Scientific Journal of Impact Factor (SJIF): 4.72
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406
International Journal of Advance Engineering and Research Development
Volume 4, Issue 11, November -2017

Applications of Set Theory in Digital Image Processing

Suresha M

Department of Computer Science, Kuvempu University, Karnataka, India.

Abstract: *Set Theory is an important language and tool in digital image processing. It has fundamental ideas in computer science from theory to practice. Many ideas and methodologies in computer science are inspired by set theory. One among those ideas, morphological image processing is the application of set theory. Basic morphological algorithms and application in finger print identification has been discussed in this chapter.*

***Keywords**: Set Theory, Morphological Operations, Finger Print, Recognition.*

1. Morphological Image Processing

The word *morphology* commonly denotes a branch of biology that deals with the form and structure of animals and plants. The same word here in the context of mathematical morphology as a tool for extracting image components.

Here explained and illustrated several important concepts in mathematical morphology. Many of the ideas introduced here can be formulated in terms of n-dimensional Euclidean space, E^n. However, here discussing on binary images (Refer Appendix I for image types) whose elements are elements of Z^2.

The language of mathematical morphology is a set theory. As such, morphology offers a unified and powerful approach to numerous image processing problems. Sets in mathematical morphology represent objects in an image. For example, the set of all white pixels in a binary image is a complete morphological description of the image. In binary images, the sets in question are numbers of the *2-D* integer space Z^2, where each element of a set is tuple (*2-D* vector) whose coordinates are the (x, y) of a white (or black, depending on the convention) pixel in the image.

The concepts of set reflection and translation are used extensively in morphology. The reflection of a set B, denoted by B^R, is defined as

$$B^R = \{w \mid w = -b, \text{ for } b \in B\} \quad (1)$$

If B is the set of pixels (*2-D* points) representing an object in an image, the B^R is simply the set of points in B whose (x, y) coordinates have been replaced by $(-x, -y)$. Figures 1(a) and 1(b) show a simple set and its reflection.

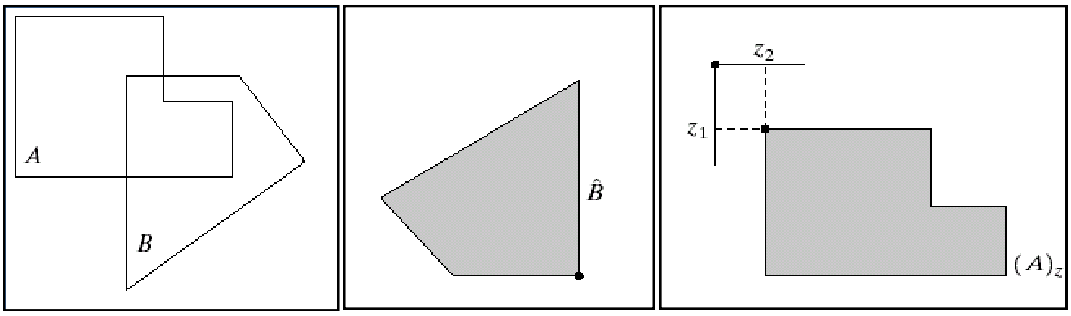


Figure 1: (a) Set A and B (b) Reflection of B (c) Translation of A

The translation of a set B by point $z = (z_1, z_2)$, denoted $(B)z$, is defined as

$$(B)_z = \{ c \mid c = b + z, \text{ for } b \in B\} \quad (2)$$

If B is the set of pixels representing an object in an image, then $(B)_z$ is the set of points in B whose (x, y) coordinates have been replaced by $(x + z_1, y + z_2)$. Figure 1(c) illustrates this concept using the set A from Fig. 1(a).

2. Structuring Elements

Set reflection and translation are employed extensively in morphology to formulate operations based on so-called structuring elements (SEs).Structuring elements can be any size and make any shape.However, for simplicity considered a rectangular structuring elements with their origin at the middle pixel.

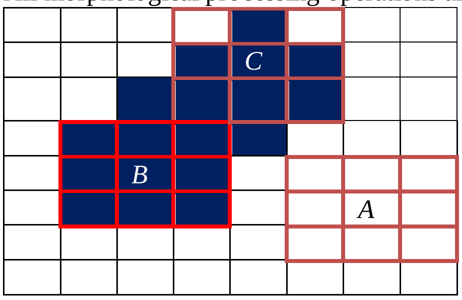
0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

1	1	1
1	1	1
1	1	1

0	1	0
1	1	1
0	1	0

Figure 2: Structuring Elements.

Fit: All *on pixels* in the structuring element cover *on pixels* in the image.
Hit: Any *on pixel* in the structuring element covers an *on pixel* in the image.
All morphological processing operations are based on these simple ideas.



Assuming the pixel value of object in this image is 1 and background pixel value is 0. The first structuring element in Figure3 is fit at *B* because the structuring element is a sub set of the object. The second structuring element in Figure 3 is fit at *C* because the structuring element is a sub set of the object. The first structuring element is not fit at *C* because structuring element is not a subset at *C* where as it is a hit. Both the structuring elements neither fit nor hit at *A*.

Figure 3: Hit and Fit

The structuring element is moved across every pixel in the original image to give a pixel in a new processed image. Some of the neighbors of pixel lie outside the digital image if its position is on the border of the image. Those pixels are considered as background. The value of this new pixel depends on the operation performed. There are two basic morphological operations: *erosion*and *dilation***.**

3. Erosion

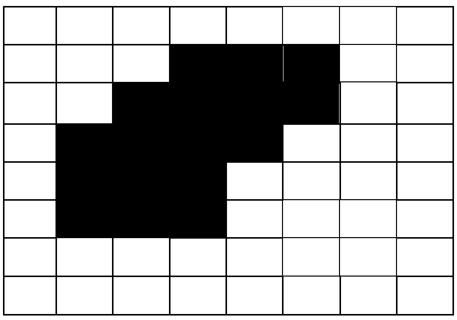
With *A* and *B* as sets in Z^2, the *erosion* of *A* by *B*, denoted $A \ominus B$, is defined as

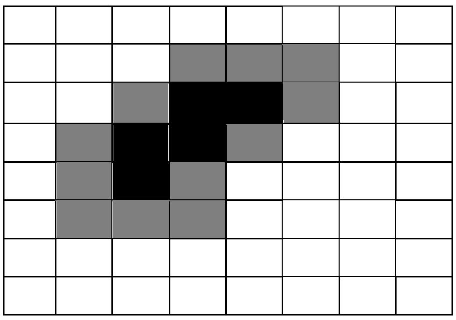
$$A \ominus B = \{z|(B)_z \subseteq A\} \quad (3)$$

This equation (3) indicates that the erosion of *A* by *B* is the set of all points *z* such that *B*, translated by *z*, is contained in *A*. The set *B* is a structuring element. Equation (3) is the mathematical formulation of the example in Figure4. The statement that *B* has to be contained in *A* is equivalent to *B* not sharing any common elements with the background; we can express erosion in the following equivalent form:

$$A \ominus B = \{z|(B)_z \cap A^c = \emptyset\} \quad (4)$$

Where A^c is the complement of *A* and $\emptyset$ is the empty set.





0	1	0
1	1	1
0	1	0

Figure 4: a) Original Image b) Processed Image (c) Structuring Element

Figure 4 shows an example of erosion. The elements of object are shown in figure 4(a) is shaded and the background is white. The border pixels of an object that are in gray color are removed from an object as shown in figure 4(b) by applying erosion with the structuring element 4(c).

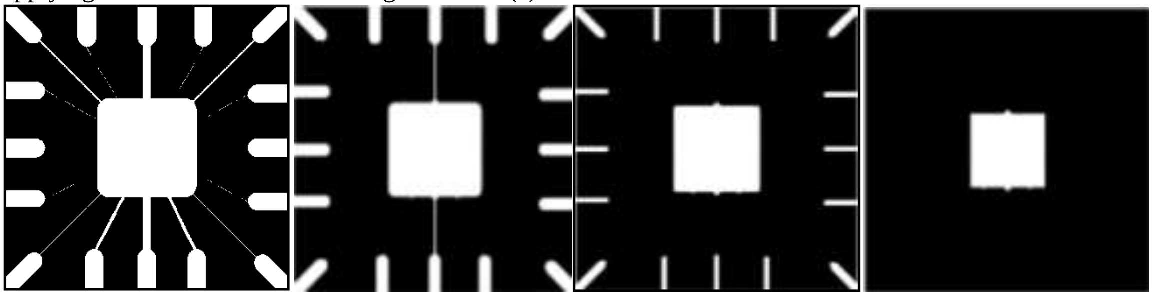


Figure 5:(a) Original Image (b)Eroding (a) with SE of size 5×5(c) Eroding (a) with SE of size 7×7 (d) Eroding (a) with SE of size 22×22 Image Courtesy: Rafael C.Gonzalez and Richard E. Woods

Suppose that we wish to remove the lines connecting the center region to the border pads in figure 5(a). Eroding the image with a square structuring element of size 5×5 whose components are all 1s removed most of the lines, as figure 5(b) shows. The reason the two vertical lines in the center were thinned but not removed completely is that their width is greater than 5 pixels. Changing structuring element size into 7× 7 and eroding the original image again did remove all the connecting lines, as figure 5(c) shows. Increasing the size of the structuring element even more would eliminate larger components. For example, the border pads can be removed with a structuring element of size 22 × 22, as figure 5(d) shows.

4. Dilation

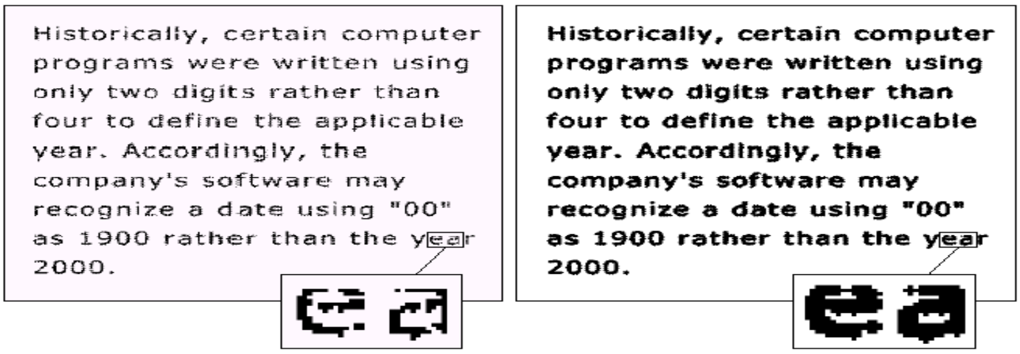
With *A* and *B* as sets in Z^2, the *dilation* of *A* by *B*, denoted $A \oplus B$, is defined as

$$A \oplus B = \{z | (B^R)z \cap A \neq \emptyset\} \tag{5}$$

This equation is based on reflecting *B* about its origin and shifting the reflection by z. The dilation of *A* by *B* then is the set of all displacements, *z*. Such that B^R and *A* overlap by at least one element. Based on this interpretation, equation (3) can be written equivalently as

$$A \oplus B = \{z | [(B^R)z \cap A] \subseteq A\} \tag{6}$$

We assume that *B* is a structuring element and *A* is the set (object in an image) to be dilated.
Unlike erosion, this is a shrinking or thinning operation, dilation “grows” or “thickens” objects in a binary image. The specific manner and extent of this thickening is controlled by the shape of the structuring element.
Example: One of the simplest applications of dilation is for bridging gaps. Figure6(a) shows the image with broken characters. Figure6(c) shows the structuring element that can be used to repair gaps. Figure6(b) shows the result of dilating the original image with this structuring element. The gaps were bridged.



0	1	0
1	1	1
0	1	0

Figure 6: (a) Original Image (b) Dilated Image (c) Structuring ElementImage Courtesy: Rafael C.Gonzalez and Richard E. Woods

5. Opening and Closing

As you have seen, dilation expands the components of an image and erosion shrinks them. In this section we discuss two other combined operations: *Opening* and *Closing*. *Opening* generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions. *Closing* also tends to smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thing gulfs, eliminates small holes, and fills gaps in the contour.
The opening of set *A* by structuring element *B*, denoted $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B \tag{7}$$

Thus, the opening *A* by *B* is the erosion of *A* by *B*, followed by a dilation of the result by *B*.
Similarly, the closing of set *A* by structuring element *B*, denoted $A \bullet B$, is defined as

$$A \bullet B = (A \oplus B) \ominus B \tag{8}$$

This says that the closing of *A* by *B* is simply the dilation of *A* by *B*, followed by the erosion of the result by *B*.

6. The Hit-Or-Miss Transformation

The morphological Hit-or-Miss transform is a basic tool for shape detection. Hit-or-Miss transformation is used to detect object of given shape and size in an image. Introducing this concept with the aid of figure7, which shows an image *A*(figure 7(a)) consisting of two shapes, denoted by *B* and *C*. The objective is to find one of the shapes, say *C*.
Hit-or-Miss Transformation is defined as:

$$A \circledast B = (A \ominus X) \cap [A^C \ominus (W-X)] \tag{9}$$

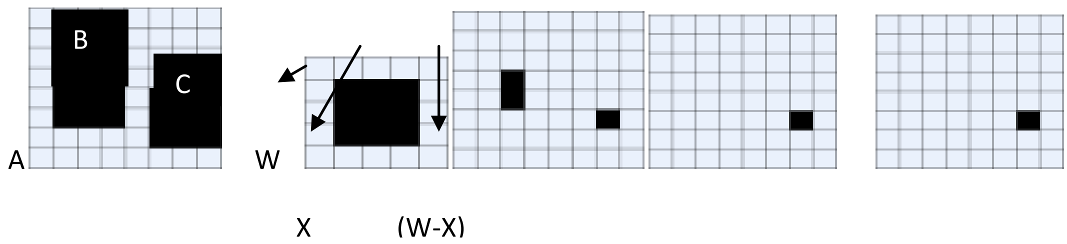


Figure 7:a) Set *A* b) SE c) $A \ominus X$ d) $A^C \ominus (W-X)$ e) $(A \ominus X) \cap [A^C \ominus (W-X)]$

Let the origin of each shape be located at its center of gravity. Let *X* be enclosed by a small window, *W*. The *local background* of *X* with respect to *W* is defined as the set difference (*W-X*), as shown in figure7(b). Figure7(c) shows the erosion of *A* by *X*. Recall that he erosion of *A*by *X* is the set of locations of the origin of *X*, such that *X* is completely contained in *A*. Interpreted in another way, $A \ominus X$ may be viewed geometrically as the set of all locations of the origin of *X* at which *X* found a match (hit) in *A*. Figure 7(d) shows the A^c eroded with (*W-X*). Figure7(e) shows the intersection of images (set of pixels) figure 7(c) and figure7(d). Figure7(d) shows the center pixel of the object *C*, we can say that object *C* is determined even though object *B* is the super set of object *C*.
We can generalize the notation somewhat by letting $B = (B_1, B_2)$, where B_1 is the set formed from elements of *B* associated with an object and B_2 is the set of elements of *B* associated with the corresponding background. From the preceding discussion, $B_1 = X$ and $B_2 = (W-X)$. With this notation, equation (9) becomes

$$A \circledast B = (A \ominus B_1) \cap [A^C \ominus B_2] \tag{10}$$

The reason for using a structuring element B_1 associated with objects and an element B_2 associated with the background is based on an assumed definition that two or more objects are distinct only if they form disjoint sets. This is guaranteed by requiring that each object have at least one pixel thick background around it. In some applications, we may be interested in detecting certain patterns (combinations) of 1s and 0s within a set, in which case a background is not required. In such instances, the hit-or-miss transform reduces to simple erosion. This simplified pattern detection scheme is used in some of the algorithms developed in the following section.
With the preceding discussion as foundation, we are now ready to consider some practical uses of morphology. When dealing with binary images, one of the principle applications of morphology is in extracting image components that are useful in the representation and description of the shape. In particular, we consider morphological algorithms for extracting boundaries, hole filling, thinning, and thickening.

7. 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,

$$\beta(A) = A - (A \ominus B) \tag{11}$$

Where*B* is a suitable structuring element.
Figure (8) illustrates the mechanism of boundary extraction. It shows a simple binary object, a structuring element *B*, and the results of using equation (11). Although the structuring element in figure8(b) is among the most frequently used, it is by no means unique.

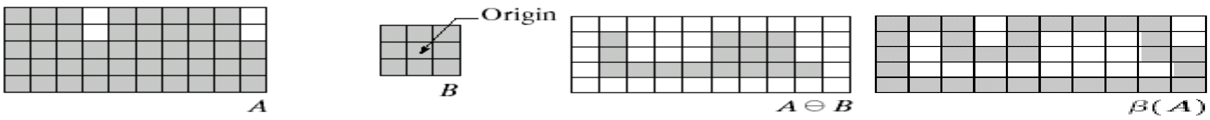


Figure 8. (a) Set *A*. (b) SE B. (c) *A* eroded by *B* (d) Boundary extracted from object in *A*

A simple image and the result of performing boundary extraction using a square 3*3 structuring element

Figure 9. (a)Original Image (b) Boundary ExtractedImage Courtesy: Rafael C.Gonzalez and Richard E. Woods

8. Hole Filling

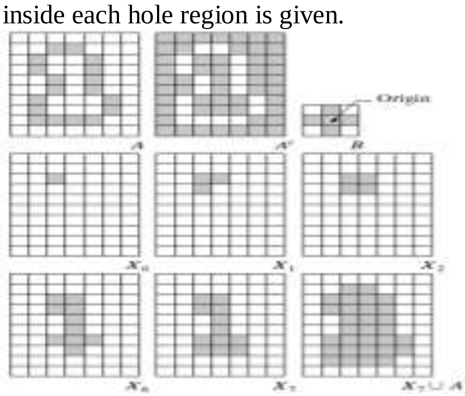
A hole may be defined as a background region surrounded by a connected border of foreground pixels. In this section, explained an algorithm based on set dilation, complementation, and intersection for filling holes in and image. Let *A* denote a set whose elements are 8-connected boundaries (Refer Appendix II), each boundary enclosing a background region (i.e., a hole). Given a point in each hole, the objective is to fill the holes with 1s.

We begin by forming an array, X_0, of 0s (the same size as the array containing *A*), except at the locations in X_0 corresponding to the given point in each hole, which we set to 1. Then, the following procedure fills all the holes with 1s:

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1, 2, 3, \ldots \qquad (12)$$

Where *B* is the symmetric structuring element in figure10(c). The algorithm terminates at iteration step *k* if $X_k = X_{k-1}$. The set X_k then contains all the filled holes. The set union of X_k and *A* contains all the filled holes and their boundaries.

The dilation in equation (12) would fill the entire area if left unchecked. However, the intersection at each step with A^c limits the result to inside the region of interest. This example only has one hole, the concept clearly applies to any finite number of holes, assuming that a point inside each hole region is given.



a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

Figure 10: Region Filling (a) Set *A*. (b) Complement of A. (c) Structuring Element B. (d) Initial point inside the boundary. (e) – (h) Various steps of Equation (12). (i) Final Result [Union of (a) and (h)].

9. Thinning

The thinning of a set *A* by a structuring element *B*, denoted $A \otimes B$, can be defined in terms of the hit-or-miss transform:

$$A \otimes B = A - (A \circledast B)$$

$$= A \cap (A \circledast B)^c \qquad (13)$$

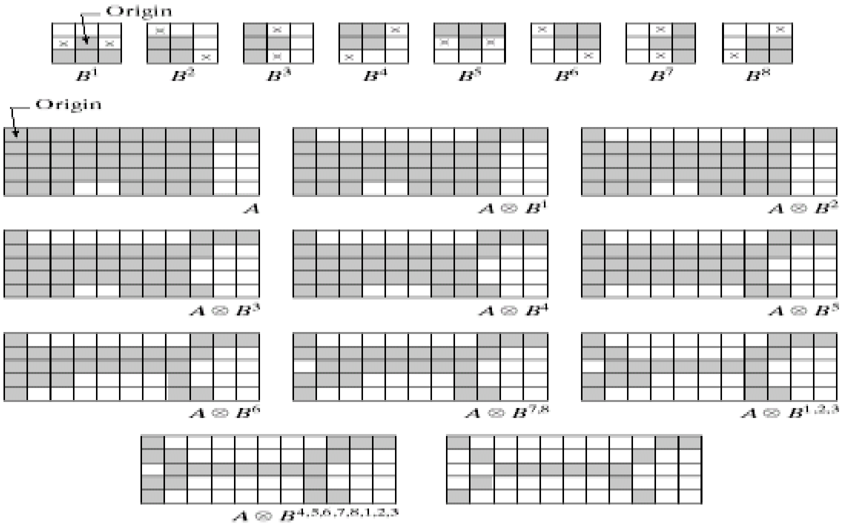
As in the previous section, we are interested only in pattern matching with the structuring elements, so no background operation is required in the hit-or-miss transform. A more useful expression for thinning *A* symmetrically is based on a sequence of structuring elements:

$$\{B\} = \{B_1, B_2, B_3, \ldots, B_n\} \qquad (14)$$

Where B_i is a rotated version of B_{i-1}. Using this concept, we now define thinning by a sequence of structuring elements as

$$A \otimes \{B\} = ((\ldots((A \otimes B_1) \otimes B_2) \ldots) \otimes B_n) \qquad (15)$$

The process is to thin *A* by one pass with B_1, then thin the result with one pass of B_2, and so on, until *A* is thinned with one pass of B_n. The entire process is repeated until no further changes occur. Each individual thinning pass is performed using equation (15).



	a	
b	*c*	*d*
e	*f*	*g*
h	*i*	*j*
k	*l*	

Figure 11: (a) SEs (b) Set A (c) Result of thinning with the first SE. (d) – (i) Results of thinning with next seven SEs. (j) Result of using the first SE again (k) Result after convergence. (l) Conversion to m-connectivity.

10. Thickening

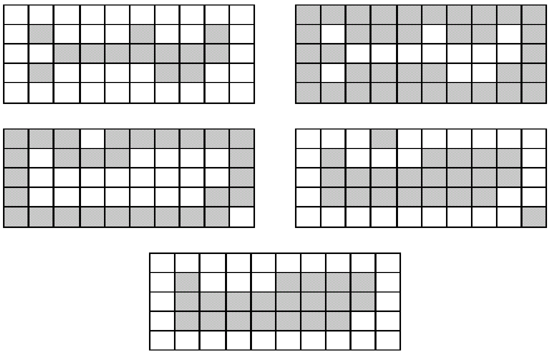
Thickening is the morphological dual of thinning and is defined by the expression

$$A \odot B = A \cup (A \circledast B) \tag{16}$$

Where B is a structuring element suitable for thickening. As in thinning, thickening can be defined as sequential operations:

$$A \odot \{B\} = ((\ldots((A \odot B_1) \odot B_2)\ldots) \odot B_n) \tag{17}$$

The structuring elements used for thickening have the same form as those shown in figure 10(a), but with all 1s and 0s interchanged. However, a separate algorithm for thickening is seldom used in practice. Instead, the usual procedure is to thin the background of the set in question and then complement the result. In other words, to thicken a set A, we form $C = A^c$, thin C, and then form C^c. Figure 12 illustrate this procedure.

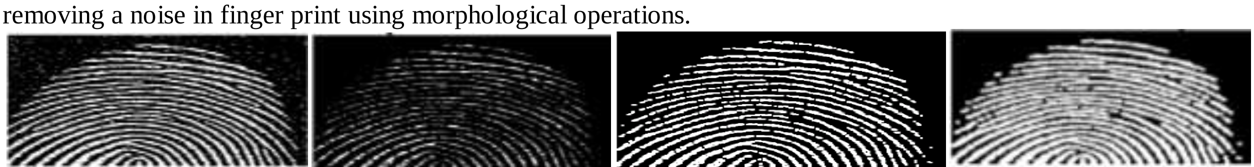


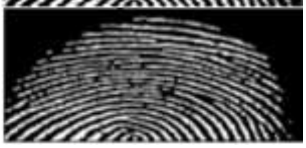
a *b*
c *d*
e

Figure 12 : Thickening Operation (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected (isolated) points.

11. Morphological Operationsin Finger Print Identification.

The image 13(a) contains noise (unwanted information) outside the finger print. Figure 13 illustrates the process of removing a noise in finger print using morphological operations.





1	1	1
1	1	1
1	1	1

a *b*
c *d*
e *f*

Figure 13: (a) Set A(Noisy Image)(b) $A \ominus B$(c) $(A \ominus B) \oplus B = A \circ B$ (d) $(A \circ B) \oplus B$(e)$[(A \circ B) \oplus B] \ominus B = (A \circ B) \bullet B$(f) *SE*

Figure 13 shows a step-by-step sequence of the operations. Figure 13(b) is the result of eroding A with the structuring element 13(c). The background noise was completely eliminated in the erosion stage of opening because in this case all noise components are smaller than the structuring element. The size of the noise elements (dark spots) contained within the fingerprint actually increased in size. The reason is that these elements are inner boundaries that increase in size as the object is eroded. This enlarge is countered by performing dilation on figure 13(b). Figure 13(c) shows the result. The noise components contained in the fingerprint were reduced in size of deleted completely. The two operations just described constitute the opening of A by B. The net effect of opening was to eliminate virtually all noise components in both the background and the fingerprint itself as shown in figure 13(c). The new gaps between the fingerprint ridges were created. In order to solve this undesirable effect performed dilation on the opening, as shown in figures 13(d). Most of the breaks were restored, but the ridges were thickened, a condition that can be remedied by erosion. The result shown in figure 13(e) constitutes the closing of the opening of figure 13(c). This final result is remarkably clean of noise specks, but it has the disadvantage that some of the print ridges were not fully repaired, and thus contain breaks.

APPENDIX I:

1. Digital Image Fundamentals

Digital images are composed of pixels (short for picture elements). Each pixel represents a color (or gray level for black and white photos) at a single point in the image, so a pixel is like a tiny dot of a particular color. By measuring the color of an image at a large number of points, we can create a digital approximation of the image from which a copy of the original can be reconstructed. Pixels are a little like grain particles in a conventional photographic image, but arranged in a regular pattern of rows and columns and store information somewhat differently. A digital image is a rectangular array of pixels sometimes called a bitmap.

Two main application areas of digital image processing are:

1. Improvement of pictorial information for human interpretation.
2. Processing of image data for storage, transmission, and representation for autonomous machine perception.

2. Representation of Digital Images

Digital images are represented in two dimensional array, $f(x, y)$ containing M rows and N columns. Where (x, y) are discrete coordinates. For notational clarity and convenience, we use inter values for these discrete coordinates; $x = 0, 1, 2. . .M\text{-}1$ and $y = 0, 1, 2. . .N - 1$.

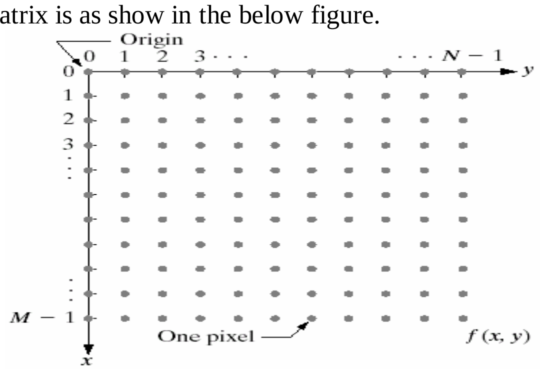
The representation of an $M\ X\ N$ numerical array as

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & . & . & . & f(0,N-1) \\ f(1,0) & f(1,1) & . & . & . & f(1,N-1) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ f(M-1,0) & f(M-1,1) & . & . & . & f(M-1,N-1) \end{bmatrix}$$

In some discussion it is advantageous to use a more traditional matrix notation to denote a digital image and its elements.

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & . & . & . & a_{0,N-1} \\ a_{1,0} & a_{1,1} & . & . & . & a_{1,N-1} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{M-1,0} & a_{M-1,1} & . & . & . & a_{M-1,N-1} \end{bmatrix}$$

The pictorial representation of a matrix is as show in the below figure.



3. Types of Digital Images

For photographic purposes, there are two important types of digital images—*color* and *black and white*. *Color* images are made up of colour pixels while *black and white* images are made of pixels in different shades of gray.

4. Gray Scale Images

A *gray scale image* is made up of pixels each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different gray levels. Since the eye can barely distinguish about 200 different gray levels. Figure *x* illustrates the different shades from black to white, we can see different shades between black and white in the below figure

Assuming 256 gray levels, each pixel can be stored in a single byte of memory.

5. Color Images

A color image is made up of pixels each of which holds three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue (sometimes referred to as *RGB*) are the primary colours for mixing light—these so-called additive primary colours are different from the subtractive primary colours used for mixing paints (cyan, magenta, and yellow). Any color can be created by mixing the correct amounts of red, green, and blue light. Assuming 256 levels for each primary, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colours.

6. Binary or Bi-level Images

Binary images use only a single bit to represent each pixel. Since a bit can only exist in two states—on or off, every pixel in a binary image must be one of two colours, usually black or white.

Gray scale and binary images can be represented with one two dimensional matrix where as color (*RGB*) image can be represented with three two dimensional matrices, one matrix for each component, i.e., for *R* component one two dimensional matrix, for *G* component one two dimensional matrix, and for *B* component one two dimensional matrix.

APPENDIX II:

Basic Relationships between Pixels called neighbours of a pixel and adjacency of pixels are discussed in this section.

1. Neighbors of a Pixel

A pixel p at coordinates (x,y) has four horizontal and vertical neighbors whose coordinates are given by

$$(x + 1, y), (x - 1 , y), (x , y + 1), (x , y - 1)$$

This set of pixels, called the 4-*neighbors* of p, is denoted by $N_4(p)$. Each pixel is a unit distance from (x, y).

The four *diagonals* neighbors of p have coordinates

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x -1, y - 1)$$

and are denoted by $N_D(p)$. The 8 neighbors of a pixel p is given by union of for neighbors and diagonal neighbors, denoted by $N_8(p)$. The coordinators are

$$(x+ 1, y), (x - 1, y), (x, y + 1), (x , y - 1), (x + 1, y + 1), (x + 1 , y - 1), (x - 1 , y + 1),(x -1, y - 1)$$

Neighbor of a pixel is illustrated in the below given figure.

$(x - 1, y - 1)$	$(x, y - 1)$	$(x + 1, y - 1)$
$(x - 1, y)$	(x, y)	$(x + 1, y)$
$(x - 1, y + 1)$	$(x, y + 1)$	$(x + 1, y + 1)$

2. Adjacencyof Pixels

Let V be the set of intensity values used to define adjacency. In a binary image, $V = \{1\}$ if we are referring to adjacency of pixels with value 1. In a gray-scale image, the idea is same, but set V typically contains more elements. For example, in the adjacency of pixels with a range of possible intensity values 0 to 255, set V could be any subset of these 256 values. We consider three types of adjacency:

(a) *4-adjacency*: Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$.

(b) *8-adjacency*: Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(p)$.

(c) *m-adjacency*: Two pixels p and q with values from V are m-adjacent if

(i) q is in $N_4(p)$, or

(ii) q is in $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V. (in other words $N_4(p) \cap N_4(q) = \emptyset$).

CONCLUSION

Set theory is a prerequisite for morphological operations on digital images. Morphological operations are proposed to extraction of boundary of objects, extraction of connected components, to fill the objects, to find out a convex hull of the object, thinning and thickening of the object, extraction of skeleton of an object etc. In this paper addressed basics of digital image and image processing to understand to the beginners and finger print identification using morphological operations were discussed.

REFERENCES

[1] Anil K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall, 1989.
[2] Glynn Winskel, "Set Theory for Computer Science", 2010.
[3] Jose Meseguer, "Set Theory in Computer Science - A Gentle Introduction to Mathematical Modeling I", 2011.
[4] Rafael C.Gonzalez and Richard E. Woods, "Digital Image Processing," *PPH*, 2008
[5] Rafael C.Gonzalez and Richard E. Woods, "Digital Image Processing using MATLAB," *PPH*, 2008