CHAPTER-4

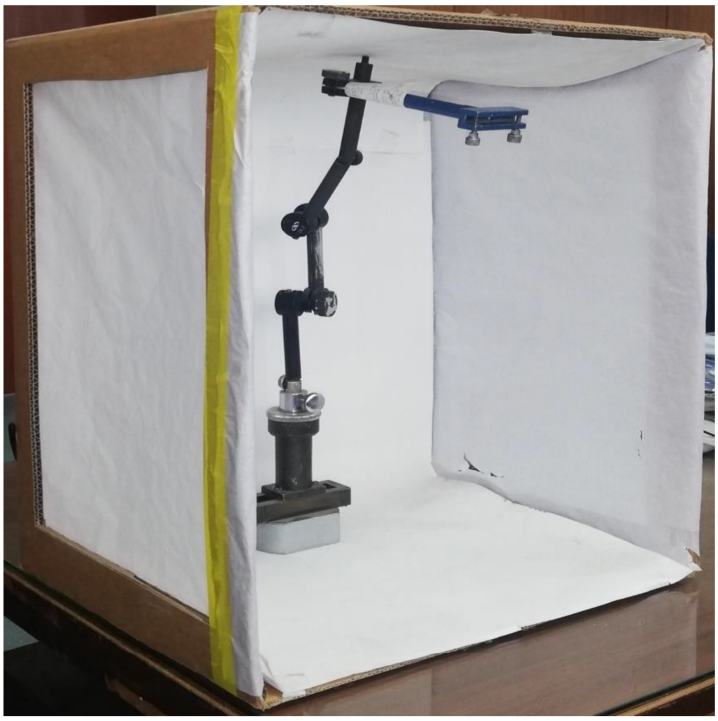
PROJECT WORK

4.1 Introduction

This Project is based on Image processing and the software used for that is MATLAB. By using IPT(Image processing toolbox) processing of hand image is done to get the length of all the fingers. A proper discussion of all the processes, commands used, code and the results are done in this chapter.

4.2 Image Acquisition

A perfect setup is required for image acquisition as image should be shadow free for image segmentation and morphological operations.

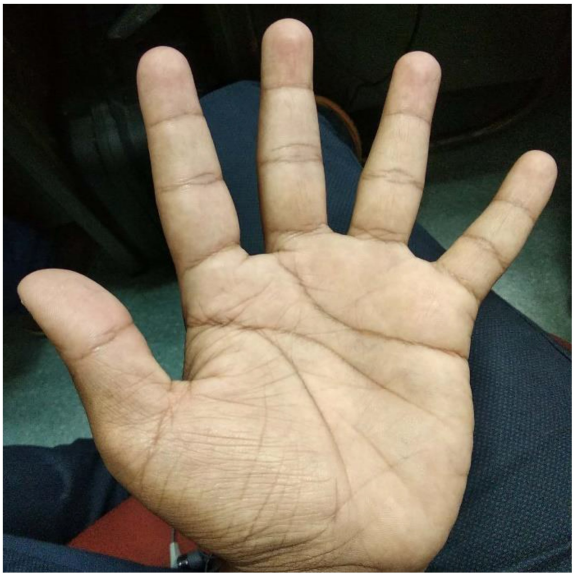


4.3 Commands Used

imread : Read image from graphics file
imshow : Display image
imfinfo : Information about graphics file
rgb2gray : Convert RGB image or colormap to grayscale
graythresh : Global image threshold using Otsu's method
im2bw : Convert image to binary image, based on threshold
bwareafilt : Extract objects from binary image by size
imcomplement : Complement image
imfill : Fill image regions and holes
strel : Morphological structuring element
imopen : Morphologically open image
bwlabel : Label connected components in 2-D binary image
ismember : Array elements that are members of set array
imclearborder : Suppress light structures connected to image border
sprint : Format data into string
title : Add title
regionprops : Measure properties of image regions
sort : Sort array elements
struct2table : Convert structure array to table
find : Find indices and values of nonzero elements
any : Determine if any array elements are nonzero

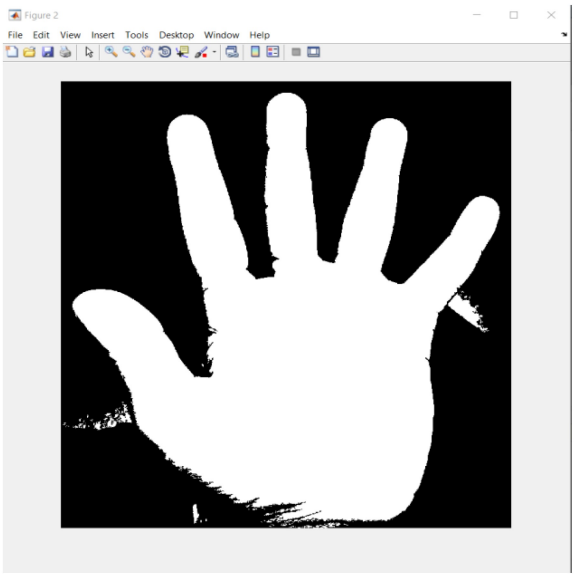
4.4 Steps

1. Read the acquiesced image by imread command and show the same by using imshow command .



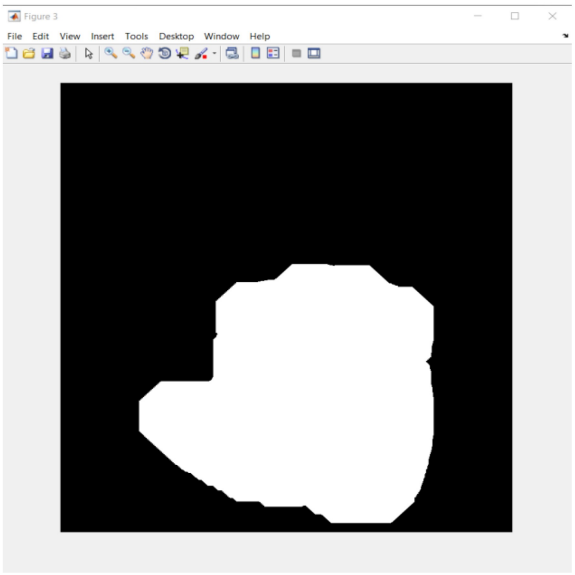
Original image

2. Convert the original image that is RGB image to GRAY image by using rgb2gray command and then do the segmentation by using graythresh and im2bw command to convert the GRAY image to BINARY image.



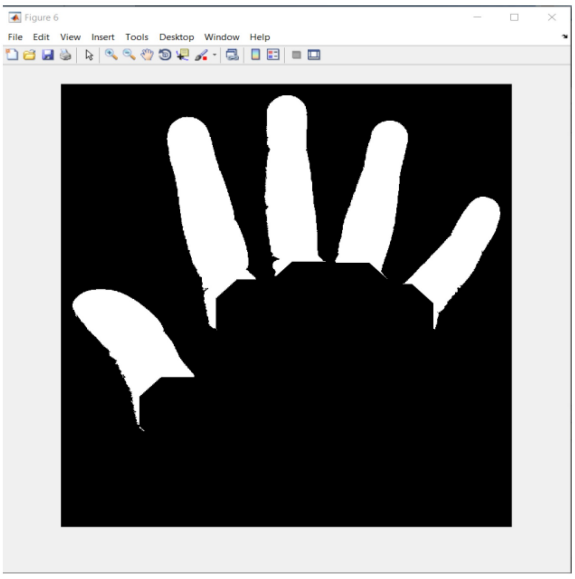
Segmented Image

3. Do the filtering and enhancement of the image to get the proper image for further operations then use the structuring element ‘disk’ of proper diameter to get the required structure by using strel command to use it for morphological operation to get the morphologically opened image.



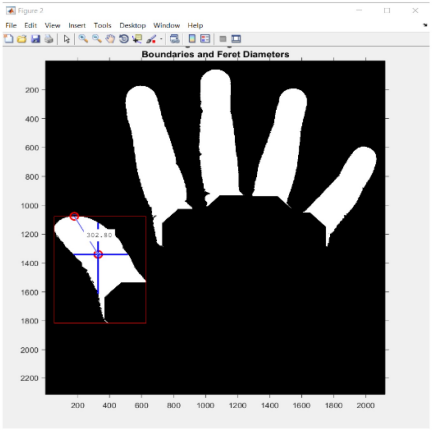
Morphologically opened image

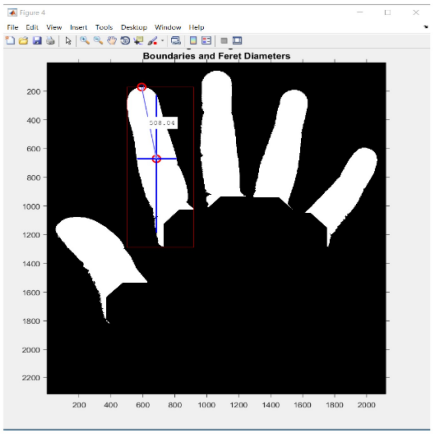
4. Subtract the morphologically opened image from the segmented one to get the required objects (hand's fingers)

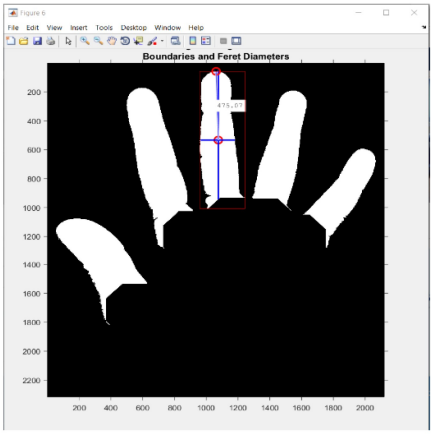


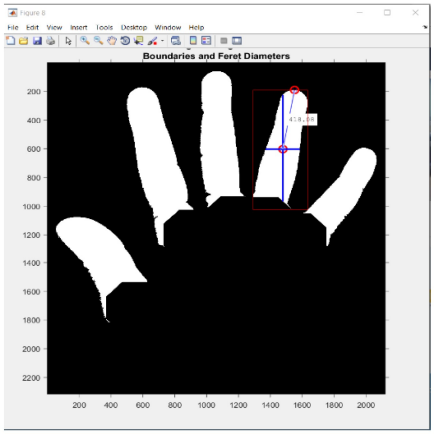
Extracted image

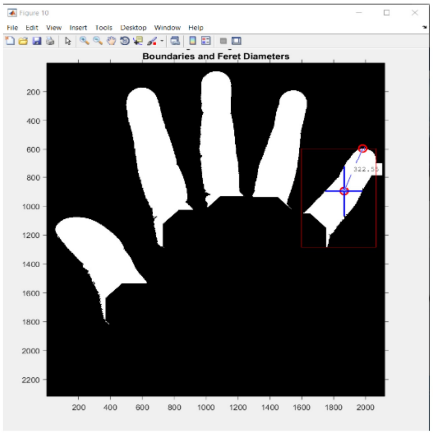
5. Label connected components in 2-D binary image. Use 'for' loop to Extract the boradest contour out of many. Use regionprops command to get the centroid and struct2table to get the centroid coordinates. Similarly find the Extrema of each bolb. Use command imdistline by using coordinates of centroid and extremas to get the distance. Results are as follows:











4.5 Code

```
%Reading image
n=imread('Image.jpg');
%showing the image
imshow(n)
%converting RGB image to GRAY image
I = rgb2gray(n);
%converting GRAY image to BIMARY image using segmentation technique
level=graythresh(I);
m = im2bw(I,level);
imshow(m)
% Extracting objects from binary image
f= bwareafilt(m,100);
% Complementing image
f=imcomplement(f);
% Filling image regions and holes
f= imfill(f,'holes');
imshow(f)
% Using Morphological structuring element for morphological operation
se=strel('disk',500);
```

```
%Morphologically open image
c=imopen(f,se);
imshow(c)
se2=strel('disk',350);
d=imopen(c,se2);
imshow(d)
%subtracting image
x=f-c;
imshow(logical(x))
% Extracting objects from binary image
v=bwareafilt(logical(x),8);
v=bwareafilt(v,5);
imshow(v)
% Labeling connected components in 2-D binary image
[L, num]=bwlabel(v);
for k = 1 : num
    thisBlob = ismember(L, k);
%% To Extract the boradest contour out of many
 % Fill holes
binaryImage = imfill(thisBlob, 'holes');
% Get rid of anything touching the edge of the image
binaryImage = imclearborder(binaryImage);
subplot(2, 2, 4);
imshow(binaryImage, []);
axis on;
hold on;
caption = sprintf('Filled, Cleaned Binary Image with\nBoundaries and Feret
Diameters');
title(caption, 'FontSize', 12);
% Copy the gray scale image to the lower left.
subplot(2, 2, 3);
imshow(n, []);
caption = sprintf('Original Image with\nBoundaries and Feret Diameters');
title(caption, 'FontSize', 12);
axis on;
hold on;
 figure;
imshow(v, []);% Label the image so we can get the average perpendicular width.
labeledImage = bwlabel(binaryImage);
% Let's find the areas
props = regionprops(labeledImage, 'Area');
allAreas = sort([props.Area], 'descend');
% Measure the area
measurements = regionprops(labeledImage, 'Area');
caption = sprintf('Original Image with\nBoundaries and Feret Diameters');
title(caption, 'FontSize', 12);
axis on;
hold on;
[labeledImage, numberOfObjects] = bwlabel(binaryImage);
measurements = regionprops(labeledImage, 'Centroid');
```

```
t = struct2table(measurements); % New with 2013 releases
xCentroids = t.Centroid(:,1);
yCentroids = t.Centroid(:,2);
% Find the column and row number nearest to the centroid
xCentroidColumns = int32(xCentroids);
yCentroidColumns = int32(yCentroids);
hold on;
% Plot centroids
for k = 1 : numberOfObjects
  plot(xCentroids(k), yCentroids(k), 'ro', 'Markersize', 10, 'linewidth', 2);
end
% Find vertical and horizontal lines
for k = 1 : numberOfObjects
  thisBlob = ismember(labeledImage, k);
  % Look at column yCentroidColumns(k) and find out
  % the top and bottom line of the blob.
  topRow = find(thisBlob(:,xCentroidColumns(k)), 1, 'first');
  bottomRow = find(thisBlob(:,xCentroidColumns(k)), 1, 'last');
   plot([xCentroidColumns(k), xCentroidColumns(k)], [topRow, bottomRow], 'b-',
'LineWidth', 2);
  % Horizontal lines
  leftColumn = find(thisBlob(yCentroidColumns(k), :), 1, 'first');
  rightColumn = find(thisBlob(yCentroidColumns(k), :), 1, 'last');
  plot([leftColumn, rightColumn], [yCentroidColumns(k), yCentroidColumns(k)], 'b-',
'LineWidth',2);
[labeledImage, numberOfObjects] = bwlabel(binaryImage);
measurements = regionprops(labeledImage, 'Extrema');
t = struct2table(measurements); % New with 2013 releases
xExtremas = t.Extrema(:,1);
yExtremas = t.Extrema(:,2);
for k = 1 : numberOfObjects
  plot(xExtremas(k),yExtremas(k), 'ro', 'Markersize', 10, 'linewidth', 2);
end
h = imdistline(gca,[xExtremas(k) xCentroids(k)],[yExtremas(k) yCentroids(k)]);
end
vertical = any(binaryImage, 2);
horizontal = any(binaryImage, 1);
row1 = find(vertical, 1, 'first'); % Y1
row2 = find(vertical, 1, 'last'); % Y2
column1 = find(horizontal, 1, 'first'); % X1
column2 = find(horizontal, 1, 'last'); % X2
boxY = [row1 row2 row1];
boxX = [column1 column2 column1 column1];
hold on;
plot(boxX,boxY, 'r-');
figure
   imshow(thisBlob, []);
```