MINI-PROJECT #8

GROUP-8

Chirag Bariya | 13EC35015
Diptanshu Jamgade|13EC35021

MINI PROJECT #8
Hand Gesture Recognition

Objective:

To develop an application in HCI and/or sign language

Introduction:

The use of hand gestures is a common means of nonverbal interaction among people and can, therefore, also provide an attractive alternative to cumbersome interface devices. The Human Computer Interaction (HCI) has led to technologies such as head tracking, face and facial expression recognition, eye tracking and gesture recognition. Gesture and gesture recognition terms are heavily encountered in HCI.

Gestures are the motion of the body or physical action by the user in order to convey some meaningful information. Gesture recognition is the process by which gesture made by the user is made known to the system. Through the use of computer vision or machine eye, there is great emphasis on using hand gesture as a substitute of new input modality in broad range applications. With the development and realization of virtual environment, current user-machine interaction tools and methods including mouse, joystick, keyboard and electronic pen are not sufficient.

Hand gesture has the natural ability to represent ideas and actions very easily. Thus using these different hand shapes, being identified by gesture recognition system and interpreted to generate corresponding event, has the potential to provide a more natural interface to the computer system.

Appearance-based models

These models don't use a spatial representation of the body anymore, because they derive the parameters directly from the images or videos using a template database. Some are based on the deformable 2D templates of the human parts of the body, particularly hands. Deformable templates are sets of points on the outline of an object, used as interpolation nodes for the object's outline approximation. One of the simplest interpolation function is linear, which performs an average shape from point sets, point variability parameters and external deformators. These template-based models are mostly used for hand-tracking, but could also be of use for simple gesture classification.

A second approach in gesture detecting using appearance-based models uses image sequences as gesture templates. Parameters for this method are either the images themselves, or certain features derived from these. Most of the time, only one (monoscopic) or two (stereoscopic) views are used.

Challenges

There are many challenges associated with the accuracy and usefulness of gesture recognition software. For image-based gesture recognition there are limitations on the equipment used and image noise. Images or video may not be under consistent lighting, or in the same location. Items in the background or distinct features of the users may make recognition more difficult.

The variety of implementations for image-based gesture recognition may also cause issue for viability of the technology to general usage. For example, an algorithm calibrated for one camera may not work for a different camera. The amount of background noise also causes tracking and recognition difficulties, especially when occlusions (partial and full) occur. Furthermore, the distance from the camera, and the camera's resolution and quality, also cause variations in recognition accuracy.

In order to capture human gestures by visual sensors, robust computer vision methods are also required, for example for hand tracking and hand posture recognition or for capturing movements of the head, facial expressions or gaze direction.

Techniques Used

1) OTSU Thresholding

Otsu's method exhibits the relatively good performance if the histogram can be assumed to have bimodal distribution and assumed to possess a deep and sharp valley between two peaks. But if the object area is small compared with the background area, the histogram no longer exhibits bimodality. And if the variances of the object and the background intensities are large compared to the mean difference, or the image is severely corrupted by additive noise, the sharp valley of the gray level histogram is degraded. Then the possibly incorrect threshold determined by Otsu's method results in the segmentation error. (Here we define the object size to be the ratio of the object area to the entire image area and the mean difference to be the difference of the average intensities of the object and the background)

From the experimental results, the performance of global thresholding techniques including Otsu's method is shown to be limited by the small object size, the small mean difference, the large variances of the object and the background intensities, the large amount of noise added, and so on.

2) Erosion

The erosion operator takes two pieces of data as inputs. The first is the image which is to be eroded. The second is a (usually small) set of coordinate points known as a structuring element (also known as a kernel). It is this structuring element that determines the precise effect of the erosion on the input image.

The mathematical definition of erosion for binary images is as follows:

Suppose that X is the set of Euclidean coordinates corresponding to the input binary image, and that K is the set of coordinates for the structuring element.
Let Kx denote the translation of K so that its origin is at x.

Then the erosion of X by K is simply the set of all points x such that Kx is a subset of X.

The mathematical definition for grayscale erosion is identical except in the way in which the set of coordinates associated with the input image is derived. In addition, these coordinates are 3-D rather than 2-D.

As an example of binary erosion, suppose that the structuring element is a 3×3 square, with the origin at its center as shown in Figure 1. Note that in this and subsequent diagrams, foreground pixels are represented by 1's and background pixels by 0's.

To compute the erosion of a binary input image by this structuring element, we consider each of the foreground pixels in the input image in turn. For each foreground pixel (which we will call the input pixel) we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel coordinates. If for every pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is. If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value.

For our example 3×3 structuring element, the effect of this operation is to remove any foreground pixel that is not completely surrounded by other white pixels (assuming 8-connectedness). Such pixels must lie at the edges of white regions, and so the practical upshot is that foreground regions shrink (and holes inside a region grow).

Erosion is the dual of dilation, i.e. eroding foreground pixels is equivalent to dilating the background pixels.

3) **Closing**

Closing is opening performed in reverse. It is defined simply as a dilation followed by an erosion using the same structuring element for both operations. See the sections on erosion and dilation for details of the individual steps. The closing operator therefore requires two inputs: an image to be closed and a structuring element. Graylevel closing consists straightforwardly of a graylevel dilation followed by a graylevel erosion.

Closing is the dual of opening, i.e. closing the foreground pixels with a particular structuring element, is equivalent to closing the background with the same element

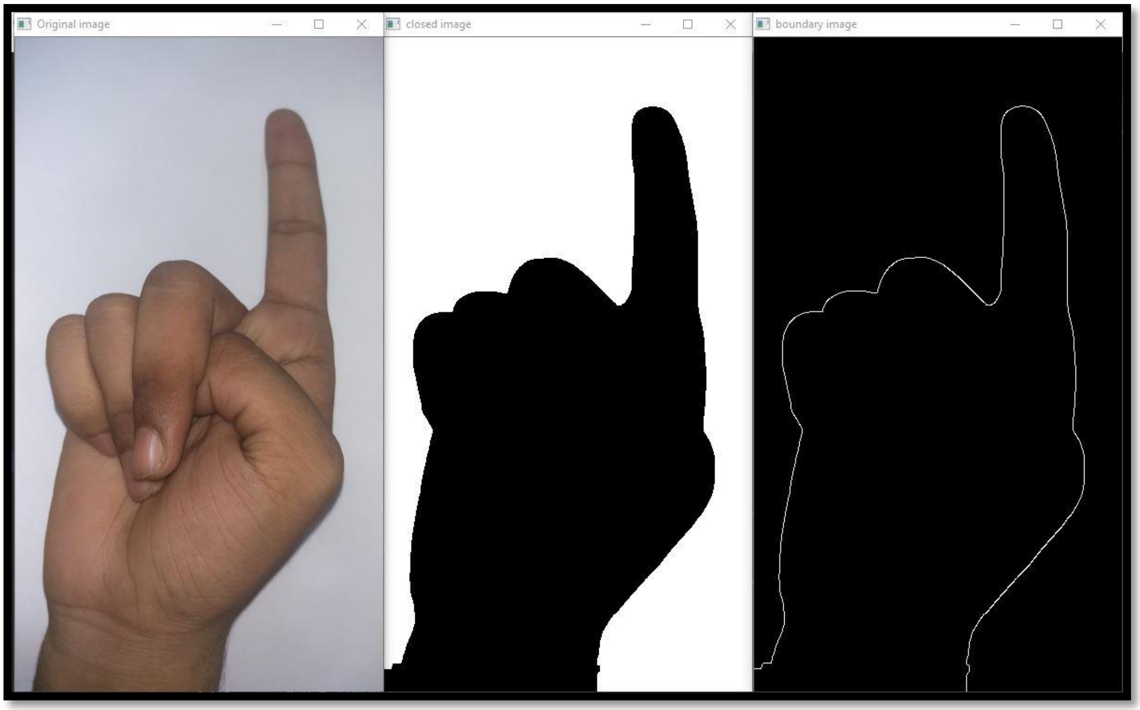
4) **Kenny Edge Detector**

The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.
6. Every step will be described in details as following. The introduction of procedure below is developed based on Prof Thomas Moeslund's lecture note for digital image processing in Indian Institute of Technology.

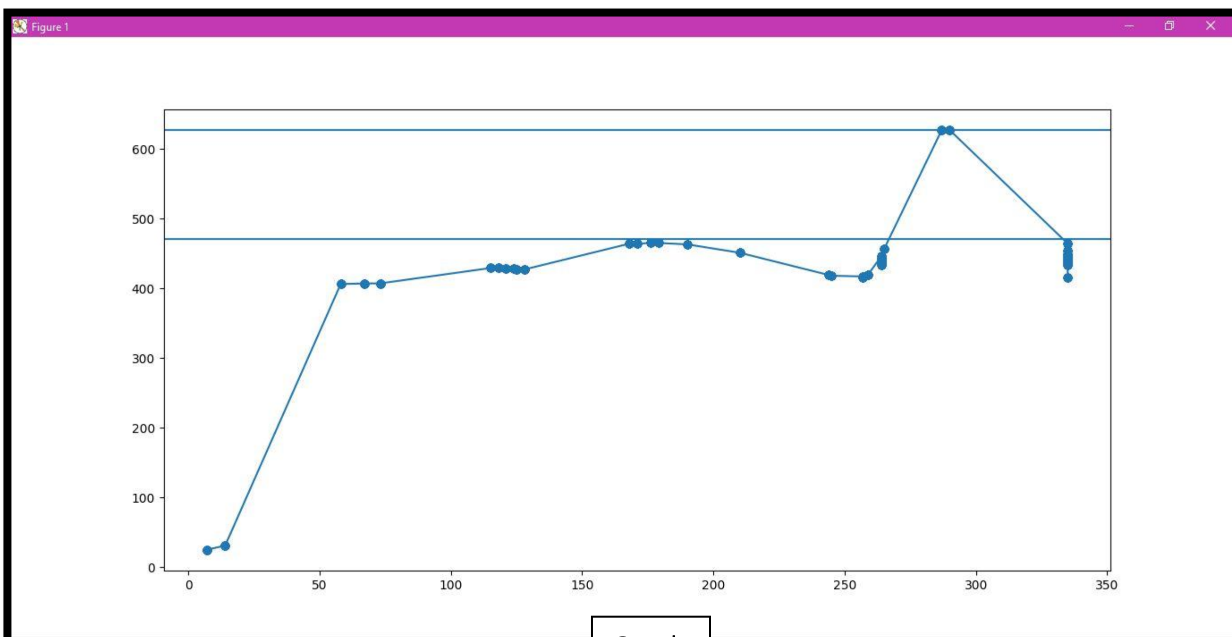
5) **Image Moments**

Moments are well-known for their application in image analysis, since they can be used to derive invariants with respect to specific transformation classes. The term invariant moments is often abused in this context. However, while moment invariants are invariants that are formed from moments, the only moments that are invariants themselves are the central moments. Note that the invariants detailed below are exactly invariant only in the continuous domain. In a discrete domain, neither scaling nor rotation are well defined: a discrete image transformed in such a way is generally an approximation, and the transformation is not reversible. These invariants therefore are only approximately invariant when describing a shape in a discrete image.

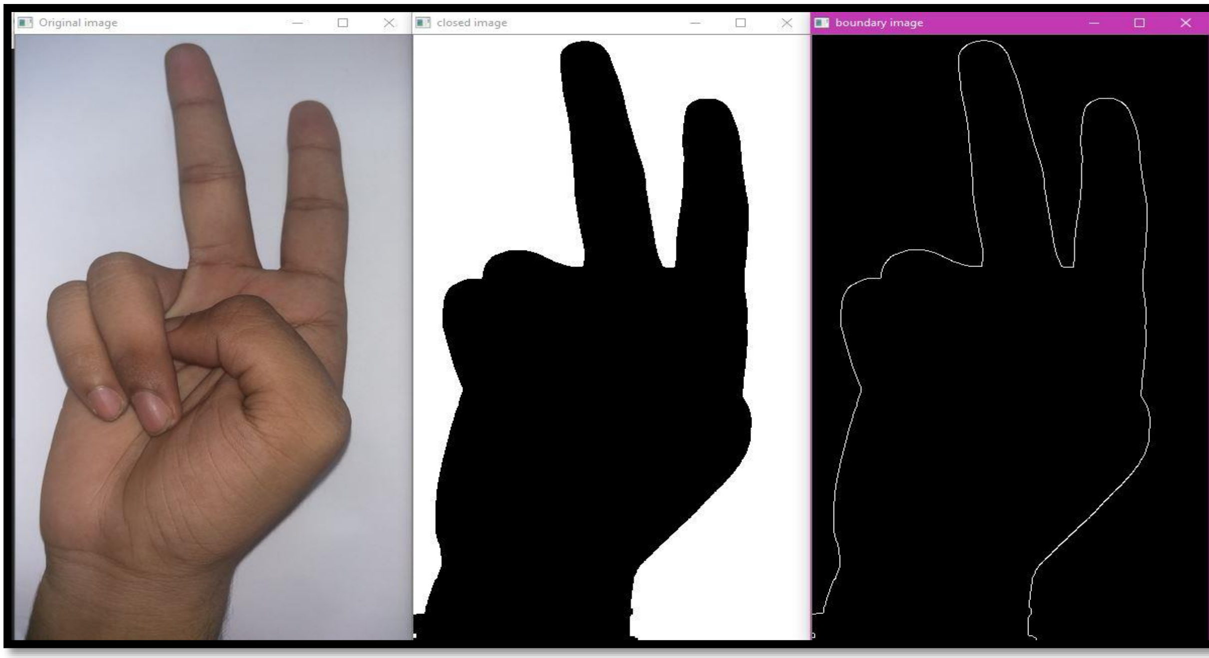


Result

Hand Images



Graph


Original image
closed image
boundary image

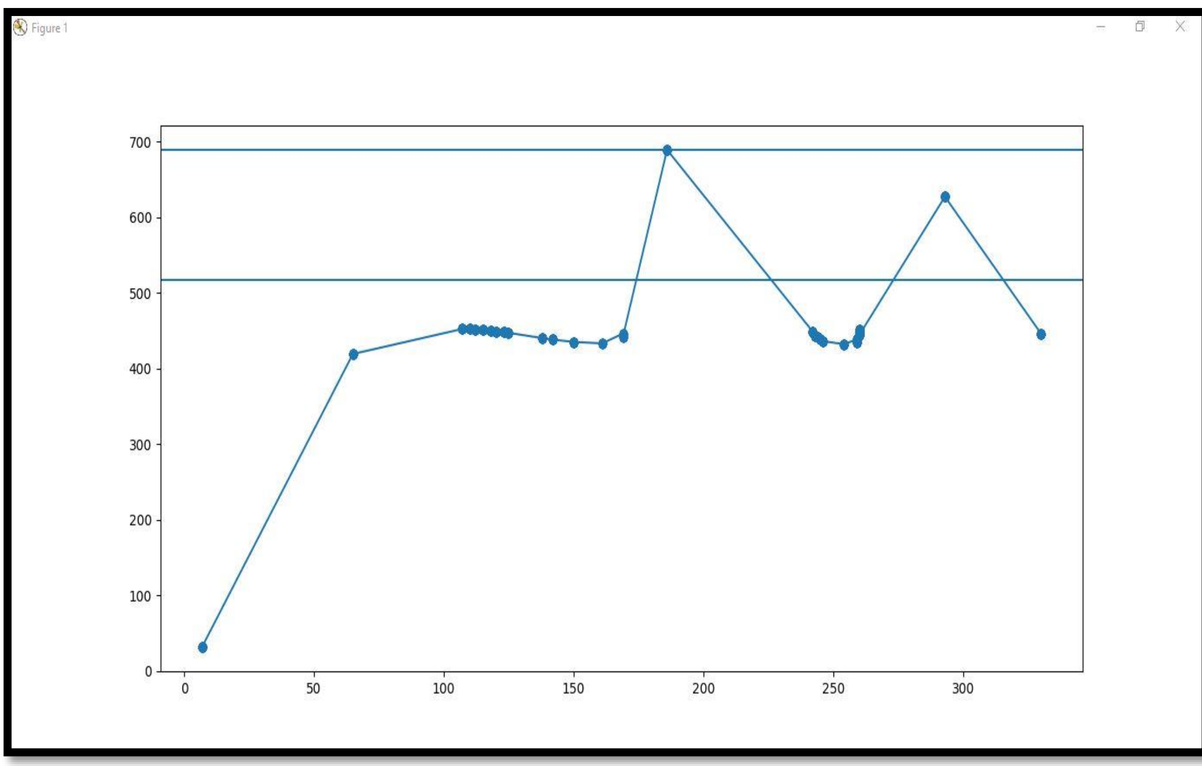
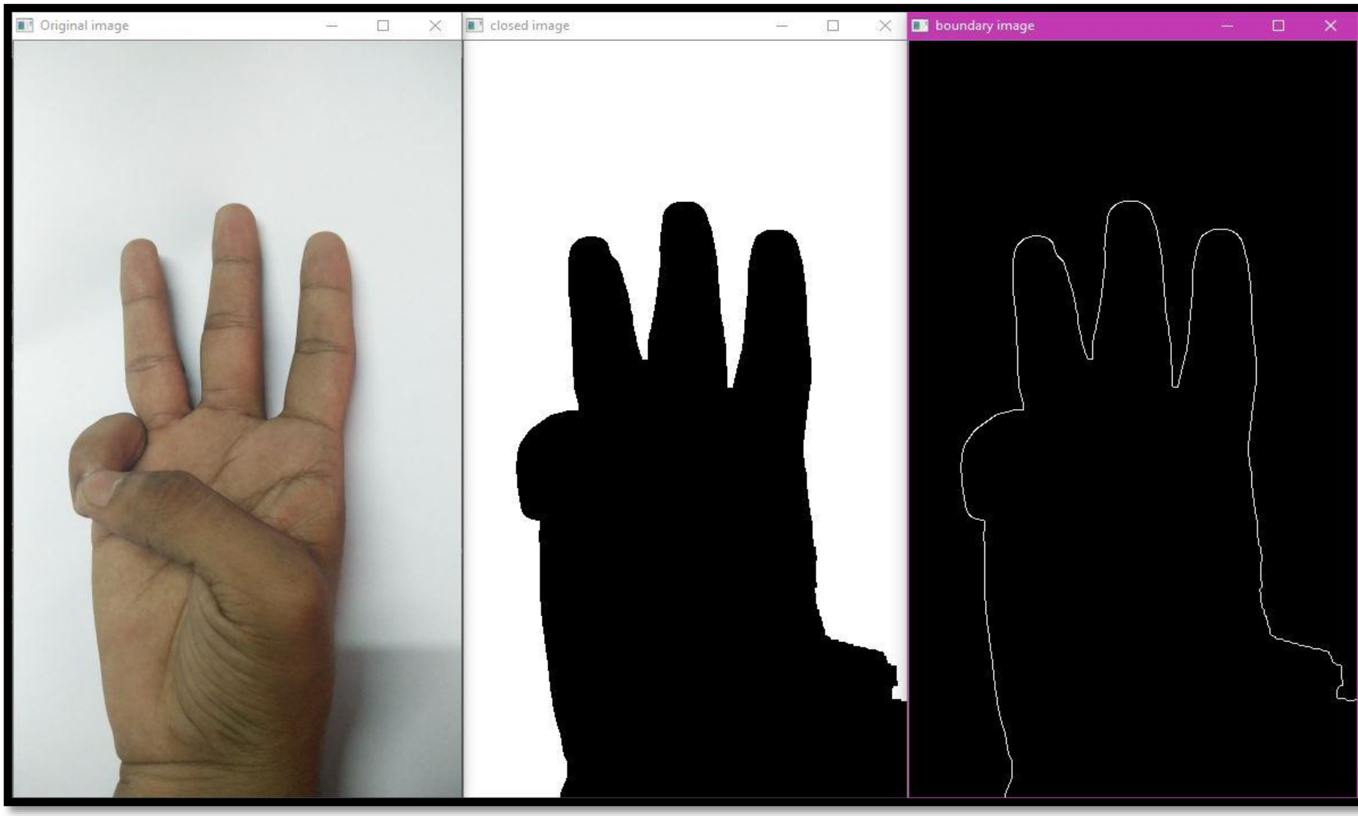


Figure 1
700
600
500
400
300
200
100
0
0
50
100
150
200
250
300


Original image
closed image
boundary image

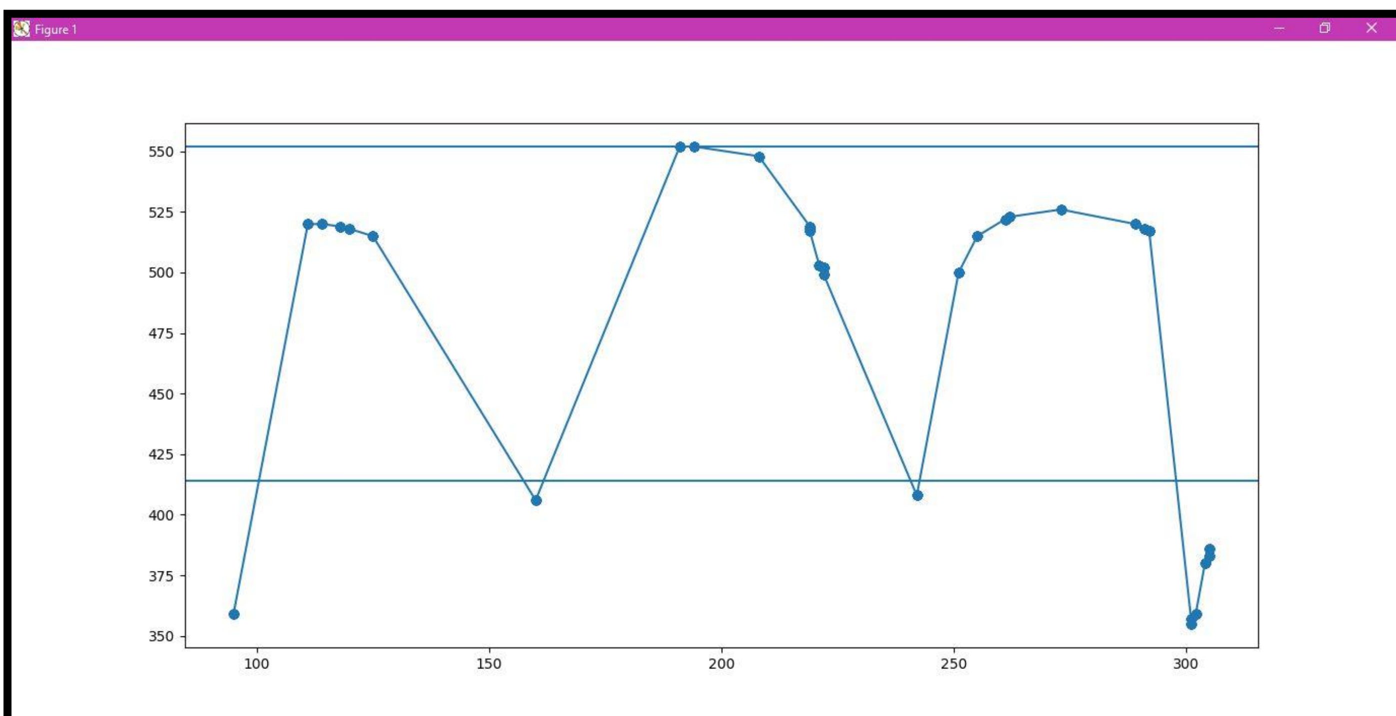
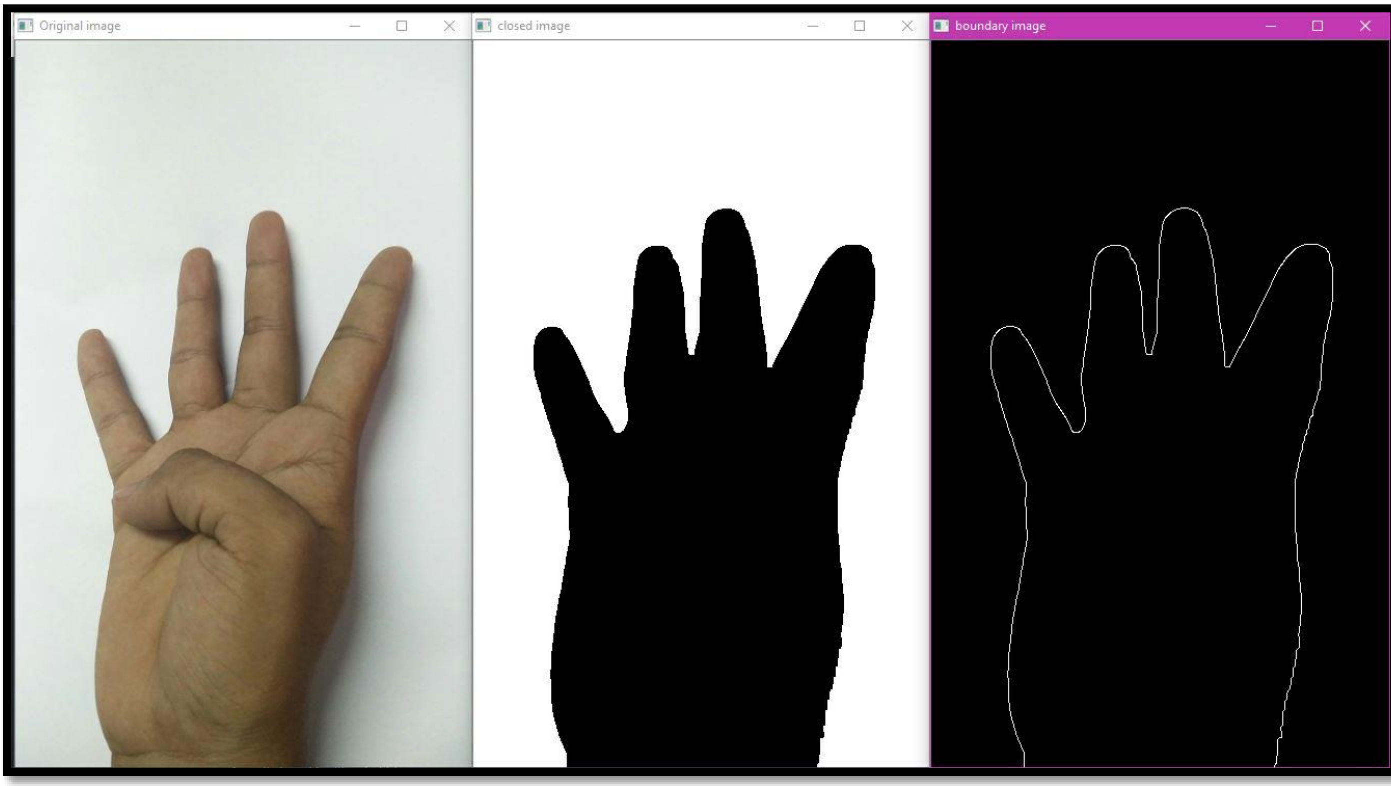


Figure 1
550
525
500
475
450
425
400
375
350
100
150
200
250
300


Original image
closed image
boundary image

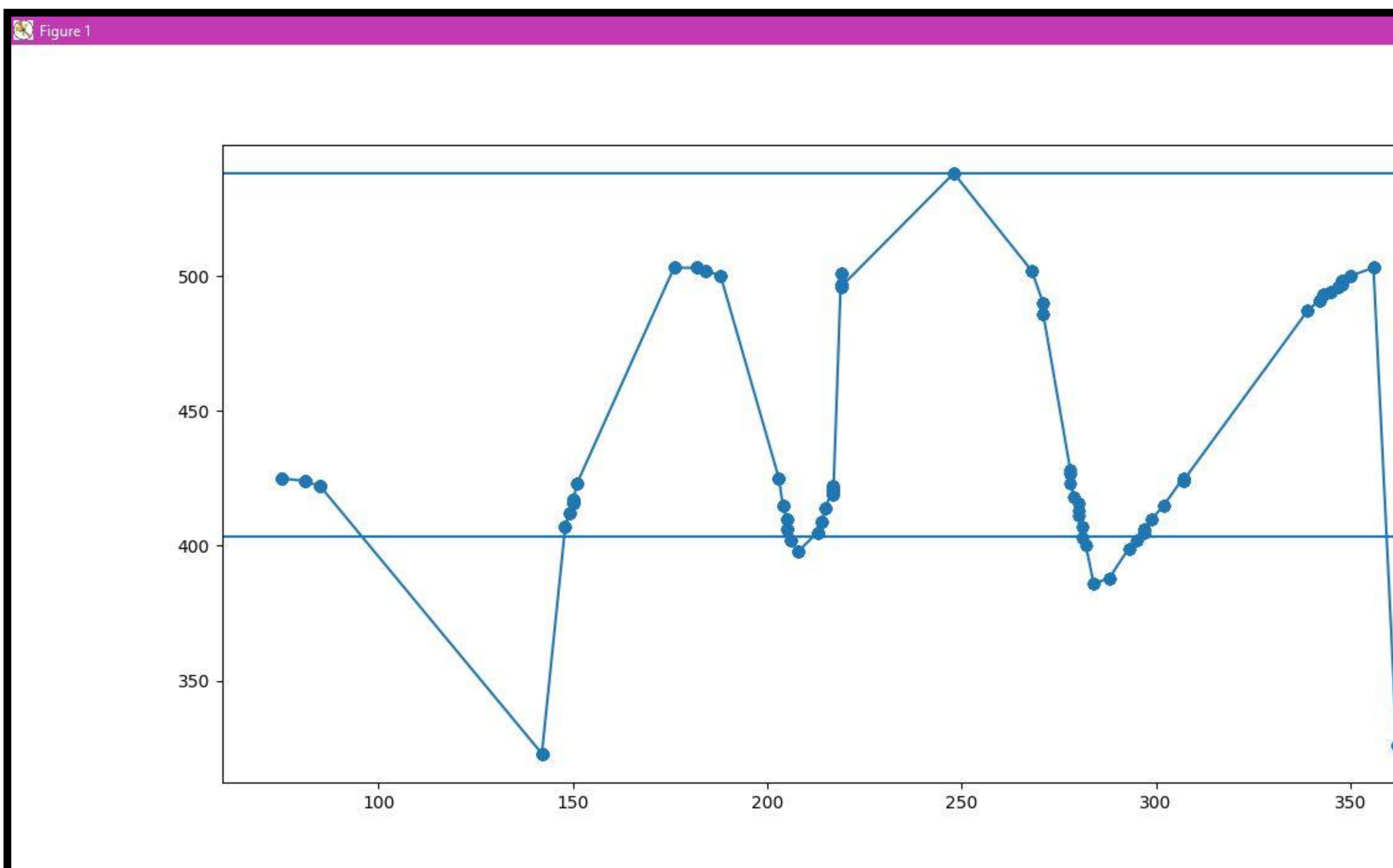
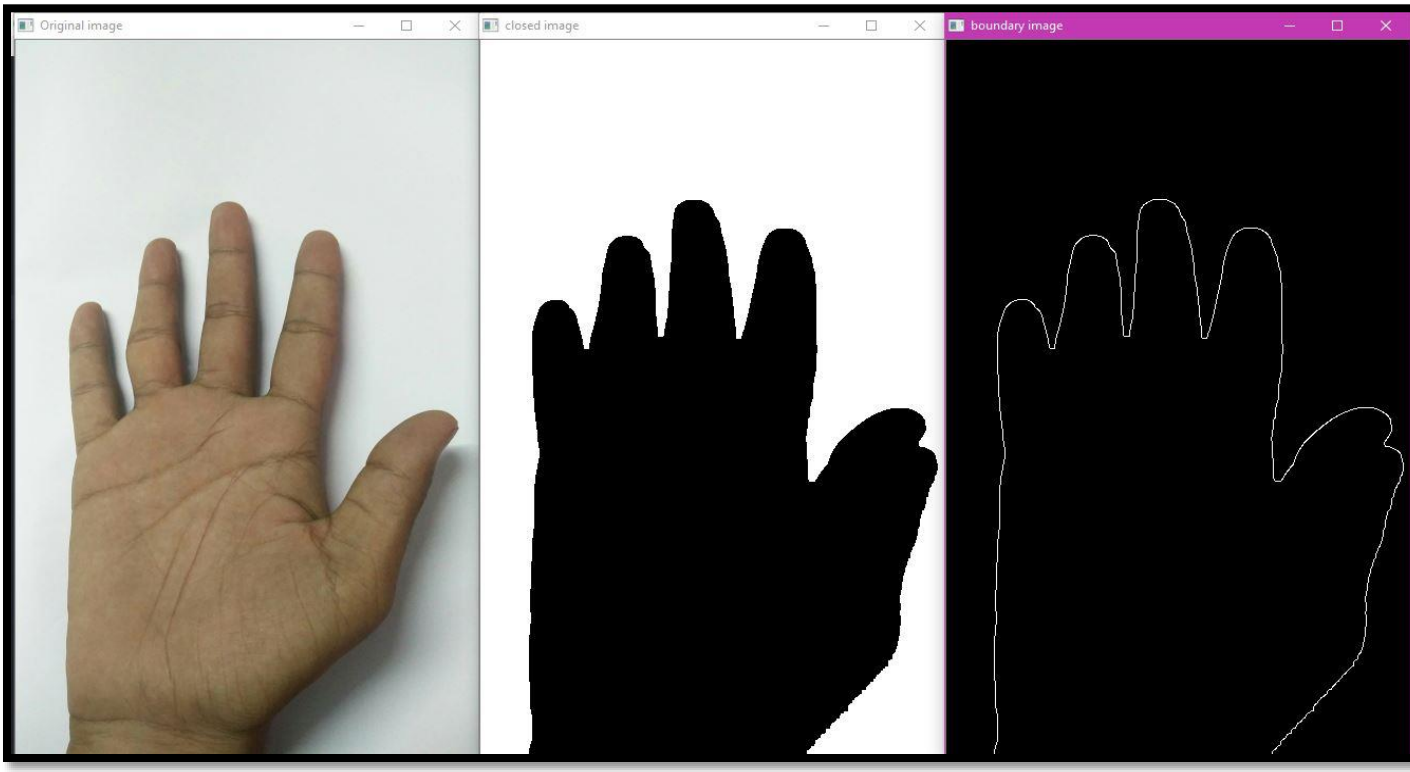


Figure 1
500
450
400
350
100
150
200
250
300
350


Original image
closed image
boundary image

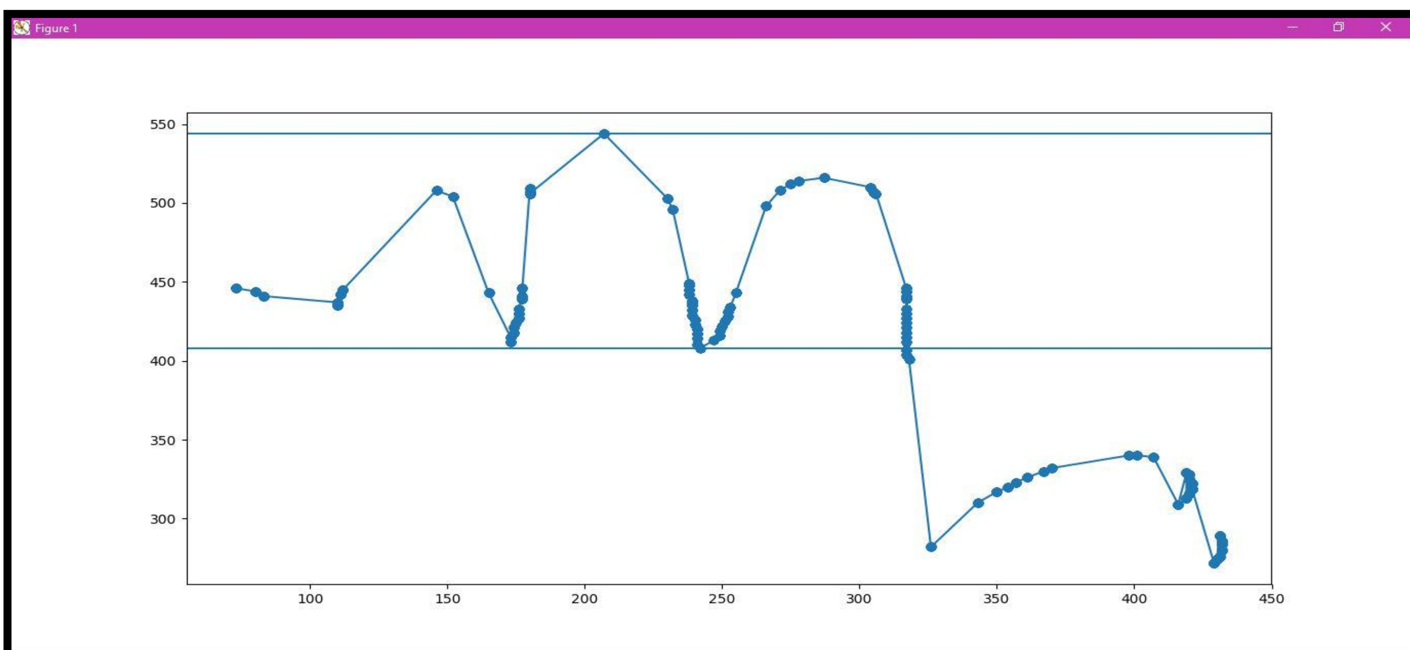


Figure 1
550
500
450
400
350
300
100
150
200
250
300
350
400
450

Bibliography

[1] Meenakshi Panwar and Pawan Singh Mehra , “Hand Gesture Recognition for Human Computer Interaction”, Proceedings of IEEE International Conference on Image Information Processing(ICIIP 2011), Waknaghat, India, November 2011. Meenakshi Panwar and Pawan Singh Mehra , “Hand Gesture Recognition for Human Computer Interaction”, Proceedings of IEEE International Conference on Image Information Processing(ICIIP 2011), Waknaghat, India, November 2011.

[2] Source “The Internet**”**