CSE585/EE555: Digital Image Processing II
Computer Project # 2
The Morphological Skeleton and Shape Analysis

Saptarashmi Bandyopadhyay ; Bharadwaj Ravichandran

02/22/2019

---

1 Objectives

The main objective of this project is to use basic mathematical morphology operations like erosion, dilation, and opening to implement the morphological skeleton algorithm and also to perform shape analysis using Size Distribution and Pattern Spectrum. Pattern Recognition has been done on a set of solid objects using the Pattern Spectrum.

2 Methods and Code Organization

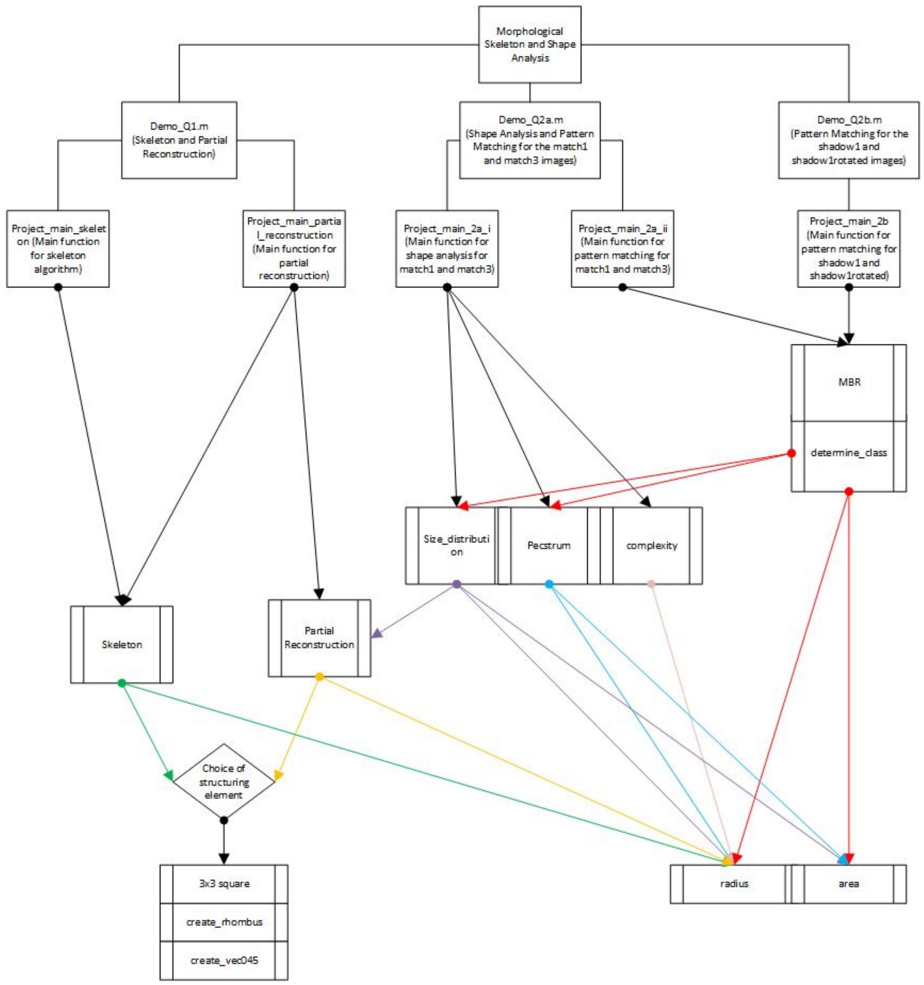


Figure 1: Overall Block diagram and code flow

Figure 1 shows the overall code flow of the project. The demo and main files setup the various parameters and functions to be called. The colored lines represent the function call to the different subroutines of the project. An arrow from Box A to Box B means that the function in Box A is calling the function in Box B. The mathematical formulae, the input and output parameters of the functions are discussed below.

The procedure to run the project codes in MATLAB has been specified clearly in the README.md file in the Submission folder of Project2.

2.1 Morphological Skeleton Algorithm

The Morphological Skeleton (Medial Axis) Algorithm is implemented to obtain a "stick-figure" representation of an object, i.e., the central axes (centerlines) of the object. It is also called as the "grass-fire" transform. Skeleton Construction can be explained in a naive manner as follows:

1. Set grassfire on object boundary δX.
2. Fire wavefronts meet at certain points.
3. When fire totally extinguished, the set of quench points $sk(X)$ gives the Skeleton (medial axis).

The mathematical description of the Morphological Skeleton Algorithm is as follows:

Consider $E = \mathbb{Z}^2$ (2-D discrete space).

$$sk(X) = \bigcup_{0 \leq n \leq N} S_n(X) \; where:$$

$$S_n(X) = (X \ominus nB) - (X \ominus nB)_B$$

$$(n^{th} \text{ skeletal component})$$

$$nB = (B \oplus B \oplus ... \oplus B), \; n = 0, 1, 2, ..., N$$

$$(\text{structuring element of radius } n)$$

$$N = max\{k : X \ominus kB \neq \phi\}$$

NB = largest homothetic of B that fits inside X.
The radius of an image has been executed by the radius.m code in MATLAB.

The pseudocode of radius.m is as follows:

Function: radius.m (X, B)

Objective: To find the radius N which is defined as $N = max\{k : X \ominus kB \neq \phi\}$

Input Parameters:

X: Binary Image
B: Structuring Element

Returned Result:

N: The maximum radius when dilated with the structuring element B (NB) produces a non-empty set on eroding the image X

Processing Flow:

```
r = 1
Until input image becomes a matrix of zeros
    Erode image X with structuring element rB
    r ← r + 1;
End
```

Restrictions/Notes:

This function requires a binary image and structuring element as input.

The following functions are called:

erosion_2.m

The following functions are calling:

skeleton.m, partial_reconstruction.m, size_distribution.m, pecstrum.m, complexity.m, determine_class.m

Assume $B = B^s$. The 0^{th} order skeletal component is given by:

$$S_0(X) = X - X_B$$

The skeleton of an image has been executed by the skeleton.m code in MATLAB.

The pseudocode of skeleton.m is as follows:

Function: skeleton.m (X, Bnum)

Objective: To generate the morphological skeleton of an input binary image based on the given structuring element.

Input Parameters:

X: Binary Image
Bnum: Structuring element choice:
(1) 3x3 square (2) Rhombus (3) Vec045

Returned Result:

skX: Skeleton of the input binary image X

Processing Flow:

```
Based on user input choose the structuring element
    (1) 3x3 square
    (2) Rhombus
    (3) Vec045
Find the maximum radius N
Let rStruct be the row_size and cStruct be column size of B
temp ← 0
Until N
    B ← B(rStruct + temp)
    Perform skeletonization as per the algorithm explained above
    if struct element is vec045
        temp ← temp + 1
    else
        temp ← temp + 2
End
```

Restrictions/Notes:

This function requires a binary image and structuring element as input

The following functions are called:

radius, erosion_2, dilation_2, create_rhombus, create_vec045

The following functions are calling:

Project_main_skeleton.m, Project_main_partial_reconstruction.m

The skeletonization function $skf(x)$ is given by:

$$skf(x) = \begin{cases} n, & \text{if } x \ \epsilon \ S_n(x) \\ \text{undefined}, & \text{otherwise} \end{cases}$$

To reconstruct X:

$$X = \bigcup_{0 \leq n \leq N} [S_n(X) \oplus nB]$$

To partially reconstruct X:

$$X = \bigcup_{k \leq n \leq N} [S_n(X) \oplus nB], \ k \geq 0$$

The partial reconstruction of an image has been executed by the partial_reconstruction.m code in MATLAB.

The pseudocode of partial_reconstruction.m is as follows:

Function: partial_reconstruction.m (X, Bnum,k)

Objective: To partially reconstruct the input image from the k^{th} Skeleton subset using a structuring element.

Input Parameters:

X: Input Binary Image
Bnum: Structuring element choice:
(1) 3x3 square (2) Rhombus (3) Vec045
k - starting from the k^{th} skeleton subset

Returned Result:

XkB: Partial Reconstructed image from the k^{th} skeleton subset.

Processing Flow:

```
Based on user input choose the structuring element
    (1) 3x3 square
    (2) Rhombus
    (3) Vec045
Find the maximum radius N
Let rStruct be the row_size and cStruct be column size of B
if k is 0
    XkB = X
    set k ← 1

temp ← 2k − 2
Until N
    B ← B(rStruct + temp)
    Perform skeletonization as per the algorithm explained above
    Create the skeleton subsets from the result of skeletonization
    if struct element is vec045
        temp ← temp + 1
    else
        temp ← temp + 2
End
```

Restrictions/Notes:

This function requires a binary image, structuring element and a k value.

The following functions are called:

radius, erosion_2, dilation_2, create_rhombus, create_vec045

The following functions are calling:

Project_main_partial_reconstruction.m, size_distribution.m

2.2 Minimum Bounding Rectangle

The Minimum Bounding Rectangle is a key component for pattern matching. Using the regionprops and the bwlabel functions of MATLAB, we were able to crop out the individual objects from the input images.The minimum bounding rectangle of an image has been executed by the MBR.m code in MATLAB.

2.3 Size Distribution

Size Distribution(Granulometry) of an object X provides the information about the area of X covered by a structuring element rB (where r is the radius). The mathematical formula for size distribution is given as:

$$U(r) = m(X_{rB}),\ r \epsilon R$$

where,

X_{rB} is opening of X by rB

$m(X_{rB})$ is area of X_{rB}

The size distribution $U(r)$ is a monotonically decreasing function.

Calculation of size_distribution of an image has been executed by the size_distribution.m code in MATLAB. There are two files of size_distribution, one which displays the graphs for shape analysis in Q.2a i and another which does not display the graphs for shape analysis in Q.2a ii and Q.2b

The pseudocode of size_distribution.m is as follows:

Function: size_distribution.m (X, B)

Objective: To generate the size distribution U of an input binary image object and plot it versus the radius.

Input Parameters:

X: Binary Image Object
B: Structuring Element

Returned Result:

U: Size Distribution

Processing Flow:

```
Get maximum radius N
Initialise U(1) to area of $X_{0B}$
Intialise R(1) to 0
temp ← 1
r ← 2

Until N+1
    R(r) ← temp
    $X_{kB}$ ← partial reconstruction with k = R(r)
    U(r) = area of $X_{kB}$
    temp ← temp + 1
End
```

Restrictions/Notes:

This function requires a binary input image and structuring element.

The following functions are called:

partial_reconstruction.m, area.m, radius.m

The following functions are calling:

Project_main_2a_i.m, determine_class.m

2.4 Pattern Spectrum (Pecstrum)

Pattern Spectrum or Pecstrum is the normalized negative derivative of the size distribution. The Pecstrum of a shape X is given as, for $r > 0$,

$$f(r) = \frac{-\frac{du(r)}{dr}}{m(X)}$$

where, $m(X) = area\{X\}$ (for normalization)

The pecstrum $f(r)$ gives the amount of area in X per component. The digital version of pecstrum is given as follows:

$$f(n) = \frac{m(X_{nB}) - m(X_{(n+1)B})}{m(X)}$$

Also, the following assumption has been considered in calculating the pecstrum value for

$$N = max\{k : X \ominus kB \neq \phi\}$$
$$f(N) = \frac{m(X_{NB})}{m(X)}$$
$$m(X_{(N+1)B}) = 0;$$

This value of f(N) has been displayed in the pecstrum graph and has been used in complexity calculation as per the formula. However, the value of f(N) has not been used for distance calculation in pecstral analysis in accordance to the formula.

Calculation of pecstrum of an image has been executed by the pecstrum.m code in MATLAB. There are two files of pecstrum, one which displays the graphs for shape analysis in Q.2a i and another which does not display the graphs for shape analysis in Q.2a ii and Q.2b

The pseudocode of pecstrum.m is as follows:

Function: pecstrum.m (X, B, U)

Objective: To calculate the pattern spectrum of object X from the size distribution U.

Input Parameters:

X: Binary image Object B: Structuring Element U: size distribution

Returned Result:

f: Pattern spectrum of object X

Processing Flow:

```
Get maximum radius N
r ← 1
until N
    f(r) ← U(r) − U(r+1)
    R(r) ← r − 1
End
f(N+1) ← U(N+1)/area(X)
R(N+1) ← N
```

Restrictions/Notes:

This function requires a binary image object and stucturing element.

The following functions are called:

radius.m, area.m

The following functions are calling:

Project_main_2a_i.m, determine_class.m

2.5 Shape Complexity

The measure of Shape Complexity is given as:

$$H(X|B) = -\sum_{i=0}^{N} f(i)\ log f(i)$$

Calculation of the shape complexity of an image has been executed by the complexity.m code in MATLAB.

The pseudocode of complexity.m is as follows:

Function: complexity.m (X, B, f)

Objective: To compute the shape complexity of an image X with structuring element B and pecstrum f.

Input Parameters:

X: Input Binary Image
B: Structuring Element
f: pecstrum of X with structuring element B

Returned Result:

H: Value of the shape complexity of X.

Processing Flow:

```
Find the maximum radius N for image X with structuring element B.

Until N+1 (since MATLAB indices start from 1)
      Compute temp -> f(i)*log2 f(i)
      Add temp values to H
End

H = -H
```

Restrictions/Notes:

This function requires a binary image, structuring element and an array of pecstrum values.

The following functions is called: radius

The following functions are calling:

Project_main_2a_i.m and Project_main_2b.m,

2.6 Pattern Matching

For Pattern Matching, a weighted Euclidean distance function is used in order to match the pecstrum of a test object $f(n)$ to the pecstrum of the reference object

$f_{R_i}(n)$. The mathematical formula for the Euclidean distance function for determining the class R_i is given as:

$$d_i = \left[\sum_{n=0}^{N-1} C_n \left(f(n) - f_{R_i}(n) \right)^2 \right]^{1/2}$$

Next, we need to find the argument i that minimizes d_i over all classes $i = 0, 1, 2, ..., N-1$, that is,

$$arg\{min_i \ d_i\} = i_{min}$$

The $\mathrm{C_n}$ has been selected as the absolute value of the difference of the area of the reference object to the area of the test object for values of n from 0 to N-1. The selection is done to reduce the impact of large differences of pecstrum values in di for the test object and its matching reference object. The area of the reference and test objects are already being calculated for pecstrum. Let X denote an image.

$$C_n(R_i) = |area(X_{R_i}) - area(X_{unknown})|$$

$\mathrm{X_{R_i}}$ is the i-th object in the reference image while $\mathrm{X_{unknown}}$ is an object in the test image.
Pattern matching by pecstral analysis has been executed by the determine_class.m code in MATLAB.

The pseudocode of determine_class.m is as follows:

Function: determine_class.m (X, X_test,B)

Objective: To match the X_test object to an object in X using the same structuring element B on both of them.

Input Parameters:

X: Cell Array of Objects in Reference Image
X_test: An object in the Test Image
B: Structuring element (3x3 square in the Project)

Returned Result:

di : Array of distance values, calculated by pecstral analysis corresponding to all the objects in the reference image for a certain object in test image
min_i: Index of the object in the reference image with minimum distance value

Processing Flow:

```
Find the size of cell array X in len_X.

Until len_X
    Find radius of each reference object X(i) in N(i).
    Find size_distribution of each X(i) in X.
    Find pecstrum of each X(i) in X and store it in f_known(i).
End

Find radius of X_test in N_unknown.
Find size_distribution of X_test.
Find pecstrum of X_test and store it in f_unknown.

Until len_X
    Find maximum of N(i) and N_unknown and store it in N_big
    Store the other radius in N_small
    Until N_small
       di_sq -> square(f_unknown(X\_test) - fknown(X(i)))
       Multiply di_sq with the absolute difference of their areas.
       Calculate the sum of temp till N_small
    End
    Repeat the process from N_small+1 to N_big
    Instead of difference, consider the pecstrum of X with radius N_big
    di -> square root(di_sq)
End
Find i for minimum of di
```

Restrictions/Notes:

This function requires a cell array of binary images which are objects in the reference image, a binary test image and a structuring element.

The following functions are called: radius, size_distribution_2, pecstrum_2, create_rhombus, create_vec045

The following function is calling:

Project_main_2b

3 Results

3.1 Structuring elements

In table 1, the 3x3 square structuring element has a grid because we cropped it from the pixel viewer of MATLAB's imtool window.

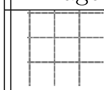
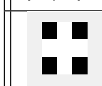
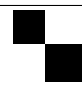
Image	Name
	3x3 Square
	Rhombus
	Vec045

Table 1: Table containing the structuring elements

Since the rhombus is drawn within a 3x3 matrix, it looks like a star. To have a better representation of a rhombus, figure 2 shows a rhombus in a 9x9 matrix.

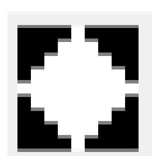


Figure 2: 9x9 Rhombus

3.2 Results of the Morphological Skeleton Algorithm and Partial Reconstruction

In this subsection, we will be showing the results of the morphological skeleton algorithm and partial reconstruction applied on two images - "penn256.gif" and "bear.gif". Also, there is something important to note before looking at the results.

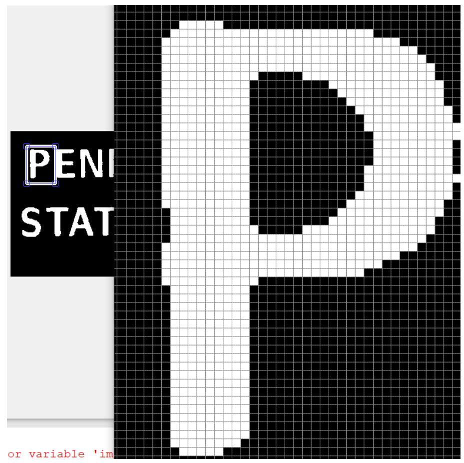


Figure 3: A closer look at the number of pixels at the top region of P

Figure 3 shows that the top part of the letter P has a pixel width varying from 5 to 7 pixels. This is one important infomration to know in order to understand the results of partial reconstruction.

3.2.1 Using 3x3 Square Structuring Element

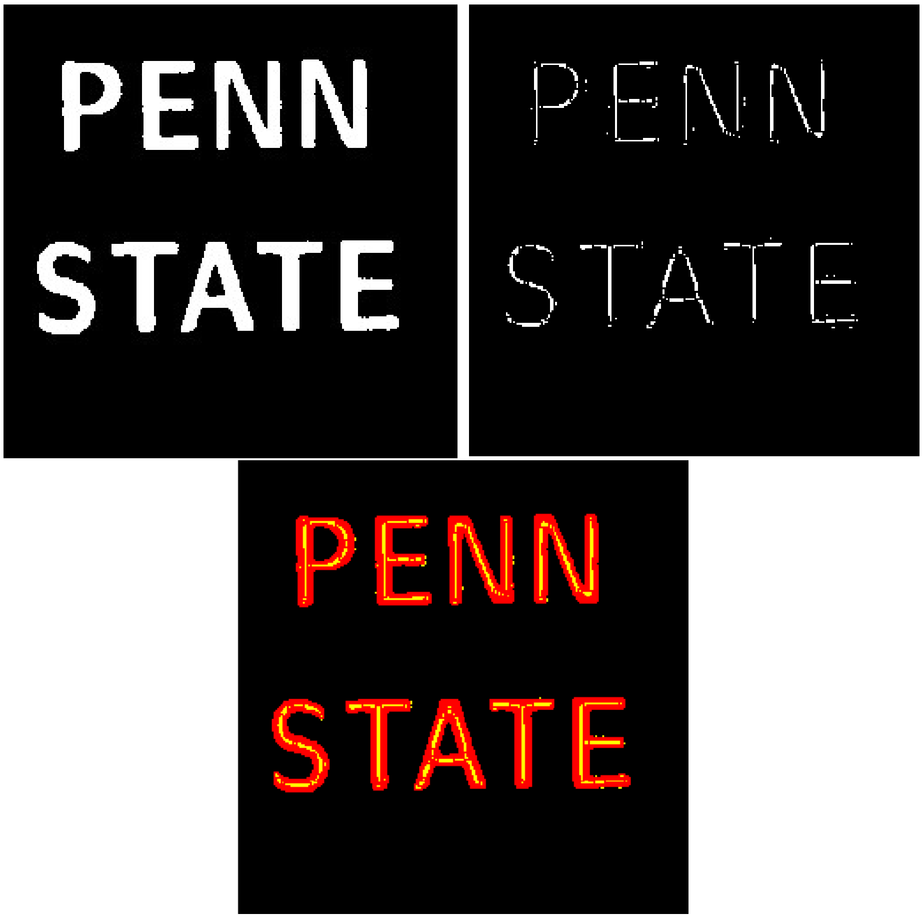


Figure 4: (a) Original Image (Top-Left) (b) Skeleton Image (Top-Right) (c) Skeleton superimposed on Original Image (Bottom-Center)

Here, we are using a 3x3 square as the structuring element. Figure 4 shows the results of the skeleton algorithm using a 3x3 square structuring element on the "penn256" image. Figure 4 (a) is the Original input binary image. Figure 4 (b) shows the output from the morphological skeleton algorithm and finally, figure 4 (c) shows the skeleton (yellow) superimposed on the original image (red).

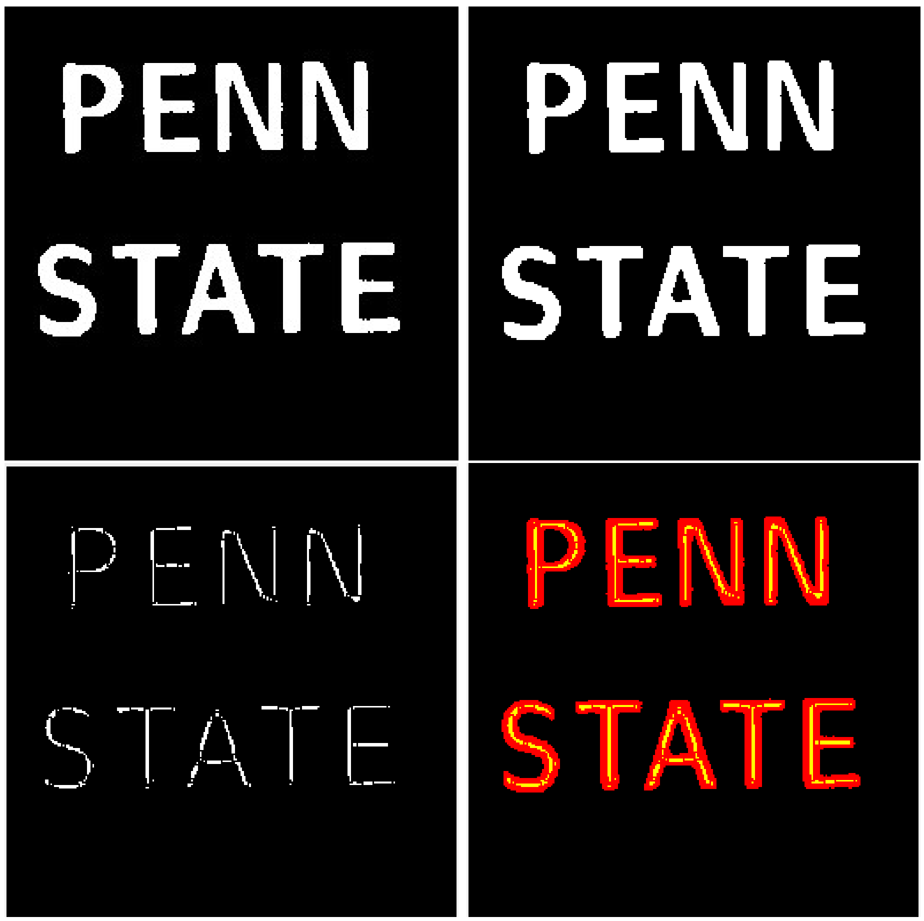


Figure 5: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 2 (Top-Right) (c) Skeleton of partial reconstructed image with k=2 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 5 shows the results of the partial reconstruction algorithm using a 3x3 square structuring element on the "penn256" image. Figure 5 (a) is the Original input binary image. Figure 5 (b) shows the partial reconstructed image with k = 2. Figure 5 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 5 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

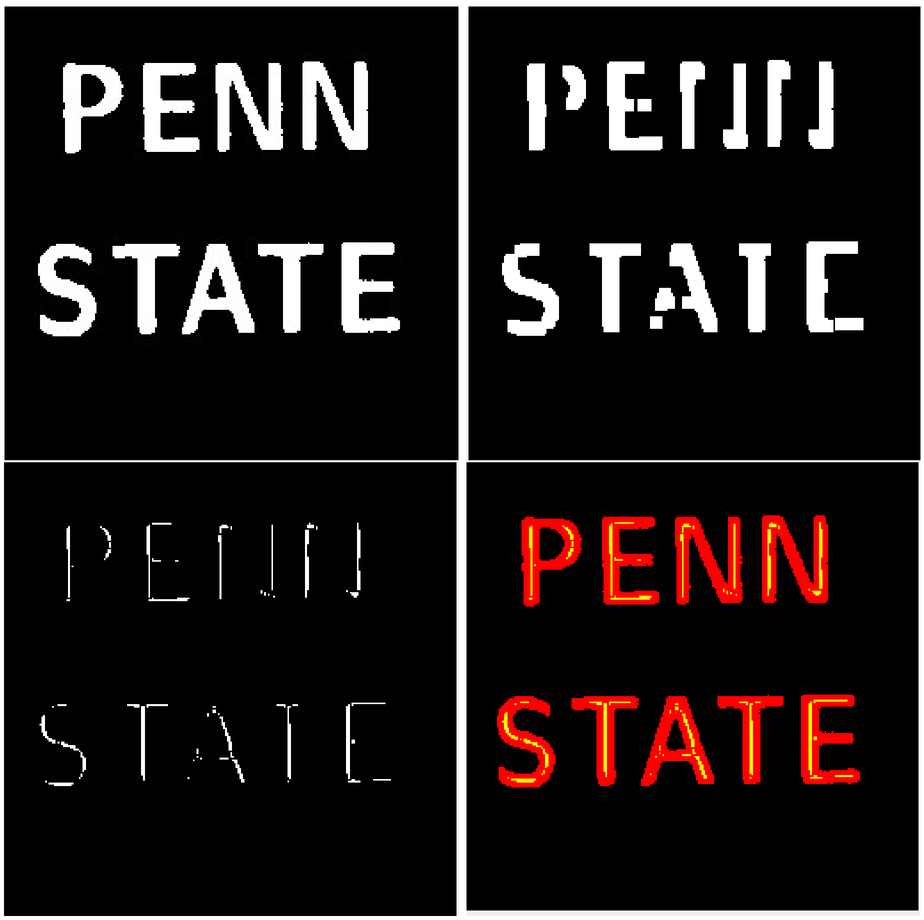


Figure 6: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 3 (Top-Right) (c) Skeleton of partial reconstructed image with k=3 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 6 shows the results of the partial reconstruction algorithm using a 3x3 square structuring element on the "penn256" image. Figure 6 (a) is the Original input binary image. Figure 6 (b) shows the partial reconstructed image with k = 3. Figure 6 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 6 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

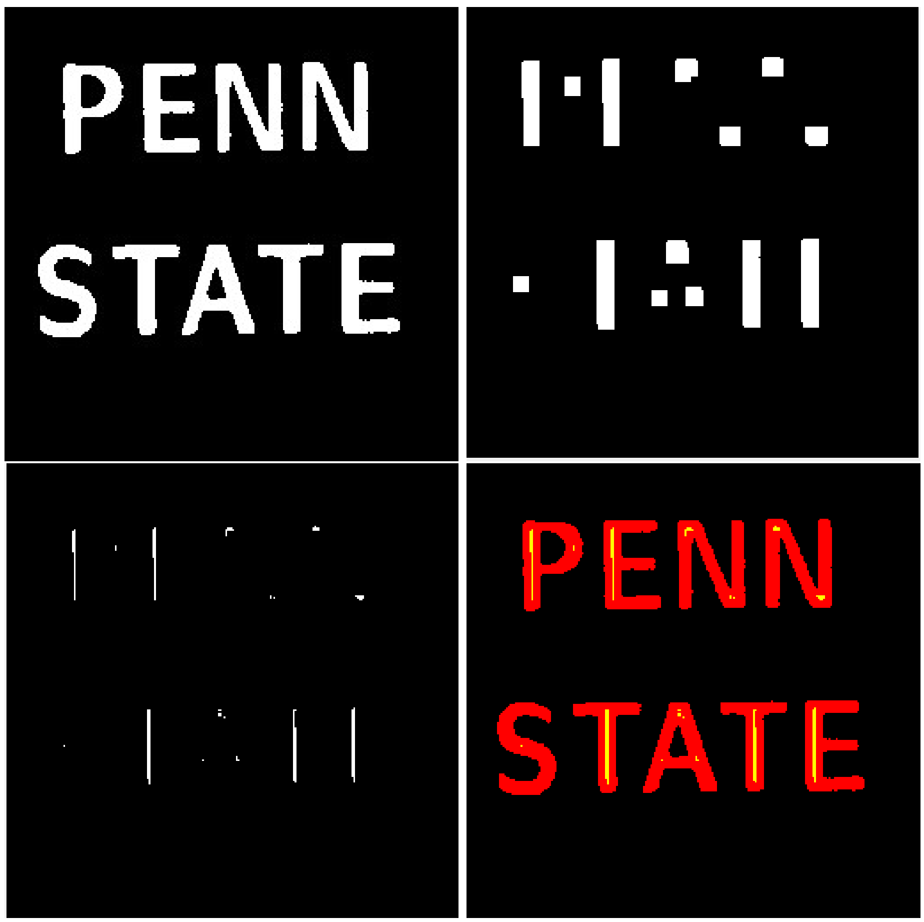


Figure 7: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 4 (Top-Right) (c) Skeleton of partial reconstructed image with k=4 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 7 shows the results of the partial reconstruction algorithm using a 3x3 square structuring element on the "penn256" image. Figure 7 (a) is the Original input binary image. Figure 7 (b) shows the partial reconstructed image with k = 4. Figure 7 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 7 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

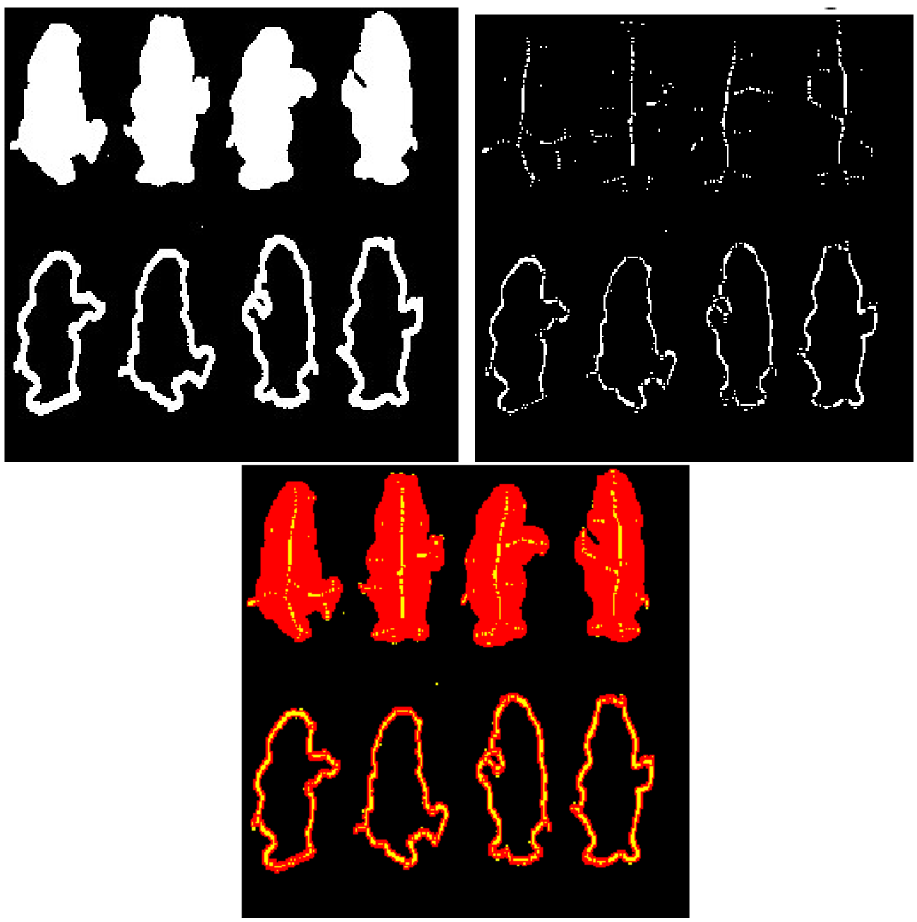


Figure 8: (a) Original Image (Top-Left) (b) Skeleton Image (Top-Right) (c) Skeleton superimposed on Original Image (Bottom-Center)

Figure 8 shows the results of the skeleton algorithm using a 3x3 square structuring element on the "bear" image. Figure 8 (a) is the Original input binary image. Figure 8 (b) shows the output from the morphological skeleton algorithm and finally, figure 8 (c) shows the skeleton (yellow) superimposed on the original image (red).

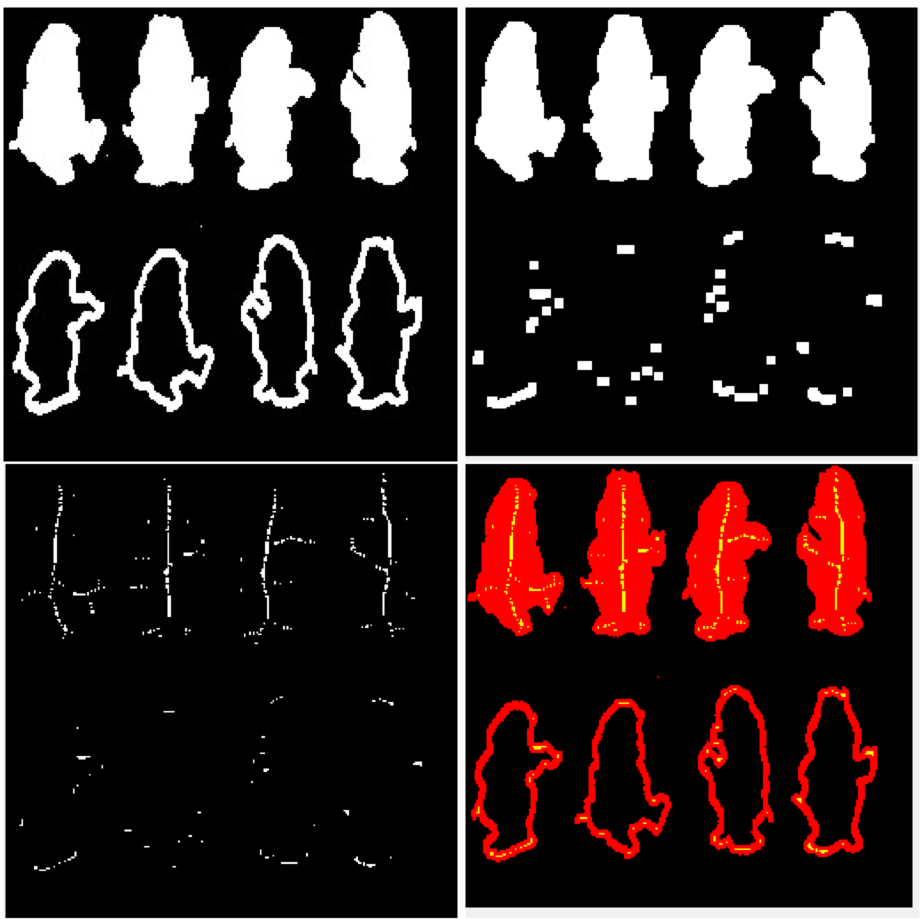


Figure 9: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 2 (Top-Right) (c) Skeleton of partial reconstructed image with k=2 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 9 shows the results of the partial reconstruction algorithm using a 3x3 square structuring element on the "bear" image. Figure 9 (a) is the Original input binary image. Figure 9 (b) shows the partial reconstructed image with k = 2. Figure 9 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 9 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

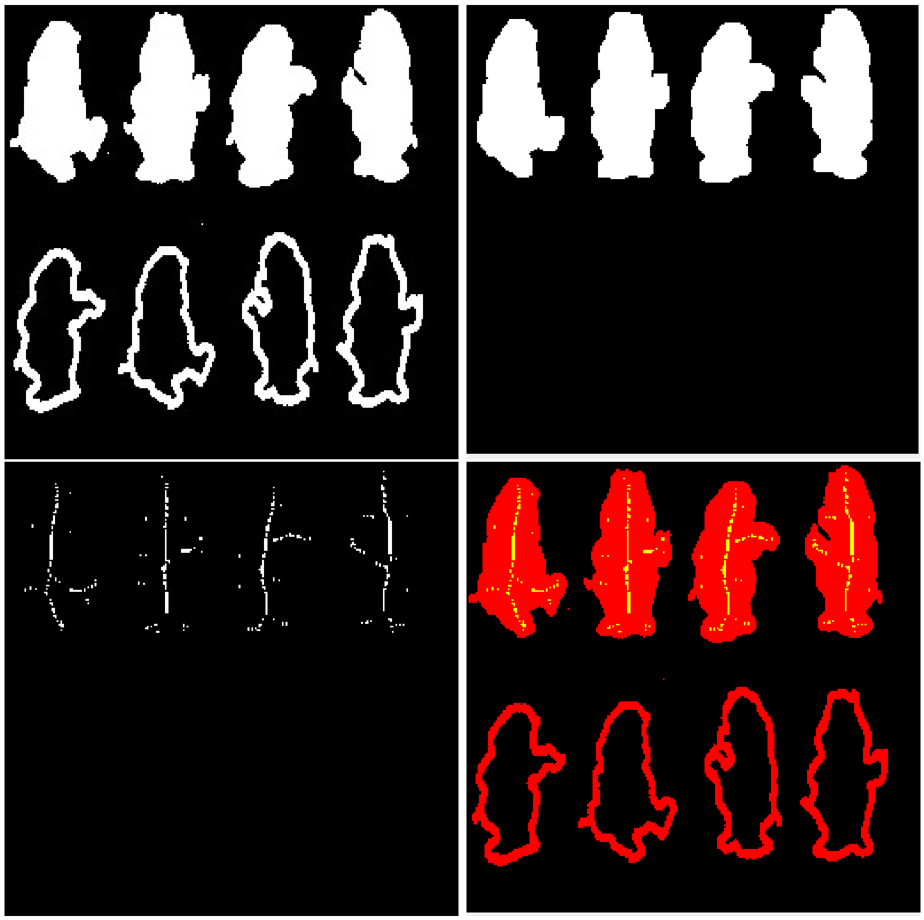


Figure 10: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 3 (Top-Right) (c) Skeleton of partial reconstructed image with k = 3 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image. (Bottom-Right)

Figure 10 shows the results of the partial reconstruction algorithm using a 3x3 square structuring element on the "bear" image. Figure 10 (a) is the Original input binary image. Figure 10 (b) shows the partial reconstructed image with k = 3. Figure 10 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 10 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

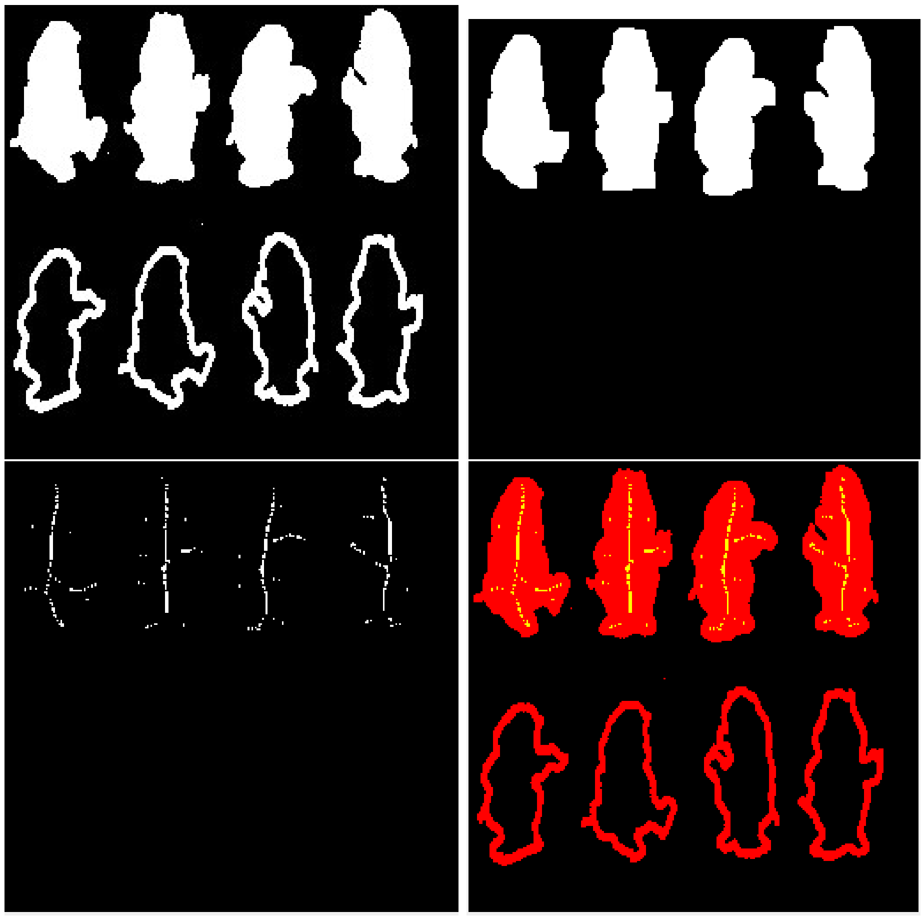


Figure 11: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 4 (Top-Right) (c) Skeleton of partial reconstructed image with k=4 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.

Figure 11 shows the results of the partial reconstruction algorithm using a 3x3 square structuring element on the "bear" image. Figure 11 (a) is the Original input binary image. Figure 11 (b) shows the partial reconstructed image with k = 4. Figure 11 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 11 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

3.2.2 Using Rhombus Structuring Element

Here, the structuring element is a rhombus inscribed within a 3x3 square. The results using it for the skeleton algorithm is as follows:

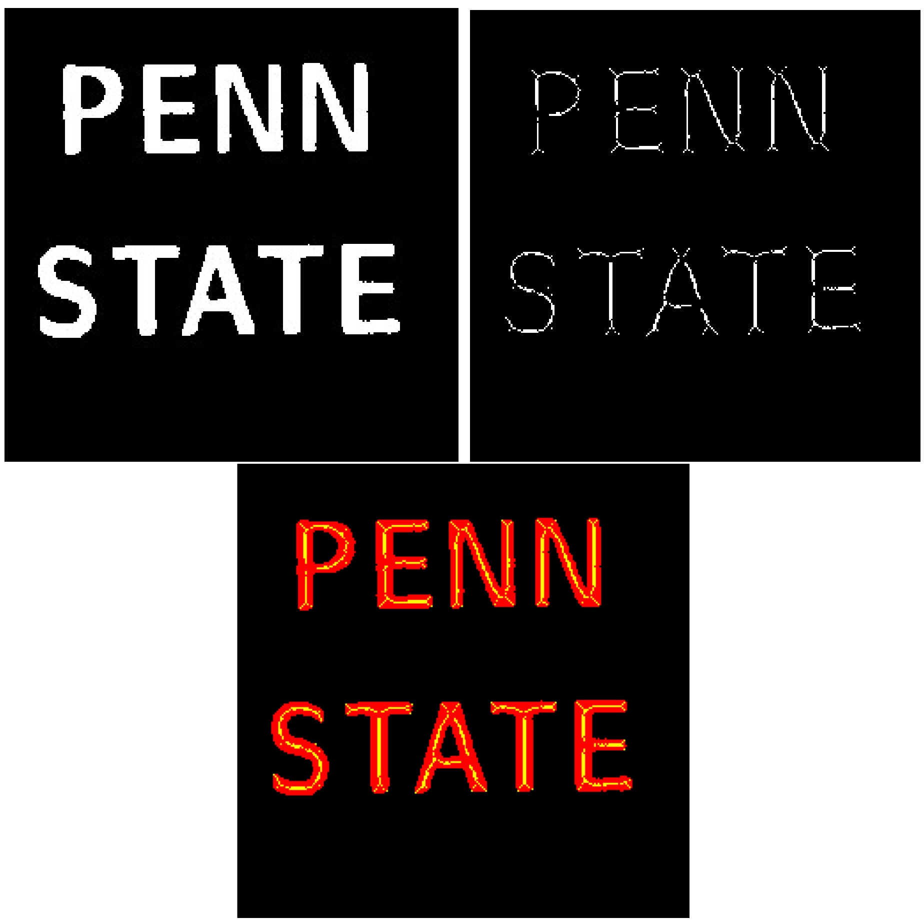


Figure 12: (a) Original Image (Top-Left) (b) Skeleton Image (Top-Right) (c) Skeleton superimposed on Original Image (Bottom-Center)

Figure 12 shows the results of the skeleton algorithm using a rhombus structuring element on the ”penn256” image. Figure 12 (a) is the Original input binary image. Figure 12 (b) shows the output from the morphological skeleton algorithm and finally, figure 12 shows the skeleton (yellow) superimposed on the original image (red).

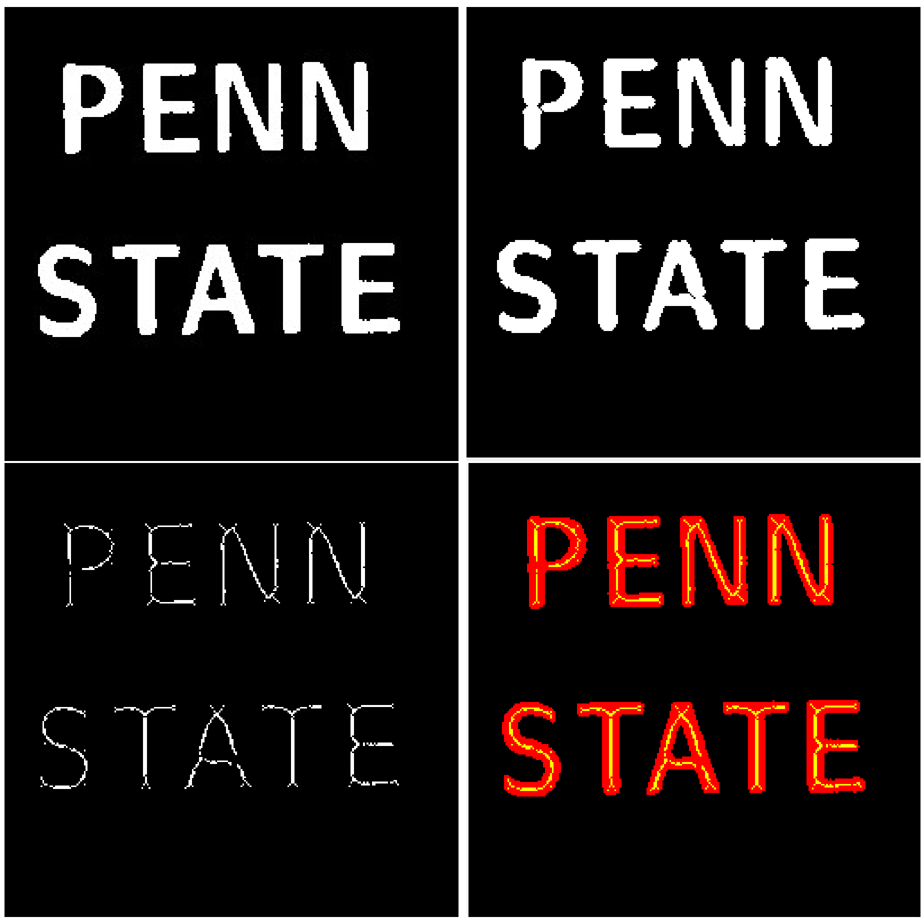


Figure 13: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 2 (Top-Right) (c) Skeleton of partial reconstructed image with k=2 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 13 shows the results of the partial reconstruction algorithm using a rhombus structuring element on the "penn256" image. Figure 13 (a) is the Original input binary image. Figure 13 (b) shows the partial reconstructed image with k = 2. Figure 13 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 13 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

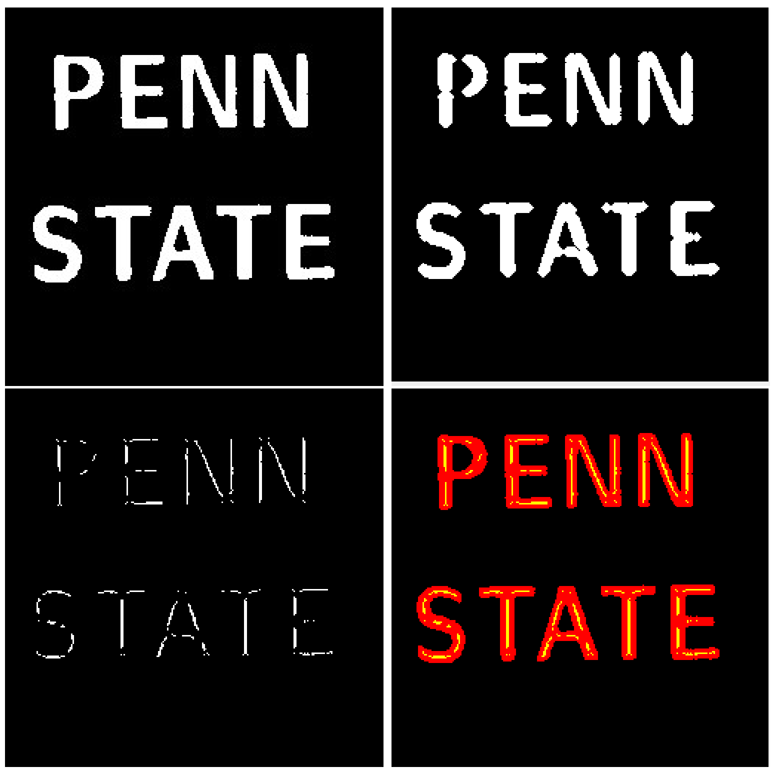


Figure 14: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 3 (Top-Right) (c) Skeleton of partial reconstructed image with k = 3 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 14 shows the results of the partial reconstruction algorithm using a rhombus structuring element on the "penn256" image. Figure 14 (a) is the Original input binary image. Figure 14 (b) shows the partial reconstructed image with k = 3. Figure 14 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 14 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

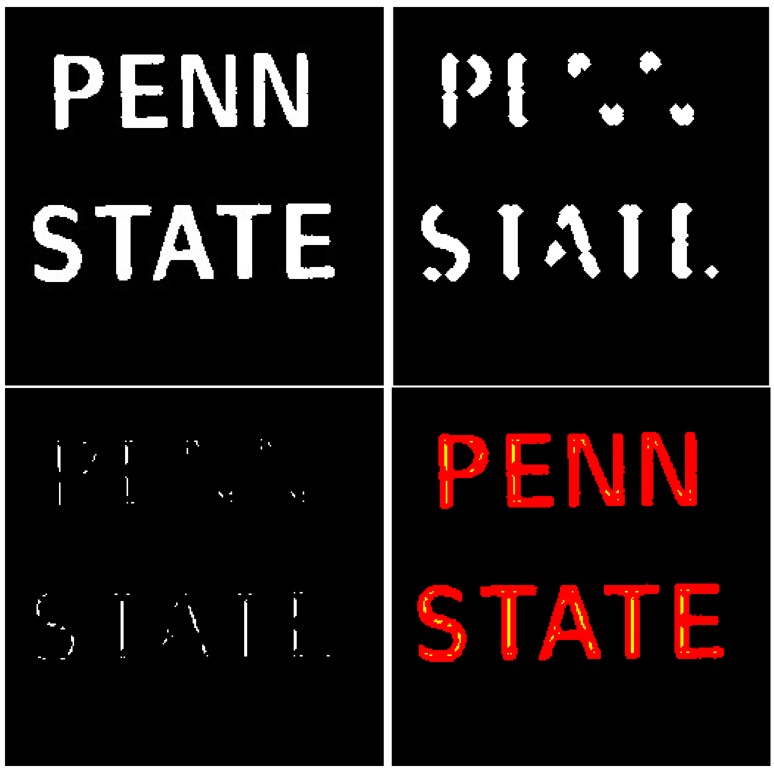


Figure 15: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 4 (Top-Right) (c) Skeleton of partial reconstructed image with k = 4 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 15 shows the results of the partial reconstruction algorithm using a rhombus structuring element on the "penn256" image. Figure 15 (a) is the Original input binary image. Figure 15 (b) shows the partial reconstructed image with k = 4. Figure 15 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 15 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

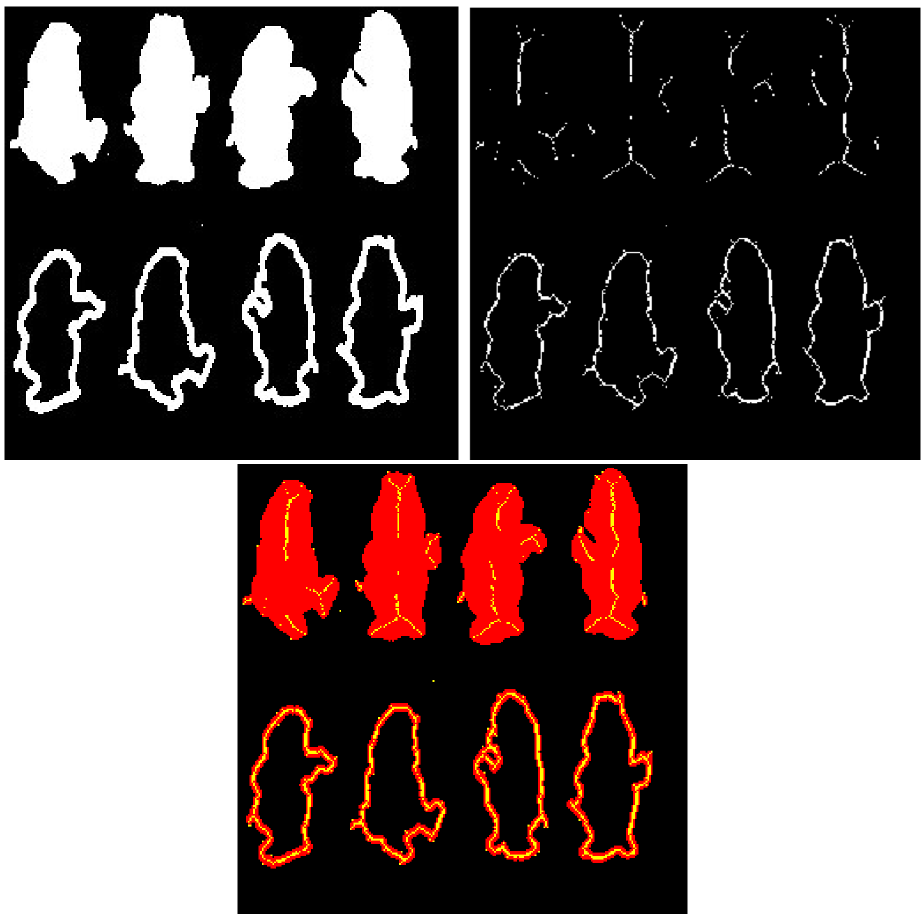


Figure 16: (a) Original Image (Top-Left) (b) Skeleton Image (Top-Right) (c) Skeleton superimposed on Original Image (Bottom-Center)

Figure 16 shows the results of the skeleton algorithm using a rhombus structuring element on the "bear" image. Figure 16 (a) is the Original input binary image.

Figure 16 (b) shows the output from the morphological skeleton algorithm and finally, figure 16 shows the skeleton (yellow) superimposed on the original image (red).

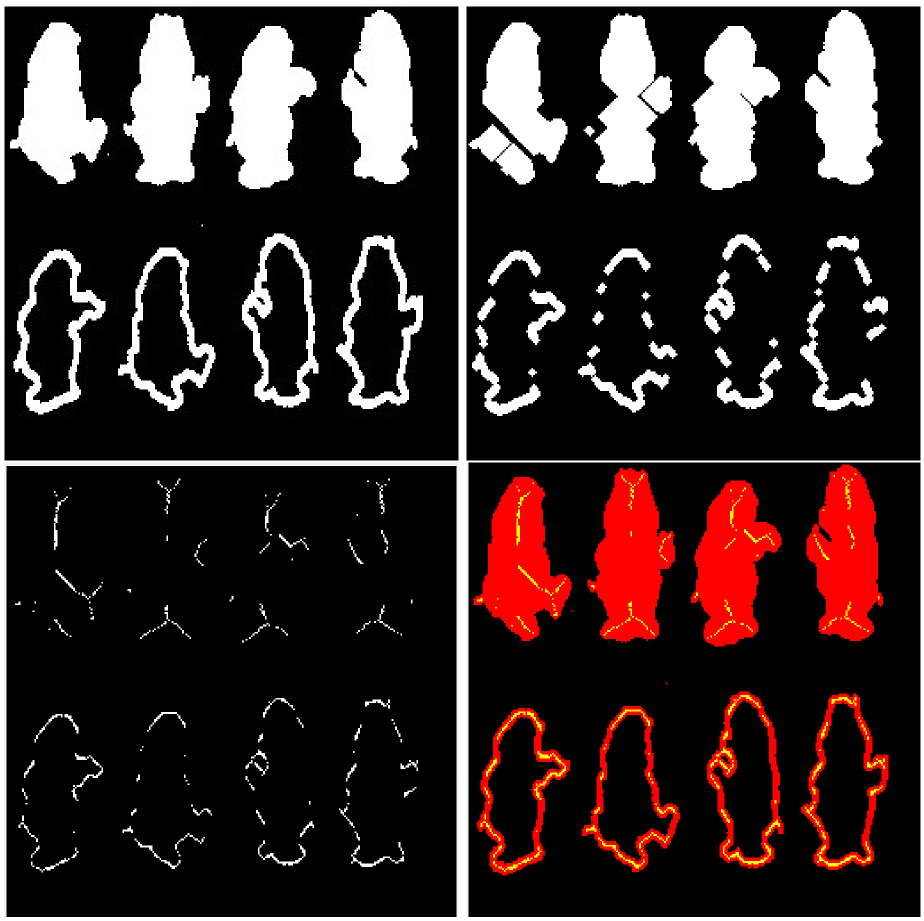


Figure 17: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 2 (Top-Right) (c) Skeleton of partial reconstructed image with k = 2 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.

Figure 17 shows the results of the partial reconstruction algorithm using a rhombus structuring element on the "bear" image. Figure 17 (a) is the Original input binary image. Figure 17 (b) shows the partial reconstructed image with k = 2. Figure 17 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 17 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

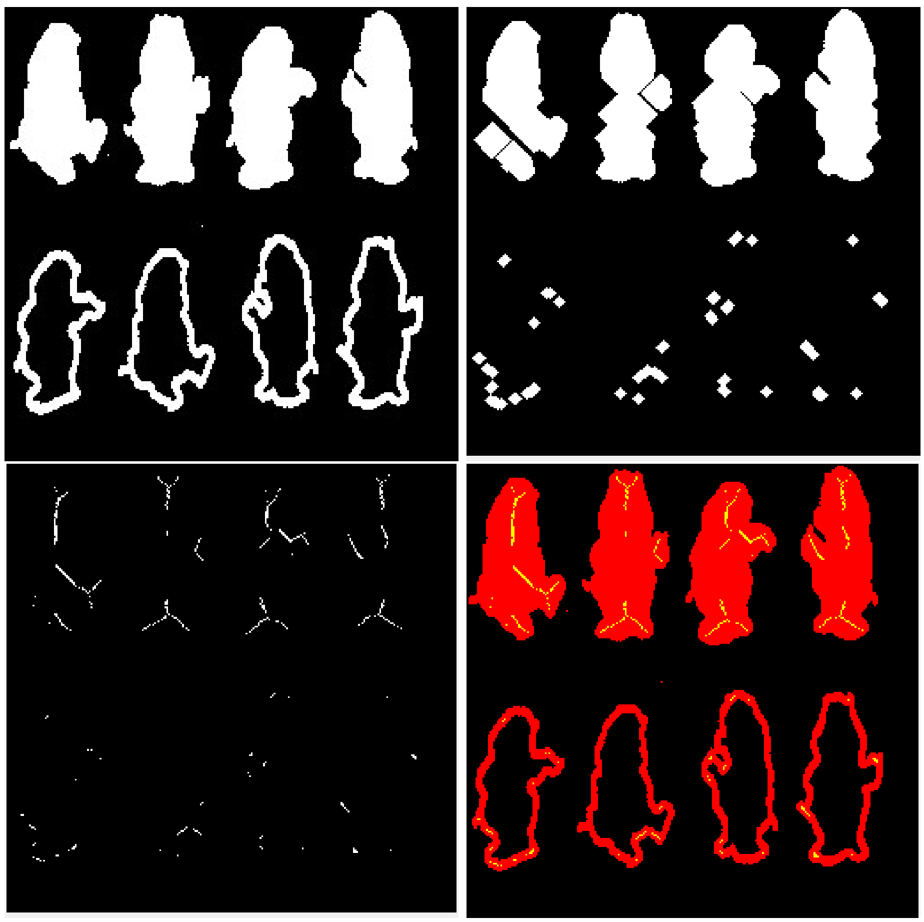


Figure 18: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 3 (Top-Right) (c) Skeleton of partial reconstructed image with k = 3 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.

Figure 18 shows the results of the partial reconstruction algorithm using a rhombus structuring element on the "bear" image. Figure 18 (a) is the Original input binary image. Figure 18 (b) shows the partial reconstructed image with k = 3. Figure 18 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 18 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

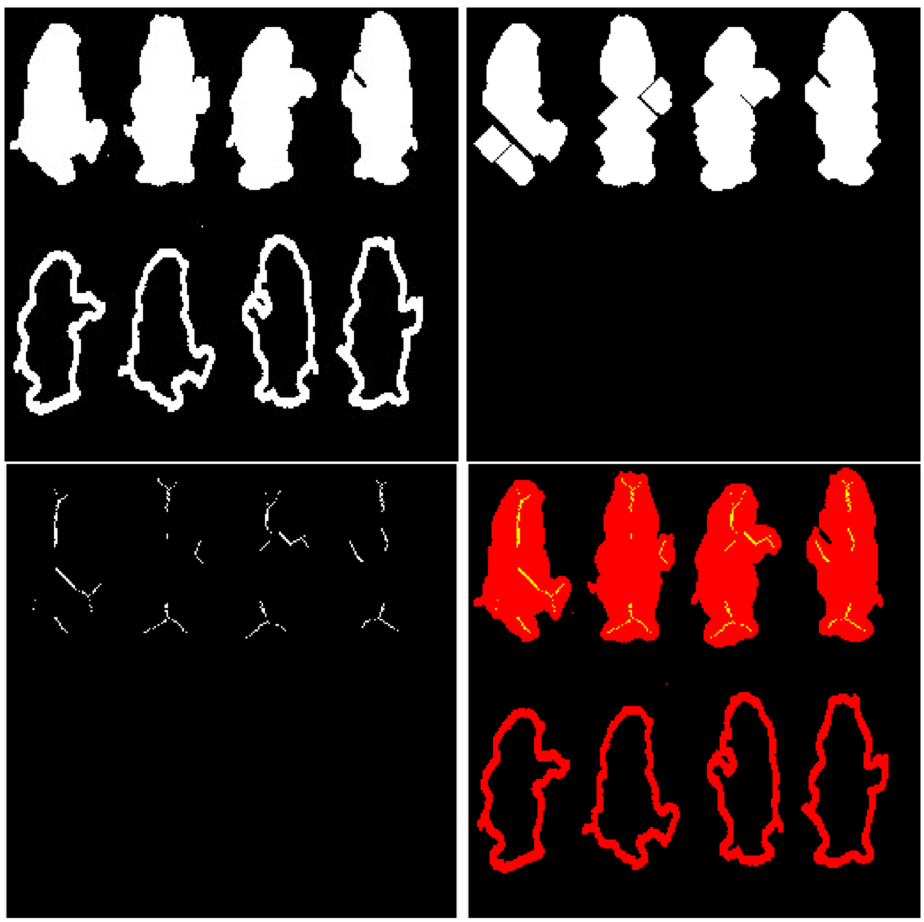


Figure 19: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 4 (Top-Right) (c) Skeleton of partial reconstructed image with k = 4 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.

Figure 19 shows the results of the partial reconstruction algorithm using a rhombus structuring element on the "bear" image. Figure 19 (a) is the Original input binary image. Figure 19 (b) shows the partial reconstructed image with k = 4. Figure 19 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 19 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

3.2.3 Using Vec045 Structuring Element

Here, the structuring element is a vec045 inscribed within a 2x2 square. The results using it for the skeleton algorithm is as follows:

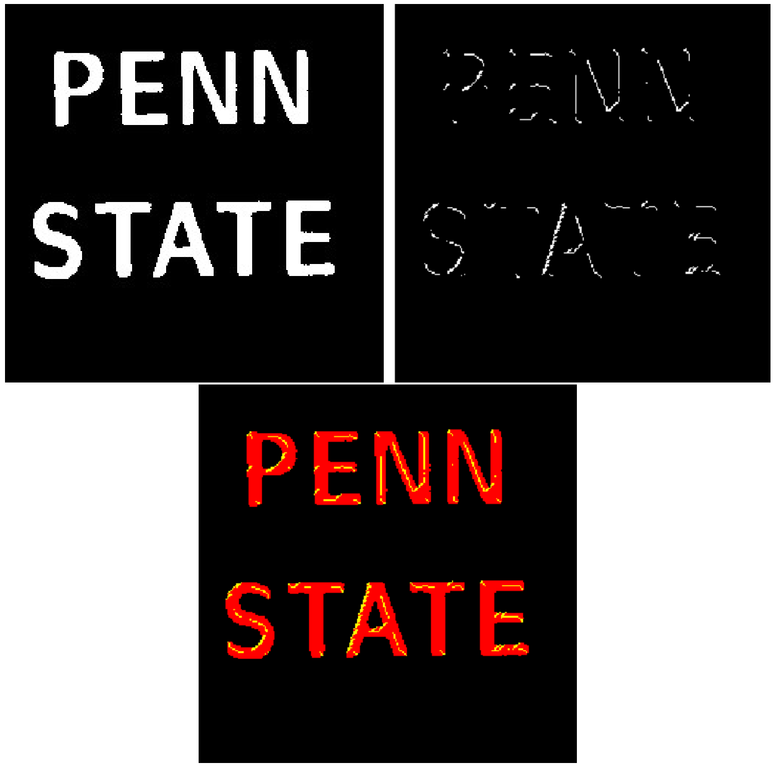


Figure 20: (a) Original Image (Top-Left) (b) Skeleton Image (Top-Right) (c) Skeleton superimposed on Original Image (Bottom-Center)

Figure 20 shows the results of the skeleton algorithm using the vec045 structuring element on the "penn256" image. Figure 20 (a) is the Original input binary image. Figure 20 (b) shows the output from the morphological skeleton algorithm and finally, figure 20 shows the skeleton (yellow) superimposed on the original image (red).

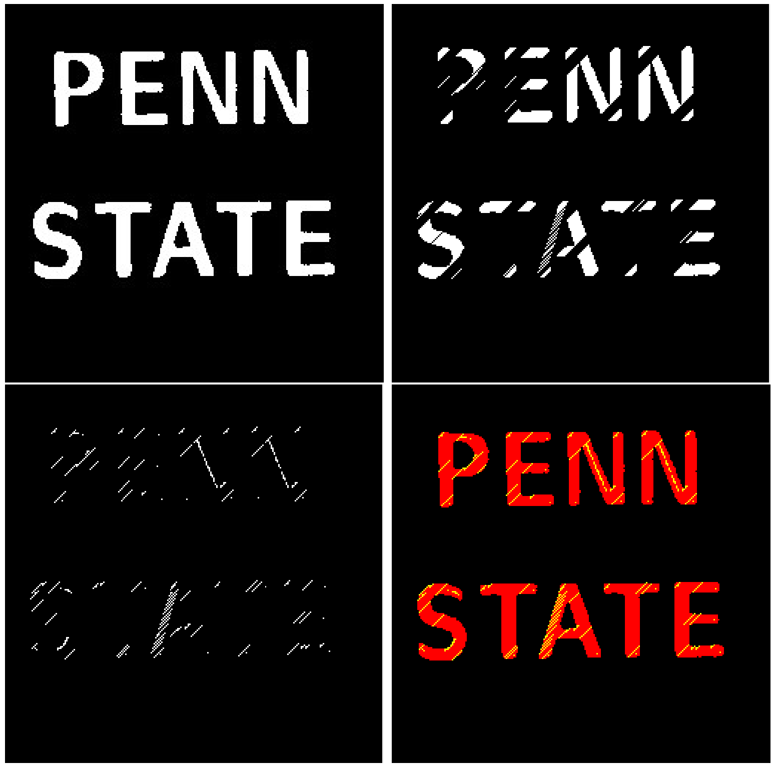


Figure 21: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 2 (Top-Right) (c) Skeleton of partial reconstructed image with k = 2 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 21 shows the results of the partial reconstruction algorithm using the vec045 structuring element on the "penn256" image. Figure 21 (a) is the Original input binary image. Figure 21 (b) shows the partial reconstructed image with k = 2. Figure 21 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 21 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

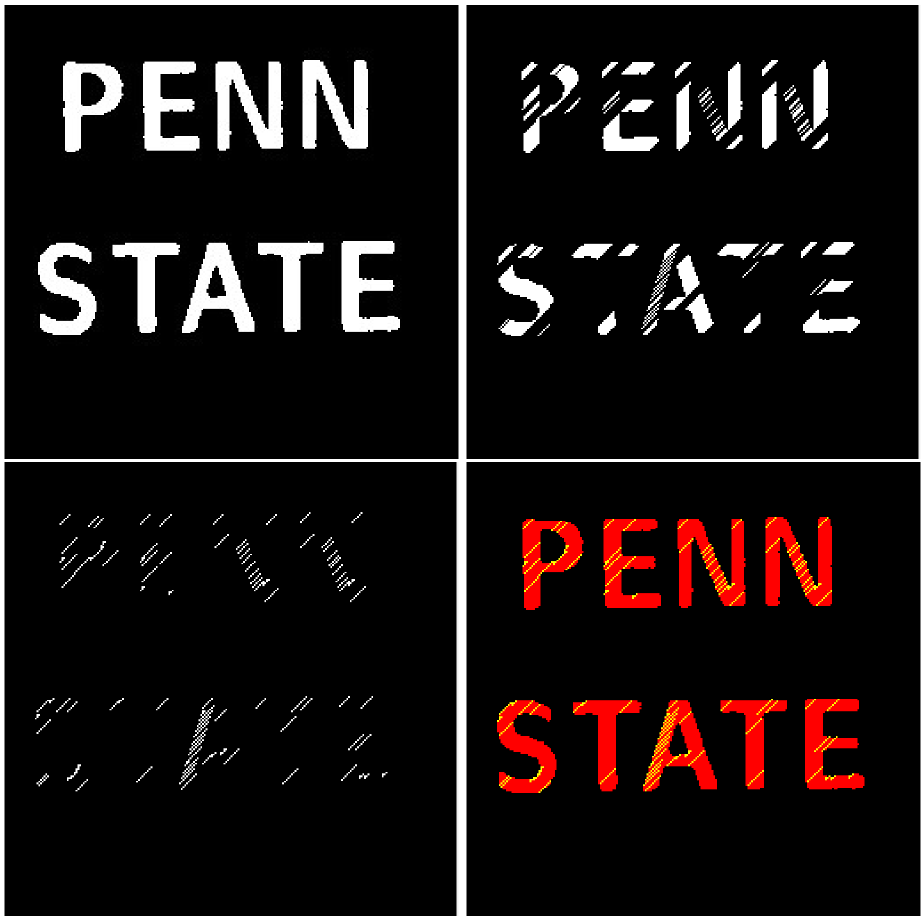


Figure 22: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 3 (Top-Right) (c) Skeleton of partial reconstructed image with k = 3 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 22 shows the results of the partial reconstruction algorithm using the vec045 structuring element on the "penn256" image. Figure 22 (a) is the Original input binary image. Figure 22 (b) shows the partial reconstructed image with k = 3. Figure 22 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 22 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

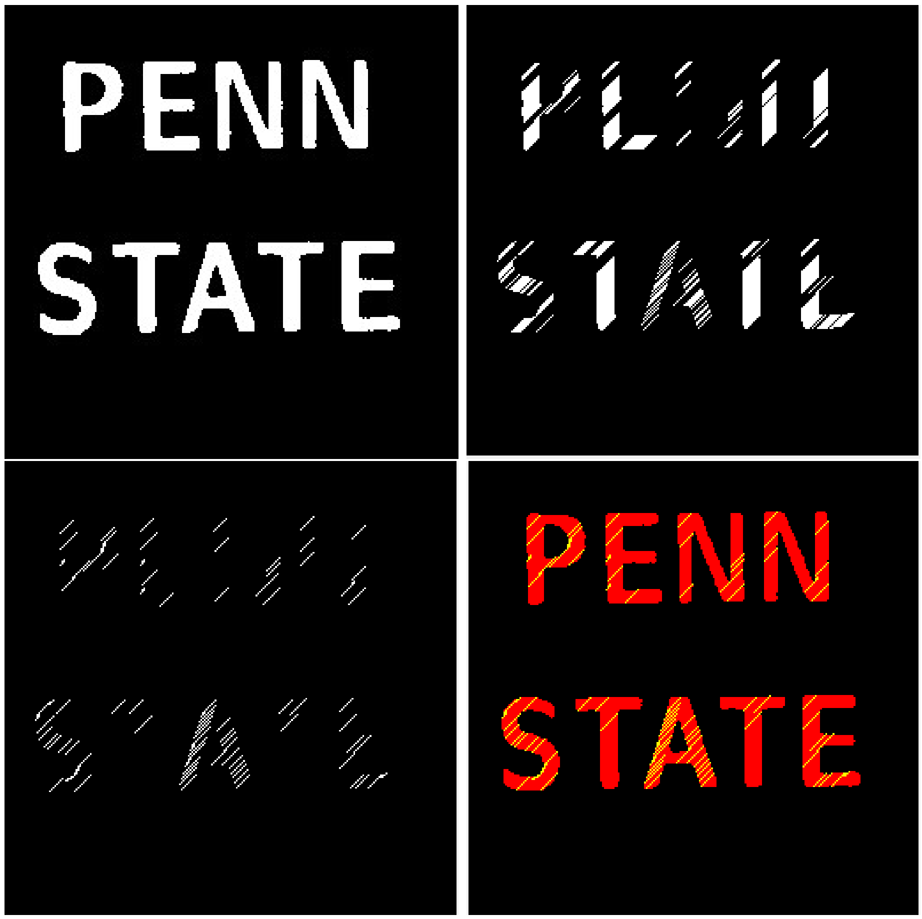


Figure 23: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 4 (Top-Right) (c) Skeleton of partial reconstructed image with k = 4 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.(Bottom-Right)

Figure 23 shows the results of the partial reconstruction algorithm using the vec045 structuring element on the "penn256" image. Figure 23 (a) is the Original input binary image. Figure 23 (b) shows the partial reconstructed image with k = 4. Figure 23 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 23 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

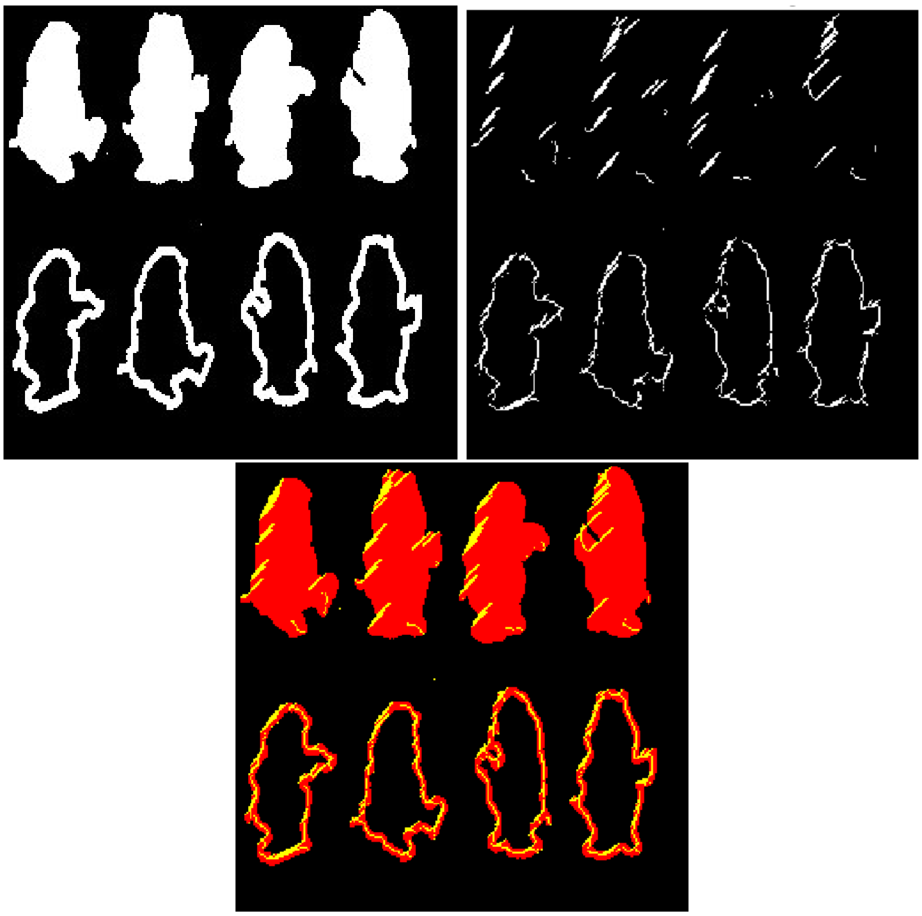


Figure 24: (a) Original Image (Top-Left) (b) Skeleton Image (Top-Right) (c) Skeleton superimposed on Original Image (Bottom-Center)

Figure 24 shows the results of the skeleton algorithm using the vec045 element on the "bear" image. Figure 24 (a) is the Original input binary image. Figure 24 (b) shows the output from the morphological skeleton algorithm and finally, figure 24 shows the skeleton (yellow) superimposed on the original image (red).

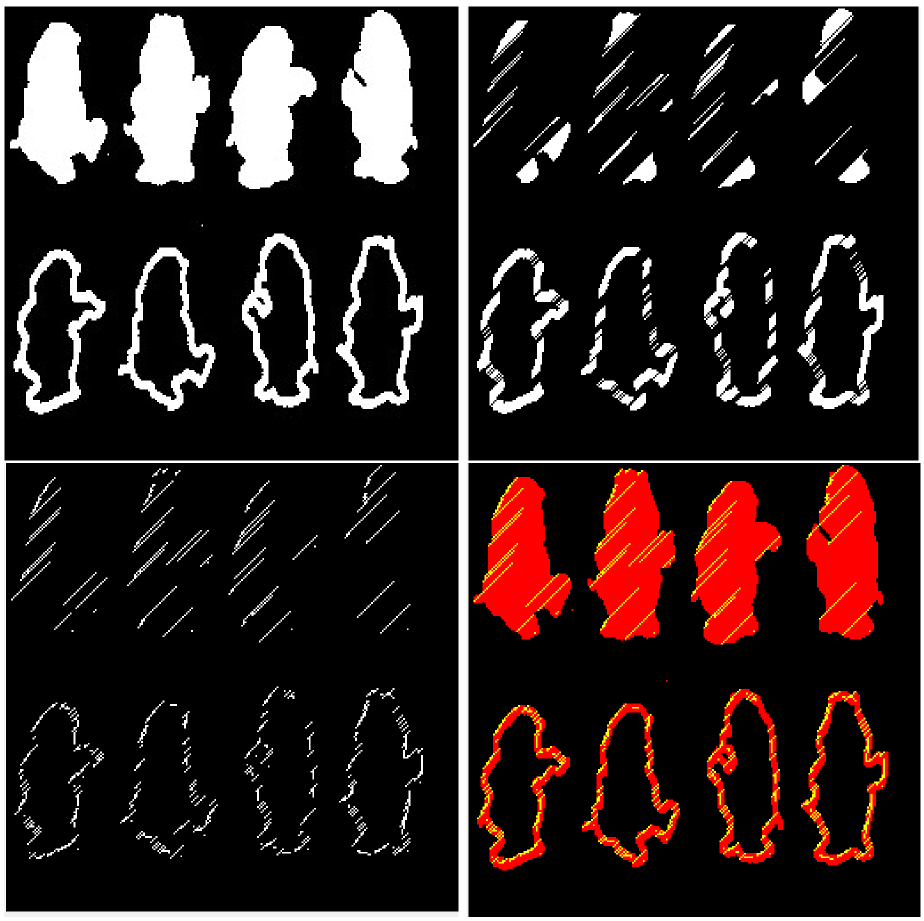


Figure 25: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 2 (Top-Right) (c) Skeleton of partial reconstructed image with k = 2 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.

Figure 25 shows the results of the partial reconstruction algorithm using the vec045 structuring element on the "bear" image. Figure 25 (a) is the Original input binary image. Figure 25 (b) shows the partial reconstructed image with k = 2. Figure 25 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 25 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

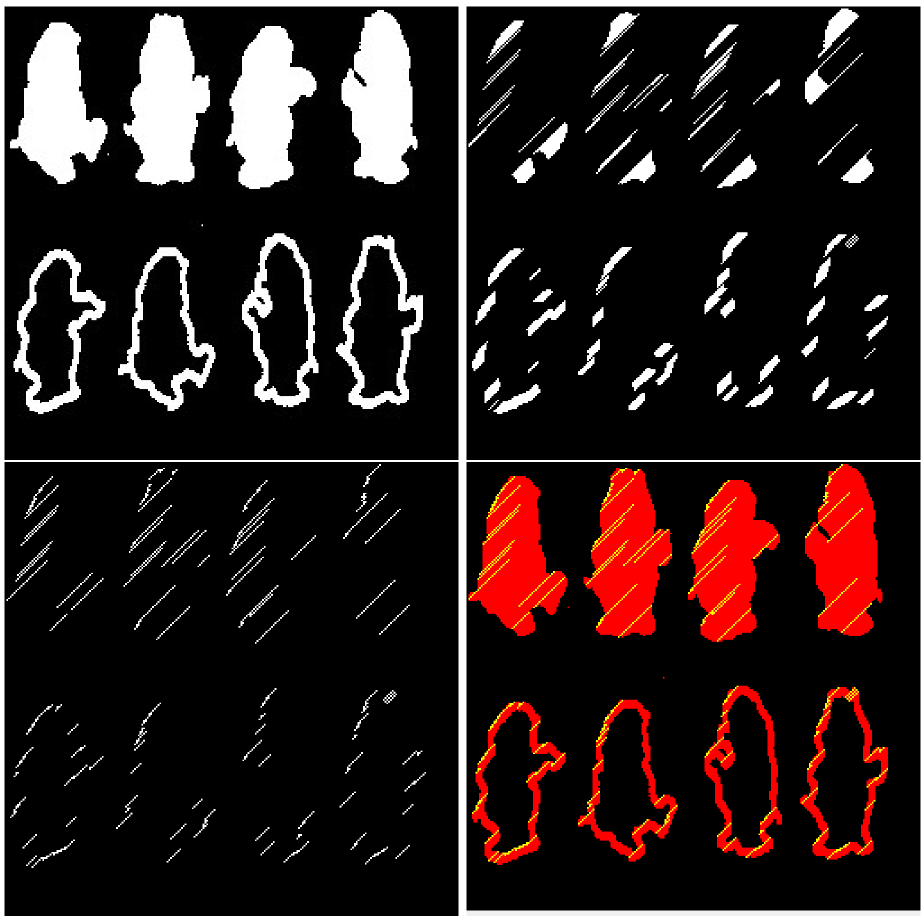


Figure 26: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 3 (Top-Right) (c) Skeleton of partial reconstructed image with k = 3 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.

Figure 26 shows the results of the partial reconstruction algorithm using the vec045 structuring element on the "bear" image. Figure 26 (a) is the Original input binary image. Figure 26 (b) shows the partial reconstructed image with k = 3. Figure 26 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 26 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

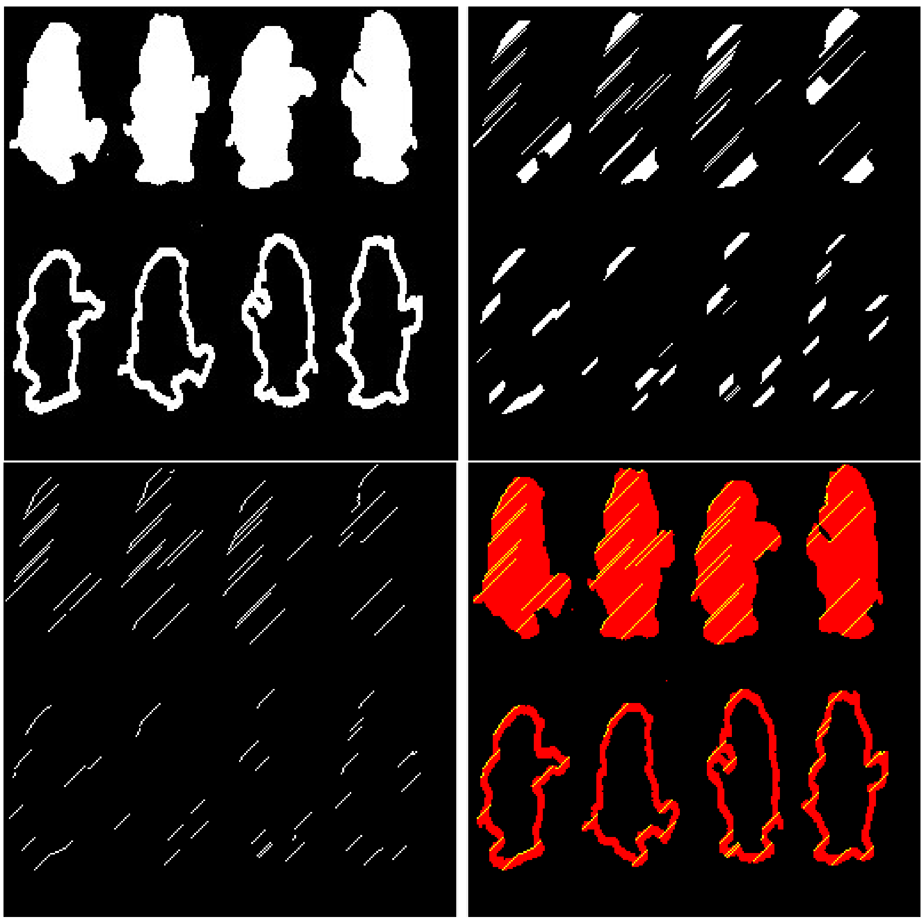


Figure 27: (a) Original Image (Top-Left) (b) Partial Reconstructed with k = 4 (Top-Right) (c) Skeleton of partial reconstructed image with k = 4 (Bottom-Left) (d) Skeleton superimposed on partial reconstructed image.

Figure 27 shows the results of the partial reconstruction algorithm using the vec045 structuring element on the "bear" image. Figure 27 (a) is the Original input binary image. Figure 27 (b) shows the partial reconstructed image with k = 4. Figure 27 (c) shows the output of (b) from the morphological skeleton algorithm and finally, figure 27 (d) shows the skeleton (yellow) superimposed on the partial reconstructed image (red).

From the results of skeletonization and partial reconstruction, we can observe that the rhombus does a very good job of showing the skeleton distinctly at the boundary of an object clearly than the other two structuring elements. The 3x3 square does a geate job of getting the medial axis of the body of the object. But, vec045 is not as effective as rhombus or 3x3 square.

3.3 Results of Shape Analysis in Q.2a

In this subsection, we are showing the results of the size distribution, pecstrum and complexity of objects from two images - "match1.gif" (Reference set), "match3.gif" (Test Set). We will be showing the size distribution plot, the pecstrum plot and a table containing complexity values (this table is shown after discussing all the objects from match1 and match3).

3.3.1 Object 1 of Reference Image match1 Using 3x3 Square Structuring Element

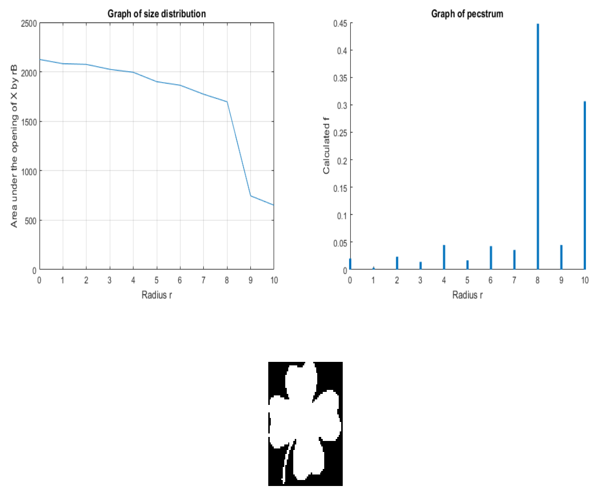


Figure 28: (a) Size distribution of Object 1 in Reference Image match1 (Top-Left) (b) Pecstrum of Object 1 in Reference Image match1 (Top-Right) (c) Object 1 in Reference Image match1 (Bottom-Center)

Figure 28 shows the results of shape analysis of Object1 of "match1" image. Figure 28 (a) is the size distribution plot of Object1. Figure 28 (b) is the size distribution plot of Object1 and figure 28 (c) is the Original Binary image of Object1.

3.3.2 Object 2 of Reference Image match1 Using 3x3 Square Structuring Element

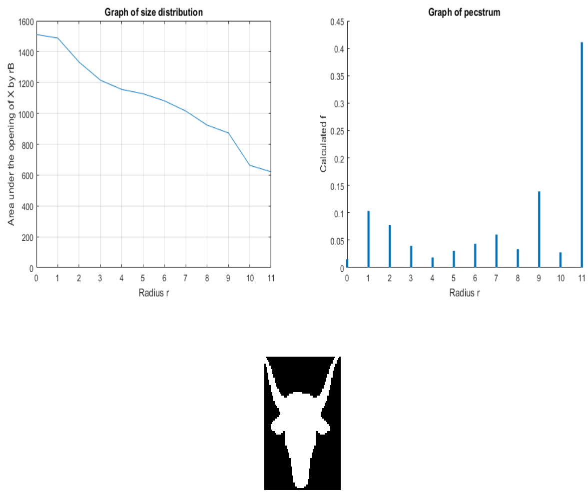


Figure 29: (a) Size distribution of Object 2 in Reference Image match1 (Top-Left) (b) Pecstrum of Object 2 in Reference Image match1 (Top-Right) (c) Object 2 in Reference Image match1 (Bottom-Center)

Figure 29 shows the results of shape analysis of Object2 of "match1" image. Figure 29 (a) is the size distribution plot of Object2. Figure 29 (b) is the size distribution plot of Object2 and figure 29 (c) is the Original Binary image of Object2.

3.3.3 Object 3 of Reference Image match1 Using 3x3 Square Structuring Element

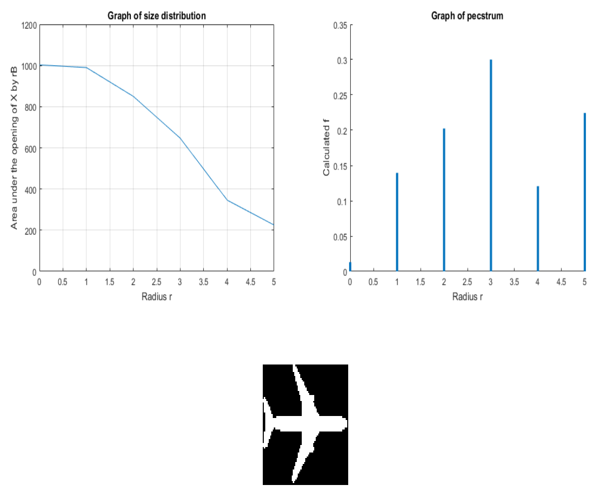


Figure 30: (a) Size distribution of Object 3 in Reference Image match1 (Top-Left) (b) Pecstrum of Object 3 in Reference Image match1 (Top-Right) (c) Object3 in Reference Image match1 (Bottom-Center)

Figure 30 shows the results of shape analysis of Object3 of "match1" image. Figure 30 (a) is the size distribution plot of Object3. Figure 30 (b) is the size distribution plot of Object3 and figure 30 (c) is the Original Binary image of Object3.

3.3.4 Object 4 of Reference Image match1 Using 3x3 Square Structuring Element

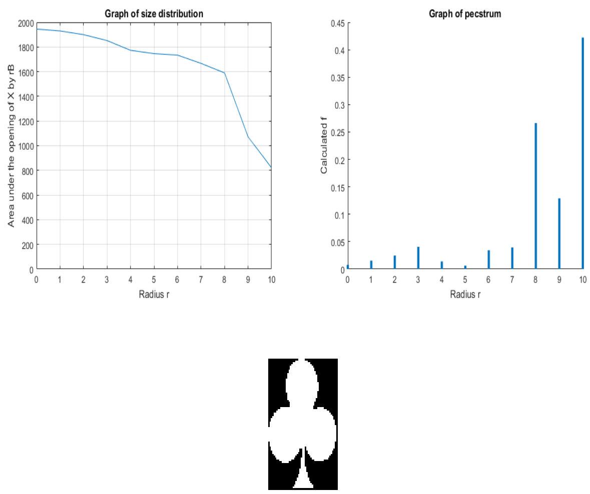


Figure 31: (a) Size distribution of Object 4 in Reference Image match1 (Top-Left) (b) Pecstrum of Object 4 in Reference Image match1 (Top-Right) (c) Object 4 in Reference Image match1 (Bottom-Center)

Figure 31 shows the results of shape analysis of Object4 of "match1" image. Figure 31 (a) is the size distribution plot of Object4. Figure 31 (b) is the size distribution plot of Object4 and figure 31 (c) is the Original Binary image of Object4.

3.3.5 Object 1 of Test Image match3 Using 3x3 Square Structuring Element

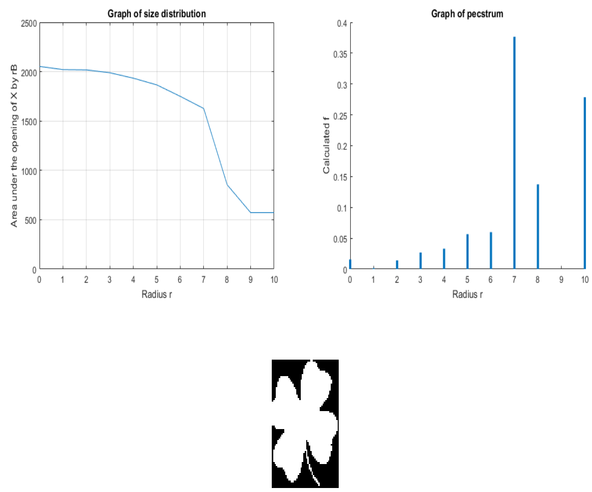


Figure 32: (a) Size distribution of Object 1 in Test Image match3 (Top-Left) (b) Pecstrum of Object 1 in Test Image match3 (Top-Right) (c) Object 1 in Test Image match3 (Bottom-Center)

Figure 32 shows the results of shape analysis of Object1 of "match3" image. Figure 32 (a) is the size distribution plot of Object1. Figure 32 (b) is the size distribution plot of Object1 and figure 32 (c) is the Original Binary image of Object1.

3.3.6 Object 2 of Test Image match3 Using 3x3 Square Structuring Element

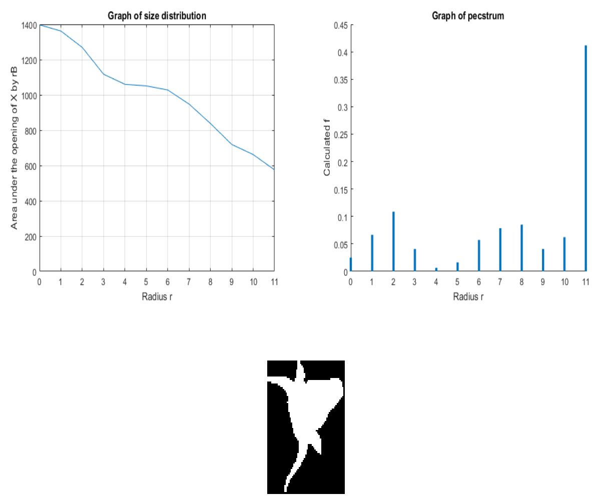


Figure 33: (a) Size distribution of Object 2 in Test Image match3 (Top-Left) (b) Pecstrum of Object 2 in Test Image match3 (Top-Right) (c) Object 2 in Test Image match3 (Bottom-Center)

Figure 33 shows the results of shape analysis of Object2 of "match3" image. Figure 33 (a) is the size distribution plot of Object2. Figure 33 (b) is the size distribution plot of Object2 and figure 33 (c) is the Original Binary image of Object2.

3.3.7 Object 3 of Test Image match3 Using 3x3 Square Structuring Element

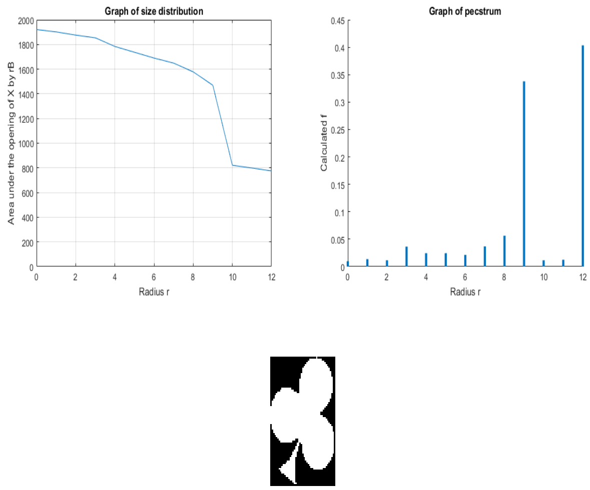


Figure 34: (a) Size distribution of Object 3 in Test Image match3 (Top-Left) (b) Pecstrum of Object 3 in Test Image match3 (Top-Right) (c) Object 3 in Test Image match3 (Bottom-Center)

Figure 34 shows the results of shape analysis of Object3 of "match3" image. Figure 34 (a) is the size distribution plot of Object3. Figure 34 (b) is the size distribution plot of Object3 and figure 34 (c) is the Original Binary image of Object3.

3.3.8 Object 4 of Test Image match3 Using 3x3 Square Structuring Element

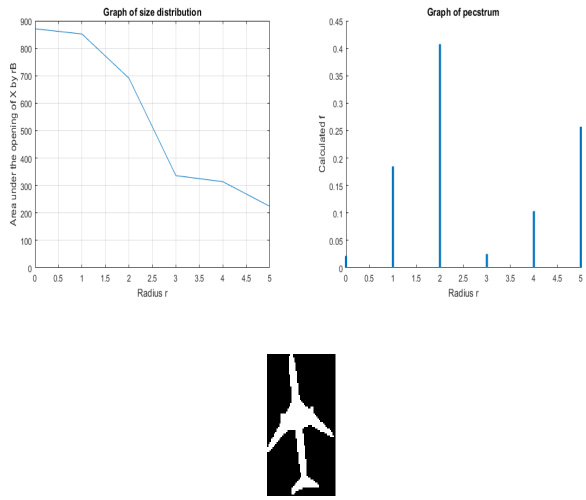


Figure 35: (a) Size distribution of Object 4 in Test Image match3 (Top-Left) (b) Pecstrum of Object 4 in Test Image match3 (Top-Right) (c) Object 4 in Test Image match3 (Bottom-Center)

Figure 35 shows the results of shape analysis of Object4 of "match3" image. Figure 35 (a) is the size distribution plot of Object4. Figure 35 (b) is the size distribution plot of Object4 and figure 35 (c) is the Original Binary image of Object4.

3.3.9 Calculation of Complexity of Each Object in Reference and Test Image in Q. 2a i

	Object1	Object2	Object3	Object4
Reference Image	2.264103	2.833744	2.317193	2.364053
Test Image	2.412078	2.866970	2.397661	2.074618

Table 2: Complexity Values of Each Object in the Reference and Test Images

It can be observed from Table 2, that the object2 (image of the steer) in the reference image ("match1.gif") is the most complex, as it has the highest complexity value (2.833744 units) among the 4 objects. [Answer to Q. 2a.i]

Object1 in the reference image is the least complex, as it has the least complexity value (2.264103 units) among the 4 objects.

Also, in the test image, Object2 is the most complex, as it has the highest complexity value (2.866970 units) among the 4 objects while object4 is the least complex, as it has the least complexity value (2.074618 units) among the 4 objects.

3.3.10 Calculation of Distance Values for Each Test Object with respect to the 4 Reference Objects in Q.2a ii

Test \ Reference	Object1	Object2	Object3	Object4
Object1	3.948936	8.912552	17.875113	4.037261
Object2	10.474921	1.382626	6.791325	5.573545
Object3	7.013527	4.706453	15.183412	1.460295
Object4	22.106141	9.823636	3.981033	17.158374

Table 3: Distance Values of Each Test Object for Pattern Recognition

In Table 3, Each test object is matched to one of the 4 objects in the reference image, if the distance calculated for that object is the minimum among all 4 objects. The test object and its matched reference objects have been displayed in Table 4.

3.3.11 Matching of Test Object with Reference Object in Q.2a ii

Test Object	Image of Test Object	Image of Reference Object	Reference Object
Object1		Object1	
Object2		Object2	
Object3		Object4	
Object4		Object3	

Table 4: Matching of Test Objects to Reference Objects by Pecstral Analysis

It can be observed from Table 4, that the clover,steer, spade and airplane in the test image correctly matches to the corresponding objects in the reference image.

3.4 Results of Shape Analysis in Q.2b

In this subsection, we are showing the results of the distance calculation during pecstral analysis of two images - "shadow1.gif", "shadow1rotated.gif". The matched images are also displayed in this section.

The "shadow1.gif" is referred to as the Reference Image while the 4 solid objects in the "shadow1.gif" are referred to as the Reference Objects.

The "shadow1rotated.gif" is referred to as the Test Image while the 4 solid objects in the "shadow1rotated.gif" are referred to as the Test Objects.

3.4.1 Calculation of Distance Values for Each Test Object with respect to the 4 Reference Objects in Q.2b

Test \ Reference	Object1	Object2	Object3	Object4
Object1	1.288888	11.640566	0.555825	7.191605
Object2	8.785265	1.496396	12.918726	3.274013
Object3	3.944866	6.414783	7.930968	0.147906
Object4	0.363720	8.620016	1.927990	3.542005

Table 5: Distance Values of Each Test Object for Pattern Recognition

Each test object matches with the reference object for which it has the least distance (di) value among all other reference objects.

3.4.2 Matching of Test Object with Reference Object in Q.2b

Test Object	Image of Test Object	Image of Reference Object	Reference Object
Object1		Object3	
Object2		Object2	
Object3		Object4	
Object4		Object1	

Table 6: Matching of Test Objects to Reference Objects by Pecstral Analysis

It can be observed that all 4 objects in the test image correctly match with their corresponding objects in the reference image.

3.5 Statistics of Each Object in the Reference and Test Images

In this subsection, we are showing the results of the size distribution, pecstrum and complexity of two images - "shadow1.gif" (reference image) and "shadow1rotated.gif" (test image). We will be showing the size distribution plot, the pecstrum plot and a table containing complexity values (this table is shown once every 4 objects discussed below). The statistics is useful to understand the mechanism behind the matching of the test objects to the reference objects by pecstral analysis.

3.5.1 Object 1 of Reference Image shadow1 Using 3x3 Square Structuring Element

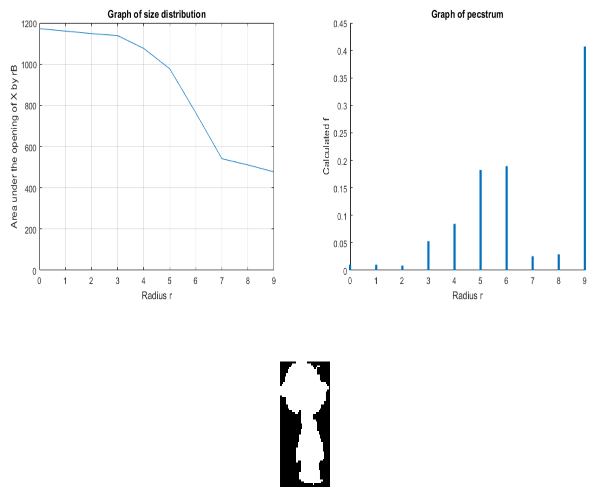


Figure 36: (a) Size distribution of Object 1 in Reference Image shadow1 (Top-Left) (b) Pecstrum of Object 1 in Reference Image shadow1 (Top-Right) (c) Object 1 in Reference Image shadow1 (Bottom-Center)

Figure 36 shows the results of shape analysis of Object1 of "shadow1" image. Figure 36 (a) is the size distribution plot of Object1. Figure 36 (b) is the size distribution plot of Object1 and figure 36 (c) is the Original Binary image of Object1.

3.5.2 Object 2 of Reference Image shadow1 Using 3x3 Square Structuring Element

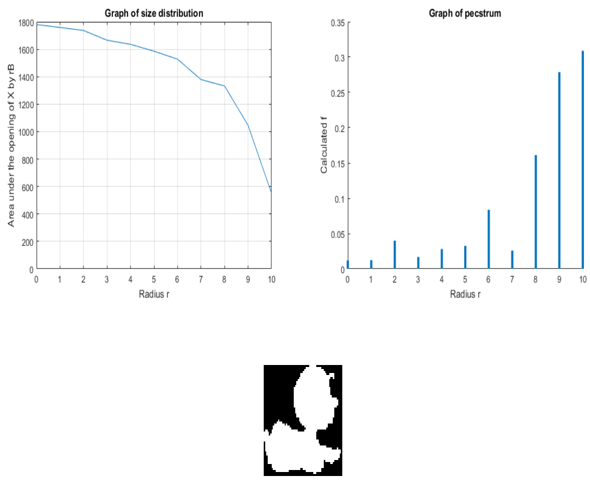


Figure 37: (a) Size distribution of Object 2 in Reference Image shadow1 (Top-Left) (b) Pecstrum of Object 2 in Reference Image shadow1 (Top-Right) (c) Object 2 in Reference Image shadow1 (Bottom-Center)

Figure 37 shows the results of shape analysis of Object2 of "shadow1" image. Figure 37 (a) is the size distribution plot of Object2. Figure 37 (b) is the size distribution plot of Object2 and figure 37 (c) is the Original Binary image of Object2.

3.5.3 Object 3 of Reference Image shadow1 Using 3x3 Square Structuring Element

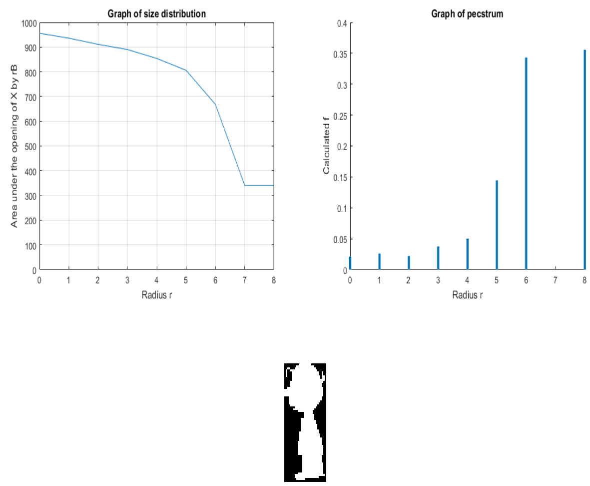


Figure 38: (a) Size distribution of Object 3 in Reference Image shadow1 (Top-Left) (b) Pecstrum of Object 3 in Reference Image shadow1 (Top-Right) (c) Object3 in Reference Image shadow1 (Bottom-Center)

Figure 38 shows the results of shape analysis of Object3 of "shadow1" image. Figure 38 (a) is the size distribution plot of Object3. Figure 38 (b) is the size distribution plot of Object3 and figure 38 (c) is the Original Binary image of Object3.

3.5.4 Object 4 of Reference Image shadow1 Using 3x3 Square Structuring Element

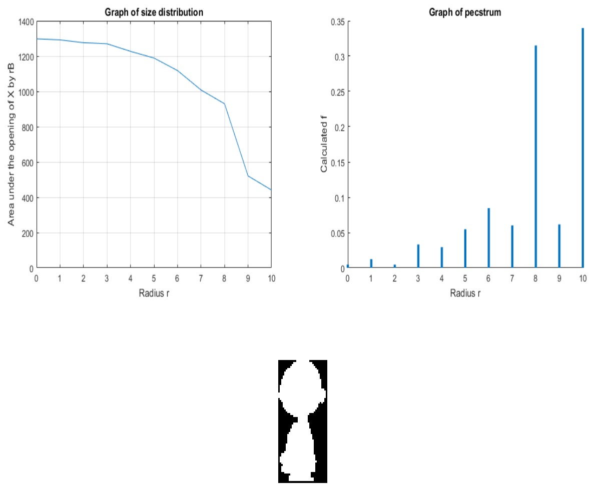


Figure 39: (a) Size distribution of Object 4 in Reference Image shadow1 (Top-Left) (b) Pecstrum of Object 4 in Reference Image shadow1 (Top-Right) (c) Object 4 in Reference Image shadow1 (Bottom-Center)

Figure 39 shows the results of shape analysis of Object4 of "shadow1" image. Figure 39 (a) is the size distribution plot of Object4. Figure 39 (b) is the size distribution plot of Object4 and figure 39 (c) is the Original Binary image of Object4.

3.5.5 Object 1 of Test Image shadow1rotated Using 3x3 Square Structuring Element

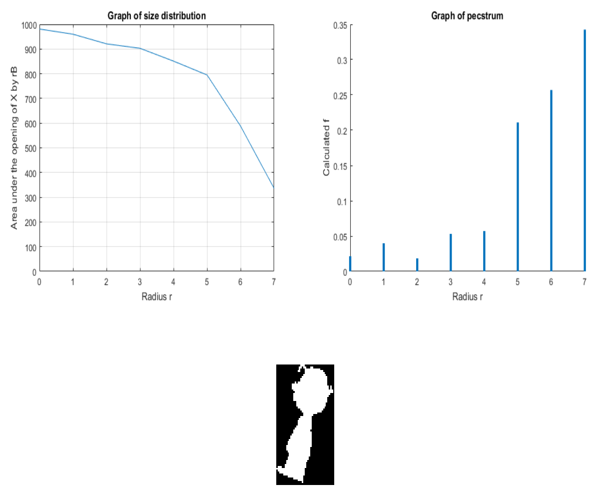


Figure 40: (a) Size distribution of Object 1 in Test Image shadow1rotated (Top-Left) (b) Pecstrum of Object 1 in Test Image shadow1rotated (Top-Right) (c) Object 1 in Test Image shadow1rotated (Bottom-Center)

Figure 40 shows the results of shape analysis of Object1 of "shadow1rotated" image. Figure 40 (a) is the size distribution plot of Object1. Figure 40 (b) is the size distribution plot of Object1 and figure 40 (c) is the Original Binary image of Object1.

3.5.6 Object 2 of Test Image shadow1rotated Using 3x3 Square Structuring Element

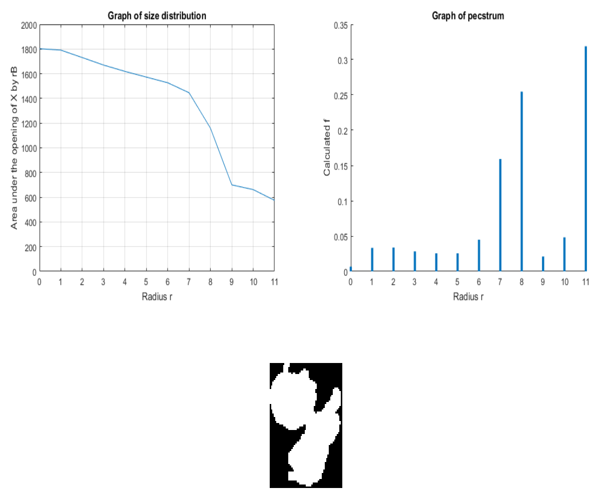


Figure 41: (a) Size distribution of Object 2 in Test Image shadow1rotated (Top-Left) (b) Pecstrum of Object 2 in Test Image shadow1rotated (Top-Right) (c) Object 2 in Test Image shadow1rotated (Bottom-Center)

Figure 41 shows the results of shape analysis of Object2 of ”shadow1rotated” image. Figure 41 (a) is the size distribution plot of Object2. Figure 41 (b) is the size distribution plot of Object2 and figure 41 (c) is the Original Binary image of Object2.

3.5.7 Object 3 of Test Image shadow1rotated Using 3x3 Square Structuring Element

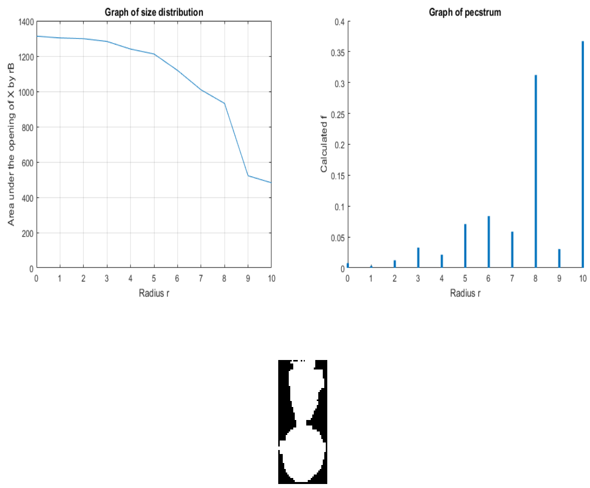


Figure 42: (a) Size distribution of Object 3 in Test Image shadow1rotated (Top-Left) (b) Pecstrum of Object 3 in Test Image shadow1rotated (Top-Right) (c) Object 3 in Test Image shadow1rotated (Bottom-Center)

Figure 42 shows the results of shape analysis of Object3 of ”shadow1rotated” image. Figure 42 (a) is the size distribution plot of Object3. Figure 42 (b) is the size distribution plot of Object3 and figure 42 (c) is the Original Binary image of Object3.

3.5.8 Object 4 of Test Image shadow1rotated Using 3x3 Square Structuring Element

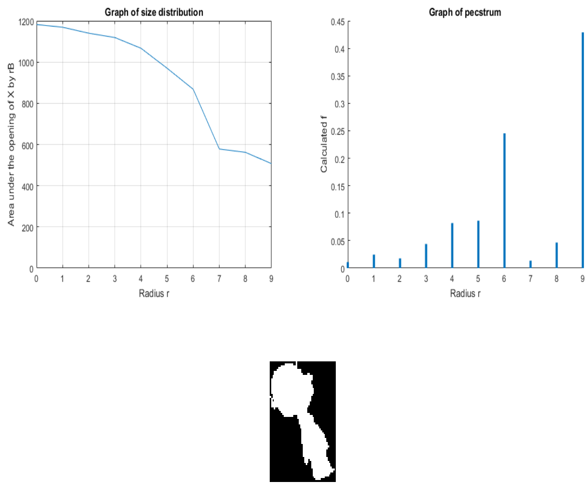


Figure 43: (a) Size distribution of Object 4 in Test Image shadow1rotated (Top-Left) (b) Pecstrum of Object 4 in Test Image shadow1rotated (Top-Right) (c) Object 4 in Test Image shadow1rotated (Bottom-Center)

Figure 43 shows the results of shape analysis of Object4 of "shadow1rotated" image. Figure 43 (a) is the size distribution plot of Object4. Figure 43 (b) is the size distribution plot of Object4 and figure 43 (c) is the Original Binary image of Object4.

3.5.9 Calculation of Complexity of Each Object in Reference and Test Images

	Object1	Object2	Object3	Object4
Reference Image	2.433484	2.641876	2.233068	2.538640
Test Image	2.376735	2.772153	2.455174	2.416492

Table 7: Complexity Values of Each Object in the Reference and Test Images

It can be observed from Table 7, that the object2 in the reference image is the most complex, as it has the highest complexity value (2.641876 units) among the 4 objects while object3 is the least complex, as it has the least complexity value (2.33068 units) among the 4 objects.

Also, in the test image, Object2 is the most complex, as it has the highest complexity value (2.772153 units) among the 4 objects while object1 is the least complex, as it has the least complexity value (2.376735 units) among the 4 objects.

4 Conclusions

The skeleton algorithm shows how different structuring elements can impact the generation of the medial axis of a binary input image. When using the rhombus, the skeleton close to the boundary was seen with a greater detail than the other structuring elements. The vec045 generated a skeleton with the least detail among the three structuring elements.

The pecstrum is useful for pattern recognition as it is different for different images with the same structuring element. Pecstrum is used over size distribution for pattern recognition as it is not a strictly monotonically decreasing function unlike size distribution.

It can be concluded from the pecstral analysis, that the selection of the constant Cn in the distance calculation for pattern recognition plays a significant role in matching the objects in the test image to the objects in the reference image. Since, the area of the reference and the test objects have to be already calculated for pecstrum, the area values can be reused in Cn as the absolute value of their differences, based on observation of the reference and test objects.