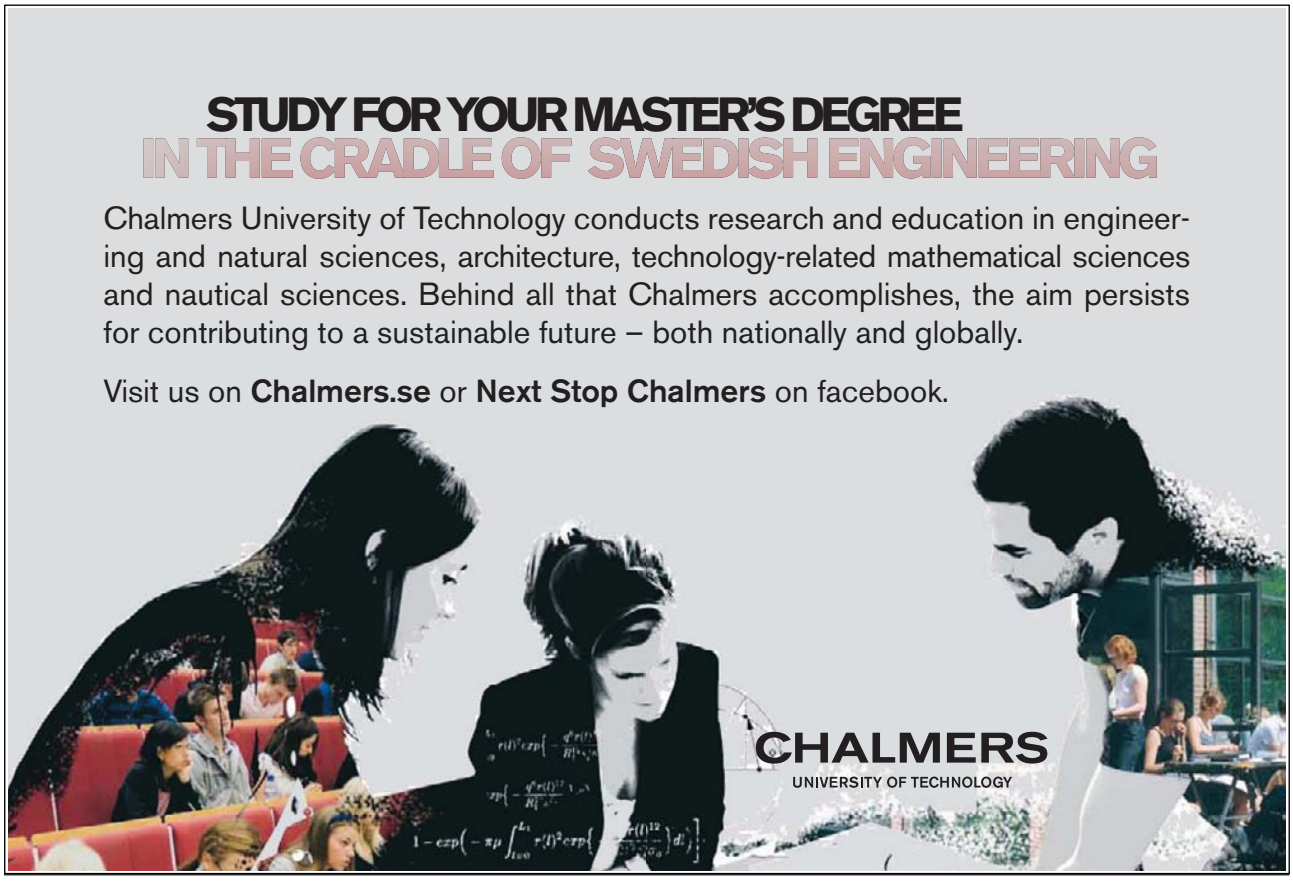


Figure 67 Segmentation Quad-tree.

The main drawback of the region splitting approach is that the final image partition may contain adjacent regions with identical properties. The simplest way to address this issue is to add a merging step to the region splitting method, leading to a **split and merge algorithm** [14] [27]:

Please click the advert



Download free ebooks at bookboon.com

1). Define a similarity criterion *P(R)* for a region *R*.
2). If a region *R* is inhomogeneous, *P(R) = FALSE*, then region *R* is split into four sub regions.
3). If two adjacent regions R_i, R_j are homogeneous, $P(R_i \cup R_j) = TRUE$, they are merged.
4). The algorithm stops when no further splitting or merging is possible.

The example of region splitting and merging for segmentation is shown in **Figure 68**.

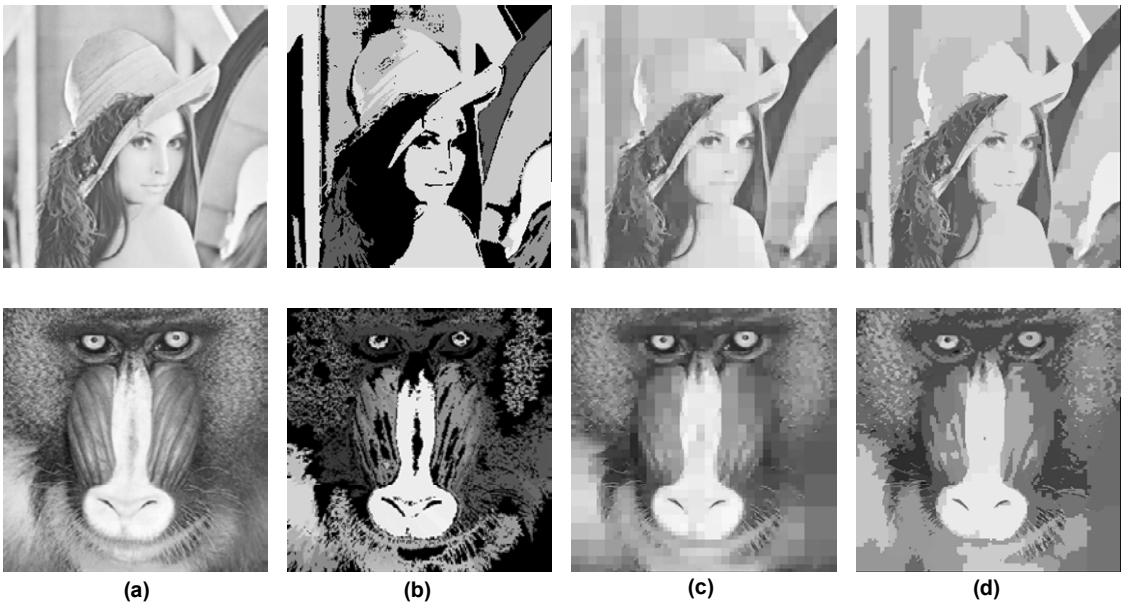
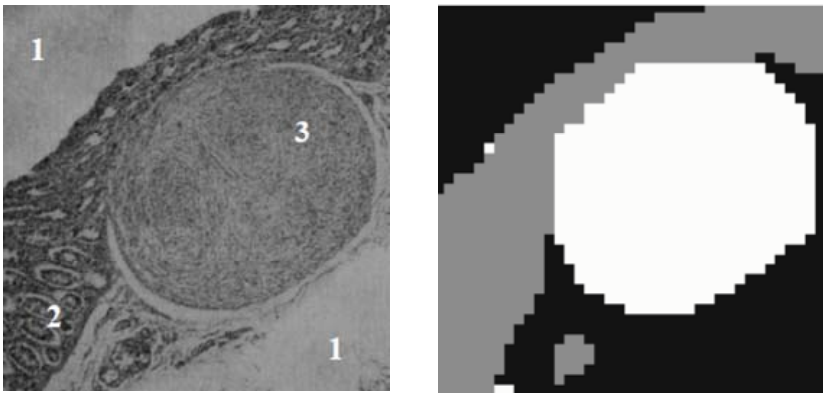


Figure 68 Two examples of region splitting and merging for segmentation. (a) Original input image; (b) Segmentation by region merge only; (c) Segmentation by region spit only; (d) Segmentation by region split and merge.

3.7 Texture-based segmentation

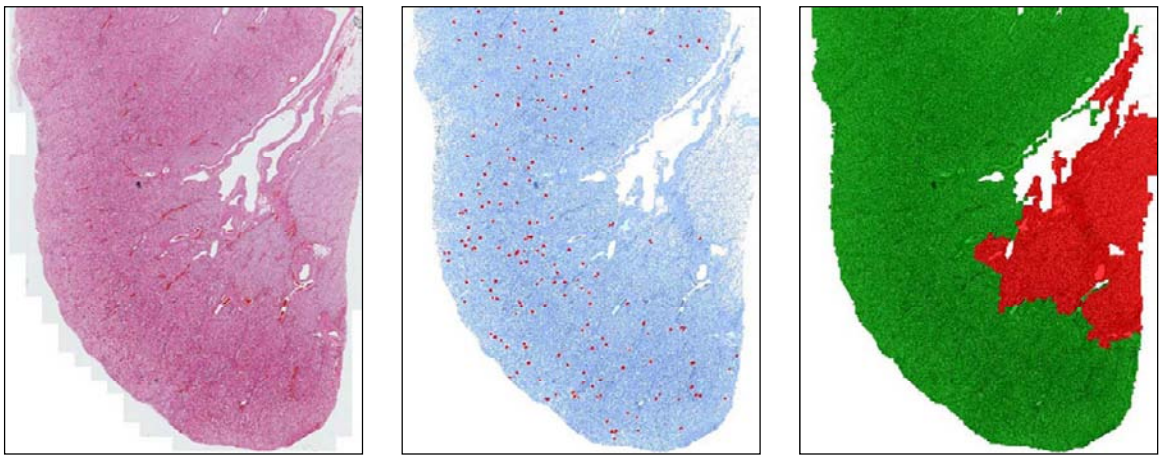
So far we have discussed segmentation methods based on image intensity; however, many images contain areas that are clearly differentiated by texture that could be used as a means of achieving segmentation. For example, in the kidney the cortex and medulla can be differentiated from each other by the density and location of structures such as glomerulus (**Figure 70**). Texture is characterized not only by the grey value at a given pixel, but also by the pattern in a neighborhood that surrounds the pixel. Texture features and texture analysis methods can be loosely divided into statistical and structural methods where the following approaches can be applied: Hurst coefficient, grey level co-occurrence matrices (**Figure 69**), the power spectrum method of Fourier texture descriptors, Gabor filters, and Markov random fields etc [25] [29] [31] [32]. An example of texture-based segmentation is given in **Figure 70**.

Download free ebooks at bookboon.com



(a) (b)

Figure 69 Segmentation of sample microscopic image representing biological tissues by using grey level co-occurrence matrices texture analysis. (a) Source image with three texture classes; (b) image after segmentation.



(a) (b) (c)

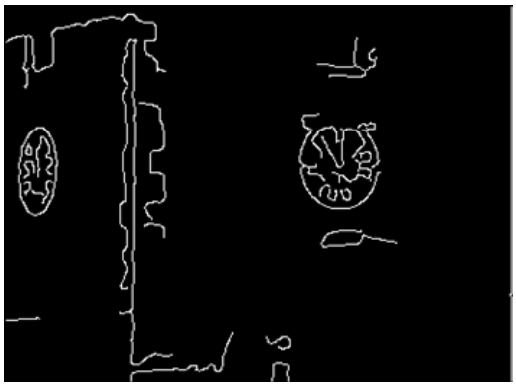
Figure 70 Texture-based image segmentation. (a) Imaging stitched mouse kidney tissue section; (b) Identification of kidney tissue from background (represented by colour blue) and finding glomerulus (represented by colour red); (c) Extract cortex (represented by colour green), and medulla (represented by colour red).

3.8 Segmentation by active contour

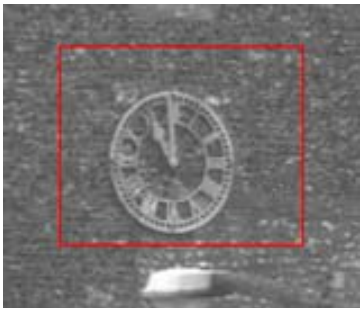
In computer vision, recognising objects often depends on identifying particular shapes in an image. For example suppose we are interested in the outlines of the clock faces. We might start by looking to see whether image edges will help - so we might try a Canny edge detector (Section 3.4.3). As it happens, with these parameters, there is a simple contour round the left clock face, but the contour of the right clock face is rather broken up. In addition, bits and pieces of other structure inevitably show up in the edge map (**Figure 71**). Clearly, using an edge detector alone, however good it is, will not separate the clock faces from other structure in the image. We need to bring more *prior knowledge (conditions)* to bear on the problem. Active contour models, or snakes, allow us to set up such general conditions, and find image structures that satisfy the conditions.

Download free ebooks at bookboon.com

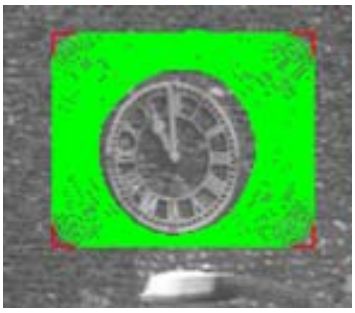
(a)



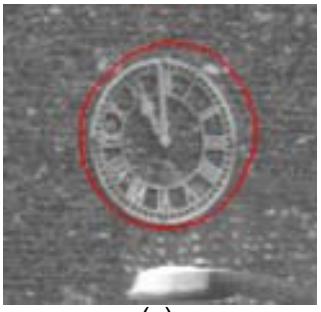
(b)



(a)



(b)



(c)

Please click the advert

GCHQ
it's an interesting world

Get under the skin of it.

Graduate opportunities
Cheltenham | £24,945 + benefits

One of the UK's intelligence services, GCHQ's role is two-fold: to gather and analyse intelligence which helps shape Britain's response to global events, and, to provide technical advice for the protection of Government communication and information systems. In doing so, our specialists – in IT, internet, engineering, languages, information assurance, mathematics and intelligence – get well beneath the surface of global affairs. If you thought the world was an interesting place, you really ought to explore our world of work.

TOP 100 · TOP GOVERNMENT EMPLOYER · INVESTORS IN PEOPLE

www.careersinbritishintelligence.co.uk

Applicants must be British citizens. GCHQ values diversity and welcomes applicants from all sections of the community. We want our workforce to reflect the diversity of our work.

Download free ebooks at bookboon.com

Figure 71 Active contour. (a) Original image of clock faces; (b) Edge detection of image (a) by using Canny filter; (c) To illustrate the active contour (snake), suppose we know that there is a clock face in the rectangular region (in red) of the image; (d) Snake to shrink, to try to form a smooth contour, and to avoid going onto brighter parts of the image; (e) The final position of the snake is shown. The snake has converged on the contour of the outside of the clock face, distorted a little by the bright flint at 1 o'clock.

In an active contour framework, object segmentation is achieved by evolving a closed contour to the object's boundary, such that the contour tightly encloses the object region (**Figure 71**). Evolution of the contour is governed by an energy functional which defines the fitness of the contour to the hypothesized object region. The snake is *active* because it is continuously evolving so as to reduce its energy. By specifying an appropriate energy function we can make a snake that evolves to have particular properties such as smoothness.

The energy function for a snake is in two parts, the *internal* and *external* energies.

$$E_{snake} = E_{internal} + E_{external} \tag{3.1}$$

The internal energy depends on the intrinsic properties of the snake, such as its length or curvature. The external energy depends on factors such as image structure and particular constraints the user has imposed. A snake used for image analysis attempts to minimize its total energy, which is the sum of the internal and external energies. Snakes start with a closed curve and minimize the total energy function to deform until they reach their optimal state. In general, the initial contour should be fairly close to the final contour but does not have to follow its shape in detail: the active contour/snake method is semi-automatic since it requires the user to mark an initial contour (**Figure 72**). The main advantage of the active contour method is that it results in closed coherent areas with smooth boundaries, whereas in other methods the edge is not guaranteed to be continuous or closed.

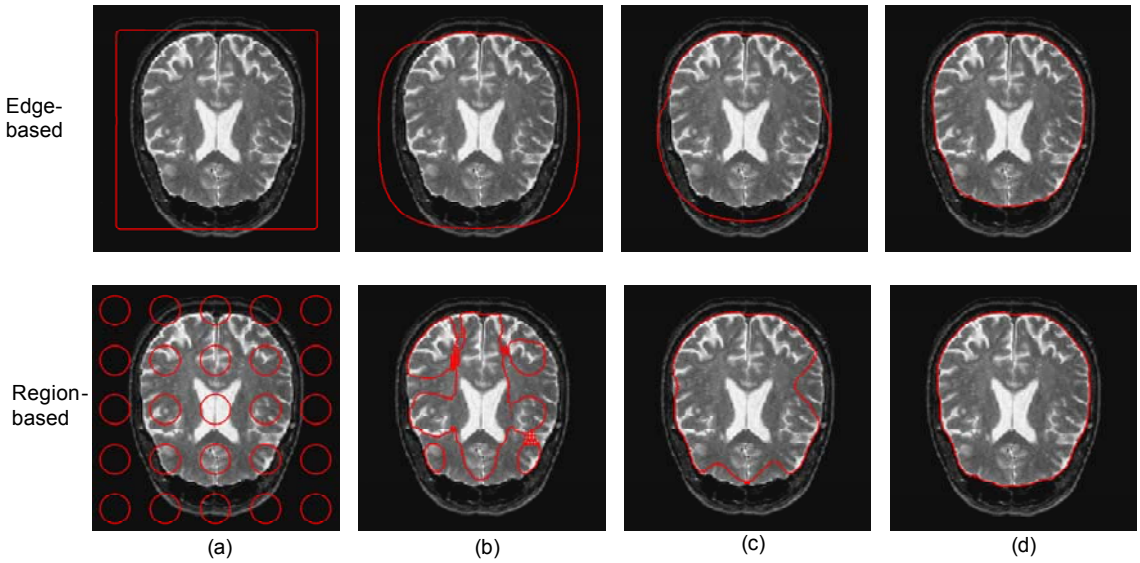


Figure 72 Active contour is semi-automatic since it requires the user to mark an initial contour. The first row is edge-based active contour (snake + level set), and the second row is region-base active contour. (a) Initial contour by user input; (b) and (c) Intermediate contour; (d) Final contour.

Download free ebooks at bookboon.com

3.9 Object-oriented image segmentation

The challenge of understanding images is not just to analyze a piece of information locally, such as intensities, but also to bring the context into play. Object oriented image analysis overcomes this operational gap between productivity and image morphology complexity, which is based on human-like cognition principle – cognition network technology (Cellenger, Definiens and [23]).

The image data is represented as image objects (**Figure 73**). Image objects represent connected regions of the image. The pixels of the associated region are linked to the image object with an "is-part-of" link object. Two image objects are neighboured to each other, if their associated regions are neighboured to each other. The neighbourhood relation between two image objects is represented by a special neighbour link object. The image is partitioned by image objects; all image objects of such a partition are called an image object level. The output of any segmentation algorithm can be interpreted as a valid image object level. Each segment of this segmentation result defines the associated region of an image object. Two trivial image object levels are the partition of the image into pixels (the pixel level) and the level with only on object covering the entire image (the scene level). Image object levels are restructured in an image object hierarchy. The image object levels of the hierarchy are ordered according to inclusion. The image objects of any level are restricted to be completely included (according to their associated image regions) in some image object on any "higher order" image object level. The image object hierarchy together with the image forms the instance cognition network that is generated from the input data.

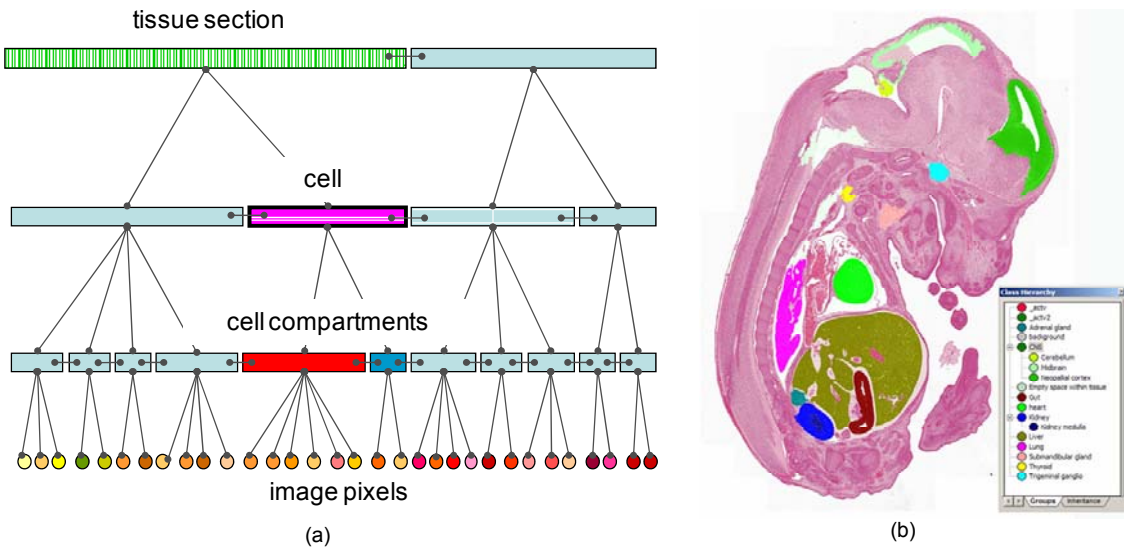


Figure 73 Object-oriented image segmentation. In this example an E14.5 mouse embryo section was processed using a rule set to uniquely identify different embryonic tissues. (a) Example of the image object hierarchies; (b) Result of image processing shown the separation of different tissue regions e.g. heart in red; liver in yellow and kidney in green.

3.10 Colour image segmentation

It has long been recognized that human eyes can discern thousands of colour shades and intensities but only tow-dozen shades of grey. It is quite often when the objects cannot be extracted using gray scale but can be extracted using colour information. Compared to grey scale, colour provides information in addition to intensity. However the literatures on colour image segmentation is not as extensively presented as that on monochrome image segmentation. Most published results of colour image segmentation [33] are based on grey level image segmentation approaches with different colour representations (see **Figure 74**).

In general, there is no standard rule to segment colour images so far.

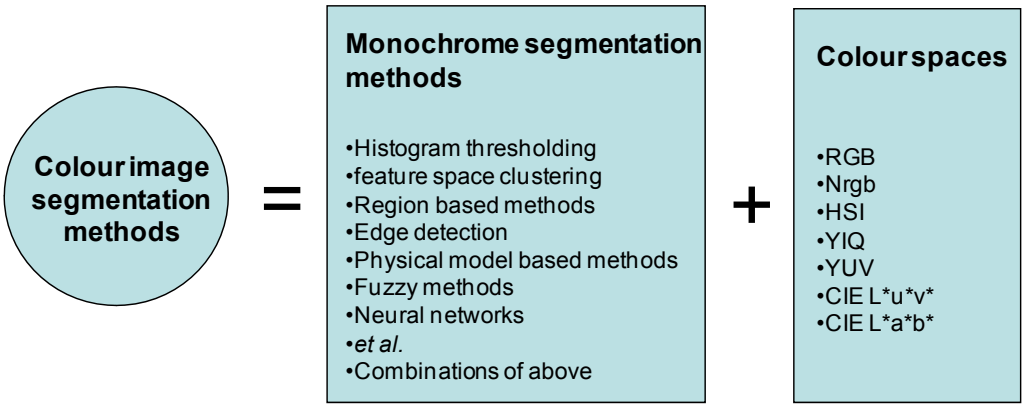


Figure 74 Strategy for colour image segmentation.

Please click the advert

Download free ebooks at bookboon.com

Summary

Image segmentation is an essential preliminary step in most automatic pictorial patter recognition and scene analysis problem. As indicated by the range of examples presented in this chapter, the choice of one segmentation technique over another is dictated mostly by the particular characteristics o the problem being considered. The methods discussed in this chapter, although far from exhaustive, are representative of techniques used commonly in practices.

References and further reading

[22]. R. Adams, L. Bischof, Seeded region growing, IEEE Trans. on PAMI, 16(6): 641-647, 1994.
[23]. P. Biberthaler, M. Athelogou, S. Langer, B. Luchting, R. Leiderer, and K. Messmer, Evaluation of murine liver transmission electron micrographs by an innovative object-based quantitative Image Analysis System (Cellenger), Eur. J. Med. Res 8: 257-282, 2003.
[24]. K. R. Castleman, Digital Image Processing, Prentice Hall, 1996.
[25]. C. H. Chen, L. F. Pau and P. S. P. Wang (editors), Handbook of Pattern Recognition & Computer Vision, World Scientific, 1993.
[26]. R. O. Duda and P. E. Hart, Use of the Hough Transformation to Detect Lines and Curves in Pictures, Comm. ACM, Vol.15, pp.11–15, January, 1972.
[27]. R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd edition, Addison-Wesley Publishing Company, Reading, MA, 2002.
[28]. R. C. Gonzalez, R. E. Woods and S. L. Eddins, Digital Image Processing Using Matlab, Pearson Prentice Hall, New Jersey, 2004.
[29]. R. M. Haralick, Statistical and structural approaches to texture, Proceedings of the IEEE, 67(5): 786-804, 1979.
[30]. N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man and Cybernetics, 9(1): 62-66, 1979.
[31]. T. R. Reed and J. M. Hans du Buf, A review of recent texture segmentation and feature extraction techniques, Computer Vision, Graphics, and Image Processing: Image Understanding, 57(3): 359-372, 1993.
[32]. J. C. Russ, The Image Processing Handbook, 5th edition, CRC Press, 2007.
[33]. W. Skarbek and A. Koschan, Colour image segmentation: a survey, Technischer Bericht 94-32, Technical University of Berlin, 1994.
[34]. P. Soille, Morphological Image Analysis – Principles and Applications, 2nd edition, Springer-Verlag, New York, 2003.
[35]. M. Sonka, and J. M. Fitzpatrick, Handbook of Medical Imaging: Medical Image Processing and Analysis, SPIE Society of Photo-Optical Instrumentation Engineering, 2000.
[36]. All illustrations and examples in this chapter have been programmed by Mathworks Matlab, and the source codes can be obtained on request to author.

Problems

(39) Explain the basis for optimal segmentation using the Otsu method.
(40) Develop a program to implement the Hough transform.
(41) Design an energy term for a snake to track lines of constant grey value.
(42) Illustrate the use of the distance transform and morphological watershed for separating objects that ouch each other.