CHAPTER 2
MORPHOLOGY

2.1 INTRODUCTION

Mathematical morphology is a well-founded non-linear theory of image processing [15, 26, 29, 87]. Its geometry-oriented nature provides an efficient framework for analyzing object shape characteristics such as size and connectivity, which are not easily accessed by linear approaches. Morphological operations take into consideration the geometrical shape of the image objects to be analyzed. The initial form of mathematical morphology is applied to binary images and usually referred to as standard mathematical morphology in the literature in order to be discriminated by its later extensions such as the gray-scale and the soft mathematical morphology. Mathematical morphology is theoretically founded on set theory. It contributes a wide range of operators to image processing, based on a few simple mathematical concepts. The operators are particularly useful for the analysis of binary images, boundary detection, noise removal, image enhancement, and image segmentation. The advantages of morphological approaches over linear approaches are

1) Direct geometric interpretation,

2) Simplicity and

3) Efficiency in hardware implementation.

An image can be represented by a set of pixels. A morphological operation uses two sets of pixels, i.e., two images: the original data image to be
analyzed and a structuring element (also called kernel) which is a set of pixels constituting a specific shape such as a line, a disk, or a square. A structuring element is characterized by a well-defined shape (such as line, segment, or ball), size, and origin. Its shape can be regarded as a parameter to a morphological operation.

Basic operation of a morphology-based approach is the translation of a structuring element over the image and the erosion and/or dilation of the image content based on the shape of the structuring element. A morphological operation analyzes and manipulates the structure of an image by marking the locations where the structuring element fits. In mathematical morphology, neighborhoods are, therefore, defined by the structuring element, i.e., the shape of the structuring element determines the shape of the neighborhood in the image.

The hardware complexity of implementing morphological operations depends on the size of the structuring elements. The complexity increases even exponentially in some cases. Known hardware implementations of morphological operations are capable of processing structuring elements only up to $3 \times 3$ pixels [28]. If higher order structuring elements are needed, they are decomposed into smaller elements. One decomposition strategy is, for example, to present the structuring element as successive dilation of smaller structuring elements. This is known as the “chain rule for dilation” [26]. But all structuring elements cannot be decomposed.
Mathematical morphology uses tools of algebra and operates with point sets, their connectivity and shape. Morphological operations are used predominantly for the following purposes:

- Image preprocessing (noise filtering, shape simplification)
- Enhancing object structure (skeletonizing, thinning, thickening, convex hull, object marking)
- Segmenting objects from the background
- Quantitative description of objects (area, perimeter, projections)

Some of the salient points regarding the morphological approach are as follows:

1. Morphological operations provide a method for the systematic alteration of the geometric content of an image while maintaining the stability of important geometric characteristics.

2. There exists a well-developed morphological algebra that can be employed for representation and optimization.

3. It is possible to express digital algorithms in terms of a very small class of primitive morphological operations.

4. There exists rigorous representation theorems by means of which, one can obtain the expression of morphological filters in terms of the primitive morphological operations.
Mathematical Morphology is based on logical transformations of the image (this is no constraint when these transformations are generalized in terms of set definitions) carried out by using the set theoretical operations. This would enable us to make several measurements on the image, like trend, directional effect and holes. The basic step in Morphology is to compare the objects which are to be analyzed with an object of known shape, termed structuring element (This forms one mode of definition). The result of comparison of an object under study (analogous to universe) with a structuring element (analogous to a defined set) causes an image transformation.

2.2 MORPHOLOGICAL PROCESSING AND TRANSFORMS

2.2.1 FUNDAMENTAL DEFINITIONS

An image is a function of two, real (coordinate) variables $a(x,y)$ or two, discrete variables $a[m,n]$. An alternative definition of an image can be based on the notion that an image consists of a set (or collection) of either continuous or discrete coordinates. In a sense the set corresponds to the points or pixels that belong to the objects in the image. This is illustrated in Fig. 2.1, which contains two objects or sets $A$ and $B$. Note that the coordinate system is required. For the moment we will consider the pixel values to be binary. Further we shall restrict our discussion to discrete space ($Z^2$).
Fig. 2.1 A binary image containing two object sets A and B.

The object A consists of those pixels \( a \) that share some common property:

\[
Object - A = \{ \alpha \mid property(\alpha) = \text{TRUE} \}
\]

As an example, object B in Fig 2.1 consists of \{[0,0], [1,0], [0,1]\}.

The background of A is given by \( A^c \) (the complement of A) which is defined as those elements that are not in A:

\[
Background - A^c = \{ \alpha \mid \alpha \notin A \}
\]

If an object A is defined on the basis of C-connectivity (C=4, 6, or 8) then the background \( A^c \) has a connectivity. The necessity for this is illustrated for the Cartesian grid in Fig. 2.2.
A binary image requiring careful definition of object and background connectivity.

The fundamental operations associated with an object are the standard set operations union, intersection, and complement \( \{ \cup, \cap, \, ^c \} \) plus translation:

* **Translation** - Given a vector \( x \) and a set \( A \), the translation, \( A + x \), is defined as:

\[
A + x = \{ \alpha + x | \alpha \in A \}
\]

Since we are dealing with a digital image composed of pixels at integer coordinate positions \( \mathbb{Z}^2 \), this implies restrictions on the allowable translation vectors \( x \).

The basic Minkowski set operations--addition and subtraction--can be defined, based on assumptions that the individual elements that comprise \( B \) are not only pixels but also vectors as they have a clear coordinate position with respect to \([0,0]\). Given two sets \( A \) and \( B \):

**Minkowski addition** -

\[
A \oplus B = \bigcup_{x \in \mathbb{Z}^2} (A + x)
\]
Minkowski subtraction - \( A \ominus H = \bigcap_{\alpha} (A + \beta) \)

2.2.2 DILATION AND EROSION

The fundamental mathematical morphology operations dilation and erosion, based on Minkowski algebra are defined as

Dilation - \( D(A,H) = A \oplus B = \bigvee_{\beta \in B} (A + \beta) \)

Erosion - \( E(A,H) = A \ominus (-B) = \bigvee_{\beta \in B^-} (A - \beta) \)

Where \( -B = \{ -\beta \mid \beta \in B \} \) These two operations are illustrated in Figure 2.3 for the objects defined in Fig. 2.1.

Fig. 2.3. A binary image containing two object sets \( A \) and \( B \). (a) Dilation \( D(A,B) \) (b) Erosion \( E(A,B) \).

While either set \( A \) or \( B \) can be thought of as an "image", \( A \) is usually considered as the image and \( B \) is called a structuring element.

Dilation, in general, causes objects to dilate or grow in size; erosion causes objects to shrink. The amount and the way that they grow or shrink depend upon the choice of the structuring element. Dilating or eroding without
specifying the structural element makes no more sense than trying to lowpass filter an image without specifying the filter. The two most common structuring elements (given a Cartesian grid) are the 4-connected and 8-connected sets, $N_4$ and $N_8$. They are illustrated in Fig. 2.4.

![Fig. 2.4 The standard structuring elements. (a) $N_4$ (b) $N_8$.](image)

$Dilation$ and $erosion$ have the following properties:

- **Commutative** - $D(A,B) = A \oplus B = B \oplus A = D(B,A)$
- **Non-Commutative** - $E(A,B) \neq E(B,A)$
- **Associative** - $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
- **Translation Invariance** - $A \oplus (B + x) = (A \oplus B) + x$
- **Duality** - $D'(A,B) = E'(A', -B)$
  
  $E'(A,B) = D(A', -B)$

With $A$ as an object and $A^c$ as the background, eq. says that the $dilation$ of an object is equivalent to the $erosion$ of the background. Likewise, the $erosion$ of the object is equivalent to the $dilation$ of the background.
Except for special cases:

Non-Inverses - \( D(E(A,B),B) \neq A \neq E(D(A,B),B) \)

Erosion has the following translation property:

Translation Invariance - \( A \ominus (B + x) = (A + x) \ominus B = (A \ominus B) + x \)

Dilation and erosion have the following important properties. For any arbitrary structuring element \( B \) and two image objects \( A_1 \) and \( A_2 \) such that \( A_1 \subset A_2 \) (\( A_1 \) is a proper subset of \( A_2 \)):

Increasing in \( A \) - \( D(A_1, B) \subset D(A, B) \)
\[ E(A_1, B) \subset E(A, B) \]

For two structuring elements \( B_1 \) and \( B_2 \) such that \( B_1 \subset B_2 \):

Decreasing in \( B \) - \( E(A, B_1) \supset E(A, B_2) \)

The decomposition theorems below make it possible to find efficient implementations for morphological filters.

Dilation - \( A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C) = (B \cup C) \oplus A \)

Erosion - \( A \ominus (B \cup C) = (A \ominus B) \cap (A \ominus C) \)

Erosion - \( A \ominus (B \ominus C) = A \ominus (B \ominus C) \)
An important decomposition theorem is due to Vincent. First, we require some definitions. A convex set (in $\mathbb{R}^2$) is one for which the straight line joining any two points in the set consists of points that are also in the set. Care must obviously be taken when applying this definition to discrete pixels as the concept of a "straight line" must be interpreted appropriately in $\mathbb{Z}^2$. A set is bounded if each of its elements has a finite magnitude, in this case distance to the origin of the coordinate system. A set is symmetric if $B = -B$. The sets $N_e$ and $N_s$ in fig 2.4 are examples of convex, bounded, symmetric sets.

Vincent's theorem, when applied to an image consisting of discrete pixels, states that for a bounded, symmetric structuring element $B$ that contains no holes and contains its own center, $[0,0] \in B$:

$$D(A,B) = A \oplus B = A \cup (\partial A \oplus B)$$

where $\partial A$ is the contour of the object. That is, $\partial A$ is the set of pixels that have a background pixel as a neighbor. The implication of this theorem is that it is not necessary to process all the pixels in an object in order to compute a dilation or (using eq.) an erosion. Therefore one has to process the boundary pixels. This also holds for all operations that can be derived from dilations and
erosions. The processing of boundary pixels instead of object pixels means that, except for pathological images, computational complexity can be reduced from \( O(N^2) \) to \( O(N) \) for an \( N \times N \) image. A number of "fast" algorithms can be found in the literature that are based on this result. The simplest dilation and erosion algorithms are frequently described as follows.

* **Dilation** - Take each binary object pixel (with value "1") and set all background pixels (with value "0") that are \( C \)-connected to that object pixel to the value "1".

* **Erosion** - Take each binary object pixel (with value "1") that is \( C \)-connected to a background pixel and set the object pixel value to "0".

Comparison of these two procedures to eq. where \( B = N_{C_{\leq 4}} \) or \( N_{C_{> 4}} \) shows that they are equivalent to the formal definitions for dilation and erosion. The procedure is illustrated for *dilation* in Fig. 2.5.

![Fig. 2.5 Illustration of dilation. Original object pixels are in gray; pixels added through dilation are in black. (a) \( B = N_{C_{\leq 4}} \) (b) \( B = N_{C_{> 4}} \).](image)
2.2.3 BOOLEAN CONVOLUTION

An arbitrary binary image object (or structuring element) $A$ can be represented as:

$$ A \leftrightarrow \sum_{k=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} a[j,k] \delta[m-j,n-k] $$

where $\Sigma$ and $*$ are the Boolean operations $\text{OR}$ and $\text{AND}$, $a[j,k]$ is a characteristic function that takes on the Boolean values "1" and "0" as follows:

$$ a[j,k] = \begin{cases} 
1 & a \in A \\
0 & a \notin A 
\end{cases} $$

and $\delta[m,n]$ is a Boolean version of the Dirac delta function that takes on the Boolean values "1" and "0" as follows:

$$ \delta[j,k] = \begin{cases} 
1 & j = k = 0 \\
0 & \text{otherwise} 
\end{cases} $$

Dilation for binary images can therefore be written as:

$$ D(A,B) = \sum_{k=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} a[j,k] h[m-j,n-k] = a \otimes h $$

which, because Boolean $\text{OR}$ and $\text{AND}$ are commutative, can also be written as
Using De Morgan's theorem:

\[(a+b) = \overline{a \cdot b} \quad \text{and} \quad (a \cdot b) = \overline{a + b}\]

on eq. together with eq. \( \), erosion can be written as:

\[E(A,B) = \bigcap_{k} \bigcup_{j} (a[m-j,n-k]\overline{+} \overline{b} [-j,-k])\]

Thus, dilation and erosion on binary images can be viewed as a form of convolution over a Boolean algebra.

when convolution is employed, an appropriate choice of the boundary conditions for an image is essential. Dilation and erosion—being a Boolean convolution—are no exception. The two most common choices are that either everything outside the binary image is "0" or everything outside the binary image is "1".

\[2.2.4 \ \text{OPENING AND CLOSING}\]

We can combine dilation and erosion to build two important higher order operations:

Opening - \(O(A,B) = A \odot B = D(E(A,B),B)\)

Closing - \(C(A,B) = A \bullet B = E(D(A,-B),-B)\)
The **opening** and **closing** have the following properties:

**Duality** - $C^c(A,B) = O(A^c,B)$

$O^c(A,B) = C(A^c,B)$

**Translation** - $O(A+x,B) = O(A,B) + x$

$C(A+x,B) = C(A,B) + x$

For the **opening** with structuring element $B$ and images $A$, $A_1$, and $A_2$, where $A_1$ is a subimage of $A_1 (A_1 \subseteq A)$:

**Antiextensivity** - $O(A,B) \subseteq A$

**Increasing monotonicity** - $O(A,B) \subseteq O(A',B)$

**Idempotence** - $O(O(A,B),B) = O(A,B)$

For the **closing** with structuring element $B$ and images $A$, $A_1$, and $A_2$, where $A_1$ is a subimage of $A_2 (A_1 \subseteq A_2)$:

**Extensivity** - $A \subseteq C(A,B)$

**Increasing monotonicity** - $C(A,B) \subseteq C(A',B)$

**Idempotence** - $C(C(A,B),B) = C(A,B)$

The two properties given by eqs. and are so important to mathematical morphology that they can be considered as the reason for defining **erosion** with -$B$ instead of $B$ in eq.
2.2.5 HIT-OR-MISS OPERATION

The hit-or-miss operator was defined by Serra but we shall refer to it as the hit-and-miss operator and define it as follows. Given an image $A$ and two structuring elements $B_1$ and $B_2$, the set definition and Boolean definition are:

$$
\text{Hitmiss}(A,B_1,B_2) = \begin{cases} 
E(A,B_1) \cap E(A,B_2) \\
E(A,B_1) \cdot E(A,B_2) \\
E(A,B_1) - E(A,B_2)
\end{cases}
$$

where $B_1$ and $B_2$ are bounded, disjoint structuring elements. Two sets are disjoint if $B_1 \cap B_2 = \emptyset$, the empty set. In an important sense the hit-and-miss operator is the morphological equivalent of template matching, a well-known technique for matching patterns based upon cross-correlation. Here, we have a template $B_1$ for the object and a template $B_2$ for the background.

The results of the application of these basic operations on a test image are illustrated below. In Fig. 2.6 the various structuring elements used in the processing are defined. The value "-" indicates a "don't care". All three structuring elements are symmetric.

$$
B = N_s = \begin{bmatrix} 
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{bmatrix}
B_1 = \begin{bmatrix} 
- & - & - \\
- & 1 & - \\
- & - & - \\
\end{bmatrix}
B_2 = \begin{bmatrix} 
- & 1 & - \\
- & 1 & - \\
- & 1 & - \\
\end{bmatrix}
$$

(a) \hspace{2cm} (b) \hspace{2cm} (c)

Fig. 2.6 Structuring elements $B$, $B_1$, and $B_2$ that are $3 \times 3$ and symmetric.
The opening operation can separate objects that are connected in a binary image. The closing operation can fill in small holes. Both operations generate a certain amount of smoothing on an object contour given a "smooth" structuring element. The opening smoothes from the inside of the object contour and the closing smoothes from the outside of the object contour. The hit-and-miss example has found the 4-connected contour pixels. An alternative method to find the contour is simply to use the relation:

\[
\text{4-connected contour - } \partial A = A - E(A, N_x) \quad \text{or} \\
\text{8-connected contour - } \partial A = A - L(A, N_x)
\]

2.2.6 SKELETON

The informal definition of a skeleton is a line representation of an object that is:

i) one-pixel thick,

ii) through the "middle" of the object, and,

iii) preserves the topology of the object.

Fig. 2.7 Counter examples to the three requirements.
In the first example, Fig 2.7(a), it is not possible to generate a line that is one pixel thick and in the center of an object while generating a path that reflects the simplicity of the object. In Fig. 2.7(b) it is not possible to remove a pixel from the 8-connected object and simultaneously preserve the topology—the notion of connectedness—of the object. Nevertheless, there are a variety of techniques that attempt to achieve this goal and to produce a skeleton.

A basic formulation is based on the work of Lantuéjoul. The skeleton subset $S_k(A)$ is defined as:

$$S_k(A) = E(A,kB) - \{ E(A,kB) \circ B \} \quad k = 0,1, \ldots, K$$

where $K$ is the largest value of $k$ before the set $S_k(A)$ becomes empty. (From eq., $E(A,kB) \circ B \subseteq E(A,kB)$.) The structuring element $B$ is chosen (in $Z^2$) to approximate a circular disc, that is, convex, bounded and symmetric. The skeleton is then the union of the skeleton subsets:

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

An elegant side effect of this formulation is that the original object can be reconstructed given knowledge of the skeleton subsets $S_k(A)$, the structuring element $B$, and $K$:

$$A = \bigcup_{k=0}^{K} (S_k(A) \circ kB)$$
This formulation for the skeleton, however, does not preserve the topology, a requirement described in eq.

An alternative point-of-view is to implement a thinning, an erosion that reduces the thickness of an object without permitting it to vanish. A general thinning algorithm is based on the hit-and-miss operation:

\[
\text{Thinning} - \quad \text{Thin}(A,B_1,B_2) = A - \text{HitMiss}(A,B_1,B_2)
\]

Depending on the choice of \(B_1\) and \(B_2\), a large variety of thinning algorithms—and through repeated application skeletonizing algorithms—can be implemented.

A quite practical implementation can be described in another way. If we restrict ourselves to a \(3 \times 3\) neighborhood, similar to the structuring element \(B = N_s\) in fig 2.6(a), then we can view the thinning operation as a window that repeatedly scans over the (binary) image and sets the center pixel to "0" under certain conditions. The center pixel is **not** changed to "0" if and only if:

i) An isolated pixel is found (e.g. Fig 2.8a),

ii) Removing a pixel would change the connectivity (e.g. Fig 2.8b),

iii) Removing a pixel would shorten a line (e.g. Fig 2.8c).

As pixels are (potentially) removed in each iteration, the process is called conditional erosion. Three test cases of equations are illustrated in Fig. 2.8. In general all possible rotations and variations have to be checked. As there are
only 512 possible combinations for a 3 x 3 window on a binary image, this can be done easily with the use of a lookup table.

![Fig. 2.8 Test conditions for conditional erosion of the center pixel.](image)

(a) Isolated pixel. (b) Connectivity pixel. (c) End pixel.

If only condition (i) is used then each object will be reduced to a single pixel. This is useful if we wish to count the number of objects in an image. If only condition (ii) is used then holes in the objects will be found. If conditions (i + ii) are used each object will be reduced to either a single pixel if it does not contain a hole or to closed rings if it does contain holes. If conditions (i + ii + iii) are used then the "complete skeleton" will be generated.

2.2.7 PROPAGATION

It is convenient to be able to reconstruct an image that has "survived" several erosions or to fill an object that is defined, for example, by a boundary. The formal mechanism for this has several names including region-filling, reconstruction, and propagation. The formal definition is given by the following algorithm. We start with a seed image $S^{(0)}$, a mask image $A$, and a structuring element $B$. We then use dilations of $S$ with structuring element $B$ and masked by $A$ in an iterative procedure as follows:

\[
\text{Iteration } k \rightarrow S^{(k)} = \left( S^{(k-1)} \ast B \right) \cap A \quad \text{until} \quad S^{(k)} = S^{(k-1)}
\]
With each iteration the seed image grows (through dilation) but within the set (object) defined by \( A \); \( S \) propagates to fill \( A \). The most common choices for \( B \) are \( N_4 \) or \( N_8 \). Several remarks are central to the use of propagation. First, in a straightforward implementation, as suggested by eq., the computational costs are extremely high. Each iteration requires \( O(N^2) \) operations for an \( N \times N \) image and with the required number of iterations this can lead to a complexity of \( O(N^3) \). Fortunately, a recursive implementation of the algorithm exists in which one or two passes through the image are usually sufficient, meaning a complexity of \( O(N^2) \). Second, although we have not paid much attention to the issue of object/background connectivity until now (fig 2.2), it is essential that the connectivity implied by \( B \) be matched to the connectivity associated with the boundary definition of \( A \) (see eqs. and ). Finally, as mentioned earlier, it is important to make the correct choice ("0" or "1") for the boundary condition of the image. The choice depends upon the application.

2.3 GRAY VALUE MORPHOLOGICAL PROCESSING

The techniques of morphological filtering can be extended to gray-level images. To simplify matters we will restrict our presentation to structuring elements, \( B \), that comprise a finite number of pixels and are convex and bounded. Now, however, the structuring element has gray values associated with every coordinate position as does the image \( A \).

* Gray-level dilation, \( D_e(x) \), is given by:

\[
D_e(A,B) = \max_{(i,j) \in A} \{d[i,j,n-k]+b[j,k]\}
\]
For a given output coordinate \([m, n]\), the structuring element is summed with a shifted version of the image and the maximum encountered over all shifts within the \(J \times K\) domain of \(B\) is used as the result. Should the shifting require values of the image \(A\) that are outside the \(M \times N\) domain of \(A\), then a decision must be made as to which model for image extension.

*Gray-level erosion, \(E_G(*)\), is given by:

\[
E_G(A, B) = \min_{[j,k] \in B} \{a[m+j, n+k] - b[j, k]\}
\]

The duality between gray-level erosion and gray-level dilation---the gray-level counterpart of eq. ---is somewhat more complex than in the binary case:

\[
Duality \quad E_G(A, B) = -D_G(-\tilde{A}, B)
\]

where "\(-\tilde{A}\)" means that \(a[j,k] -> -a[-j,-k]\).

The definitions of higher order operations such as gray-level opening and gray-level closing are:

\[
Opening \quad O_G(A, B) = D_G(E_G(A, B), B)
\]

\[
Closing \quad C_G(A, B) = -O_G(-A, -B)
\]

The important properties that were discussed earlier such as idempotence, translation invariance, increasing in \(A\), and so forth are also
applicable to gray level morphological processing. The details can be found in Giardina and Dougherty.

In many situations the seeming complexity of gray level morphological processing is significantly reduced through the use of symmetric structuring elements where $h[j, k] = h[-j, -k]$. The most common of these is based on the use of $B = constant = 0$. For this important case and using again the domain $[j, k] \subset B$, the definitions above reduce to:

\[
Dilation - D_v(A, B) = \max_{[j,k] \subset B} \{A[m-j,n-k]\} = \max(A)
\]

\[
Erosion - E_v(A, B) = \min_{[j,k] \subset B} \{A[m-j,n-k]\} = \min(A)
\]

\[
Opening - O_v(A, B) = \max_B(\min(A))
\]

\[
Closing - C_v(A, B) = \min_B(\max(A))
\]

The remarkable conclusion is that the maximum filter and the minimum filter are gray-level dilation and gray-level erosion for the specific structuring element given by the shape of the filter window with the gray value "0" inside the window.

For a rectangular window, $J \times K$, the two-dimensional maximum or minimum filter is separable into two, one-dimensional windows. Further, a one-dimensional maximum or minimum filter can be written in incremental form.
This means that gray-level dilations and erosions have a computational complexity per pixel that is $O(constant)$, that is, independent of $J$ and $K$.

The operations defined above can be used to produce morphological algorithms for smoothing, gradient determination and a version of the Laplacian. All are constructed from the primitives for gray-level dilation and gray-level erosion and in all cases the maximum and minimum filters are taken over the domain $[j,k] \in B$.

### 2.3.1 MORPHOLOGICAL SMOOTHING

This algorithm is based on the observation that a gray-level opening smooths a gray-value image from above the brightness surface given by the function $a[m,n]$ and the gray-level closing smooths from below. We use a structuring element $B$ based on eqs. and.

$$ \text{Morphsmooth}(A,B) = C_*(O_*(A,B),B) = \min(\max(\max(\min(A)))) $$

Here we suppressed the notation for the structuring element $B$ under the max and min operations to keep the notation simple.

### 2.3.2 MORPHOLOGICAL GRADIENT

For linear filters the gradient filter yields a vector representation with a magnitude and direction. The version presented here generates a morphological estimate of the gradient magnitude:

$$ \text{Gradient}(A,B) = \frac{1}{2} (D_G(A,B) - E_G(A,B)) $$
2.3.3 MORPHOLOGICAL LAPLACIAN

The morphologically based Laplacian filter is defined by:

\[
\text{Laplacian} (A, B) = \frac{1}{2} \left( (D_G(A, B) - A) - (A - E_G(A, B)) \right)
\]

\[
= \frac{1}{2} (D_G(A, B) + E_G(A, B) - 2A)
\]

\[
= \frac{1}{2} (\max(A) + \min(A) - 2A)
\]

2.4 BASIC MORPHOLOGICAL OPERATIONS ON THE TEXTURES

The results of different morphological operation on all six tree bark textures and cloth textures are shown from the figures 2.9 to 2.38 and 2.39 to 2.68 respectively.
Fig. 2.9 Morphological operators on TBK-1 (a) Result of Dilation (b) Result of Erosion
Fig. 2.10 Morphological operators on TBK-1 (a) Result of Opening (b) Result of Closing
Fig. 2.11 Morphological operators on TBK-1 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.12 Morphological operators on TBK-1 (a) Result of Thickining (b) Result of Gradient
Fig. 2.13 Morphological operators on TBK-2 (a) Result of Dilation (b) Result of Erosion
Fig. 2.14 Morphological operators on TBK-2 (a) Result of Opening (b) Result of Closing
Fig. 2.15 Morphological operators on TBK-2 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.16 Morphological operators on TBK-2 (a) Result of Thickining (b) Result of Gradient
Fig. 2.17 Morphological operators on TBK-3 (a) Result of Dilation (b) Result of Erosion
Fig. 2.18 Morphological operators on TBK-3 (a) Result of Opening (b) Result of Closing
Fig. 2.19 Morphological operators on TBK-3 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.20 Morphological operators on TBK-3 (a) Result of Thickining (b) Result of Gradient
Fig. 2.21 Morphological operators on TBK-4 (a) Result of Dilation (b) Result of Erosion
Fig. 2.22 Morphological operators on TBK-4 (a) Result of Opening (b) Result of Closing
Fig. 2.23 Morphological operators on TBK-4 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.24 Morphological operators on TBK-4 (a) Result of Thickining (b) Result of Gradient
Fig. 2.25 Morphological operators on TBK-5 (a) Result of Dilation (b) Result of Erosion
Fig. 2.26 Morphological operators on TBK-5 (a) Result of Opening (b) Result of Closing
Fig. 2.27 Morphological operators on TBK-5 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.28 Morphological operators on TBK-5 (a) Result of Thickening (b) Result of Gradient
Fig. 2.29 Morphological operators on TBK-6 (a) Result of Dilation (b) Result of Erosion
Fig. 2.30 Morphological operators on TBK-6 (a) Result of Opening (b) Result of Closing
Fig. 2.31 Morphological operators on TBK-6 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.32 Morphological operators on TBK-6 (a) Result of Thickening (b) Result of Gradient
Fig. 2.33 Morphological operators on TCL-1 (a) Result of Dilation (b) Result of Erosion
Fig. 2.34 Morphological operators on TCL-1 (a) Result of Opening (b) Result of Closing
Fig. 2.35 Morphological operators on TCL-1 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.36 Morphological operators on TCL-1 (a) Result of Thickining (b) Result of Gradient
Fig. 2.37 Morphological operators on TCL-2 (a) Result of Dilation (b) Result of Erosion
Fig. 2.38 Morphological operators on TCL-2 (a) Result of Opening (b) Result of Closing
Fig. 2.39 Morphological operators on TCL-2 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.40 Morphological operators on TCL-2 (a) Result of Thickining (b) Result of Gradient
Fig. 2.41 Morphological operators on TCL-3 (a) Result of Dilation (b) Result of Erosion
Fig 2.42 Morphological operators on TCL-3 (a) Result of Opening (b) Result of Closing
Fig. 2.43 Morphological operators on TCL-3 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.44 Morphological operators on TCL-3 (a) Result of Thickining  (b) Result of Gradient
Fig. 2.45 Morphological operators on TCL-4 (a) Result of Dilation (b) Result of Erosion
Fig. 2.46 Morphological operators on TCL-4 (a) Result of Opening (b) Result of Closing
Fig. 2.47 Morphological operators on TCL-4 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.48 Morphological operators on TCL-4 (a) Result of Thickening (b) Result of Gradient
Fig. 2.49 Morphological operators on TCL-5 (a) Result of Dilation (b) Result of Erosion
Fig. 2.50 Morphological operators on TCL-5 (a) Result of Opening (b) Result of Closing
Fig. 2.51 Morphological operators on TCL-5 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.52 Morphological operators on TCL-S (a) Result of Thickening (b) Result of Gradient
Fig. 2.53 Morphological operators on TCL-6 (a) Result of Dilation (b) Result of Erosion
Fig. 2.54 Morphological operators on TCL-6 (a) Result of Opening (b) Result of Closing
Fig. 2.55 Morphological operators on TCL-6 (a) Result of Hit or Miss (b) Result of Thinning
Fig. 2.56 Morphological operators on TCL-6 (a) Result of Thickining (b) Result of Gradient
2.5 SUMMARY

Mathematical morphology stresses the role of shape in image pre-processing, segmentation, and object description. It constitutes a set of tools that have solid mathematical background and lead to fast algorithms. The basic entity is a point set. Morphology operates using transformations that are described using operators in a relatively simple non-linear algebra. Mathematical morphology constitutes a counterpart to traditional signal processing based on linear operators (such as convolution). Mathematical morphology is usually divided into binary mathematical morphology, which operates on binary images (2D point sets), and gray-level mathematical morphology, which acts on gray-level images (3D point sets).

In images, morphological operations are relations of two sets. One is an image and the second a small probe, called a structuring element, that systematically traverses the image; its relation to the image in each position is stored in the output image. Fundamental operations of mathematical morphology are dilation and erosion. Dilation expands an object to the closest pixels of the neighborhood. Erosion shrinks the object. Erosion and dilation are not invertible operations; their combination constitutes new operations—opening and closing. Thin and elongated objects are often simplified using a skeleton that is an archetypical stick replacement of original objects. The skeleton constitutes a line that is in the middle of the object.
The Morphological operations on textures clearly show uniform patterns in cloth textures and whereas more number of regions with different topologies are exhibited by the tree bark textures. This factors clearly co-insides with the nature of these textures.