

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.

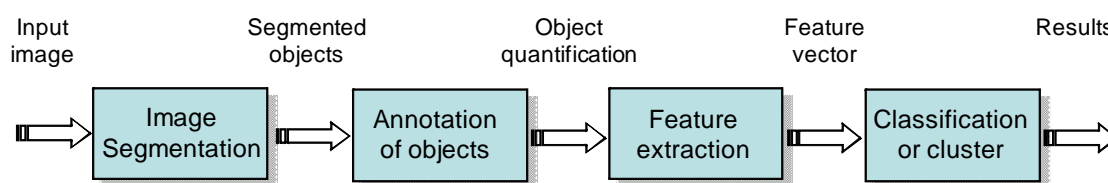


Figure 44 A typical image analysis pipeline.

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**.

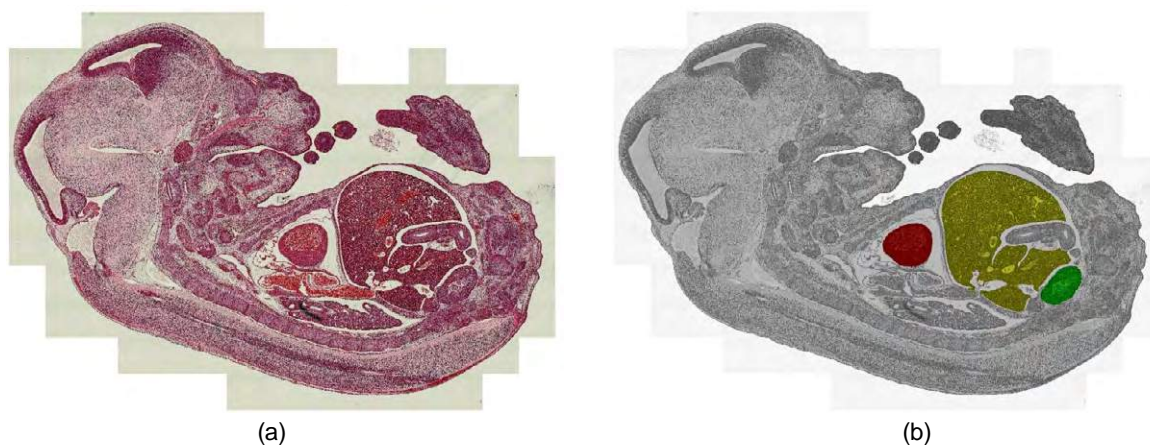


Figure 45 An example of image segmentation. (a) A stained image of a sliced mouse embryo at day 14. (b) Segmented image of (a), where heart, liver and kidney had been identified and marked by colour red, yellow and green.

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.

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.

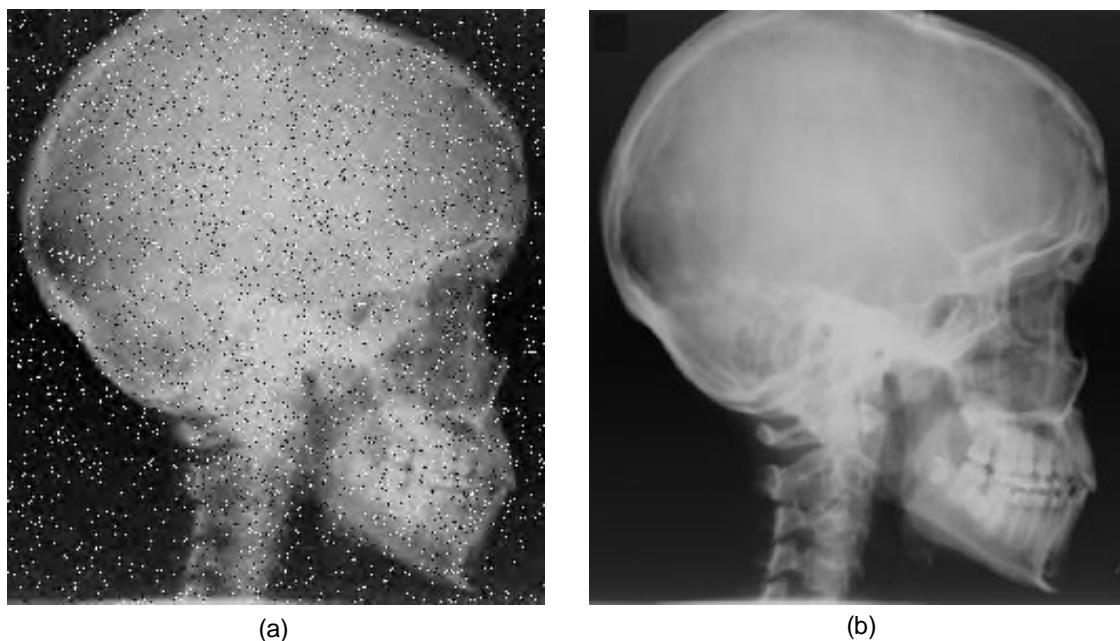


Figure 46 Image smooth by a 3×3 median filter. (a) An X-ray image of skull displaying background noise, so called salt and pepper noise; (b) Median filtered image of (a) showing noise removal. For the purposes of printed display the noise in image (a) has been exaggerated.

3.2.2 Background correction by top-hat filter

The top-hat filter (see details in Chapter 2) can be used to remove background artifacts 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).

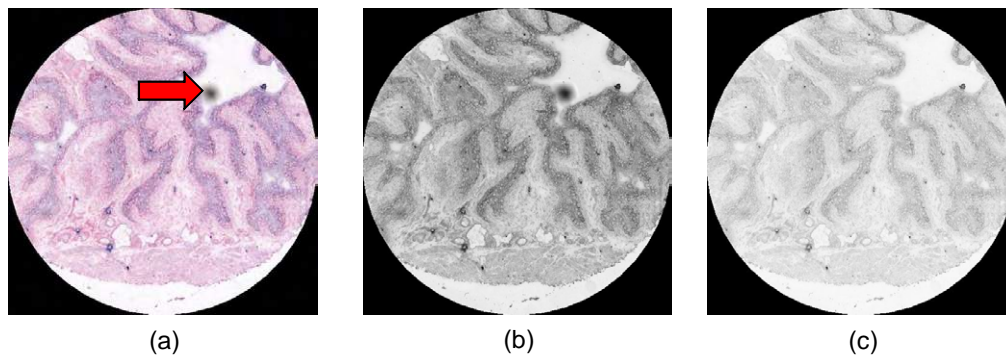


Figure 47 Background correction by top-hat filter. (a) A core image of tissue microarray with background noise, which includes a large spot artefact (red arrow); (b) Image showing colour intensities of the image in (a); (c) Top-hat filtered image showing the removal of the spot artefact.

3.2.3 Illumination correction by low-pass filter

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.

360°
thinking.

Deloitte.

Discover the truth at www.deloitte.ca/careers

© Deloitte & Touche LLP and affiliated entities.



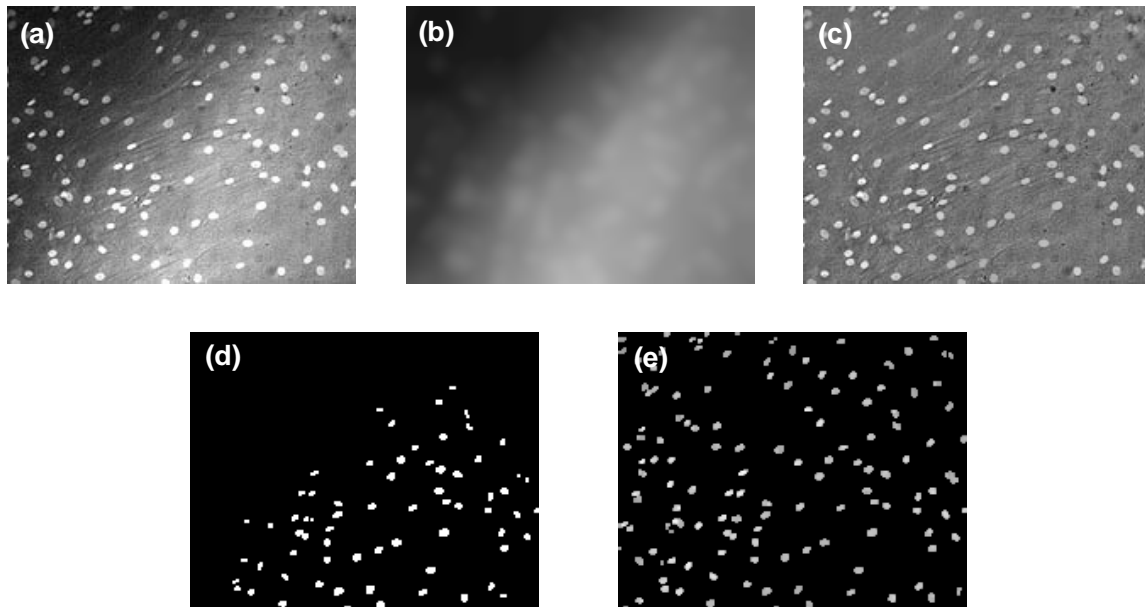


Figure 48 Low-pass filter correction of an image of nuclei with uneven illumination. (a) A field of nuclei with uneven illumination; (b) Estimated background image generated from a low-pass filter on (a); (c) Correction of uneven illumination by subtracting (b) from (a); (d) Segmentation image by thresholding 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.
- 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

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

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

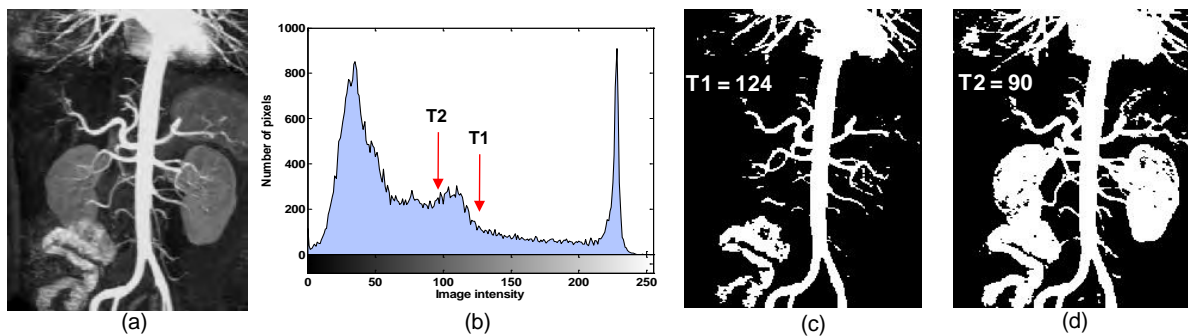


Figure 49 Fundamentals of image histogram thresholding. (a) An MR angiography image showing the aorta and other blood vessels; (b) Intensity histogram of the image in (a); (c) The binary image thresholded with a threshold value $T_1 = 124$, pixels labelled 1 (in white) corresponding to objects; (d) The binary image thresholded with a threshold value $T_2 = 90$.

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_w^2 , the between-class variance δ_B^2 .

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_B^2 / δ_T^2 , with δ_T^2 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:

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

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

with

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

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:

$$t_1 \leftarrow t_1 + e_1 \quad (3.3.4)$$

$$t_2 \leftarrow t_2 + e_2$$

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

SIMPLY CLEVER

ŠKODA



We will turn your CV into
an opportunity of a lifetime

Do you like cars? Would you like to be a part of a successful brand?
We will appreciate and reward both your enthusiasm and talent.
Send us your CV. You will be surprised where it can take you.

Send us your CV on
www.employerforlife.com



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

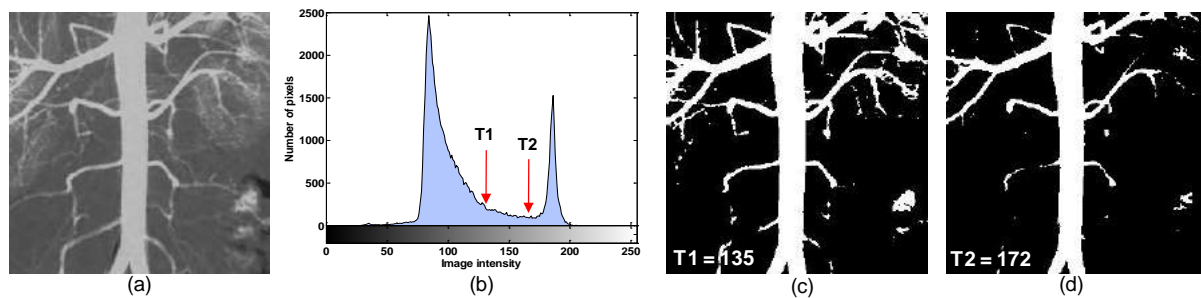


Figure 50 Otsu optimal method vs. conventional thresholding method. (a) An image of blood vessels; (b) Intensity histogram of image in (a); (c) Thresholded image with a threshold value $T1 = 135$ using Otsu optimal method; (d) Thresholded image with a threshold value $T2 = 172$ using the conventional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

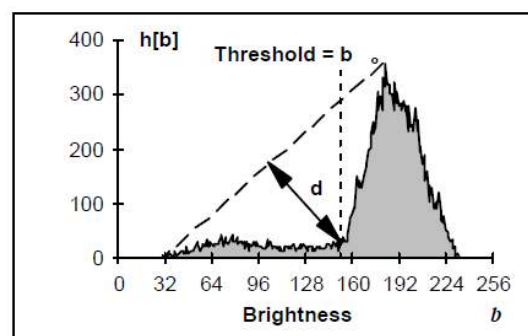


Figure 51 The histogram triangle algorithm is based on finding the value of b that gives the maximum distance d .

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**.

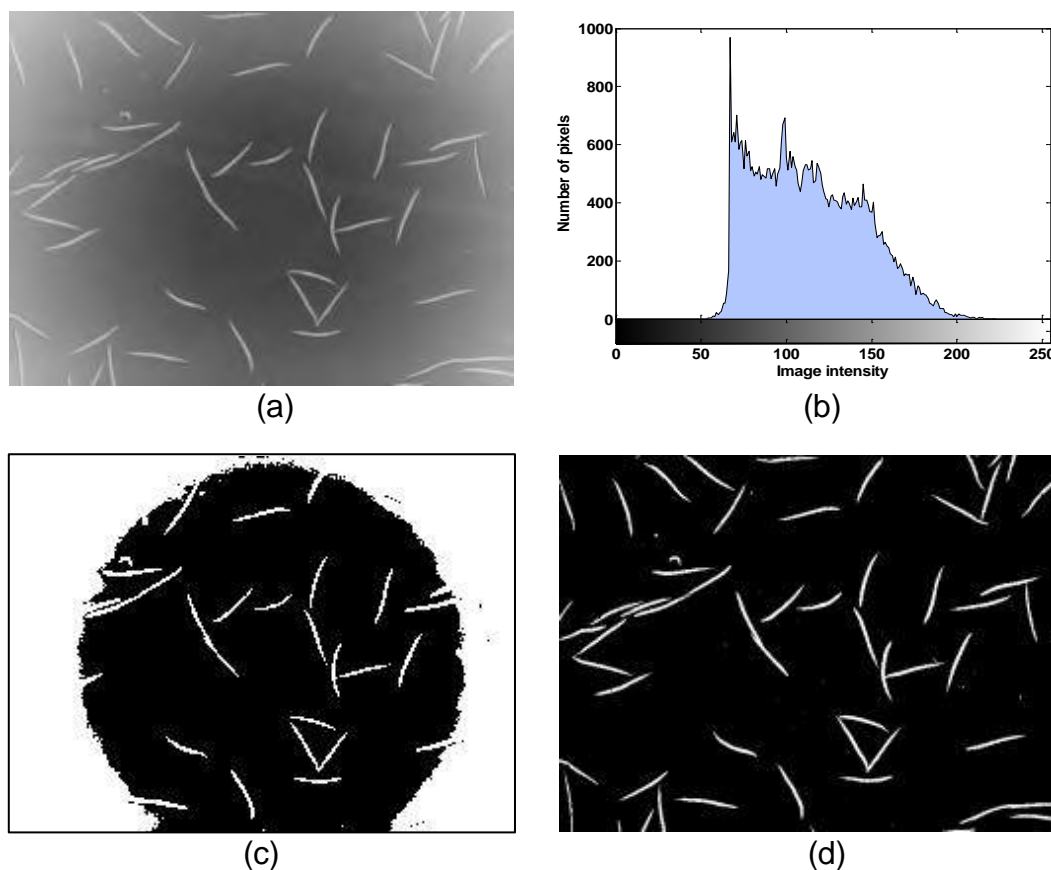


Figure 52 Adaptive local threshold vs. Otsu threshold method. (a) Original image – a microscope image of *C. elegans*, note that it has uneven illumination background; (b) Intensity histogram of image (a); (c) Segmentation image of (a) using Otsu thresholding, where threshold value is 117 and it fails 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 consists 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 objects. The success of thresholding depends critically on the selection of an appropriate threshold.

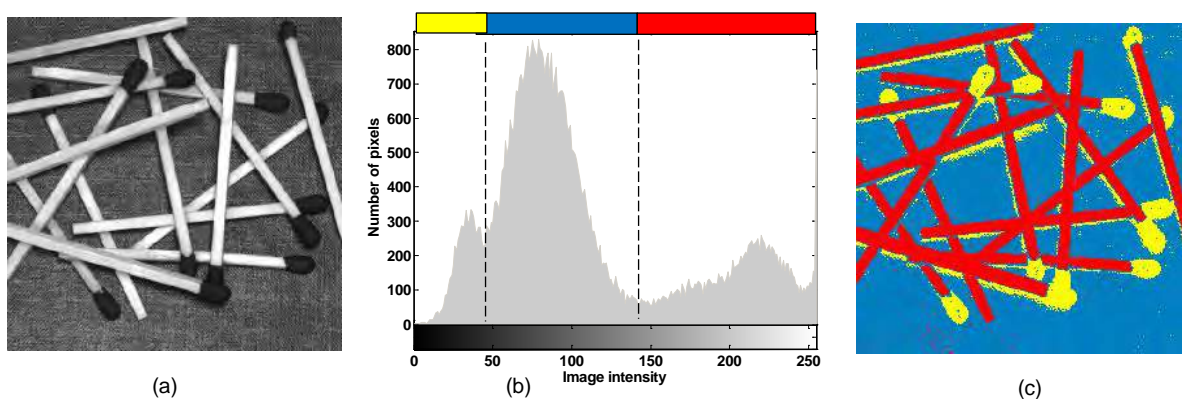


Figure 53 Multiple thresholding. (a) A grey level image of some randomly placed match; (b) Intensity histogram of image in (a); (c) Multiple thresholded image with corresponding threshold value $T_1 = 45$ and $T_2 = 134$.

I joined MITAS because
I wanted **real responsibility**

The Graduate Programme
for Engineers and Geoscientists
www.discovermitas.com

Month 16

I was a construction supervisor in the North Sea advising and helping foremen solve problems

Real work
International opportunities
Three work placements



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 $+45^\circ$; 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, indicating a zero response from the mask in areas of constant intensity.

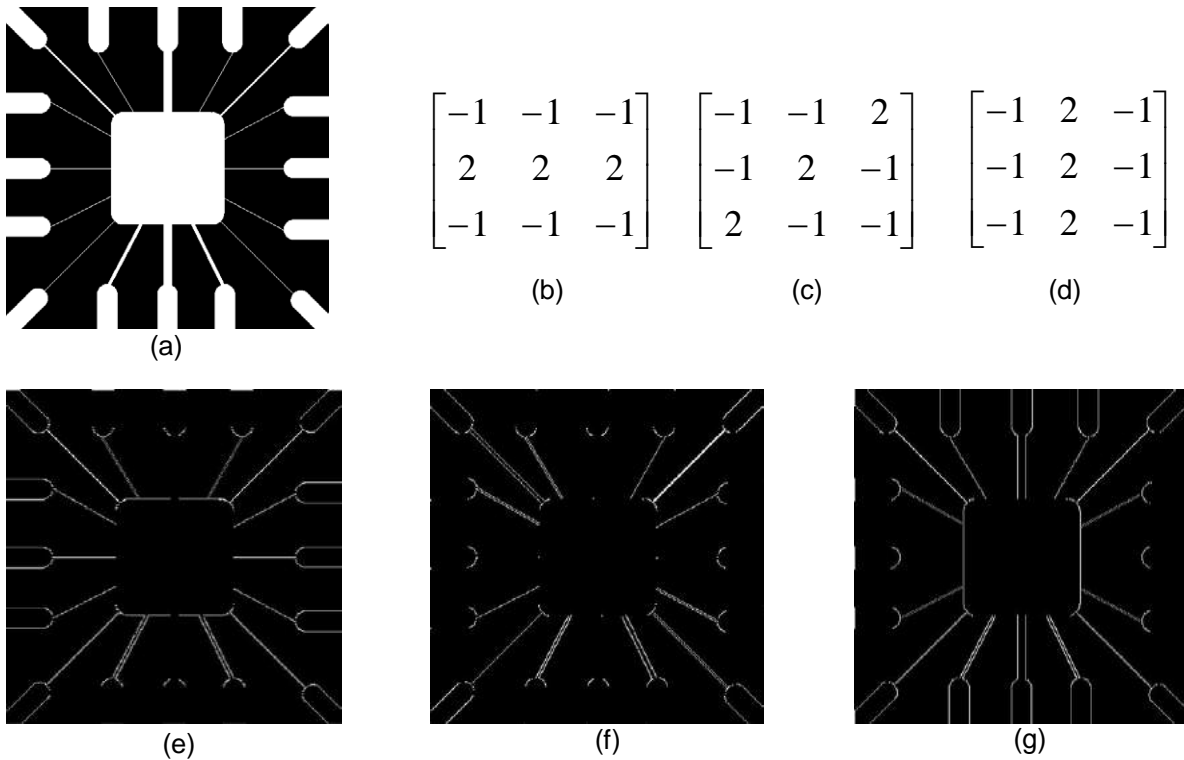


Figure 54 Line detection. (a) Image of a wire-bond mask; (b) Horizontal line detector mask; (c) $+45^\circ$ line detector mask; (d) Vertical line detector mask; (e) Result of processing with the horizontal line detector mask; (f) Result of processing with the $+45^\circ$ line detector mask; (g) Result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

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

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.

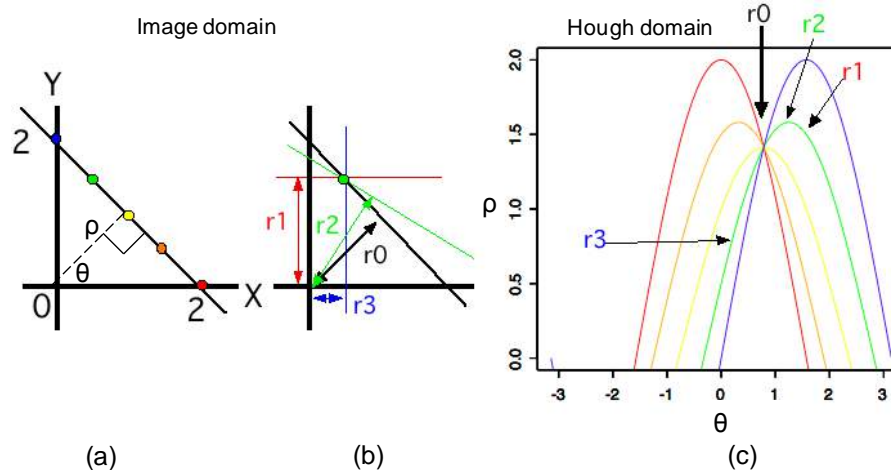


Figure 55 The principle of Hough Transform for line detection. (a). Five points on a straight line in an image domain; (b) To see how to build a sine curve in the Hough domain, we use the green point; For this single point, various lines that pass through it can be drawn (e.g. red, green, black, and 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 r_1, r_2, r_3 , and r_4 are points corresponding to perpendiculars in (b), respectively. The r_0 point is the crossing point of all sine curves, which represents the line in the image domain (a) that lies on all five points.

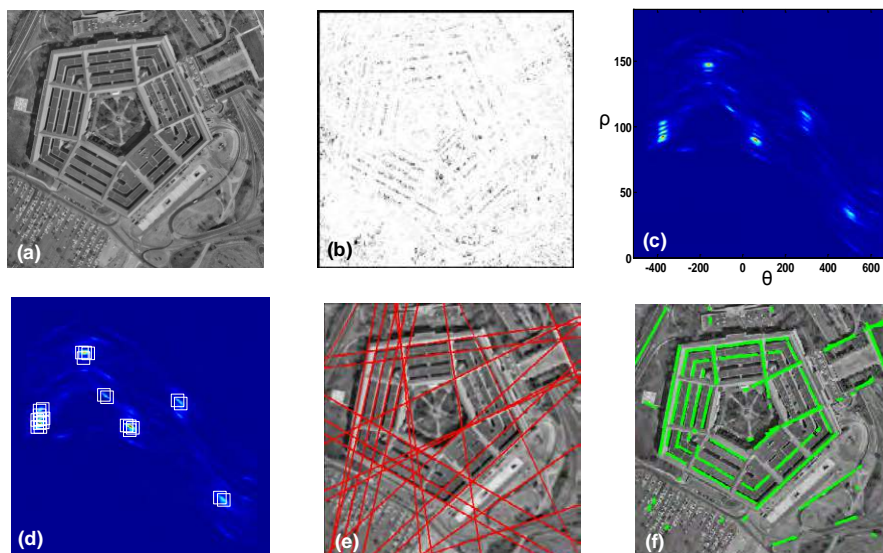


Figure 56 Line detection by Hough Transform. (a) A satellite image of Pentagon; (b) Gradient magnitude image of (a); (c) Hough transformation of the gradient field image (b). Note that a thresholding on the gradient magnitude (b) is performed before the voting process. In other words, pixels with gradient 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.

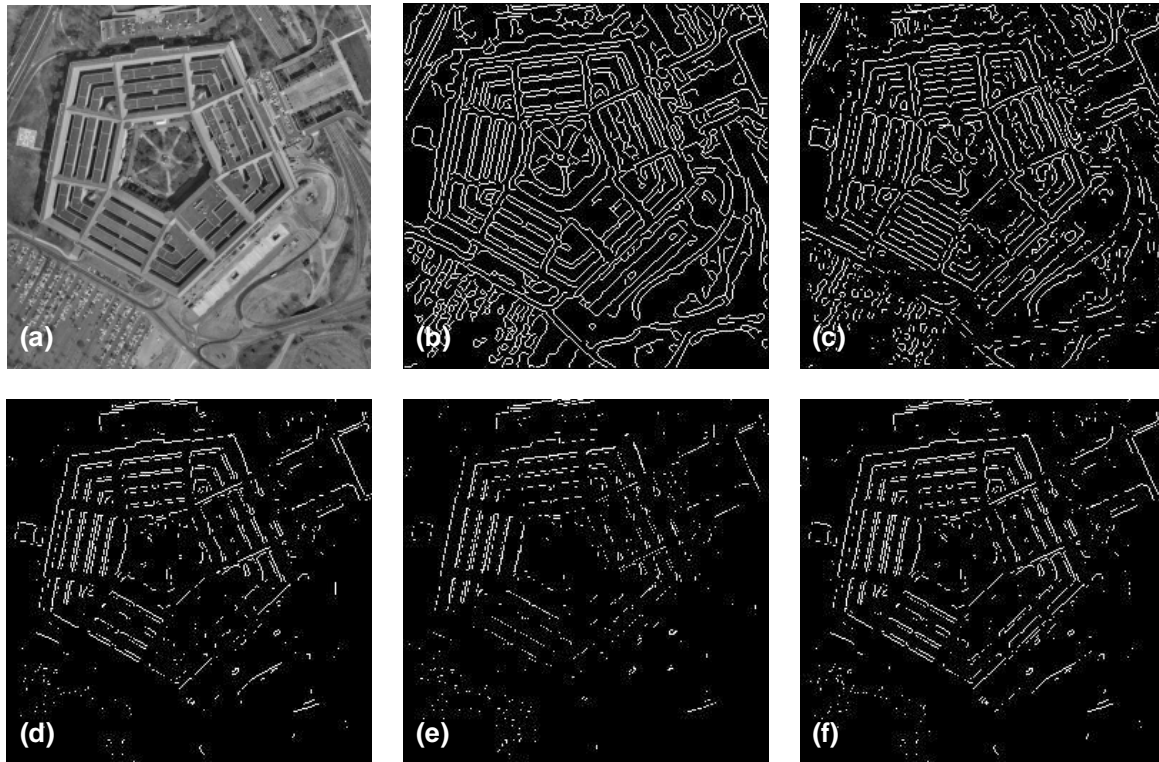


Figure 57 Examples of edge detectors. (a) A satellite image of Pentagon; (b) Canny filter; (c) Laplacian filter; (d) Prewitt filter; (e) Roberts filter; (f) Sobel filter.

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.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.4.2)$$

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, but 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:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.4.3)$$



ie business school

#1 EUROPEAN BUSINESS SCHOOL
FINANCIAL TIMES 2013

#gobeyond

MASTER IN MANAGEMENT

Because achieving your dreams is your greatest challenge. IE Business School's Master in Management taught in English, Spanish or bilingually, trains young high performance professionals at the beginning of their career through an innovative and stimulating program that will help them reach their full potential.

- Choose your area of specialization.
- Customize your master through the different options offered.
- Global Immersion Weeks in locations such as London, Silicon Valley or Shanghai.

Because you change, we change with you.

www.ie.edu/master-management | mim.admissions@ie.edu | f t in YouTube



Roberts filter is specified by:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (3.4.4)$$

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

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

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

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

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 it has two effects: It smooths 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 $\phi(x, y)$, 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:

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

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**).

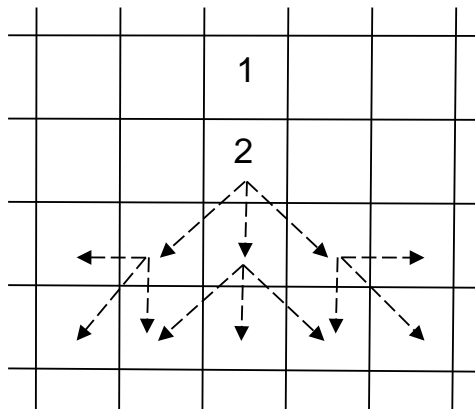


Figure 58 Boundary tracking. In one implementation, find boundary pixel (1); search eight neighbours to find next pixel (2); continues in broadly the same direction as in the previous step, with deviations of one pixel to either side permitted, to accommodate curvature of the boundary, repeat final step 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, a dam, or watershed, is built. These watersheds represent the segmentation of the image. This is illustrated in **Figure 59**.

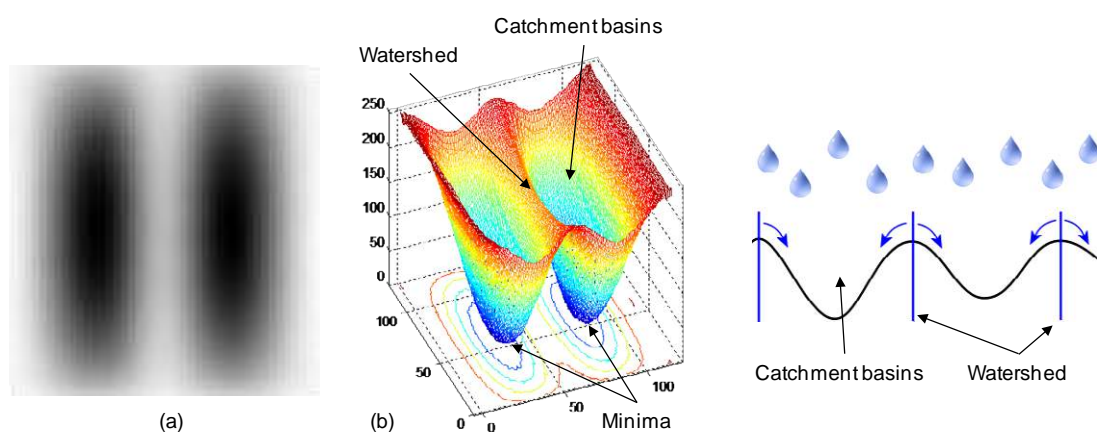


Figure 59 Watershed principle. (a) Synthetically generated grey scale image of two dark blobs; (b) Understanding the watershed transform requires that you think of an image as a topographic surface. If you imagine that bright areas are “high” and dark areas are “low,” then it might look like the surface. 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**.

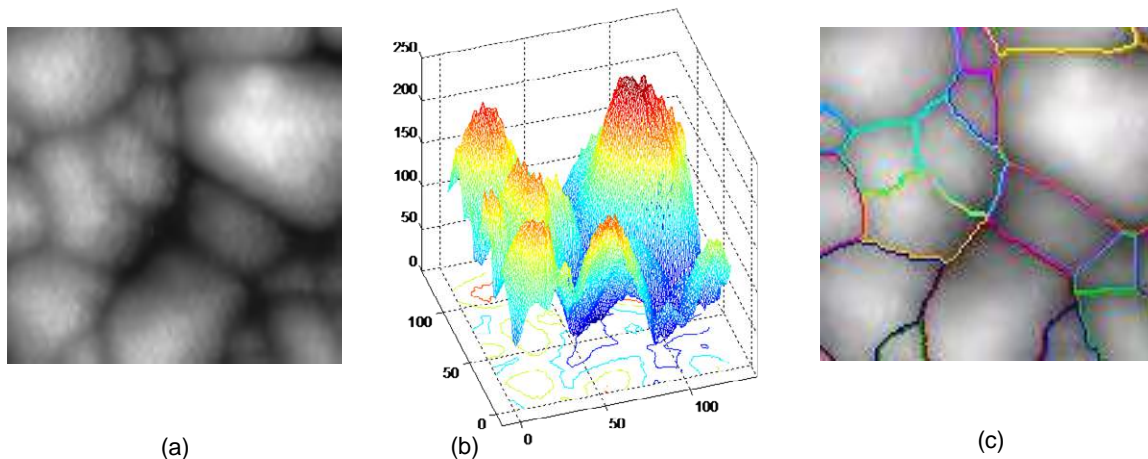


Figure 60 Watershed segmentation. (a) Original gray scale image; (b) Surface representation of the image (a); (c) Watershed segmentation applied and separated feature outlines superimposed on the original image.

"I studied English for 16 years but...
...I finally learned to speak it in just six lessons"

Jane, Chinese architect

ENGLISH OUT THERE

Click to hear me talking before and after my unique course download



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.

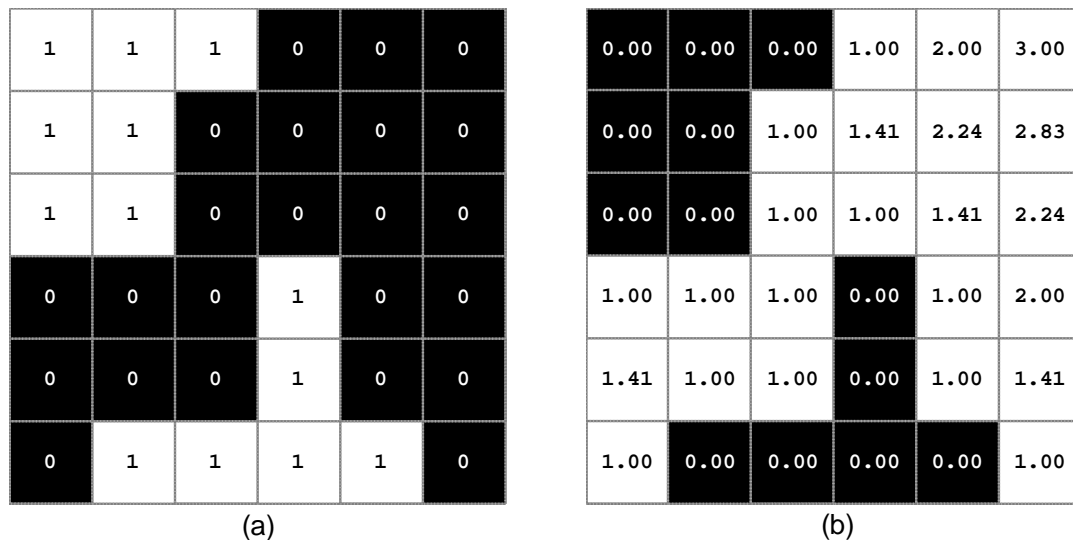


Figure 61 Principle of distance transformation. (a) A binary image matrix; (b) Distance transform of the binary image. Note that 1-valued pixels have a distance transform value of 0.

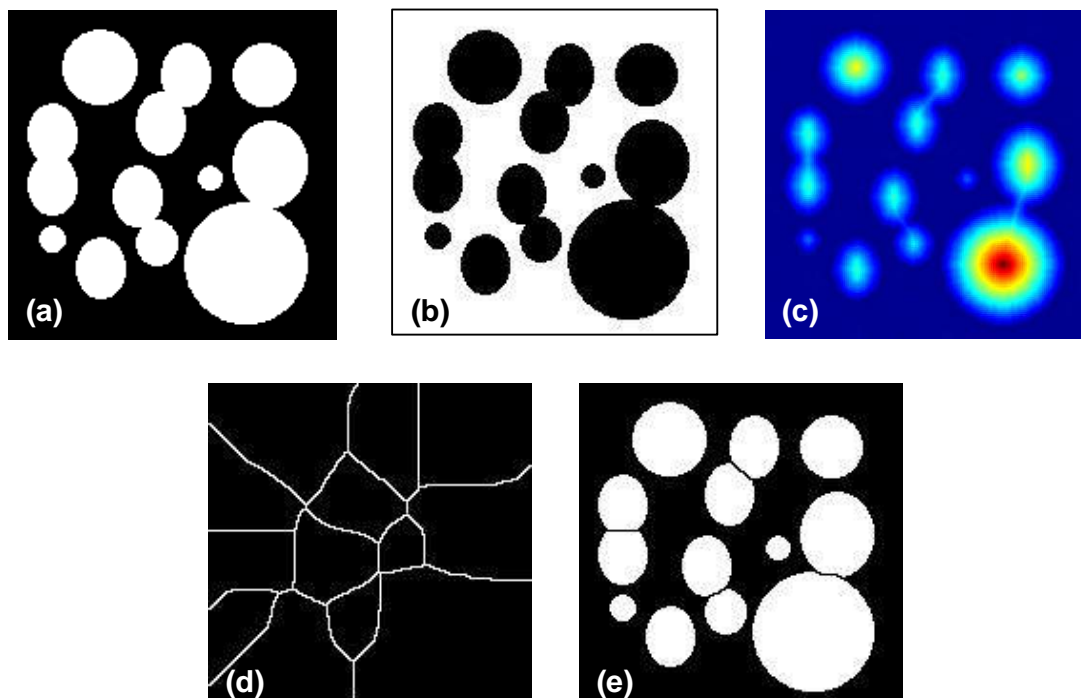


Figure 62 Watershed segmentation using the distance transformation. (a) Original binary image of circular blobs, some of which are touching each other; (b) Complement of image in (a); (c) Distance transform of image in (b); (d) Watershed ridge lines on the negative of the distance transform; (e) Watershed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

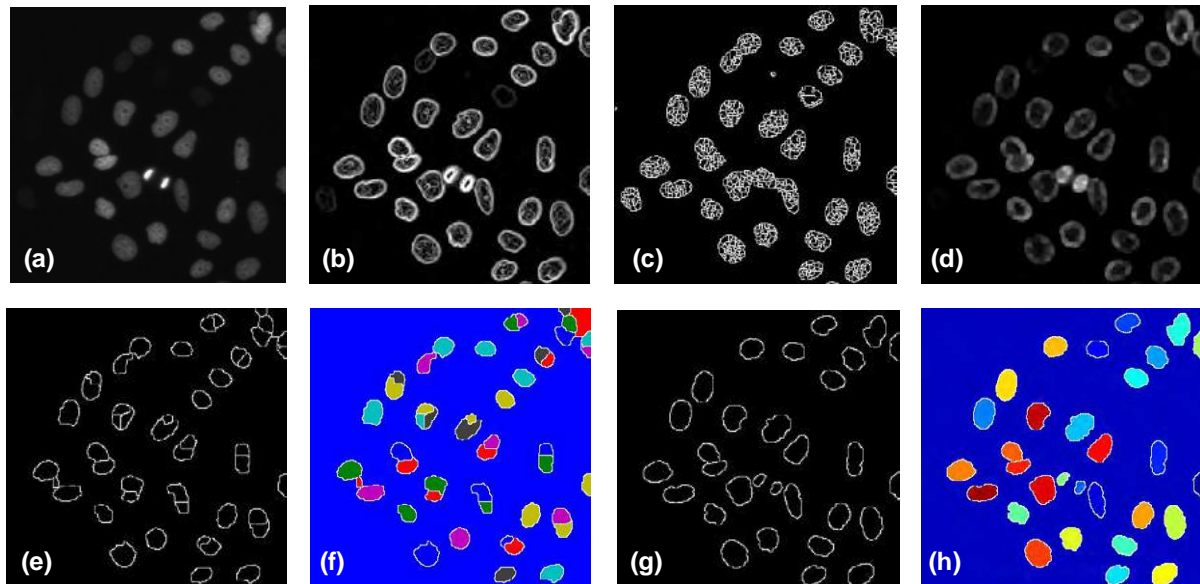


Figure 63 Watershed segmentation using the gradient field. (a) Grey scale image of nuclei from siRNA screening; (b) Gradient magnitude image using Sobel edge filter; (c) Watershed transform of the gradient magnitude image (b), the watershed ridge lines show severe over-segmentation. There are too many 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 practice, 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**).

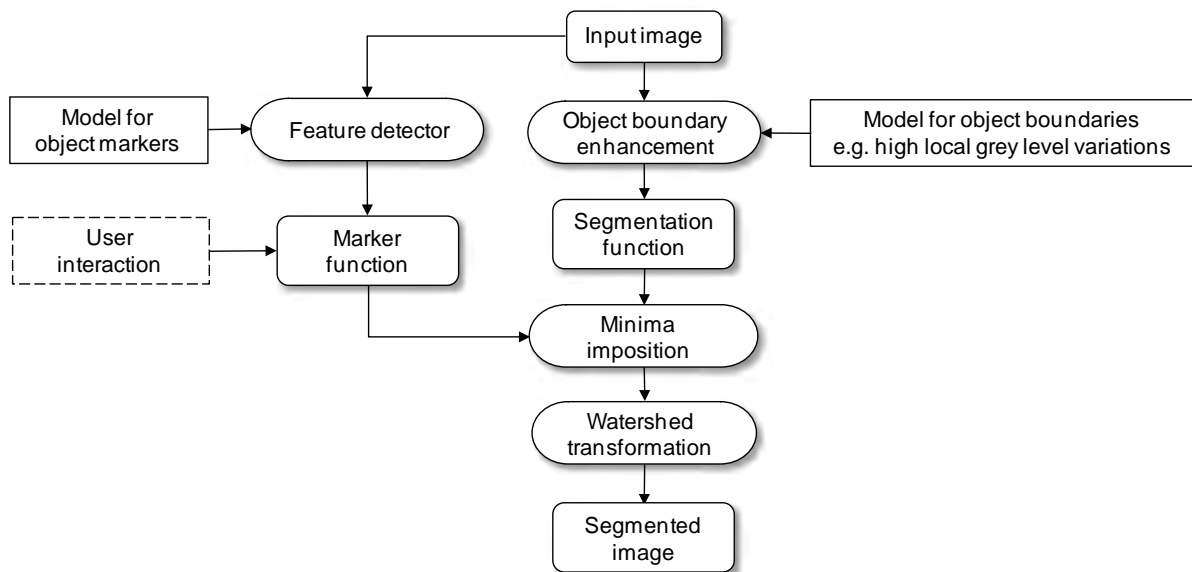


Figure 64 The schematic of marker-controlled watershed segmentation.

Excellent Economics and Business programmes at:



**university of
 groningen**



**“The perfect start
 of a successful,
 international career.”**

CLICK HERE
 to discover why both socially
 and academically the University
 of Groningen is one of the best
 places for a student to be

www.rug.nl/feb/education



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**).

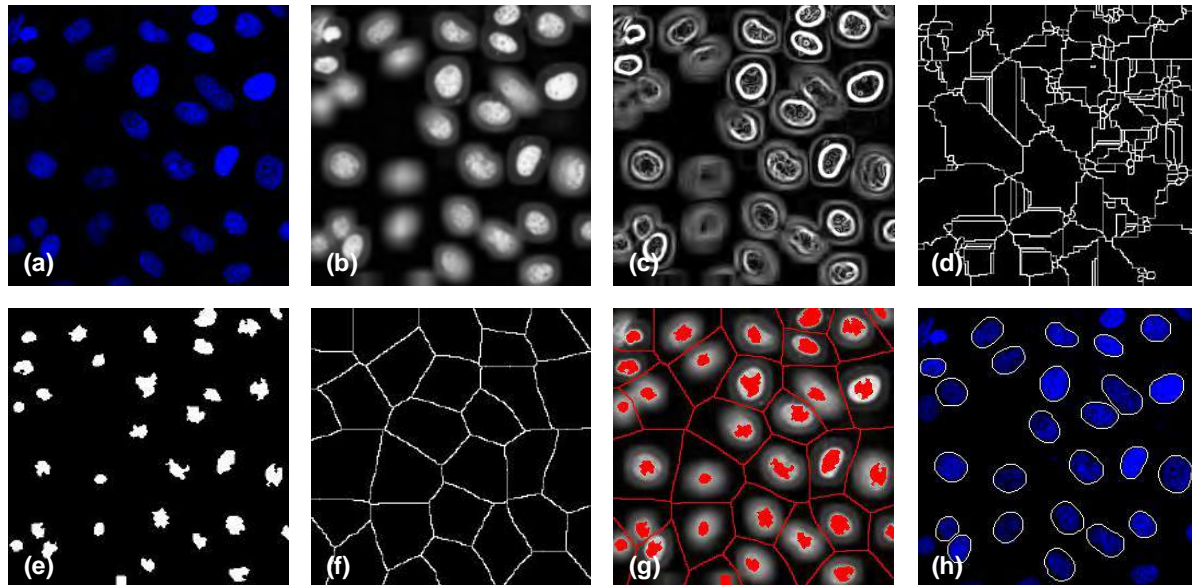


Figure 65 Example of marker-controlled watershed segmentation. (a) An image of nuclei from a siRNA screening; (b) Convert the RGB colour image into a grey level image and also apply a smooth filter in order to remove the noise; (c) Sobel edge detection on image (b); (d) Over-segmentation resulting from 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.

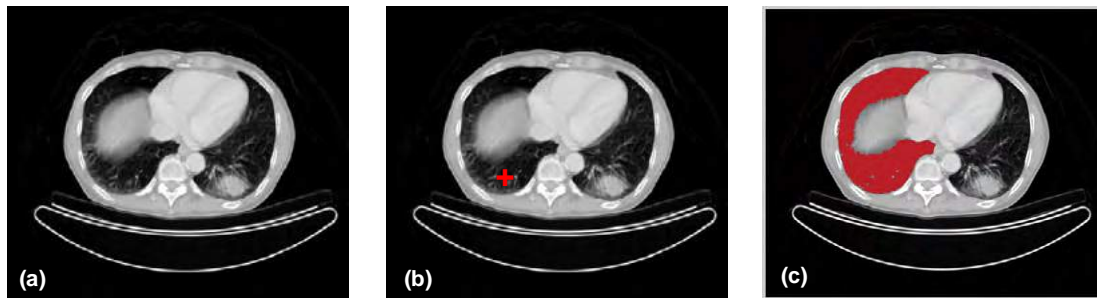


Figure 66 Seeded region growing. (a) Original input CT image; (b) A seed (marked by red “+”) had been selected for a region to be segmented; (c) Segmentation mask by seeded region growing.

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 .
5. 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.
6. 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**).

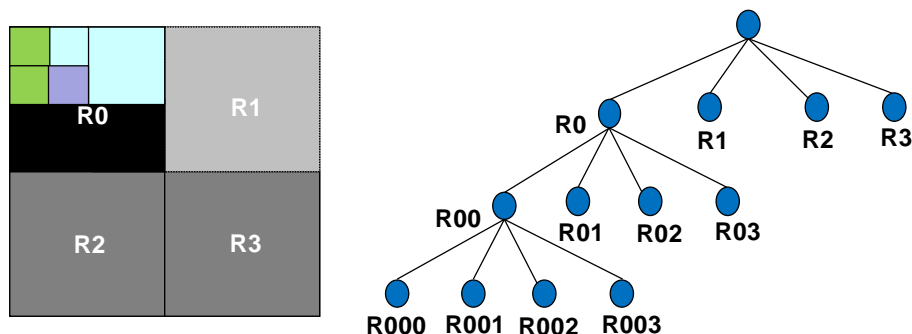


Figure 67 Segmentation Quad-tree.

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]:

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**.

LIGS University

based in Hawaii, USA

is currently enrolling in the
Interactive Online **BBA, MBA, MSc,**
DBA and PhD programs:

- ▶ enroll **by October 31st, 2014** and
- ▶ **save up to 11%** on the tuition!
- ▶ pay in 10 installments / 2 years
- ▶ Interactive **Online education**
- ▶ visit www.ligsuniversity.com to
find out more!

Note: LIGS University is not accredited by any
nationally recognized accrediting agency listed
by the US Secretary of Education.
More info [here](#).



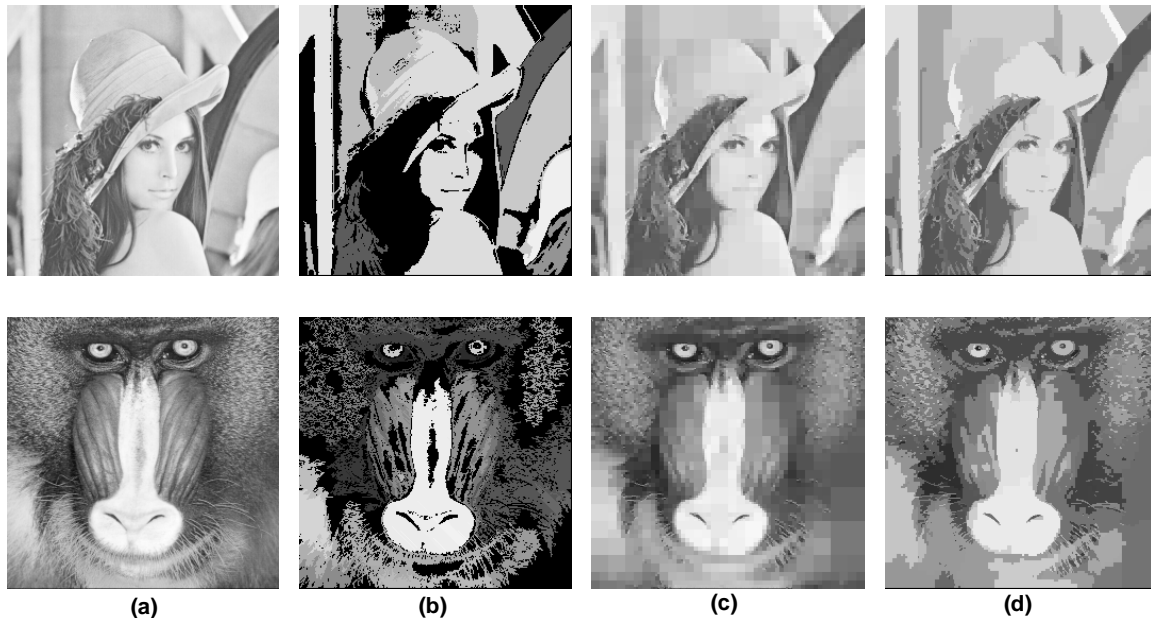


Figure 68 Two examples of region splitting and merging for segmentation. (a) Original input image; (b) Segmentation by region merge only; (c) Segmentation by region spit only; (d) Segmentation by region split and merge.

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**.

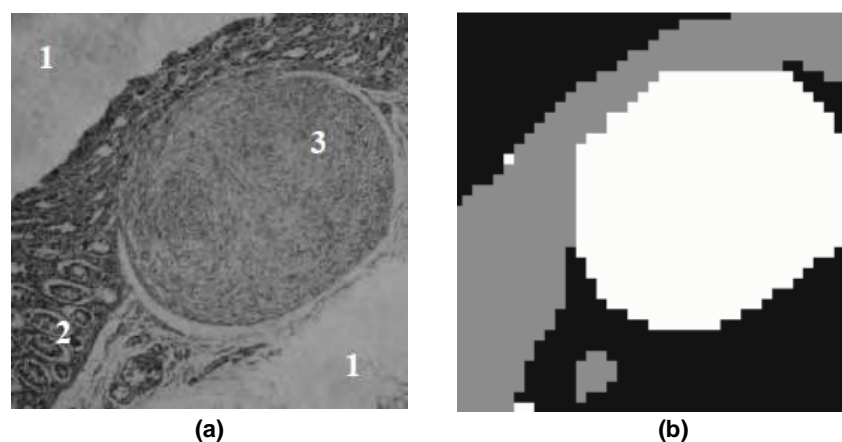


Figure 69 Segmentation of sample microscopic image representing biological tissues by using grey level co-occurrence matrices texture analysis. (a) Source image with three texture classes; (b) image after segmentation.

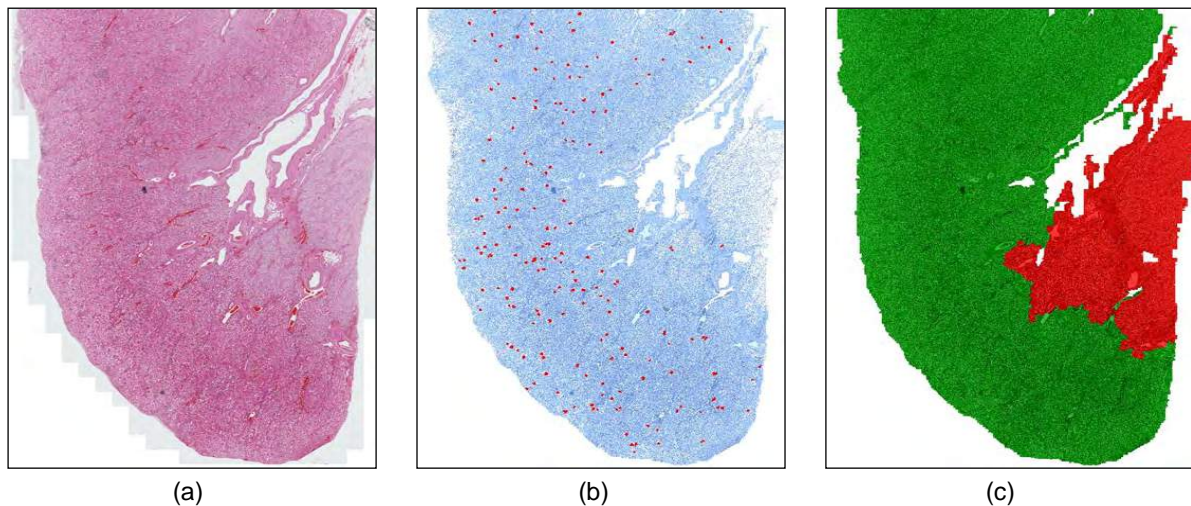


Figure 70 Texture-based image segmentation. (a) Imaging stitched mouse kidney tissue section; (b) Identification of kidney tissue from background (represented by colour blue) and finding glomerulus (represented by colour red); (c) Extract cortex (represented by colour green), and medulla (represented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

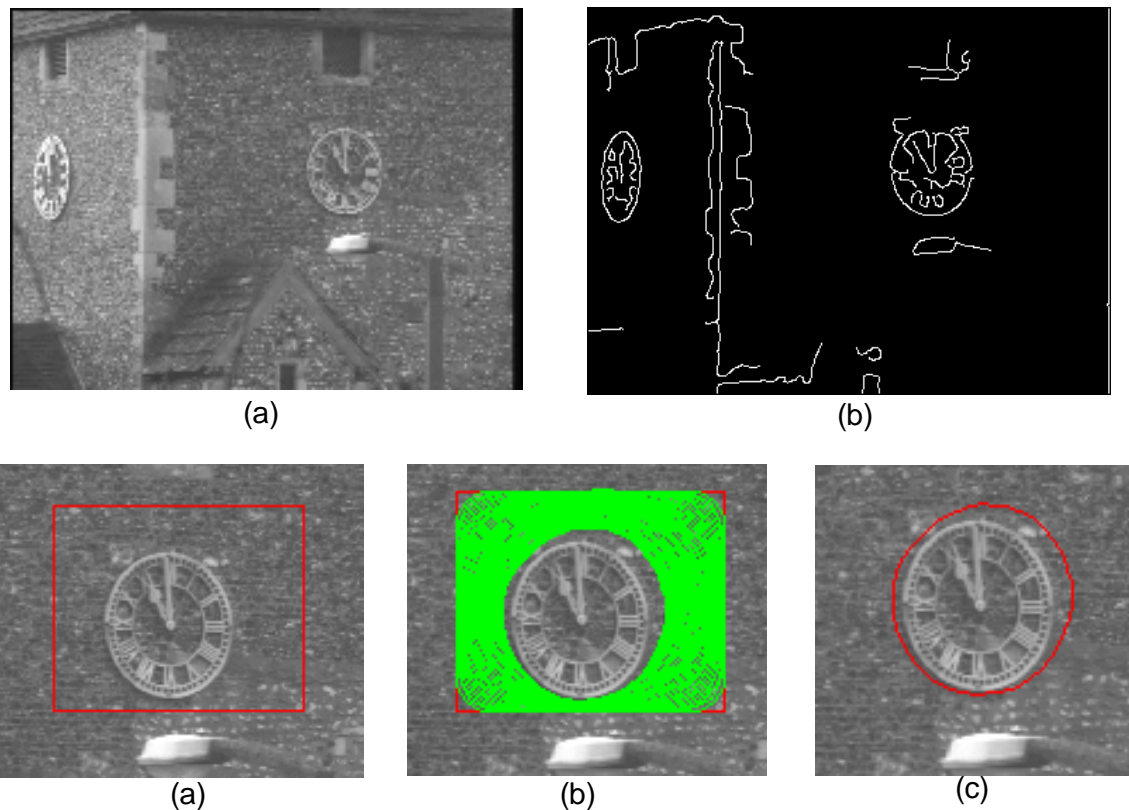


Figure 71 Active contour. (a) Original image of clock faces; (b) Edge detection of image (a) by using Canny filter; (c) To illustrate the active contour (snake), suppose we know that there is a clock face in the rectangular region (in red) of the image; (d) Snake to shrink, to try to form a smooth contour, 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.

$$E_{snake} = E_{internal} + E_{external} \quad (3.1)$$

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.



.....Alcatel-Lucent 

www.alcatel-lucent.com/careers

What if
you could
build your
future and
create the
future?

One generation's transformation is the next's status quo.
In the near future, people may soon think it's strange that
devices ever had to be "plugged in." To obtain that status, there
needs to be "The Shift".



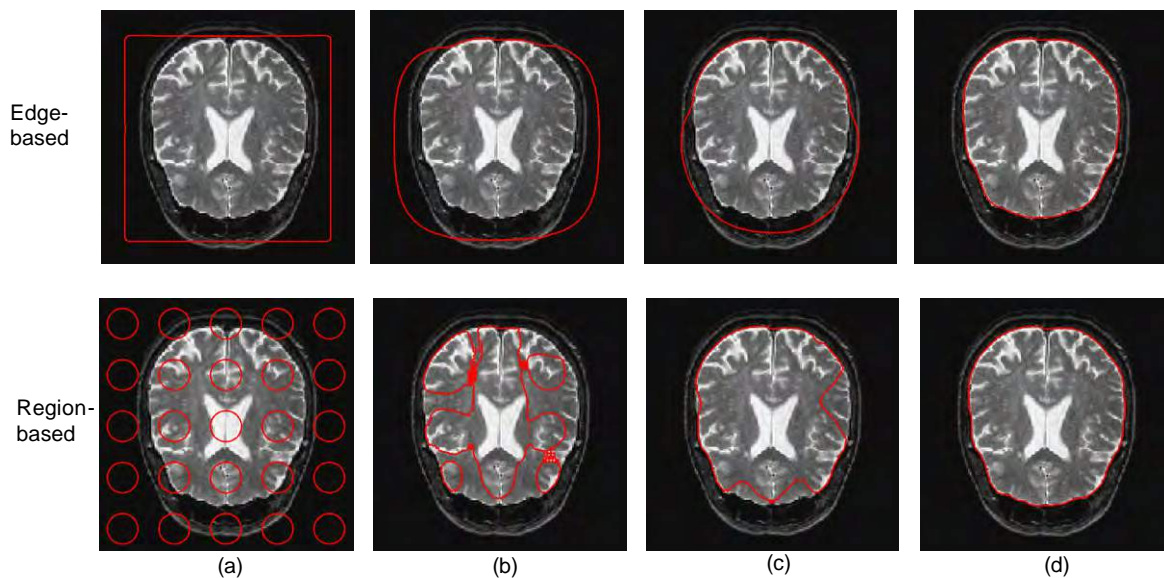


Figure 72 Active contour is semi-automatic since it requires the user to mark an initial contour. The first row is edge-based active contour (snake + level set), and the second row is region-base active contour. (a) Initial contour by user input; (b) and (c) Intermediate contour; (d) Final contour.

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.

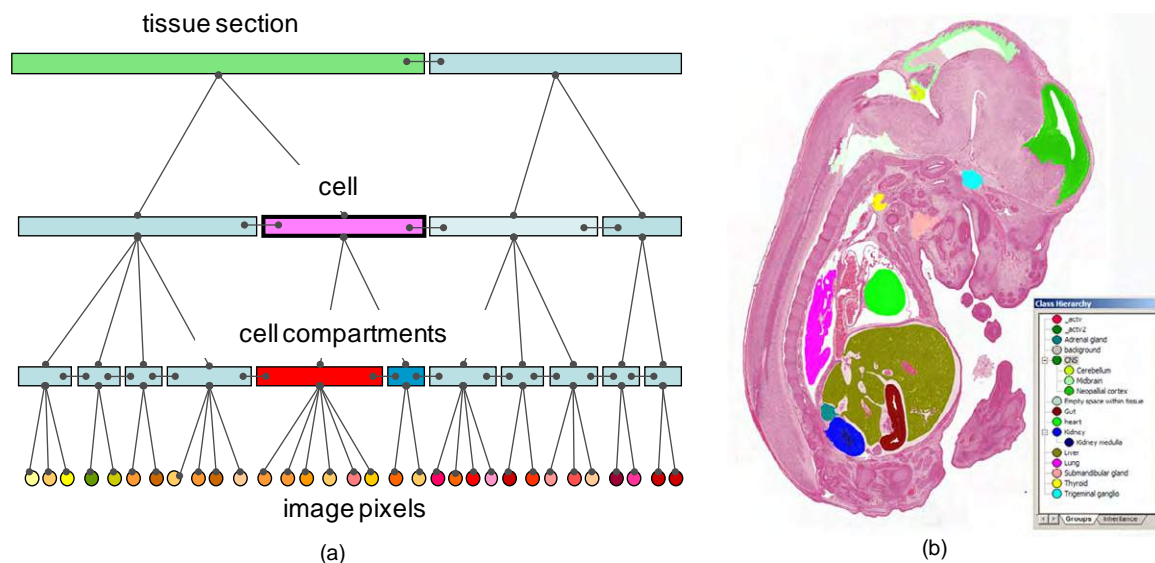


Figure 73 Object-oriented image segmentation. In this example an E14.5 mouse embryo section was processed using a rule set to uniquely identify different embryonic tissues. (a) Example of the image object hierarchies; (b) Result of image processing shown the separation of different tissue regions e.g.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

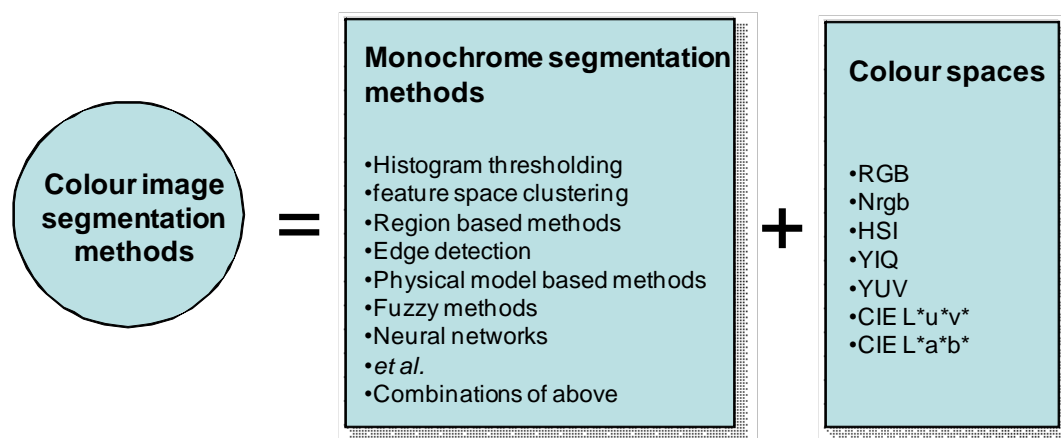


Figure 74 Strategy for colour image segmentation.

3.11 Summary

Image segmentation is an essential preliminary step in most automatic pictorial pattern 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 of the problem being considered. The methods discussed in this chapter, although far from exhaustive, are representative of techniques used commonly in practice.

3.12 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. Athellogou, 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.



Maastricht University
Leading in Learning!

**Join the best at
the Maastricht University
School of Business and
Economics!**



Top master's programmes

- 33rd place Financial Times worldwide ranking: MSc International Business
- 1st place: MSc International Business
- 1st place: MSc Financial Economics
- 2nd place: MSc Management of Learning
- 2nd place: MSc Economics
- 2nd place: MSc Econometrics and Operations Research
- 2nd place: MSc Global Supply Chain Management and Change

Sources: Keuzegids Master ranking 2013; Elsevier 'Beste Studies' ranking 2012; Financial Times Global Masters in Management ranking 2012

Visit us and find out why we are the best!
Master's Open Day: 22 February 2014

**Maastricht
University is
the best specialist
university in the
Netherlands
(Elsevier)**

www.mastersopenday.nl



- [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.

3.12 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 touch each other.