Gesture Recognition

EE 610 Term Project

Nachiket Deo (10D070006)
Sumeet Fefar (10D070001)
Chandresh (10D070002)

***Abstract*— Prime motive for the project was to develop a prototype which recognizes hand gestures using colored markers on finger tips.**

I. Introduction:

In this project we aim at recognizing hand gestures using 3 different color markers on 3 fingertips. The motion of these markers is tracked for translation, rotation and resizing of an object. The idea used for tracking of the markers is thresholding and filtering in the HSV space.

II. Process:

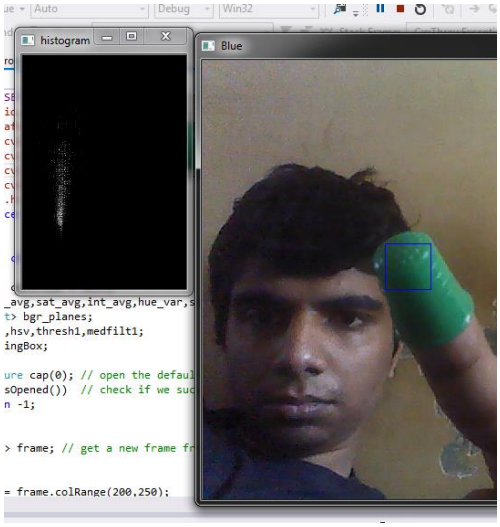
The flow of the project can basically be divided into three different steps:

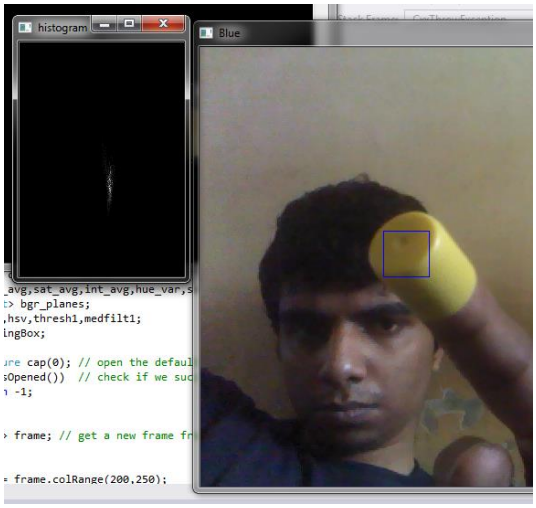
A. Calibration:

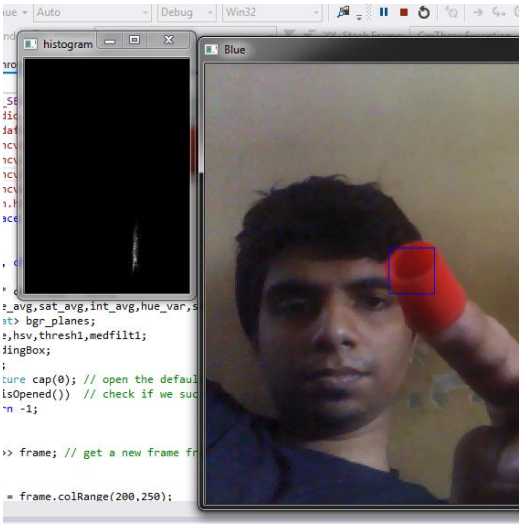
The livefeed from the camera is transformed from RGB to HSV. Now, in order to track the colored markers, thresholds need to be set in the HSV space so as to filter only those pixels which correspond to the color of the respective marker. Hence, the calibration step is carried out to determine thresholds for each of the markers.

Three Bounding boxes are provided in the calibration window where the user holds the three markers.

The subimage in each bounding box is then transformed to HSV. Histograms of Hue, Saturation and Value are then calculated for each bounding box. The respective means and the variances are then used for setting the bounds.







The Hue-Saturation Histogram for Green, Yellow and Red(x axis:Hue ,y axisSaturation):

The thresholds set are between:

$(\mu_h-2\sigma_h, \mu_s-2\sigma_s, \mu_i-2\sigma_i)$ to $(\mu_h+2\sigma_h, \mu_s+2\sigma_s, \mu_i+2\sigma_i)$

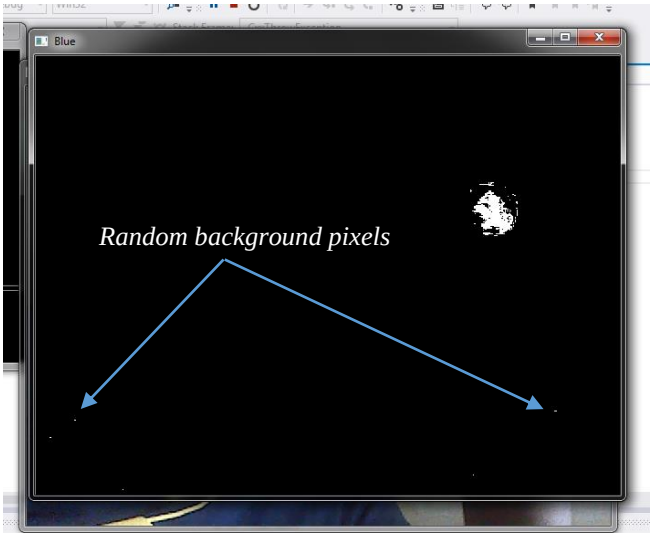
Where μ and σ are the mean and variance, respectively.

Calibration for setting thresholds is a better way than fixed (hard-coded) thresholds as the lightings, colors of the markers, and nature of the camera may vary.

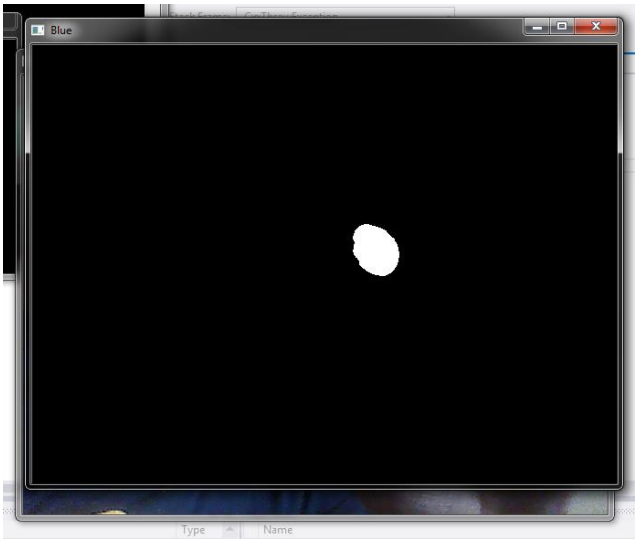
B. Tracking:

Calibration gives the HSV thresholds for each of the markers. The live field is transformed to HSV and thresholded with these values to obtain three binary images, one for each of the markers. In these images, those pixels falling within the range of the HSV threshold are set set to 1 while others are set to 0.

Each of these images are filtered using a 21x21 median filter. The median filter emphasizes the marker's pixels and gets rid of random background pixels falling within the HSV threshold.



Before applying median filter



After applying median filter

Centroids of each median filtered binary image are then calculated to give the positions of the three markers. These are then further used to recognize the gesture.

Tracking the centroids

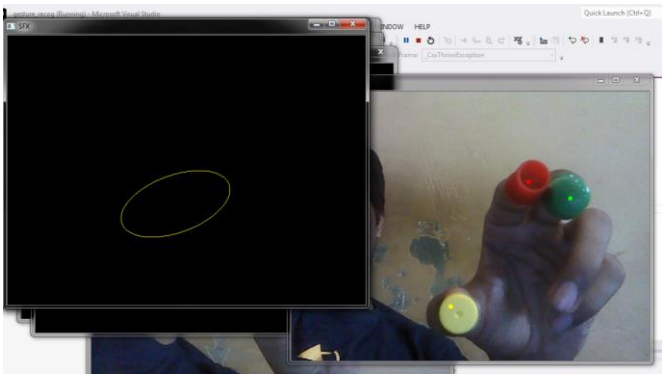
C. Recognition:

The centroid locations obtained from tracking are iteratively updated. These are used for recognizing gestures.

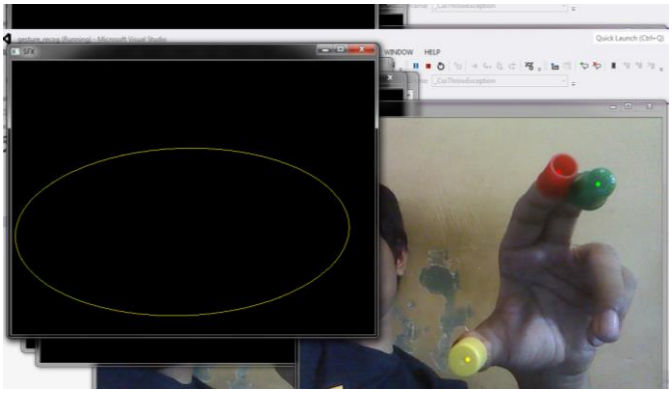
When no two centroids are close to each other, the gesture declared is the default/NO gesture.

When the index finger and middle finger marker centroids are within a pre-determined distance threshold, the **rotate and zoom** gesture is activated. This gesture remains active while these two centroids are within the set distance.

The angle ‘θ’ made by the thumb and midpoint of the index and middle finger is obtained using the slope of the line joining them. Δθ values are obtained for consecutive iterations where the **rotate and zoom** gesture is active. These Δθ values are used for creating the required effect i.e. rotation of the ellipse (clockwise for +ve Δθ, anti-clockwise for –ve Δθ). The distance between the thumb and the mid-point of the middle and index finger ‘l’ is also calculated when the **rotate and zoom** gesture is active. The Δl values for consecutive iterations are used for creating the required effect i.e. zooming in of the ellipse when Δl is –ve and zooming out when Δl is +ve.

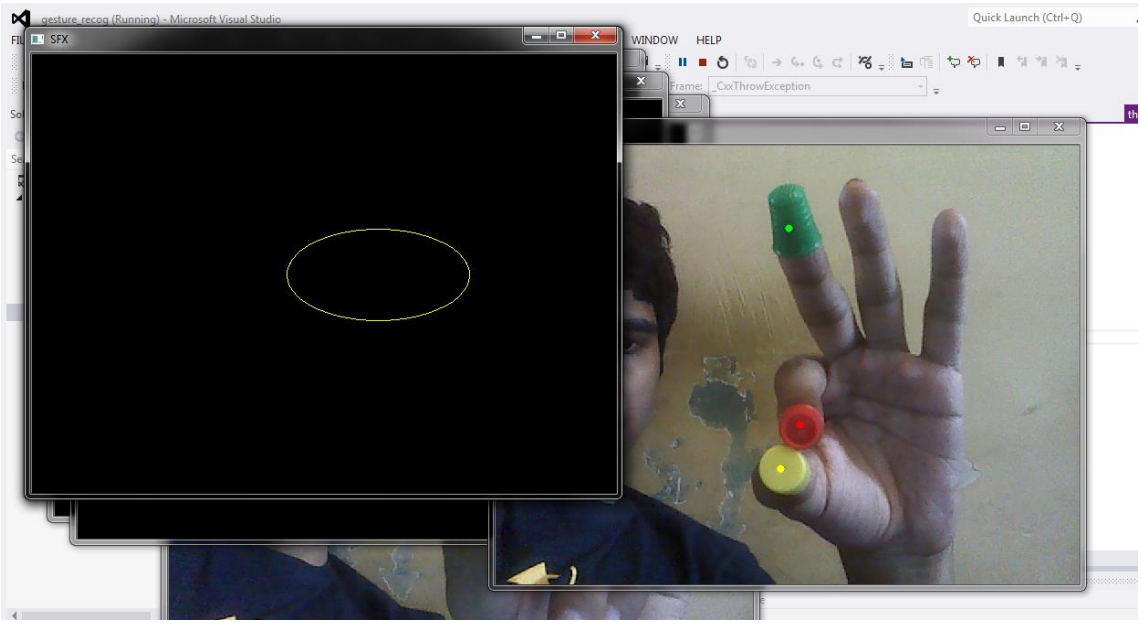


Rotating the ellipse



Zooming in the ellipse

When the thumb and index finger marker centroids are within a pre-determined distance threshold, the **move** gesture is activated.
The midpoint of the thumb and index centroids is tracked when the **move** gesture is active i.e. it's location (x,y) is calculated.
Again Δx and Δy for consecutive iterations are used for creating the required effect i.e. translational motion of the ellipse.



Move Gesture

III. Further Possibilities:

1. Adaptive Filtering:

The HSV thresholds for each of the markers are calibrated only at the beginning. Now, this works well if the overall lightning conditions of the environment remain static over time. Whereas if the lighting conditions keep varying this scheme may not perform well. For this purpose we need to adapt the light conditions in real time. This can be achieved by constantly updating the HSV thresholds (in real time) for each marker.

2. **Contour Detection:**

   Edge and contour detection could be used for further robustness of marker detection. The Canny edge detection algorithm can be applied to the HSV thresholded binary images, followed by contour detection. This would allow for detection and separation of background objects falling within the same hue range of the markers.