Morphology Based Microarray Image Segmentation

4.1. Morphology:
Morphology is the branch of biology it deals with the form and structure of animals and plants. In mathematical morphology is used to extract image components that are useful in the representation and description of region shape, such as boundary extraction, skeleton, convex hull. Also is the part of interesting in morphological techniques for pre-or-post-processing, such as morphological filtering, thinning, and pruning. [1, 2, 3] Morphology is biological term refers to study of form and structure, in imaging; the term is not used so generically. Morphological operators often take a binary image and a structuring element as input and combine them using a set operator (intersection, union, inclusion, complement). They process objects in the input image based on information of its shape, which are encoded in the structuring element. All morphological processing
depends on the concept of fitting structuring elements. The mathematical details are explained in Mathematical morphology. Usually, the structuring element is sized $3 \times 3$ and has its origin at the center pixel. It is shifted over the image and at each pixel of the image its elements are compared with the set of the underlying pixels. Mathematical Morphology refers to a branch of nonlinear image processing and analysis that concentrates on the geometric structure within an image, it is mathematical in the sense that the analysis is based on set theory, topology, lattice, random functions, etc. [4]. A morphological operator is a set operation performed between the input image and a given set, called structuring element. Usually the structuring elements used to compute the basic morphological operations (dilation, erosion) are isotropic, and then they do not take into account the local morphological properties of the input image [5]. Whereas erosion and dilation are considered the primary morphological operations and the operations of opening and closing are secondary operations and are implemented using erosion and dilation operations [1, 3]. Both of these operators take two pieces of data as input: an image to be eroded or dilated, and a structuring element (also known as a kernel). The two pieces of input data are each treated as representing sets of coordinates in a way that is slightly different for binary and grayscale images. For a binary image, white pixels are normally taken to represent foreground regions, while black pixels denote background. (Note that in some implementations this convention is reversed, and so it is very important to set up input images with the correct polarity for the implementation being used). Then the set of coordinates corresponding to that image is simply characterized to set the two-dimensional Euclidean coordinates of all the foreground pixels in the image, with an origin normally taken in one of the corners so that all coordinates have positive elements. For a grayscale image, the intensity value is taken to represent height above a base plane, so that the grayscale image represents a surface in three-dimensional Euclidean space. Then the set of coordinates associated with this image surface is simply the set of three-dimensional Euclidean coordinates of all the points within this surface and also all points below the surface, down to the base plane. Note that even when only considering points with integer coordinates, this is a lot of points, so usually algorithms are employed that do not need to consider all the points [6, 7]. A simple and general-purpose approach relying on a single parameter and leading to the segmentation of
arbitrary binary patterns into seven categories: core, islet, loop, bridge, perforation, edge, and branch. All categories are obtained by applying a series of operators originating from mathematical morphology (Serra, 1982). He uses the notations and definitions detailed in (Soille, 2003). Let $i$ be a binary image in the square grid with foreground pixels set to 1 and background pixels set to 0. He uses the notion of path connectivity to establish whether a group of foreground (respective background) pixels is connected or not. To avoid the connectivity paradox of raster grids, he assumes that the foreground is 8-connected and therefore the background is 4-connected (or vice versa). Since the analysis is performed on a raster grid, a size of 1 is equal to the distance separating the center of two 4-adjacent pixels (i.e. the width of a pixel). [6]

4.2. Shape Analysis Applications: following table shows the shape analysis applications of various research fields and their examples as shown in table 1.1 [7].

Table 4.1: shape analysis applications:

<table>
<thead>
<tr>
<th>Research Field</th>
<th>Example of Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neuroscience</td>
<td>Morphological taxonomy of neural cells, investigations about the interplay between form and functions, comparisons between cells of different cortical areas and between cells of different species, modeling of biologically realistic cells, simulation of neural structures.</td>
</tr>
<tr>
<td>Document Analysis</td>
<td>WWW, OCR, Multimedia databases, historical documents</td>
</tr>
<tr>
<td>Visual art</td>
<td>Video restoration, special effects, video tracking, games, computer graphics, image synthesis</td>
</tr>
<tr>
<td>Internet</td>
<td>Content-based information retrieval, watermarking, graphics design and usability</td>
</tr>
<tr>
<td>Medicine</td>
<td>Tumor recognition, quantification of change and/or deformation of anatomical structures (e.g. endocardial contour of left ventricle of heart, corpus callosam), morphometric analysis, identification of genetic pathologies, laparoscopy.</td>
</tr>
<tr>
<td>Biology</td>
<td>Morphometric based evaluation comparison, taxonomy.</td>
</tr>
<tr>
<td>Physics</td>
<td>Analysis of particle trajectories, crystal growth, polymers, etc.</td>
</tr>
</tbody>
</table>
Interesting shape analysis problem arises involving the following difficulties questions:
1. How many morphological classes of neurons are there?
2. How can we assign morphological classes to neural cells?
3. How reliable are the shape features and the identification? [7]

### 4.3. Shape Features:
Shape of an object refers to its profile and physical structure. These characteristics can be represented by the boundary, region, moment and structural representations this can be used for matching shapes, recognizing objects, or for making measurement of shapes. The following figure 4.1 shows list of several useful features of shapes. [8]

**Fig. 4.1:** Classification of Shape Representation.

### 4.4. Structuring Element:
The mathematical morphology is a method to quantitatively describe operations effective for the shape of objects in an image, and has recently attracted much attention [7]. The
mathematical morphology describes such operations by combinations of basic set operations between an image and a small object called a structuring element. It is very attractive for this purpose because it efficiently deals with geometrical features such as size, shape, contrast, or connectivity that can be considered as segmentation oriented features [9]. One of the advantages of using morphological approach is its low computational cost. Filters based on opening & closing by partial reconstruction can efficiently achieve the simplification for segmentation. The size of structuring element is progressively decreased to allow the introduction of more local information to improve the segmentation [4]. In all morphological operations like, \( A \ast B, A \bullet B \ldots \) and so on, where \( B \) changes the feature, structure of the image but very few researchers concentrate on this \( B \). It is nothing but a structuring element.

Structuring element is a small grid representing pixels, which are either set (1), not set (0), or “don’t care”. It is applied to images to change the structure of the image content. The center pixel of the structuring element, called the origin, which recognize the pixel of interest. The morphological function use this code to get coordinates of the origin of the structuring elements of any size and dimension.

\[
\text{Origin} = \text{floor} \left( \left( \text{size (neighborhood)} + 1 \right) / 2 \right) \ldots (4.1)
\]

The structuring element (SE) is generally of square dimension of size 3x3, 5x5 and sometimes greater depending upon the application. Structuring elements that fit into a 3x3 grid with its origin at the center are the most commonly seen type. [10]

\[
\begin{array}{ccc}
X3 & X2 & X1 \\
X4 & X & X0 \\
X5 & X6 & X7
\end{array}
\]

**Fig. 4.2.** 3x3 Structuring Element

Every pixel in the input image is evaluated with its eight neighborhoods to produce a resulting output pixel value. Structuring elements are also known as morphological mask/Kernel, which play an important role in morphological operation like opening and closing.

The structuring element consists of a pattern specified as the coordinates of a number of discrete points relative to some origin. Normally Cartesian coordinates are used and so a
convenient way of representing the element is as a small image on a rectangular grid. Figure 4.3 shows a number of different structuring elements of various sizes. In each case a ring around that point marks the origin. The origin does not have to be in the center of the structuring element, but often it is.

![Some examples of structuring elements](image)

**Fig. 4.3:** Some examples of structuring elements

Note that each point in the structuring element may have a value. In the simplest structuring elements used with binary images for operations such as erosion, the elements only have one value, conveniently represented as a one. More complicated elements, such as those used with thinning or grayscale morphological operations may have other pixel values. An important point to is that although a rectangular grid is used to represent the structuring element, not every point in that grid is part of the structuring element in general. Hence the elements shown in Figure 4.3 contain some blanks. In many texts, these blanks are represented as 0's, but this can be confusing and so this is avoid it here.

When a morphological operation is carried out, the origin of the structuring element is typically translated to each pixel position in the image in turn, and then the points within the translated structuring element are compared with the underlying image pixel values. The details of this comparison and the effect of the outcome depend on which morphological operator is being used. [10]

**4.4.1. Basic Structuring Elements:**

Structuring element (SE), of the type specified by shape. Depending on shape, structuring element can take additional parameters. There are two types of structuring element. [11]

I. Flat Structuring Element.

II. Non-flat Structuring Element.
Flat structuring element have maximum two parameters, this may used and vary as per requirement, whereas, non-flat structuring element have one additional parameter i.e. height, that means non-flat structuring element may used for 3D images. In this research work, all flat and non-flat structuring elements are applied on various microarray images and results mentioned in chapter 5.

4.4.2. Types of Structuring Element:
There are nine types of flat structuring element out of which, some need at least one parameter and some need minimum two, to give appropriate results. Non-flat structuring elements are only two with additional parameter i.e. height [10, 12].

4.4.2.1. Flat Structuring Elements are of following shapes
- Arbitrary
- Pair
- Diamond
- Disk
- Periodic Line
- Rectangle
- Line
- Square
- Octagon

4.4.2.2. Non-flat Structuring Elements are of following shapes
- Arbitrary
- Ball

4.4.2.1. Flat Structuring Elements
4.4.2.1.1. Arbitrary:
Arbitrary creates a flat structuring element where NHOOD specifies the neighborhood. NHOOD is a matrix containing 1's and 0's; the location of the 1's defines the neighborhood for the morphological operation. The center (or origin) of NHOOD is its center element, given by

\[
\text{floor} \left( \frac{(\text{size} (\text{NHOOD}) +1)/2}{\ldots} \right) \quad (4.2)
\]
4.4.2.1.2. Pair

Pair creates a flat structuring element containing two members. One member is located at the origin. The vector OFFSET specifies the second member's location. OFFSET must be a two-element vector of integers.

4.4.2.1.3. Diamond

Diamond creates a flat, diamond-shaped structuring element, where R specifies the distance from the structuring element origin to the points of the diamond. R must be a nonnegative integer scalar.
4.4.2.1.4. Disk
Disk creates a flat, disk-shaped structuring element, where \( R \) specifies the radius. \( R \) must be a nonnegative integer. \( N \) must be 0, 4, 6, or 8. When \( N \) is greater than 0, the disk-shaped structuring element is approximated by a sequence of \( N \) periodic-line structuring elements. When \( N \) equals 0, no approximation is used, and the structuring element members consist of all pixels whose centers are no greater than \( R \) away from the origin. If \( N \) is not specified, the default value is 4.

![Disk shaped Structuring element with radius R=3](image)

Fig. 4.8: Disk shaped Structuring element with radius \( R=3 \)

![Disk shaped Structuring element with radius R=5](image)

Fig. 4.9: Disk shaped Structuring element with radius \( R=5 \)

4.4.2.1.5. Periodic line
Periodic line creates a flat structuring element containing \( 2P+1 \) members. \( V \) is a two-element vector containing integer-valued row and column offsets. One structuring element member is located at the origin. The other members are located at \( 1V, -1V, 2V, -2V, ..., PV, -PV \).

![Periodic Line shaped S. E.](image)

Fig. 4.10: Periodic Line shaped S. E.

4.4.2.1.6. Rectangle
Rectangle creates a flat, rectangle-shaped structuring element, where \( MN \) specifies the size. \( MN \) must be a two-element vector of nonnegative integers. The first element of
MN is the number of rows in the structuring element neighborhood; the second element is the number of columns.

\[ MN = [3 \ 5] \]

**Fig. 4.11:** Rectangle shaped S.E.

### 4.4.2.1.7. Line

Line creates a flat, linear structuring element, where LEN specifies the length, and DEG specifies the angle (in degrees) of the line, as measured in a counterclockwise direction from the horizontal axis. LEN is approximately the distance between the centers of the structuring element members at opposite ends of the line.

**Fig. 4.12:** Line shaped S.E. with angle 45

### 4.4.2.1.8. Square

Square creates a square-structuring element whose width is W pixels. W must be a nonnegative integer scalar.

**Fig. 4.13:** Square S.E. with width 3
4.4.2.1.9. Octagon

Octagon creates a flat, octagonal structuring element, where $R$ specifies the distance from the structuring element origin to the sides of the octagon, as measured along the horizontal and vertical axes. $R$ must be a nonnegative multiple of 3.

![Octagon Shaped S.E. with radius 3](image)

**Fig. 4.14:** Octagon shaped S.E. with radius 3

4.4.2.2. Non-Flat Structuring Elements

4.4.2.2.1. Arbitrary

Arbitrary creates a non-flat structuring element, where NHOOD specifies the neighborhood. HEIGHT is a matrix the same size as NHOOD containing the height values associated with each nonzero element of NHOOD. The HEIGHT matrix must be real and finite valued.

4.4.2.2.2. Ball

Ball creates a non-flat, ball-shaped structuring element (actually an ellipsoid) whose radius in the X-Y plane is $R$ and whose height is $H$. Note that $R$ must be a nonnegative integer, $H$ must be a real scalar, and $N$ must be an even nonnegative integer. When $N$ is greater than 0, the ball-shaped structuring element is approximated by a sequence of $N$ non-flat, line-shaped structuring elements. When $N$ equals 0, no approximation is used, and the structuring element members consist of all pixels whose centers are no greater than $R$ away from the origin. The corresponding height values are determined from the formula of the ellipsoid specified by $R$ and $H$. If $N$ is not specified, the default value is 8. In this work, four types of flat and flat structuring elements are applied on microarray images to find out exact shape and size of structuring element suitable for microarray image feature extraction.
4.5. Five Basic Structuring Elements:

![Fig. 4.15: Five basic structuring elements used for binary morphology. The origin of each element is at center and the X’s indicate “don’t care” values.](image)

**Table 4.2: Examples of Morphological Operations:**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Use of operation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dilation</td>
<td>To grow image regions</td>
<td><img src="image" alt="Example" /></td>
</tr>
<tr>
<td></td>
<td>(Dilate, Grow, Expand)</td>
<td></td>
</tr>
<tr>
<td>Erosion</td>
<td>To shrink image regions</td>
<td><img src="image" alt="Example" /></td>
</tr>
<tr>
<td></td>
<td>(Erode, Shrink, Reduce)</td>
<td></td>
</tr>
<tr>
<td>Opening</td>
<td>Structured removal of image region boundary pixels</td>
<td><img src="image" alt="Example" /></td>
</tr>
<tr>
<td>Closing</td>
<td>Structured filling in of image region boundary pixels</td>
<td><img src="image" alt="Example" /></td>
</tr>
<tr>
<td>Hit and Miss Transform</td>
<td>Image pattern matching and marking</td>
<td><img src="image" alt="Example" /></td>
</tr>
<tr>
<td>Thinning</td>
<td>Structured erosion using image pattern matching</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Thickening</td>
<td>Structured dilation using image pattern matching</td>
<td></td>
</tr>
<tr>
<td>Skeletonization</td>
<td>Finding skeletons of binary regions</td>
<td></td>
</tr>
</tbody>
</table>

### 4.5.1. Properties of Morphology Operators

#### 4.5.1.1. Translation

Let $A$ and $B$ be subsets of $\mathbb{Z}^2$. The translation of $A$ by $z$ is denoted $A_z$ and is defined as

\[ (A)_z = \{ w \mid w = a + z, \text{for } a \in A \} \]

#### 4.5.1.2. Reflection

The reflection of $B$, denoted $\hat{B}$, is defined as

\[ \hat{B} = \{ w \mid w = -b, \text{for } b \in B \} \]

#### 4.5.1.3. Difference

Set of points that belongs to $A$ but not to $B$

\[ A - B = \{ w \mid w \in A, w \notin B \} = A \cap B^c \]

#### 4.5.1.4. Thinning

Thins set $A$. The first two equations give the basic definition of thinning. The last two equations denote thinning by a sequence of structuring elements.

\[ A \otimes B = A - (A \ast B) \]

\[ A \cap (A \ast B)^c \]

\[ A \otimes \{B\} = ((...((A \otimes B^1) \otimes B^2)...) \otimes B^n) \]

\[ \{B\} = \{B^1, B^2, B^3, ...B^n\} \]

#### 4.5.1.5. Thickening

Thickens set $A$. (See preceding comments on sequences of structuring elements.)
\[ A \cdot B = A \cup (A \ast B) \] ....... ........ ........ ...(4.7.1) 
\[ A \cdot \{ B \} = ((A \cdot B^1) \cdot B^2) \cdot B^n \] ........ (4.7.2)

4.5.1.6. Pruning

Is essential complement to thinning and skeletonizing algorithms because procedures tend to leave parasitic components that need to be "cleaned up" by post processing.

\[ X_4 \] is the results of pruning set A. The number of times that the first equation is applied to obtain \( X_1 \) must be specified. Structuring elements V are used for the first two equations.

In the third equation \( H \) denotes structuring element I.

\[ X_1 = A \otimes \{ B \} \] ....... ........ ........ .......(4.8.1)
\[ X_2 = \bigcup_{k=1}^{8} (X_1 \ast B^k) \] ....... ........ ........ ..(4.8.2)
\[ X_3 = (X_2 \oplus H) \cap A \] ....... ........ ........ (4.8.3)
\[ X_4 = X_1 \cup X_3 \] ....... ........ ........ .......(4.8.4)

4.5.2. Applications of Gray Scale Morphology:

Although there are some applications requiring basic gray-scale morphological operations, most applications of morphology are developed for binary images. A list of binary morphological applications follows.

- Boundary extraction
- Region filling
- Extraction of connected components
- Convex Hull
- Thinning
- Thickening
- Skeletons
- Pruning

All these applications are carried out by applying a series of basic operations with the different type of structuring elements. For gray-scale morphological case, it can be expanded to

- Smoothing
- Morphological gradient
- Top hat, Bottom-hat Transformations
4.5.2.1. Boundary Extraction:
The boundary of a set A, denoted by $\beta (A)$, can be obtained by first eroding A by B and then performing the set difference between A and its erosion. That is,

$$\beta (A) = A - (A - B)$$

Where B is a suitable structuring element. Fig 16 illustrates the mechanics of boundary extraction. It shows simple binary object, a structuring element B, and the result of using above equation.

4.5.2.2. Region Filling
It is based on set dilations, complementation, and intersections. A denotes a set containing a subset whose elements are 8-connected boundary points of a region. Beginning with a point p inside the boundary, the objective is to fill the entire region with 1’s.

$$X_k = (X_k - 1 \oplus B) \cap A^2 k = 1,2,3 \ldots \ldots \ldots$$

4.5.2.3. Smoothing
One of the best solutions to achieve smoothing is to perform a morphological opening operation followed by the closing operation. The net result of these two operations is to remove or attenuate both bright and dark artifacts and noise.

4.5.2.4. Morphological Gradient
The removal of small dark and bright artifacts, dilation and erosion often is used to compute the morphological gradient of an image, denoted $g$; for linear filters the gradient filter yields a vector representation with a magnitude and direction.

$$g = (f \circ b) - (f - b)$$ \hspace{1cm} (4.11)

### 4.5.2.5. Top-hat Transformation

When we subtract the opened image from the original image ($f$) is known as top-hat transform of an image, denoted $h$, is defined as,

$$h = f - (f \circ b)$$ \hspace{1cm} (4.12)

### 4.5.2.6. Bottom-hat Transform

When we subtract the closed image from the original image ($f$) is known as Bottom-hat Transform $h$ and which is defined as,

$$h = f - (f \bullet b)$$ \hspace{1cm} (4.13)

Where, as before, $f$ is the input image and $b$ is the structuring element function, This transform, which owns its original name to the use of a cylindrical or parallelepiped structuring element function with a flat top is useful for enhancing detail in the presence of shading.

### 4.5.2.7. Texture:

Texture is observed in structural patterns of surfaces of objects such as wood, grain, sand, grass, and cloth. The term texture generally refers to repetition of basic texture elements called texels. A texel contains several pixels, whole placement could be periodic quasi-periodic or random, whereas artificial textures are often deterministic or periodic. Texture may be coarse, fine, smooth, granulated, rippled, regular, irregular, or linear. In image analysis, texture is broadly classified into two categories, statistical and structural. The classification of texture is shown in following fig. 4.18. [8]

Textural segmentation performs,

1. Closing the image by using successively larger structuring elements than small blobs: as closing tends to remove dark details from an image, thus the small blobs are removed from the image, leaving only a light background on the left and larger blobs on the right.
2. Opening with a structuring element that is large in relation to the separation between the large blobs: opening removes the light patches between the blobs, leaving dark region
on the right consisting of the large dark blobs and now equally dark patches between these blobs. [1, 2]

**Classification of Texture**

- **Statistical:**
  - ACF
  - Transform
  - Edgness
  - Concurrence matrix
  - Texture transforms
  - Random field models

- **Structural:**
  - Periodic:
    - Primitives:
      - Gray levels
      - Shape
      - Homogeneity
    - Placement rules:
      - Period
      - Adjacency
      - Closest distances
  - Random:
    - Edge density
    - Extreme density
    - Run length

- **Other:**
  - Mosaic Model

**Fig. 4.17:** Classification of texture.

**4.5.2.8. Granulometry:**
Granulometry is to determine the size distribution of particles in an image. Also it to performs, opening operation with structuring element of increasing size on the original image. The difference between the original image and its opening is computed after each pass when a different structuring element is completed. At the end of the process, these differences are normalized and then used to construct a histogram of particle-size distribution. This approach is based on the idea that opening operations of a particular size have the most effect on regions of the input image that contains particles of similar size. [1, 2]

**4.5.3. Morphology for Gray-Scale Images:**
In this section, the binary morphology is extended to the gray-scale images the basic operations of dilation, erosion, opening, and closing case [1, 3, 7]. In other words, Mathematical morphology is a non-linear image processing approach, which is based on
the application of lattice theory to spatial structures [13]. The image is depicted in four different graphical representations: a) the pixel values mapped in gray scale: low values are dark and high values are bright gray tones; b) the pixel values also mapped in gray scale but in a reverse order: low values are bright and high values are dark gray tones; c) the same but as a top-view shading surface; and d) a mesh plot of the same surface. Through the discussions below, we use the ‘Maximum’ and ‘Minimum’ filters to define gray-scale morphological operators such as top hat and bottom hat [4]. Using these concepts, gray-scale morphology can be easily extended from binary morphology with the same concept. The differences between binary and gray-scale morphology results from the definitions of dilation and erosion because other operations basically depend on these. Except for these definitions, gray-scale morphology is fairly similar to the binary case. Hence, in this section, definitions for gray-scale opening and closing as well as some of examples for gray-scale morphological operations are merely given. Also, some expansions of the morphological procedure are mentioned at below [7].

For a grayscale image, the intensity value is taken to represent height above a base plane, so that the grayscale image represents a surface in three-dimensional Euclidean space. Figure 18 shows such a surface. Then the set of coordinates associated with this image surface is simply the set of three-dimensional Euclidean coordinates of all the points within this surface and also all points below the surface, down to the base plane. Note that even when we are only considering points with integer coordinates, this is a lot of points, so usually algorithms are employed that do not need to consider all the points.

Figure 4.18: Simple gray level image and the corresponding surface in image space.
The structuring element is already just a set of point coordinates (although it is often represented as a binary image). It differs from the input image coordinate set in that it is
normally much smaller, and its coordinate origin is often not in a corner, so that some coordinate elements will have negative values. Note that in many implementations of morphological operators, the structuring element is assumed to be a particular shape (e.g. a 3×3 square) and so is hardwired into the algorithm. [10]

4.5.3.1. Grayscale Dilation and Erosion:

4.5.3.1.1. Dilation:

Gray-scale dilation of f by b, denoted by \( f \oplus b \), is defined as

\[
(f \oplus b)(s,t) = \max \{ f(s-x,t-y) + b(x,y) \mid (s-x, (t-y) \in D_f; (s,y) \in D_b) \} \quad (14)
\]

Where \( D_f \) and \( D_b \) are the domains of f and b, respectively. Similarly gray-scale erosion can be defined as an extension of binary erosion.

4.5.3.1.2. Erosion:

Gray-scale erosion, denoted by \( f \ominus b \), is defined as

\[
(f \ominus b)(s,t) = \min \{ f(s+x, t+y) - b(x,y) \mid (s+x, (t+y) \in D_f; (s,y) \in D_b) \} \quad (15)
\]

Where \( D_f \) and \( D_b \) are the domains of each image or function.

Specific concepts and operation procedures are already explained in binary morphology. Gray-scale dilation and erosion are duals with respect to function completion and reflection. That is, the relation between these can be expressed as

\[
(f \Theta b)^c (s,t) = (f^c \oplus \hat{b})(s,t) \quad (16)
\]

Where \( f^c = -f(x,y) \) and \( \hat{b} = b(-x,-y) \). The minimum operator will interrogate a neighborhood with a certain domain and select the smallest pixel value to become the output value. This has the effect of causing the bright areas of an image to shrink or erode.

Following fig.4.19 shows Original object pixels are in gray; pixels added through dilation are in black.

![Fig. 4.19: Illustration of dilation](a) Dilation with 4-Connected (b) Dilation with 8-Connected.)
4.5.3.2. Grayscale Opening and Closing:
The expressions for opening and closing of grayscale images have the same form as their binary counterparts. The opening of image $f$ by sub image (structuring element) $b$, denoted $f \circ b$, is

$$f \circ b = (f \Theta b) \oplus b$$

As in the binary case, opening is simply the erosion of $f$ by $b$, followed by a dilation of the result by $b$. Similarly, the closing of $f$ by $b$, denoted, $f \bullet b$, is

$$f \bullet b = (f \oplus b) \Theta b$$

The opening and closing for gray-scale images are duals with respect to complementation and reflection. That is,

$$(f \bullet b)^c = f^c \circ b$$

4.5.3.2.1. Properties of Opening and Closing:
Grayscale opening operation satisfies following properties:

(i) $f \circ b \downarrow f$

(ii) Iff $f \downarrow f z$, then $(f \circ b) \downarrow (f z \circ b)$

(iii) $(f \circ b) \circ b = (f \circ b)$

The notation $e \sqsubseteq r$ indicates that the domain of $e$ is a subset of the domain of $r$, and also that $e(x, y) \leq r(x, y)$ for any $(x, y)$ in the domain of $e$.

Similarly, Grayscale closing operation satisfies following properties:

(i) $f \bullet b \downarrow f$

(ii) Iff $f \downarrow f z$, then $(f \bullet b) \downarrow (f z \bullet b)$

(iii) $(f \bullet b) \bullet b = (f \bullet b)$

4.5.3.2.2. Effect of Opening:
1. The structuring element is rolled underside the surface of $f$.
2. All the peaks that are narrow with respect to the diameter of the structuring element will be reduced in amplitude and sharpness.
3. So, opening is used to remove small light details, while leaving the overall gray levels and larger bright features relatively undistributed.
4. The initial erosion removes the details, but it also darkens the image.
5. The subsequent dilation again increases the overall intensity removed by erosion.
4.5.3.2.3. Effect of Closing:
1. The structuring element is rolled on top of the surface of f.
2. Peaks essentially are left in their original form (assume that their separation at the
   narrowest points exceeds the diameter of the structuring element).
3. So, closing is used to remove small dark details, while leaving bright features
   relatively undistributed.
4. The initial dilation removes the dark details and brightens the image.
5. The subsequent erosion darkens the image without reintroducing the details
   totally removed by dilation. [1, 2]

4.5.4. Morphology for Binary Images:
The theoretical foundation of binary mathematical morphology is set theory. In binary
images, those points in the set are called the ‘foreground’ and those in the complement
set are called the ‘background’. Besides dealing with the usual set-theoretic operations of
union and intersection, morphology depends heavily on the translation operation. For
convenience, ‘U’ denotes the set-union, ‘∩’ denotes set-intersection and ‘+’ inside the set
notation refers to vector addition in the following equations. We need two general
definitions that are used extensively to extend morphological operations [1, 2].
Binary morphology can be seen as a special case of gray level morphology in which the
input image has only two gray levels at values 0 and 1. Erosion and dilation work (at least
conceptually) by translating the structuring element to various points in the input image,
and examining the intersection between the translated kernel coordinates and the input
image coordinate. For instance, in the case of erosion, the output coordinate set consists
of just those points to which the origin of the structuring element can be translated, while
the element still remains entirely ‘within’ the input image. Virtually all other
mathematical morphology operators can be defined in terms of combinations of erosion
and dilation along with set operators such as intersection and union. Some of the more
important are opening, closing and skeletonization. [10]

4.5.4.1. Binary Dilation & Erosion:
In fundamental to morphological processing the basic two operations is dilation and
erosion.
**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".

### 4.5.4.1.1. Dilation:
One of the simplest applications of dilation is for bridging gaps.

Dilation is one of the elementary operators of Mathematical Morphology, that is, a building block for a large class of operators, the dilation of \(A\) by \(B\), denoted \(A \oplus B\), which defines as follows

\[
A \oplus B = \{ z \mid (\hat{B})z \cap A \neq \emptyset \} \quad \text{(4.22)}
\]

Where \(A\) and \(B\) as sets in \(Z^2\), \(\hat{B}\) is reflection of set \(B\) about the origin and shifting this reflection by \(z\). Based on this interpolation,

\[
A \oplus B = \{ z \mid (\hat{B})z \cap A \neq \emptyset \} \quad \text{(4.23)}
\]

Fig. 4.20: (a) Set \(A\). (b) Structuring Element. (c) Dilation of \(A\) by \(B\).

### 4.5.4.1.2. Erosion:
The erosion is one of the elementary operators of Mathematical morphology, that is, it is one of the building blocks of a large class of operators, which can be defined as

\[
A \ominus B = \{ z \mid (B)z \subseteq A \} \quad \text{(4.24)}
\]

Fig. 4.21: Illustration of (a) Set \(A\) (b) Structuring Element (c) Erosion \(A\) by \(B\).
4.5.4.2. Binary Opening & Closing:
Generally opening is smoothes the contour of an object, breaks narrow isthmuses, and
eliminates thin protrusions. Closing also tends to smooth section of contours but, as
opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small
holes, and fills gaps in the contour.

4.5.4.2.1. Opening:
The opening of set $A$ by structuring element $B$, denoted $(A \circ B)$, is defined as a
composition of erosion and dilation by

\[ (A \circ B) = (A \Theta B) \oplus B \]

![Fig. 4.22: A simple illustration of morphological opening.](image)

4.5.4.2.2. Closing:
The dual operation to opening is closing, which is defined as dilation followed by an
erosion. The closing of set $A$ by structuring element $B$, denoted by $(A \bullet B)$, is defined as,

\[ (A \bullet B) = (A \Theta B) \Theta B \]

In other words, the closing of $A$ by $B$ is simply the dilation of $A$ by $B$, followed by the
erosion of the result by $B$.

The opening operation has a simple geometric interpretation as shown in fig.4.23. Suppose that we view the structuring element $B$ as a (flat) "rolling ball". The boundary of
$A$ as $B$ is rolled around the inside of this boundary. This geometric fitting property of the
opening operation leads to a set-theoretic formulation, which states that the opening of $A$
by $B$ is obtained by taking the union of all translates of $B$ that fit into $A$. That is, opening
can be expressed as a fitting process from the outside only corners that produce into the
image such that,

\[ (A \circ B) = \cup \{ ( B )_z \mid ( B )_z \subseteq A \} \]

Where, $\cup \{ \cdot \}$ denotes the union of all sets inside the braces.
Fig. 4.23: A simple illustration of the basic geometrical properties of morphological closing.

4.5.4.3. Properties of Opening and Closing:
The opening operation satisfies the following properties:

(i) \( A \ominus B \) is a subset (subimage) of \( A \)

(ii) If \( C \) is a subset of \( D \), then \( C \ominus B \) is a subset of \( D \ominus B \) \( \ldots \ldots \) \( (28) \)

(iii) \( (A \ominus B) \ominus B = A \ominus B \).  

Similarly, the closing satisfies the following properties:

(i) \( A \) is a subset (subimage) of \( A \bullet B \)

(ii) If \( C \) is a subset of \( D \), then \( C \bullet B \) is a subset of \( D \bullet B \) \( \ldots \ldots \) \( (29) \)

(iii) \( (A \bullet B) \bullet B = A \bullet B \).

Note that, from the condition (iii) in both cases that multiple openings or closings of a set have no effect after the operator has been applied once. [1, 2]

4.6. Hit-or-Miss Transformation:
This transforms are useful for shape detection. Another used to look for particular patterns of foreground and background pixels; it is very simple object recognition technique. All other morphological operations can be derived from it.

Input: 1. Binary Image. 2. Structuring Element, containing 0s and 1s.
The objective is to find the location of one of the shapes, says \( X \). let the origin of each shape be located at its center of gravity. Let, \( X \) is enclosed by a small window, \( W \). The local background of \( X \) with respect to \( W \) is defined as the set difference (\( W - X \)).

If \( B \) denotes the set composed of \( X \) and its background, the match (or set matches) of \( B \) in \( A \), denoted \( A \odot B \) as,

\[ A \odot B = (A \Theta X) \cap [A^{C} \Theta (W - X)] \ldots \ldots \ldots \ldots (4.30) \]
4.7. Image Segmentation by a Morphological Watershed:

Many others have mentioned the watershed segmentation is very useful to microarray images. But we have checked the results using such technology this is could not workout for microarray images result as shown in fig. 4.25 (a), (b) and table 4.3. In watershed segmentation the same spot region are grouped together & it consider a same region is an only single spot. The black spot region is a single region and white spots region is another single region. In table 4.3 shows the various statistical microarray image results such as mean, STD & Number of objects of original and resultant.

![Fig.4.24: Illustration of (a) Original image (b) Using watershed](image)

<table>
<thead>
<tr>
<th>Radius</th>
<th>Original Image</th>
<th>Resultant Image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>STD</td>
</tr>
<tr>
<td>Micro</td>
<td>43.8288</td>
<td>0.4926</td>
</tr>
<tr>
<td>Micro10</td>
<td>36.6497</td>
<td>0.4808</td>
</tr>
<tr>
<td>Micro25</td>
<td>63.3049</td>
<td>0.5227</td>
</tr>
<tr>
<td>Micro68</td>
<td>52.0790</td>
<td>0.5338</td>
</tr>
<tr>
<td>Micro91</td>
<td>7.9828</td>
<td>0.4210</td>
</tr>
<tr>
<td>Micro101</td>
<td>20.4692</td>
<td>0.3454</td>
</tr>
<tr>
<td>Micro105</td>
<td>98.4486</td>
<td>0.3428</td>
</tr>
</tbody>
</table>

**Summary:** In this chapter we have studied and applied various morphology-based techniques & the watershed to microarray images, but this method also gave weak results. Because microarray images is contains very complex spots & the problems of microarray is that the two spots are merged together, spot contains whole, etc. In watershed segmentation the same spot region are grouped together & it consider a same region is an only single spot. That's why we have suggested another morphology-based technique, which was helpful to such problems; this is given in detail in the next chapter 5.
References:
[5] Romulus Terebes, Olivier Lavialle, Pierre Baylou, Monica Borda, Ioan Nafornta, "ADAPTIVE MORPHOLOGICAL OPERATORS",
DIRECTIONAL
[13] Francisco Ortiz, "Gaussian noise removal by color morphology and polar color models", 