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*:

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

It follows that the threshold operator maps any gray-tone image i3nto 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

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

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

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

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∪B, for example, is equivalent to the "OR" operation for binary image; and the "*intersection*" operator, A∪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.

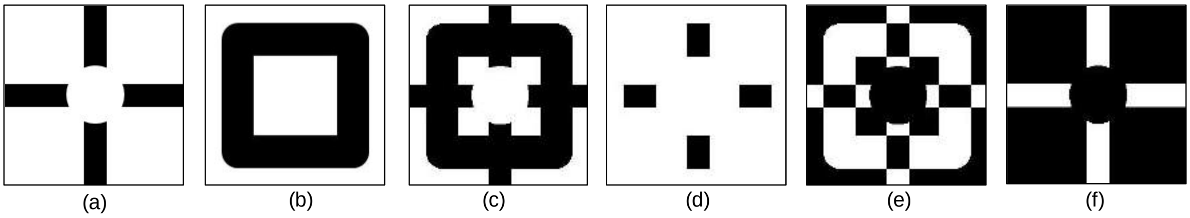


Figure 15 Basic Boolean logical operators. (a) Binary image A; (b) Binary image B; (c) A AND B; (d) A OR B; (e) A XOR B; (f) NOT A.

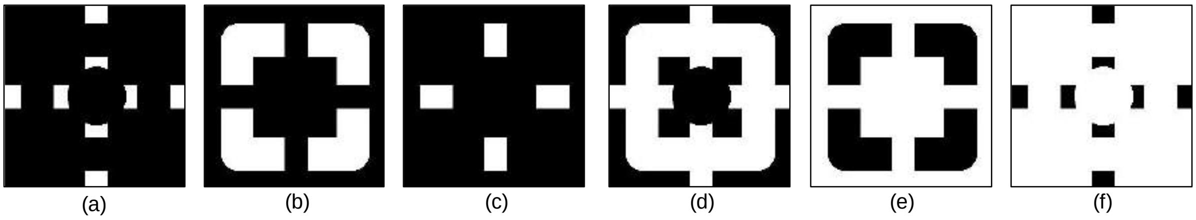


Figure 16 Combined Boolean logical operators. (a) (NOT A) AND B; (b) A AND (NOT B); (c) (NOT A) AND (NOT B); (d) NOT (A AND B); (e) (NOT A) OR B; (f) A OR (NOT B).

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.
- **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.

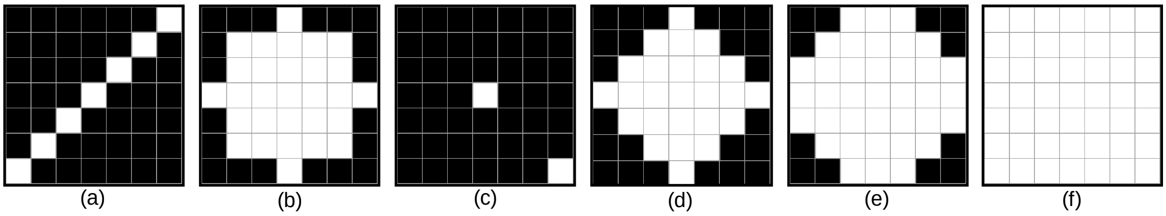


Figure 17 Some typical structure elements. (a) A line segment SE with the length 7 and the angle 45°; (b) A disk SE with the radius 3; (c) A pair of points SE containing two points with the offset 3; (d) A diamond-shaped SE; (e) A octagonal SE; (f) A 7×7 square S.

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.

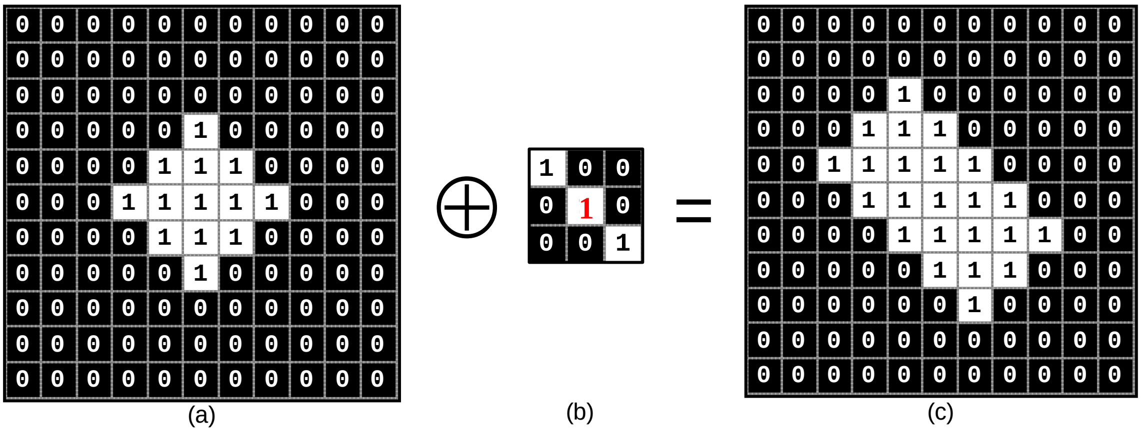


Figure 18 Illustration of morphological dilation. (a) Original binary image with a diamond object; (b) Structure element with three pixels arranged in a diagonal line at angle of 135°, the origin of the structure element is clearly identified by a red 1; (c) Dilated image, 1 at each location of the origin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

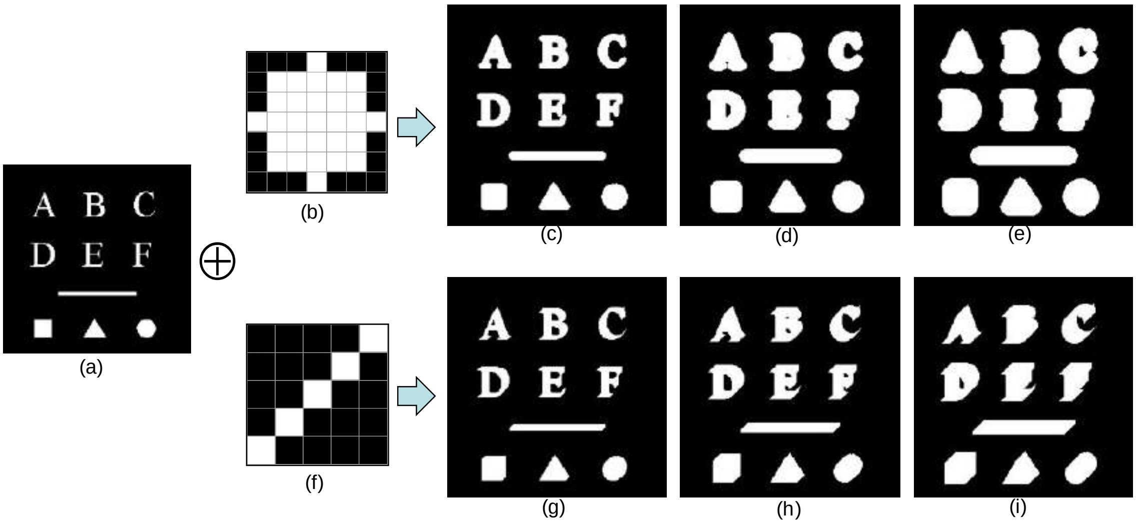


Figure 19 Example of morphological dilation. (a) A binary input image; (b) A disk structure element; (c) Dilated image of (a) by SE (b); (d) After twice dilation by SE (b); (e) After three times dilation by SE (b); (f) A line structure element; (g) Dilated image of (a) by SE (f); (h) After twice dilation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

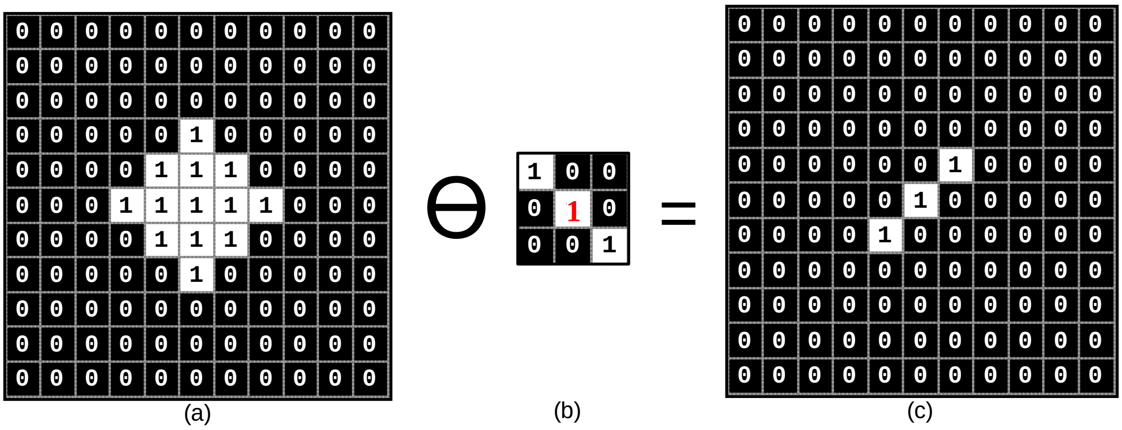


Figure 20 Illustration of morphological erosion. (a) Original binary image with a diamond object; (b) Structure element with three pixels arranged in a diagonal line at angle of 1350, the origin of the structure element is clearly identified by a red 1; (c) Eroded image, a value of 1 at each location 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).

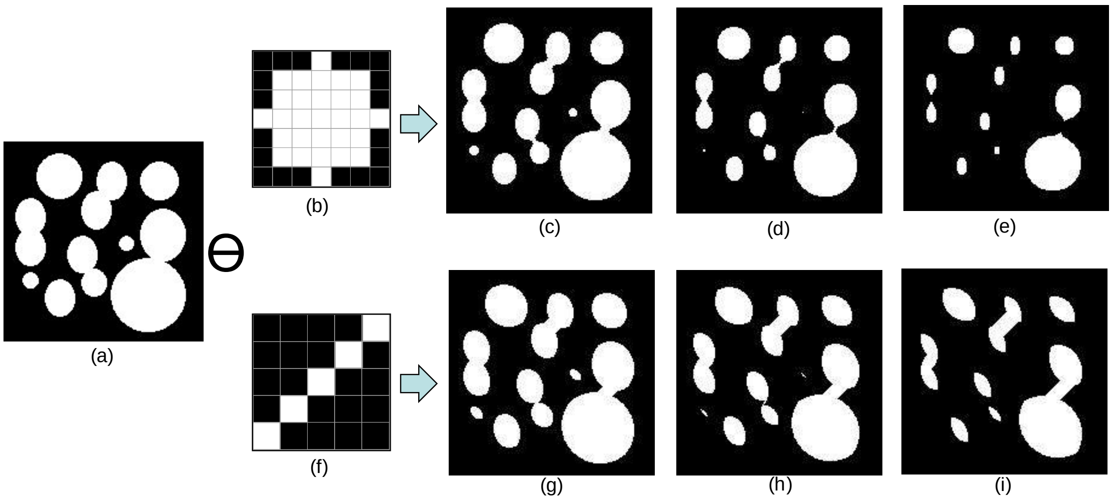


Figure 21 Example of morphological erosion. (a) A binary input image; (b) A disk structure element; (c) Eroded image of (a) by SE (b); (d) After twice erosion by SE (b); (e) After three times erosion by SE (b); (f) A line structure element; (g) Eroded image of (a) by SE (f); (h) After twice erosion by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

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

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

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

The dilation distributes the point-wise maximum operator ⊕ 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.

- ***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 $A\Omega B$:

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

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**.

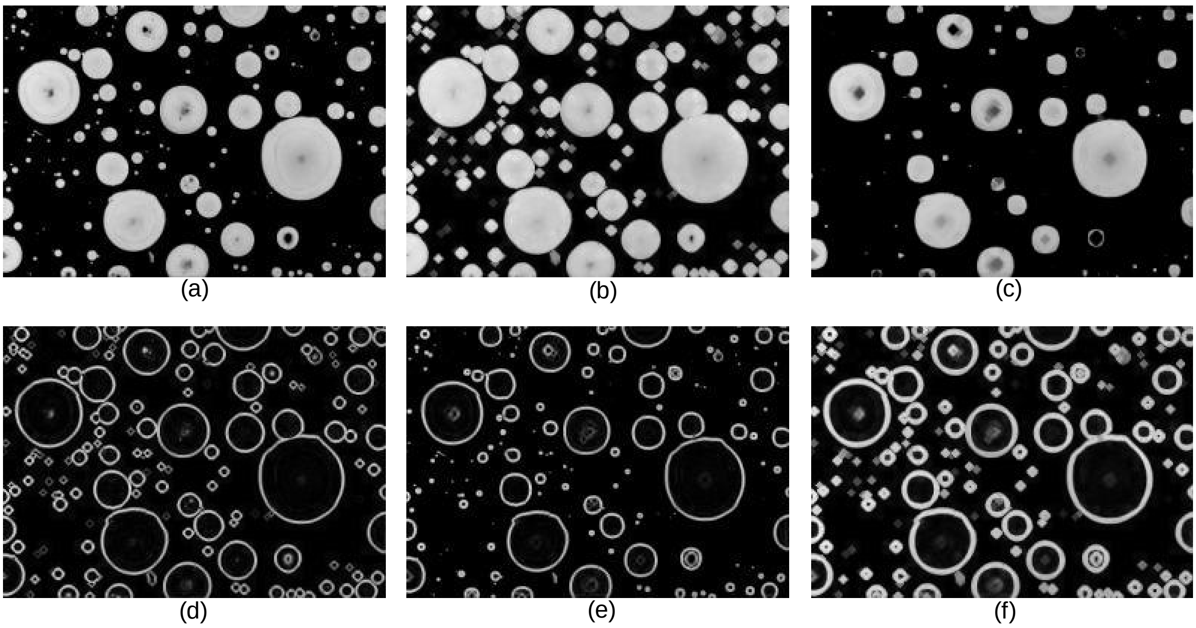


Figure 22 Morphological gradients to enhance the object boundaries. (a) Original image A of enamel particles; (b) Dilated image A by B: A B, note that structure element B is a 5×5 disk; (c) Eroded image A by B: $A\Theta B$; (d) External gradient $A\Omega+B$ = (A B)-A; (e) Internal gradient $A\Omega^{-}B = A-(A\Theta B)$; **(f)** Morphological gradient $A\Omega B = (A\oplus B)-(A\Theta B)$.

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: $A\text{o}B$ 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**.

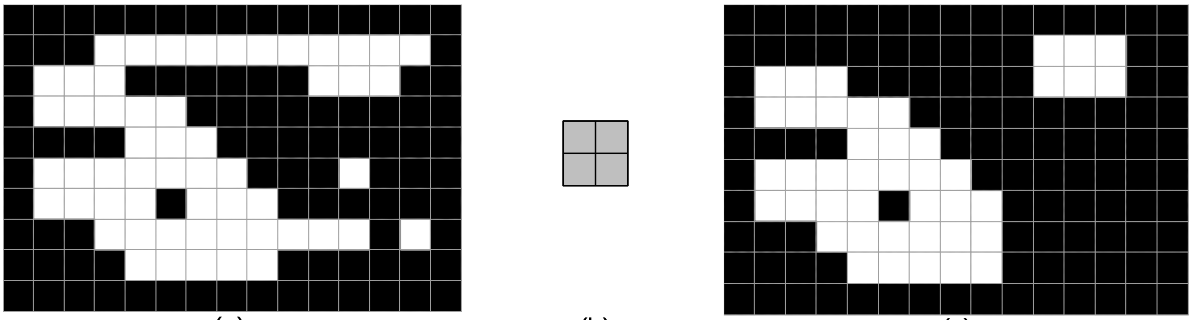


Figure 23 Illustration of opening. (a) A 10×15 discrete binary image (the object pixels are the white pixels); (b) A 2×2 structure element; (c) Opening of image (a) by SE (b). All object pixels that cannot be covered by the structure element when it fits the object pixels are removed.

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.

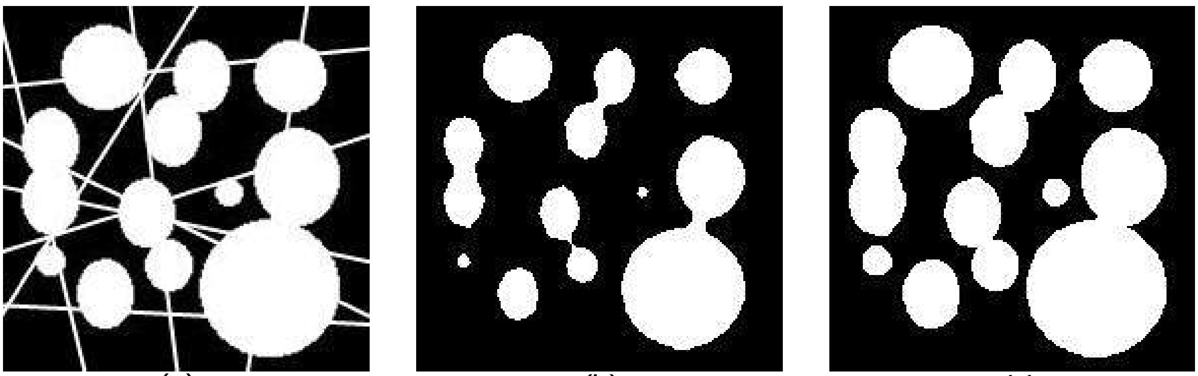


Figure 24 Shape matching by opening. (a) An original binary image A with some disks and lines; (b) Eroded image $A\Theta B$ by a disk structure element B, where the radius of the disk is 5 pixels; (c) Dilated image of (b) by the same disk structure element B: $(A\Theta B)\oplus B$.

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**.

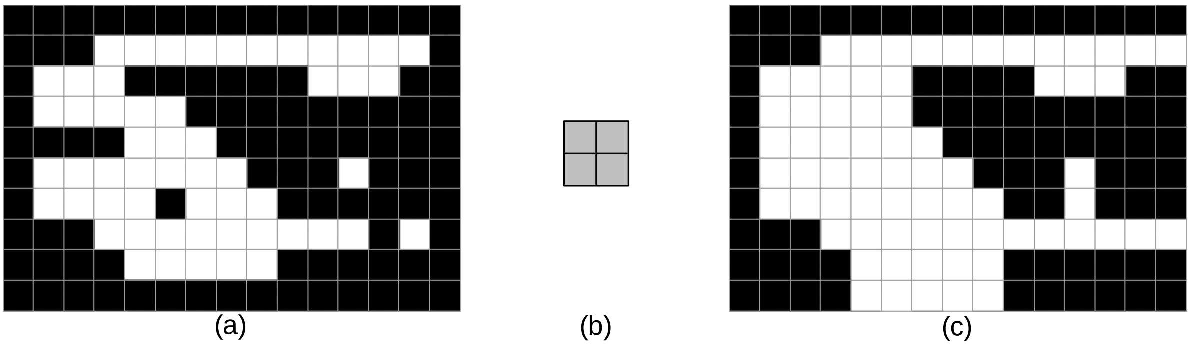


Figure 25 Illustration of closing. (a) A 10×15 discrete binary image (the object pixels are the white pixels); (b) A 2×2 structure element; (c) Closing of image (a) by SE (b). All background pixels that cannot be covered by the structure element when it fits the background are added to the object pixels.

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.

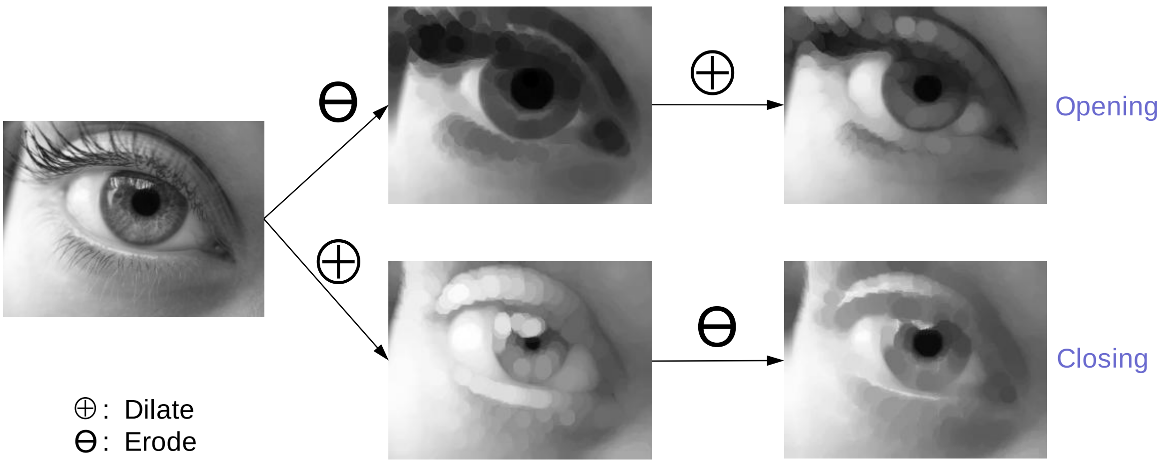


Figure 26 Opening and closing of a gray-scale image with an 8×8 disk SE.

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**.

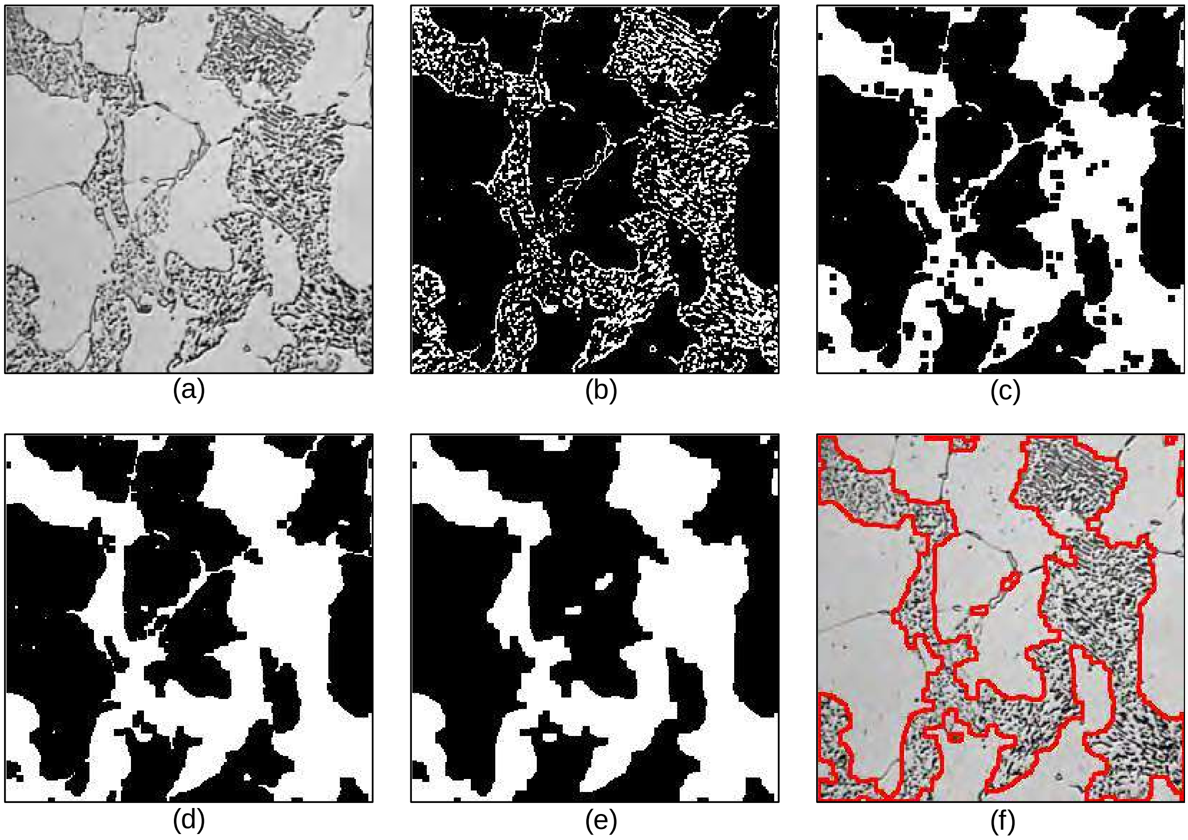


Figure 27 An example of image closing and opening. (a) Original gray level image of chemically etched metallographic specimen: dark regions are iron carbide (image courtesy to J.C. Russ); (b) Intensity histogram threshold applied to image (a); (c) Closing image (b) by a disk structure element with radius 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**.

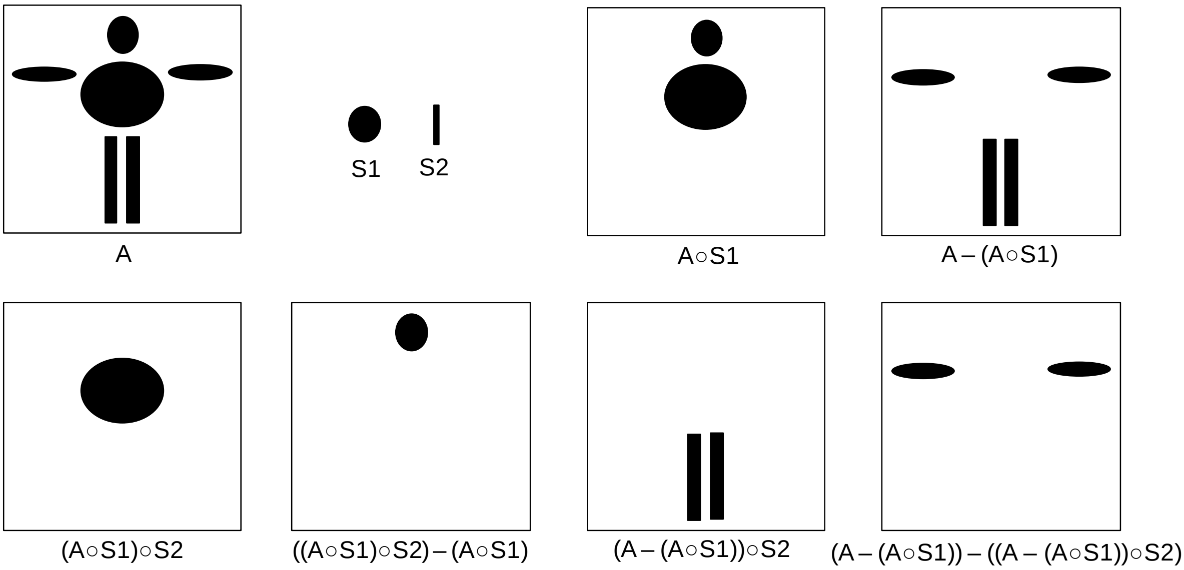


Figure 28 A series of opening and image subtractions in order to decompose an object into its constituent parts.

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.

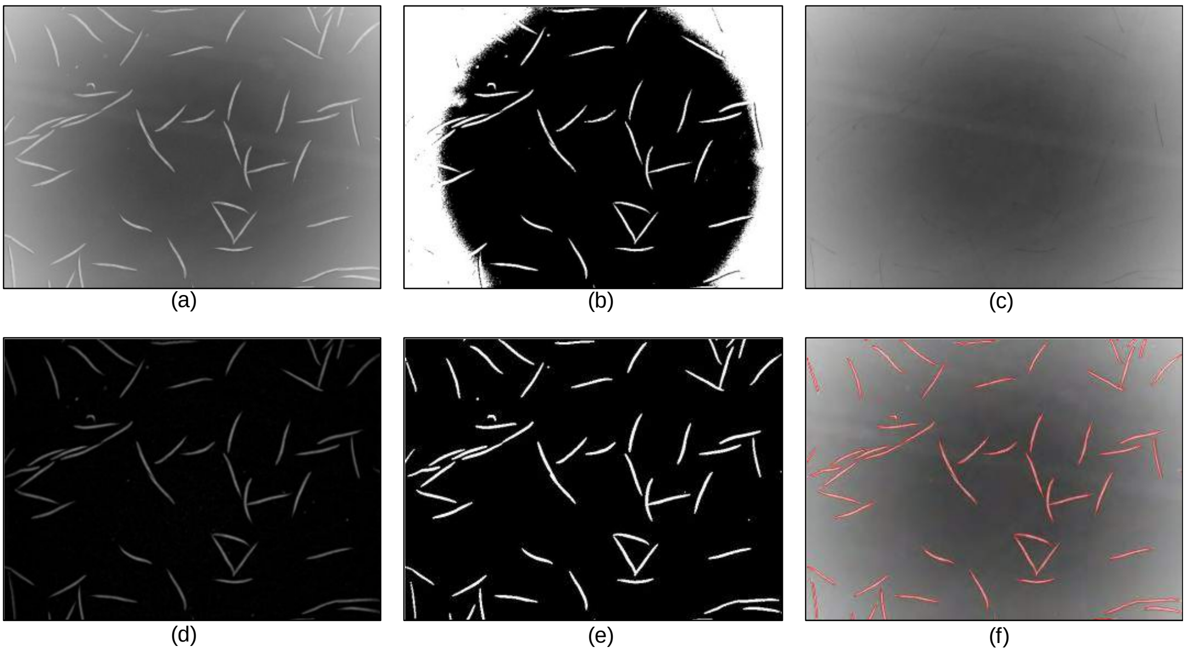


Figure 29 Top-hat transformation. (a) A gray level image of C elegant, where there is non-uniform illumination background; (b) Intensity threshold image of (a), where the object has been over-segmented; (c) Opening image of (a) by a large disk structure element with radius 5; (d) Top-hat transformation 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**.

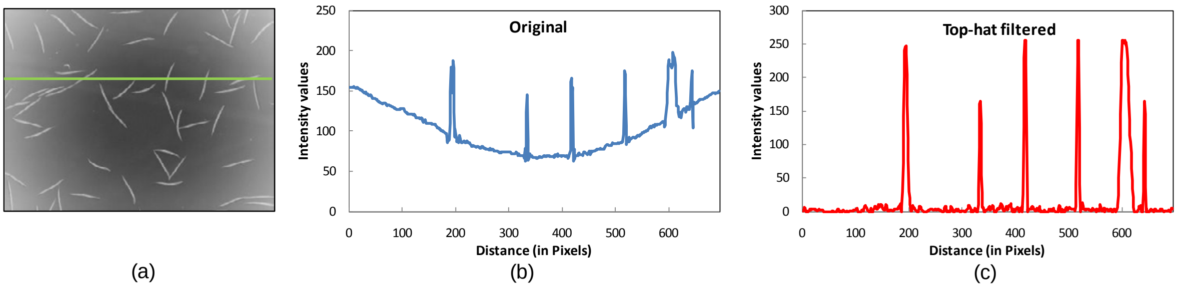


Figure 30 Illustration of top-hat transformation on a 1-D line profile. (a) A gray level image of C elegant, where there is a non-uniform illumination background, and a line profile is marked in green; (b) The original line profile across the image in (a); (c) Top-hat filtered line profile, note that 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⊗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:

$$B = \begin{array}{|c|c|c|}\hline 0 & 0 & X \\ \hline 0 & 1 & 1 \\ \hline X & 1 & X \\ \hline \end{array} \qquad B1 = \begin{array}{|c|c|c|}\hline & & \\ \hline & 1 & 1 \\ \hline & 1 & \\ \hline \end{array} \qquad B2 = \begin{array}{|c|c|c|}\hline 1 & 1 & \\ \hline 1 & & \\ \hline & & \\ \hline \end{array}$$

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**.

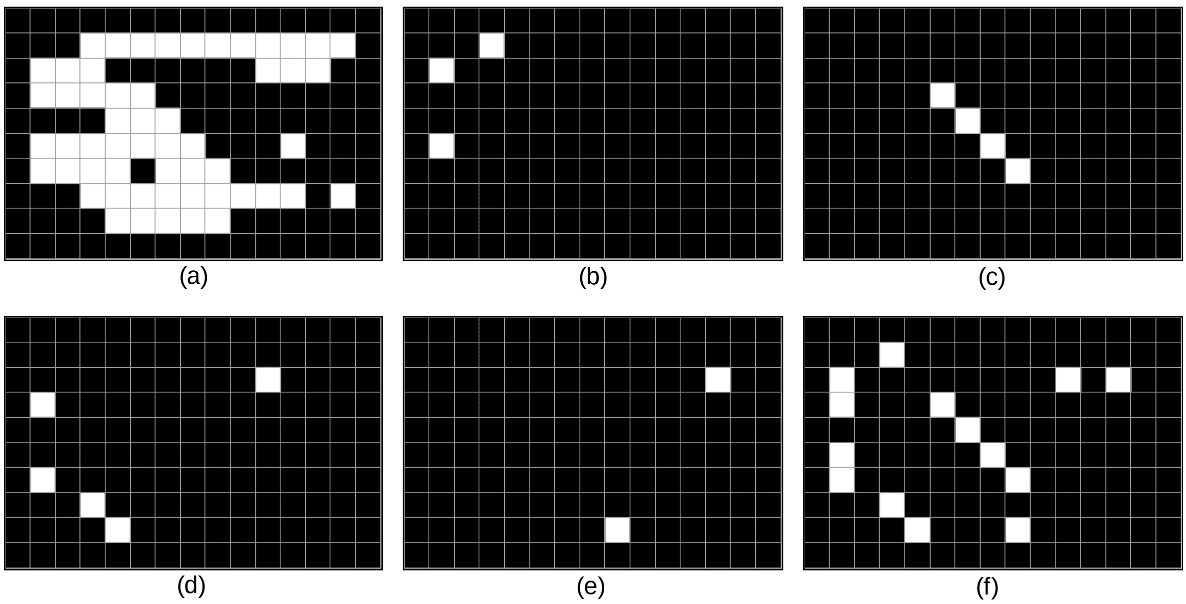


Figure 32 Corner detection by using hit-or-miss transform. (a) A 10 × 15 discrete binary image (the object pixels are the white pixels; (b) Detection for top left corners by hit-or-miss transformation of image(a) by structure element in **Figure 31**; (c) Detection for top left corners; (d) Detection 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 $AØB$, is defined as:

$$A\phi B = A - (A \otimes B) \qquad (2.5.1)$$

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:

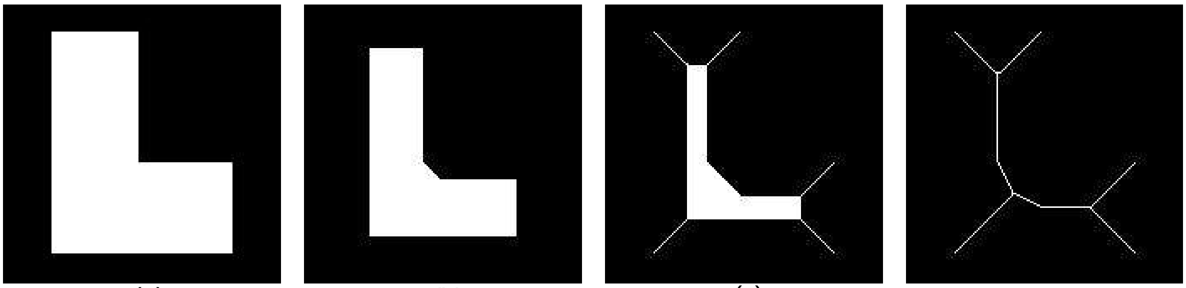
where the origin in each case is the centre element.

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

(2.5.2)

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

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.

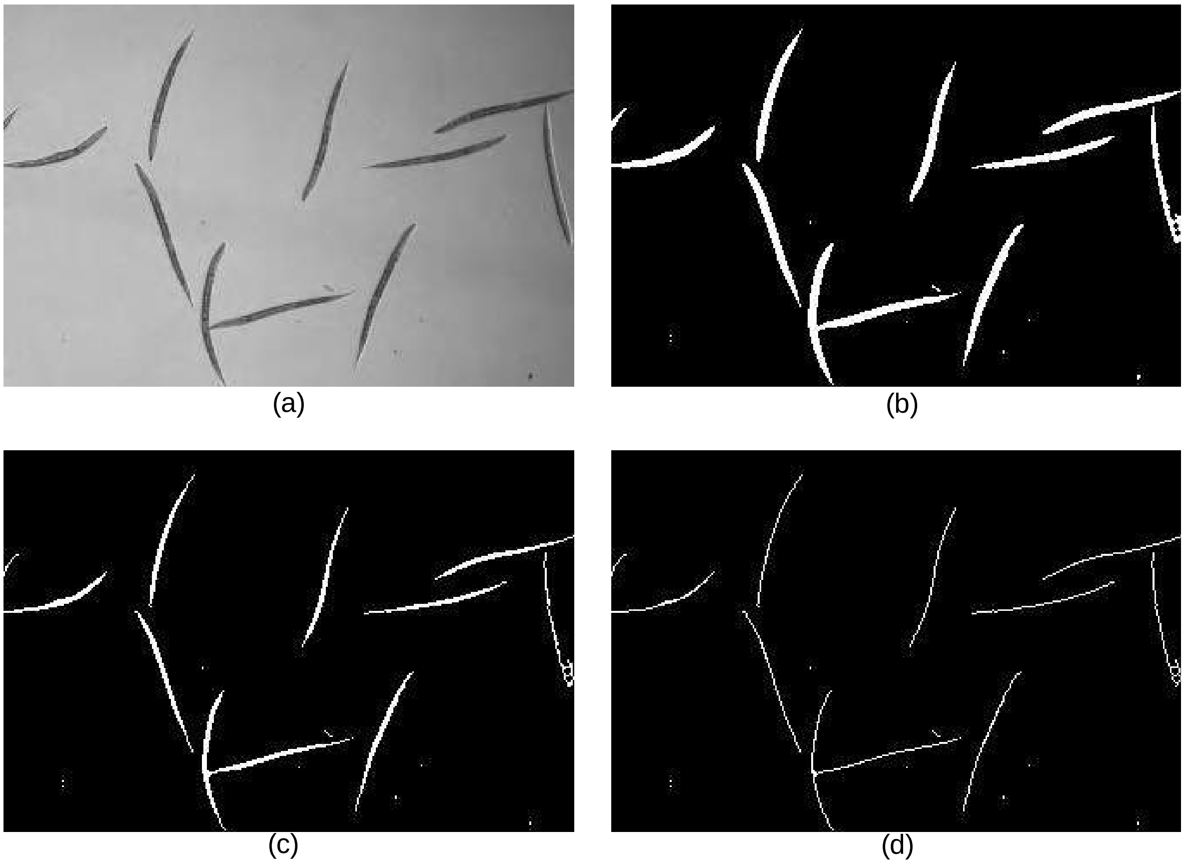


Figure 34 Thinning a group of C elegans. (a) An image of a group of C elegant; (b) Top-hat filtering image (a) followed by intensity thresholding; (c) Thinning of image (b) once; (d) Thinning of image (b) to generate a single-pixel-wide line.

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.

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.

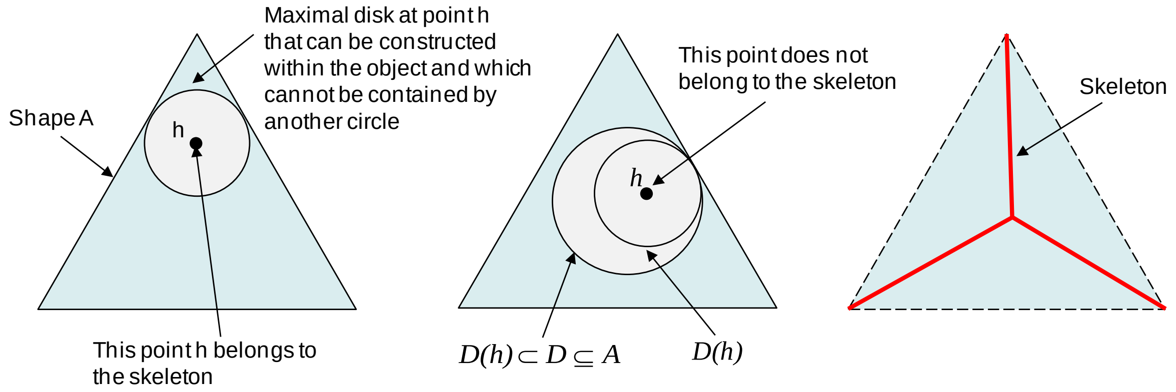


Figure 35 Illustrating the definition of the skeleton of an object: construction of the Euclidean skeleton for a triangular shape.

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**.

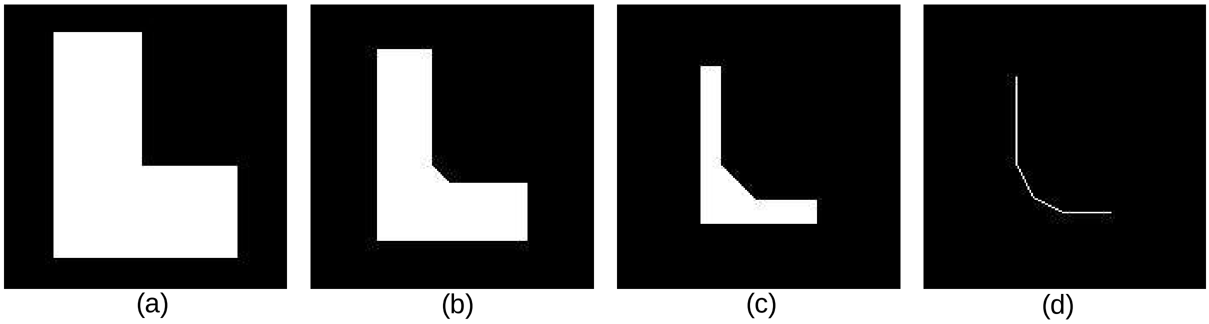


Figure 36 Illustration of skeleton for line detection. (a) A L-shaped original binary image; (b) After 10 iterations of skeletonization; (c) After 20 iterations of skeletonization; (d) The skeletonization process is repeated until no further change occurs, e.g. convergence.

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.

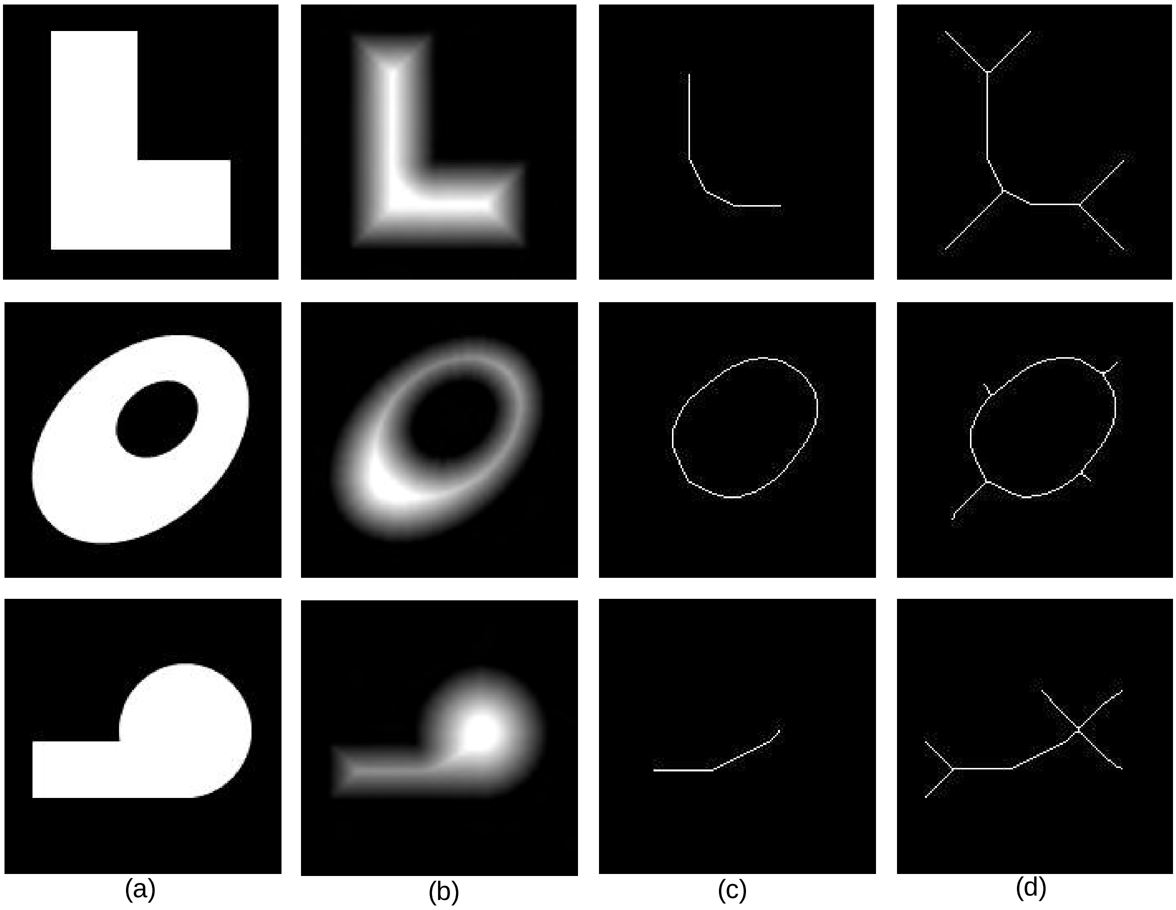
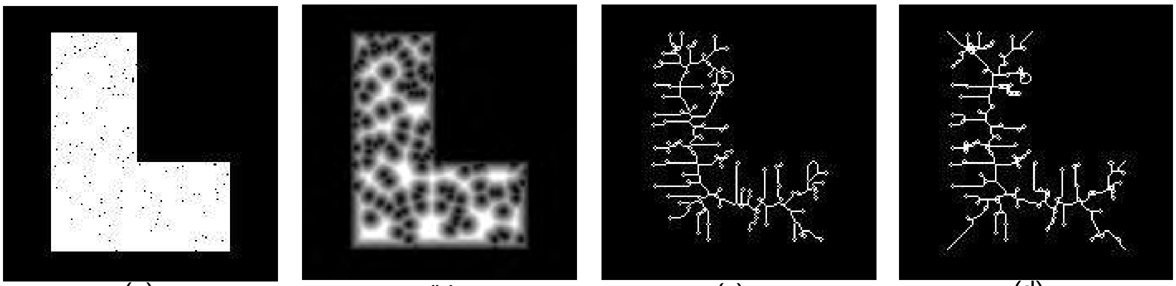


Figure 37 Skeleton vs. thinning. (a) Original binary images with various shapes; (b) Distance transformation of image (a); (c) Skeleton images; (d) Thinning images.

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.

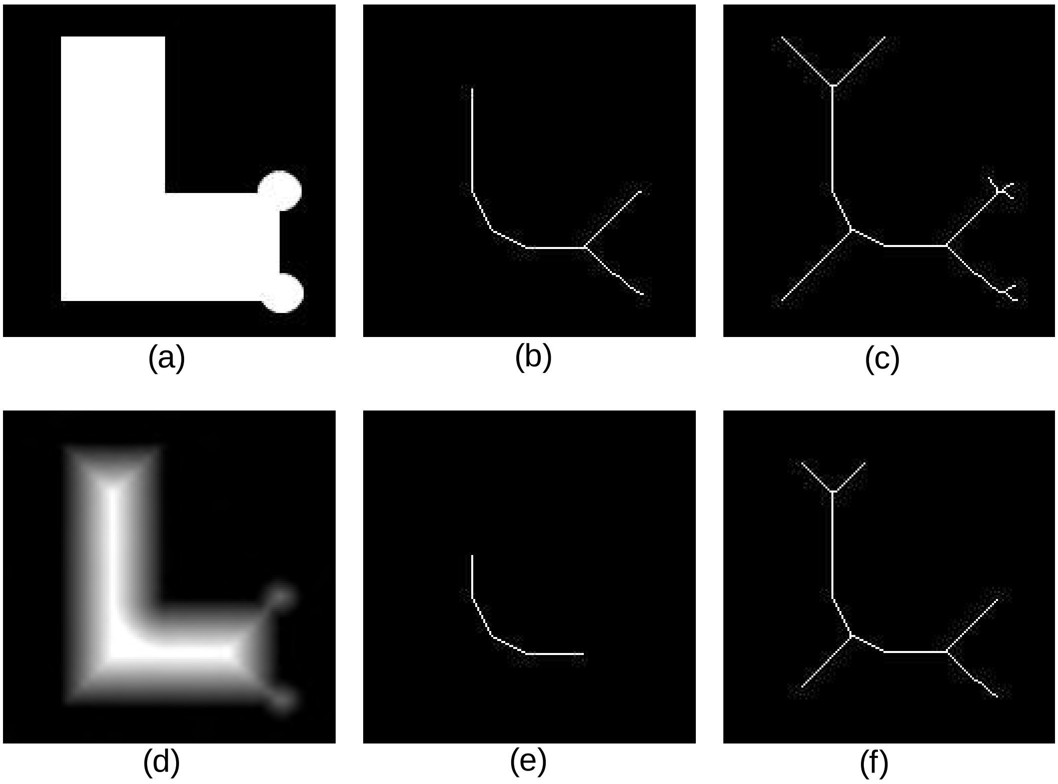


Figure 39 Illustration of pruning. (a) A L-shaped original binary image with some boundary distortion on the bottom right of L; (b) Skeletonization of image (a); (c) Thinning of image (a); (d) Distance transformation of image (a); (e) After 10 iterations of pruning on image (b); (f) After 30 iterations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

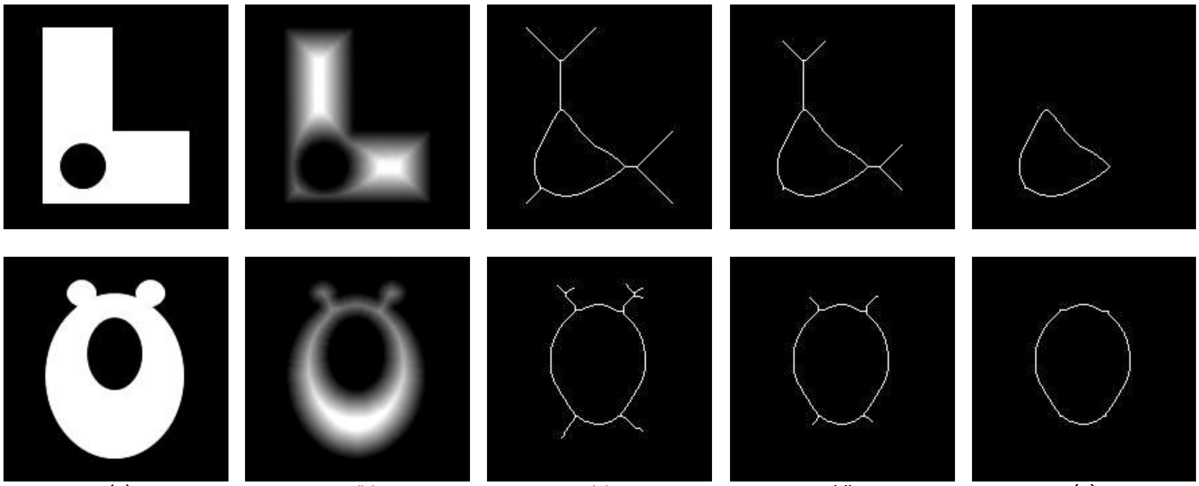


Figure 41 Pruning until convergence removes all pixels except those that form a closed loop. (a) Various shaped original binary image; (b) Distance transformation of image (a); (c) Thinning of image (a); (d) After 10 iterations of pruning on image (c); (e) Pruning on image (c) until convergence.

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}

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

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.

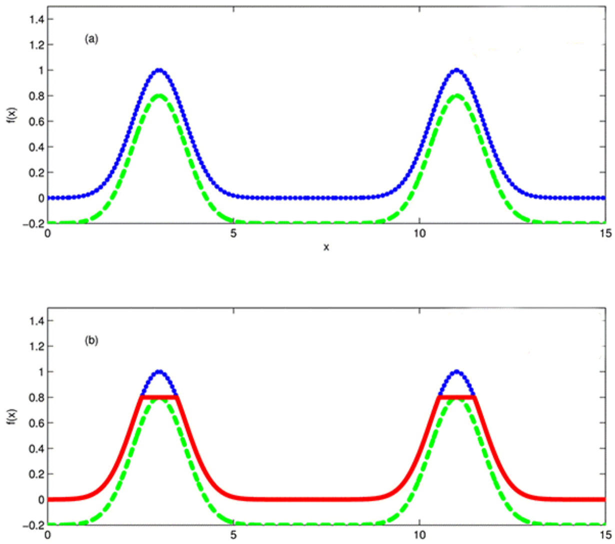


Figure 42 Illustration of morphological reconstruction in 1-D to extract the local maxima. (a) The marker in green is obtained by subtracting a small value of 0.2 from the original signal in blue; (b) Obtain the reconstructed signal in red by using morphological reconstruction, where the difference between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

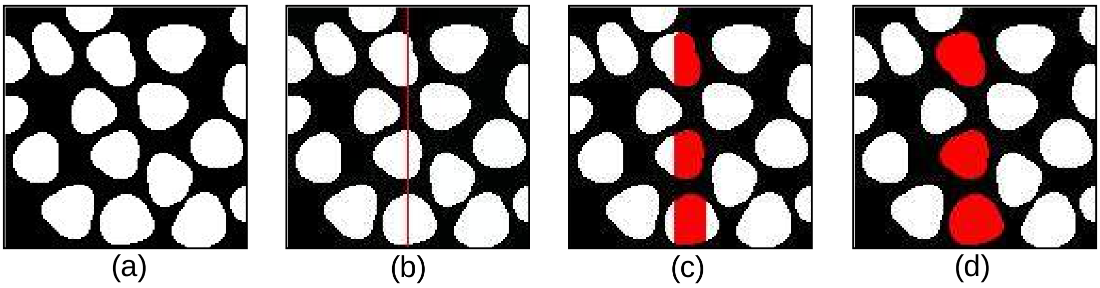


Figure 43 Example of morphological reconstruction.(a) A binary image of the blobs as the mask image; (b) A vertical red line as the marker superimposed on the mask image (a); (c) After 8 successive conditional dilations, the marker gets wider, invading the mask image, mimic the same behaviour of the 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.

2.9 Summary

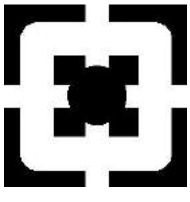
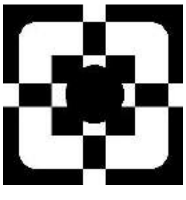
The morphological image processing introduced in this chapter is a powerful tool for extracting or modifying features from an image. The basic morphological operators, such as dilation, erosion and reconstruction, are particularly useful to analysis of binary images, although they can be extended for 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.

2.10 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.

2.11 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 $A\Theta B = (A\Theta B)\bullet B$.

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.