**ACKNOWLEDGEMENT**

**ACKNOWLEDGEMENT**

We express our sincere thanks to our Secretary **Dr.R.Prabhakar** and our Principal **Dr.V.Selladurai** for providing us a greater opportunity to carry out our work. The following words are rather very meagree to express our gratitude to them. This work is the outcome of their inspiration and product of plethora of their knowledge and rich experience.

We record the deep sense of gratefulness to **Prof.K.S.Palanisamy**, Head of the Department of Computer Science and Engineering & Information Technology, for his encouragement during this tenure.

We equally tender our sincere thankfulness to our project guides **Mrs.D.Prathiba** and **Mr.B.Arunkumar**, Department of Computer Science and Engineering & Information Technology, for his valuable suggestions and guidance during this course.

During the entire period of study, the entire staff members of the Department of Computer Science and Engineering & Information Technology have offered ungrudging help. It is also a great pleasure to acknowledge the unfailing help we have received from our friends.

It is matter of great pleasure to thank our parents and family members for their constant support and co-operation in the pursuit of this Endeavour.

**ABSTRACT**

**ABSTRACT**

The focus of this research is divided into two different independent fields. First is to recognize hand gestures captured from a webcam in real-time, and then using the gestures to control real-world applications (in our case, apply its output to control Operating System operations). Gestures can be used to communicate much more information by itself compared to mice, keyboard etc. This comes under static hand recognition system. Second is to dynamically track the hand movements and control the mouse using these movements. Further functionalities in static as well as dynamic implementation can be increased to make the project much more effective.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ACKNOWLEDGEMENT** | **1** |
|  | **ABSTRACT** | **3** |
|  | **LIST OF TABLES** | **7** |
|  | **LIST OF FIGURES** | **8** |
|  | **LIST OF ABBREVIATION** | **9** |
| **1** | **INTRODUCTION** | **10** |
| **2** | **LITERATURE SURVEY** | **12** |
| **3** | **SYSTEM SPECIFICATION** | **15** |
|  | 3.1HARDWARE SPECIFICATION | 16 |
|  | 3.2SOFTWARE SPECIFICATION | 16 |
| **4** | **SYSTEM ANALYSIS** | **17** |
|  | 4.1 EXISTING SYSTEM | 18 |
|  | 4.2PROPOSED SYSTEM | 18 |
| **5** | **SYSTEM IMPLEMENTATION** | **19** |
|  | 5.1MODULE LIST | 20 |
|  | 5.2MODULE DESCRIPTION | 20 |
|  | 5.2.1STATIC RECOGNITION | 20 |
|  | 5.2.2 DYNAMIC RECOGNITION | 22 |
|  | 5.2.3 CANNY EDGE DETETCTION | 24 |
|  | 5.2.3 K CURVATURE ALGORITHM | 24 |
| **6** | **RESULTS** | **30** |
|  | 6.1SNAPSHOTS | 31 |
|  | 6.1.1INPUT CHOICE | 31 |
|  | 6.1.2INPUT VALUES | 32 |
|  | 6.1.3 STATIC RECOGNITION OUTPUT | 32 |
|  | 6.1.4 DYNAMIC RECOGNITION OUTPUT | 33 |
| **7** | **CONCLUSION AND FUTURE SCOPE** | **34** |
| **8** | **REFERENCES** | **36** |

**LIST OF TABLES**

**TABLE NO. TITLE PAGENO.**

3.1 HARDWARE SPECIFICATION 16

3.2 SOFTWARE SPECIFACTION 16

**LIST OF FIGURES**

**FIGURE NO. TITLE PAGE NO.**

5.1 FLOW CHART OF STATIC 21

RECOGNITION

5.2 FLOW CHART OF DYNAMIC 23

RECOGNITION

5.3 Illustration of Finger Tip 25

Detection

5.4 CONVEXITY DEFECT 25

5.5 FINGER TIP LOCATION 26

5.6 K CURVATURE RESULT 27

5.7 RAW RESULTS OF K CURVATURE 28

5.8 ILLUSTRATION OF K CURVATURE 28

5.9 CENTROID MEASUREMENT 29

6.1 ILLUSTRATION OF OUTPUT 31

6.2 PROVIDING INPUT VALUES 32

6.3 ILLUSTRATION OF STATIC

RECOGNITION 32

6.4 ILLUSTRATION OF DYNAMIC

RECOGNITION 33

**LIST OF ABBREVIATIONS**

HGR Hand Gesture Recognition

KCA K Curvature Algorithm

CED Canny Edge Detection

HCI Human Computer Interface

**INTRODUCTION**

**1. INTRODUCTION**

Nowadays, computers in all the fields have become very important as they provide an effective and possibly more accurate output. However, the interaction with computers is more or less done in a static manner specifically using mouse and keyboard. We humans have always aimed to make computers more intellectual, intelligent and possessing highly accurate predictive nature. Interactions with computers have to be improvised. Computers will be more intellectual if they start understanding human gestures. Gestures can be used to communicate much more information by itself compared to mice, keyboard etc.

Gesture-based interaction was firstly proposed by M. W. Krueger as a new form of human-computer interaction in the middle of the seventies and there has been a growing interest in it recently. As a special case of human computer interaction, human robot interaction is imposed by several constraints the background is complex and dynamic; the lighting condition is variable; the shape of the human hand is deformable; the implementation is required to be executed in real time and the system is expected to be user and device independent.

Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via using mathematical algorithms. Gestures originate from any bodily motion or state but commonly originate from face or hand. Current focuses in the field include a motion recognition form the face and hand gesture recognition. Many approaches have been made using cameras and computer vision algorithms to interpret sign language. However, the identification and recognition of the posture, proxemics and human behaviours is also the subject of gesture recognition techniques. Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and human than primitive text user interface or even GUI which still limit.

The majority of input to keyboard and mouse will stop gesture recognition enables human to interface with the machine (HMI) and interact naturally without any mechanical devices. Using the concept of gesture recognition is possible to point a finger at the computer screen so that the cursor will move accordingly. This could potentially make conventional input devices such as mouse, keyboards and even touch – screen redundant. Gesture recognition can be conducted with techniques from computer vision and image processing.

**LITERATURE SURVEY**

**2. LITERATURE SURVEY**

**Real time hand tracking and 3D gesture recognition for interactive interfaces using HMM:**

The authors have introduced a system which identifies the hand gestures and captures the gestures of user having glove with colour. For noise elimination this system uses 3D kalman filters and two colour cameras for 3D reconstruction. Hidden Markov Model is used for 3D dynamic gesture recognition. This HMM is used for reducing the spatio-temporal variability. The spatio temporal variability mean the duration and shape of every gesture differs when capturing the image. This system has two parts First is the segmentation. Using HMM it is hard to discover the start and end points so they have reconstructed the model with Baum-Welch algorithm. The second part is the gesture spotting and is implemented by Adaptive Threshold model. The accuracy of 98.75% was achieved with 160 trails of 8 defined gestures.

**Model based Segmentation and recognition of dynamic gestures in continuous video stream:**

In this paper the authors mainly concentrated on the segmentation and recognition of continuous gestures which effect from the spatiotemporal variations. The authors used two types of gestures one is two arm movements for contour extraction, and another one includes a single hand movement hands contour used as the feature vector. They proposed a Multi scale Gesture Model. This model proposes 3 approaches which mainly differs in endpoint Localization. First proposed approach is used to find the end points with Multi scale search and Motion detection strategy. Second proposed approach is used to locate the end points of the fingers roughly with Dynamic Time wrapping. Third approach is based on Dynamic programming. Using these three approaches the recognition of the hand ranges from the 88% to 96%. The author mentioned that the third approach is the best for the recognition of hand gesture in continuous video streams.

**ANN for Gesture Recognition using Accelerometer Data:**

The authors introduced an Artificial Neural network application used for the classification and gesture recognition. The gesture recognition is done through the Wi-Fi remote, this remote will rotate in X, Y, Z directions. To reduce the computational cost and memory consumption the gesture recognition is processed in two levels. In first level User Authentication is done for gesture recognition. Accelerometer- Based gesture recognition method is used. In second level without any kind of signal processing for gesture recognition Fuzzy automata algorithm has been proposed. After recognizing the data of the gestures, the data was normalized and filtered by k-means and Fast Fourier transform algorithm. Using this Dynamic Bayesian Network The recognition accuracy has increased up to 95%.

**A Real-Time Hand Gesture Recognition System for Daily Information Retrieval from Internet:**

In this paper the system is proposed in such a way that with the hand movements the daily information is retrieved from the internet. Principal component analysis is used for the identifying the hand. Using YcbCr colour spaces skin colour detection and CAMSHIFT algorithm is used to detect and track the hand gestures. The position and the region of the hand is detected from the skin detection. It keeps on detecting the skin region until the condition of tracking trigger is enough. The CAMSHIFT algorithm is used when the tracking trigger condition is enough. Segmentation and normalization is done through the PCA. The experimental proves that the 93.1% of accuracy rate is achieved for hand gesture recognition. For processing a single frame the total time taken was in between 0.1sec to 0.3 sec

**Robust Part-Based Hand Gesture Recognition Using Kinect Sensor:**

Inexpensive depth camera -A Kinect sensor is used to build a robust part based hand gesture recognition, in this paper. As Kinect sensors are of low resolution it is hard to identify the hand, but they can capture large objects easily. To deal with the noisy hand gestures which are captured by Kinect sensors, the authors are proposed a novel distance metric known as Finger Earth Movers distance. Only the fingers are matched with FEMD but not the whole hand. The noisy hand shapes are managed in a better way, as FEMD can differentiate the hand gestures with small differences. This system works perfectly and efficiently in uncontrolled environments. The accuracy of 93.2% is achieved with the experimental result.

**SYSTEM SPECIFICATION**

**3. SYSTEM SPECIFICATION**

**3.1 Hardware Specification**

Table: 3.1 Hardware Specifications

|  |
| --- |
| Processor : Dual core |
| RAM : 2 GB |
| Hard disk : 160 GB |

**3.2 Software Specification**

Table: 3.2 Software Specifications

|  |
| --- |
| Operating system : Windows XP/7/8/8.1/10 |
| Language : MATLAB |
| Development kit : MATLAB R2012a |
| Development tool : MATLAB |

**SYSTEM ANALYSIS**

**4. SYSTEM ANALYSIS**

**4.1 Existing System**

The human computer interaction is the most opportunistic field to work on. The HCI includes the interaction using the sign languages. The interaction with the machine by using the sign language needs the accurate recognition of the hand gesture by the system. Various hand gesture techniques can be divided in two categories i.e. vision based and the data glove based technique. The data glove based technique needs the data glove i.e. extra hardware; it leads to the enhanced cost of the system. Due to this the vision based technique are used. This paper implements a vision based hand gesture recognition technique by using the neural network and the SIFT. The rest paper is divided in four sections. First section describes the related work i.e. the work already has been done in the domain. The next section introduces the proposed system. The proposed system is implemented using the MATLAB in the next section of the paper. The final section describes the results and the future scope.

**4.2. Proposed system**

The proposed work completes the process in two phases one is the training phase and the other is testing phase. The work is also explained by using the following flowchart. The features of the input gesture are extracted by using the SIFT i.e. scale invariant feature extraction. The SIFT is also applied on the each image within the database. The extracted features of the input images are compared with the features of the selected database image. If the number of points matched are greater than the threshold value then image is stored as the matched image. The threshold value is calculated by using the neural network. The neural network gets the fixed threshold and the gesture as the input value in the testing phase and provides the threshold value on the basis of the training done. The neural network is trained by using the database images.

**SYSTEM IMPLEMENTATION**

**5. IMPLEMENTATION**

**5.1. Module list**

* Static Recognition System
* Dynamic Gesture System
* Canny Edge detection
* K – Curvature Algorithm

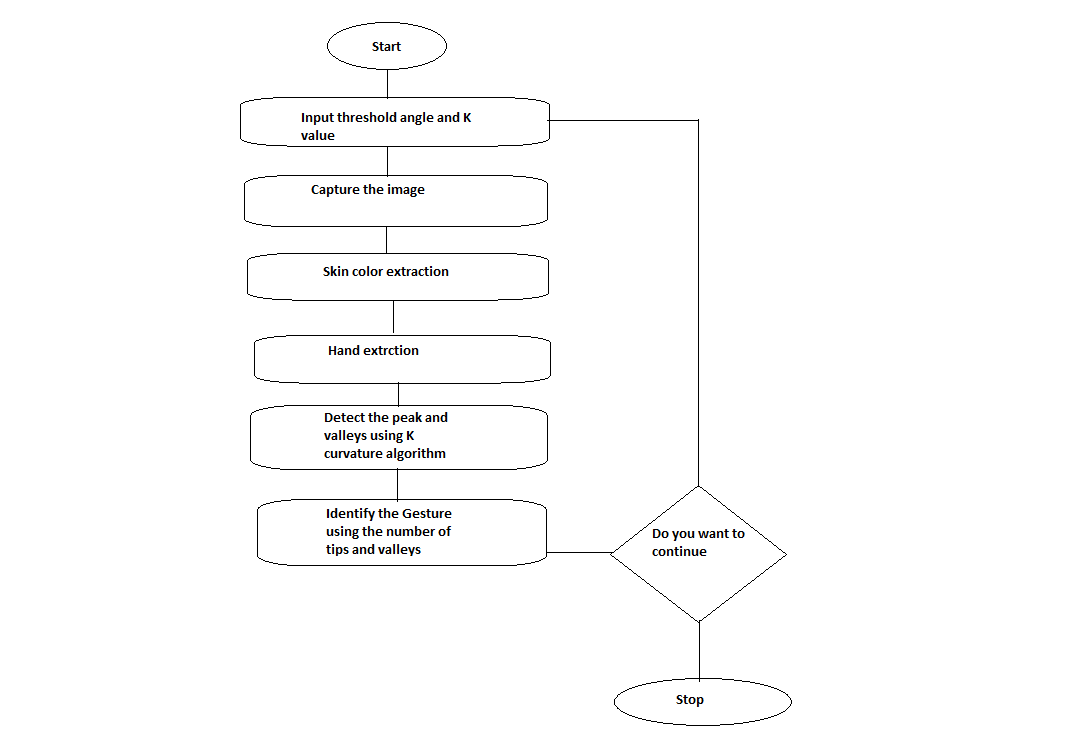
**5.2 Module description**

**5.2.1 Static Recognition System**

computers in all the fields have become very important as they provide an effective and possibly more accurate output. However, the interaction with computers is more or less done in a static manner specifically using mouse and keyboard.

We humans have always aimed to make computers more intellectual, intelligent and possessing highly accurate predictive nature. Interactions with computers have to be improvised. Computers will be more intellectual if they start understanding human gestures. Gestures can be used to communicate much more information by itself compared to mice, keyboard etc.

The static hand gesture recognition is used to find out the static hand movements like the number of fingers in the hand and performs application according to that. The peaks and valleys are extracted using the k curvature method. Using the co- ordinate values of tips and valleys we plotted the captured image. From the number of peaks and valleys we can identify the number of fingers in the current hand gesture.



**Fig.5.1 Static Recognition - Flowchart**

**5.2.2 Dynamic Recognition System**

The main objective of this section is to track the hand movement and feasibly apply it to control mouse pointer movements. This system tracks the hand motion that is it identifies whether the hand moves right, left, up or down. Apart from this some dynamic gestures can be recognized to perform mouse clicking operations.

The centroids of the binary image can be calculated using the MatLab function region props (BW, properties) here the properties will be centroid. The result will be the vertical and horizontal (x, y) coordinates of the centroid. Since the input binary image BW will contain only one white region (hand extracted image) the result of the function region props (BW, properties) will be the centroid of the hand.

By passing this centroid value to the java.awt.Robot we can change the position of the mouse pointer. If the x coordinate of the centroid crosses a threshold value then the dynamic gesture ‘right’ can be detected else ‘left’. Similarly if y coordinate crosses a threshold, the ‘up’ can be detected else ‘down’.

The application of the dynamic hand gesture recognition system is the virtual mouse in which the mouse functions according to our hand movements as in fig. 6 and fig 7.

This section represents the results of the program used to detect the number of fingertips using K Curvature Algorithm. Input is a hand image with 5 fingers. The method recognizes the number of tips and valleys present which if we count will come to know that it is 9.



**Fig.4.2 Dynamic Recognition – Flow chart**

**5.2.3 Canny Edge Detection**

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

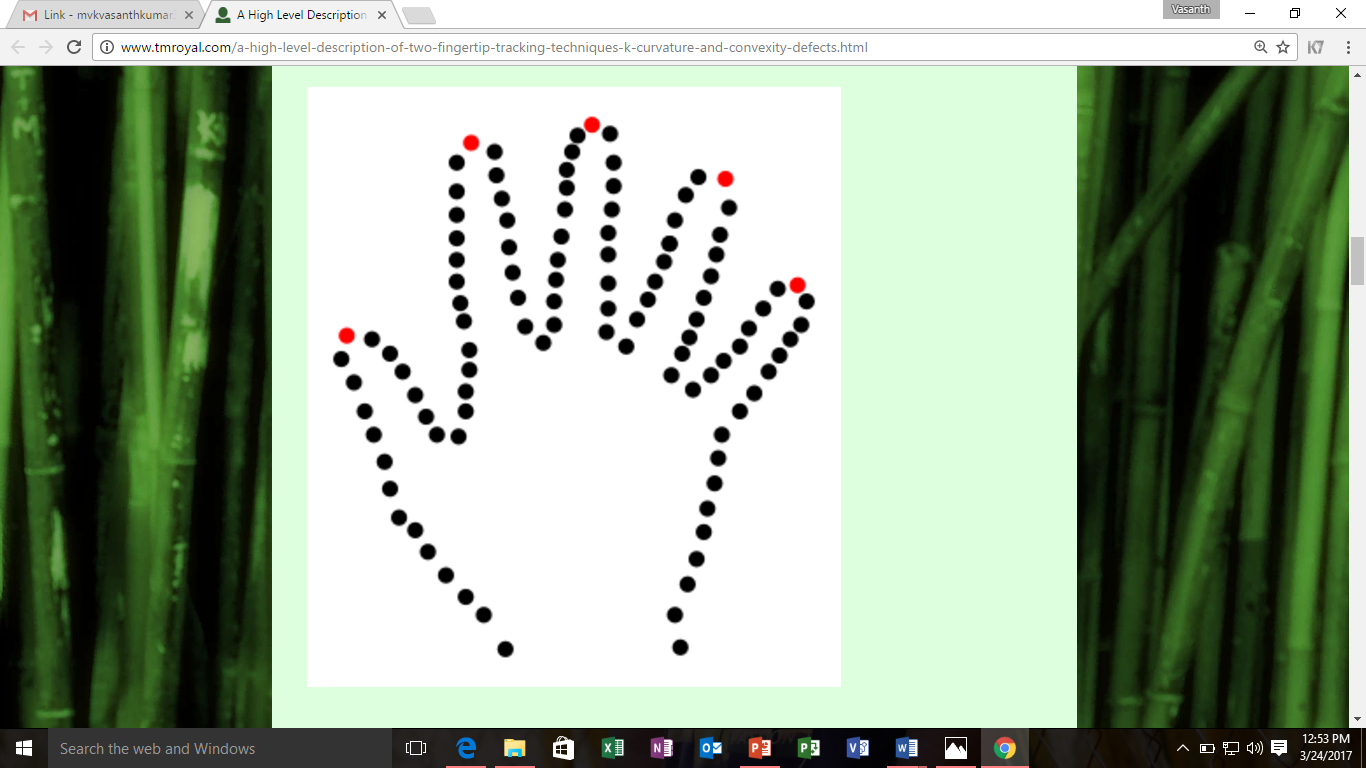
* + 1. **K – Curvature Algorithm**

Fingertip tracking has a number of compelling applications and, thus, is of interest to a great number of people. It is surprising that there are so few high level descriptions of techniques for fingertip tracking. Here I describe two methods for fingertip tracking: k-curvature and convexity defects. While this post is technical, it does not give implementation details. Rather, it is a high level description of these two techniques.

**Prerequisites for Fingertip Tracking**

The first step in tracking fingertips is locating the hand. This can be accomplished through foreground-background segmentation. There are a fair number of techniques for this kind of segmentation. Some of them are rather sophisticated. I have used simple background subtraction in the past. These details are beyond the scope of this post.

When the hand is properly segmented from the background, the hand is represented by a series of points as illustrated below. One possible formulation of the goal of a fingertip tracking algorithm is the identification of the points that correspond to the location of the fingertip relative to the rest of the frame. In other words, the goal is to identify the points marked red in the image below.

****

**Fig.5.3 Illustration of Finger Tip Detection**

**Convexity Defects**

One way of finding these points is to find the convexity defects in this set of points.A curve that is convex is one that curves outward rather than inward as is the case with a concave curve. If a curve is described as a series of points, those points corresponding to the convex parts of the curve are called its convex hull. A convexity defect is a set of points that are not also in the set comprising the convex hull.

**Concave, Convex, Convex Hull and Convexity Defects**

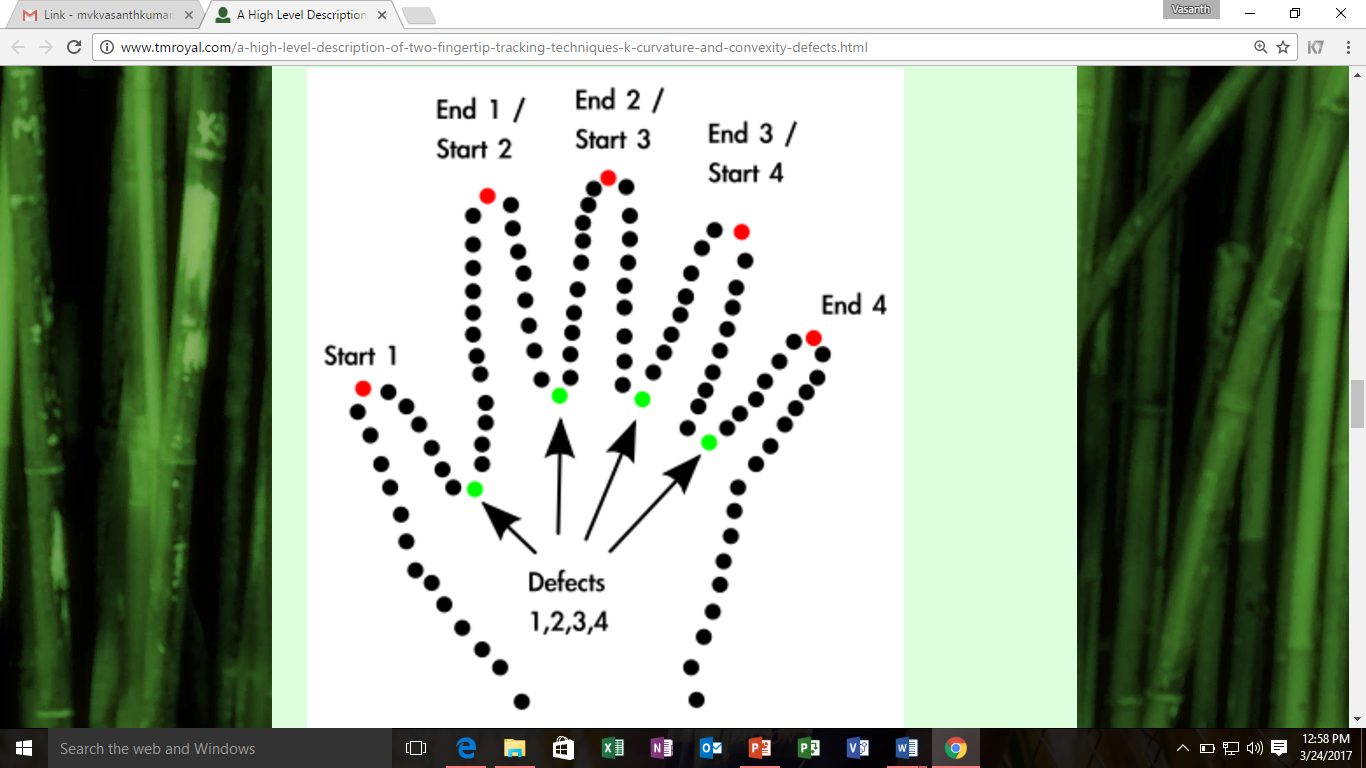


**Fig.5.4 Convexity Defects**

There are two reasons that convex curves are important to finger tracking. First, OpenCV has efficient algorithms for both determining the convex hull and finding its convexity defects. Second, the convexity defects of the points that represent the outline of a hand give clues as to the location of the finger tips. How is this accomplished?

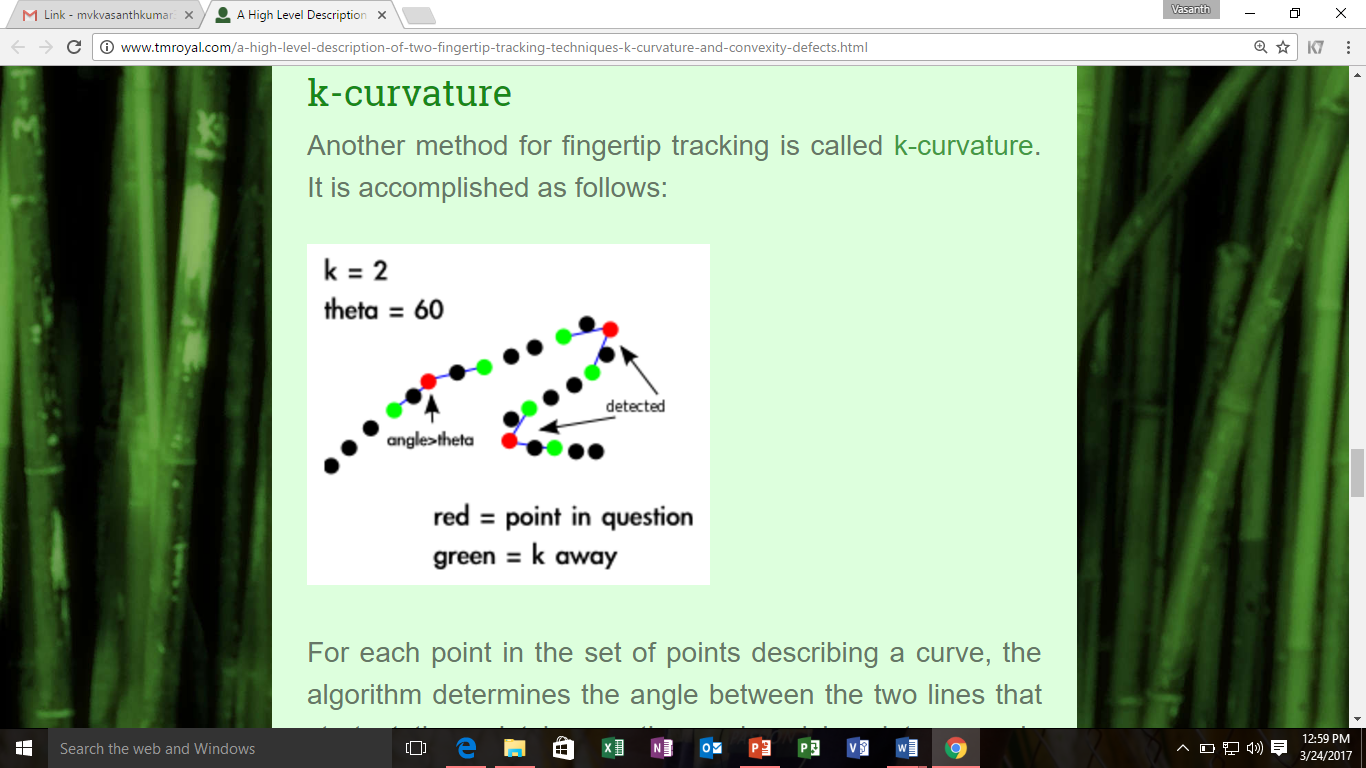
Below is an illustration of the outputs of OpenCV's convexity defect functions. It identifies those places on the curve that are concave. In the case of the hand, this is the space between fingers. Most importantly, the end points of these convexity defects correspond to fingertip locations.

There is an important limitation of using convexity defects for fingertip tracking. Given a hand with a single finger, there are no convexity defects. This means that this approach requires more than one finger to be extended to function properly.



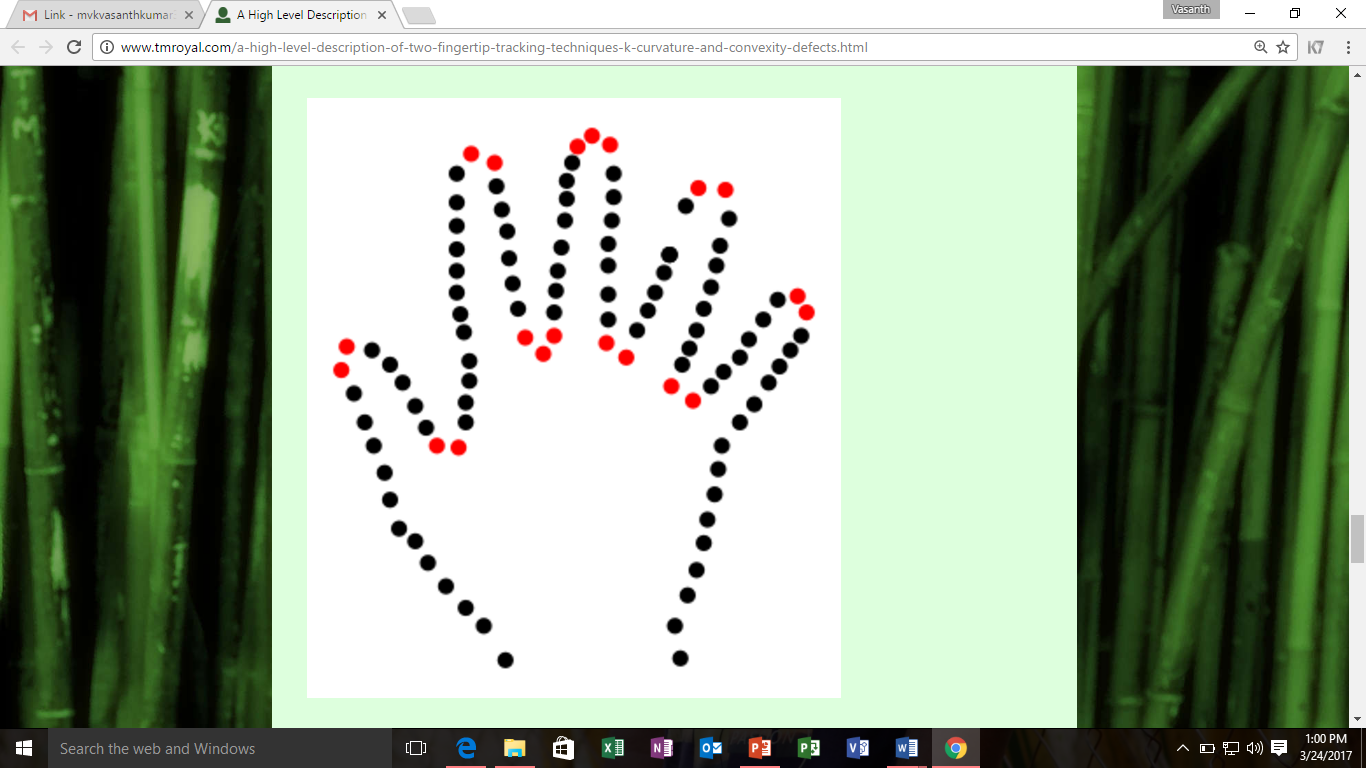
**Fig.5.5 Fingertip location**

**k-curvature**

Another method for fingertip tracking is called k-curvature. It is accomplished as follows:

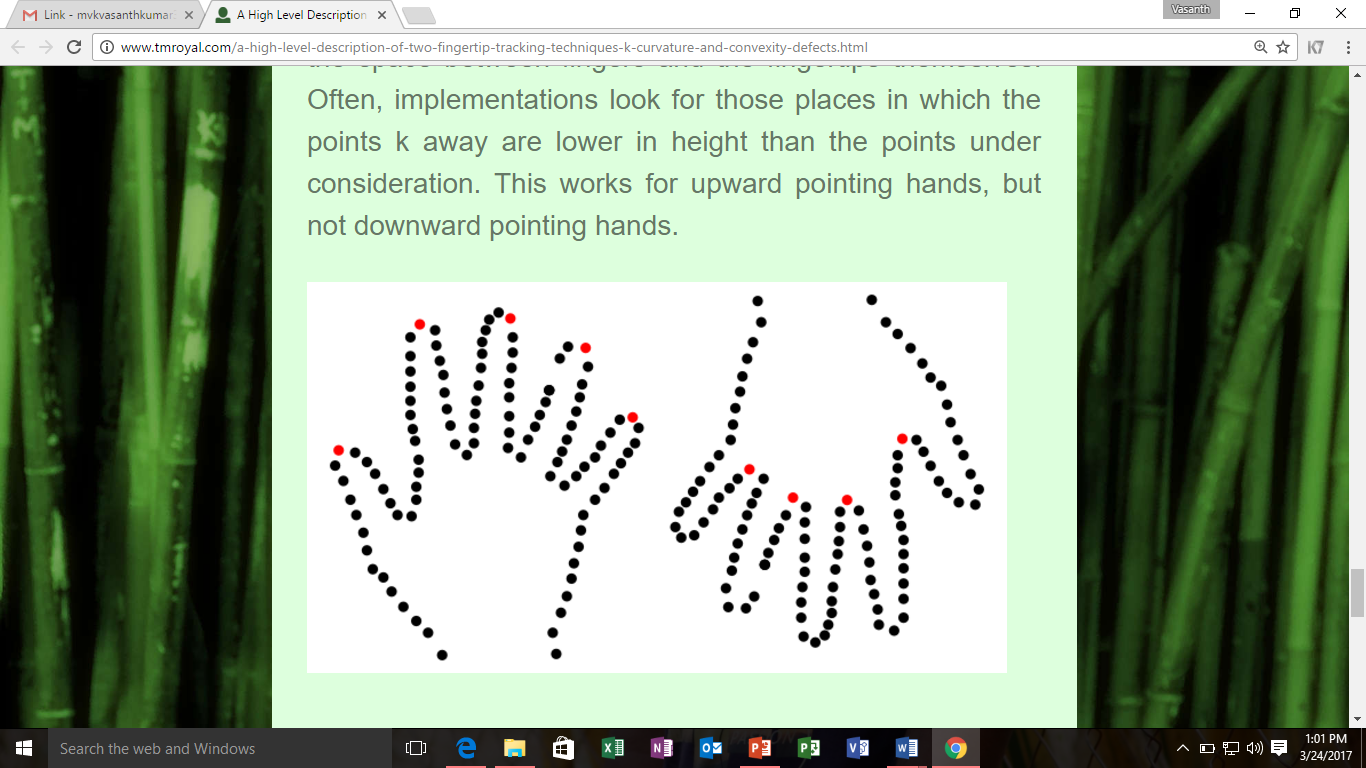
**Fig.5.6 Illustration of k-curvature results for select points**.

For each point in the set of points describing a curve, the algorithm determines the angle between the two lines that start at the point in question and end k points away in either direction. This is illustrated above. If the angle is determined to be below a certain threshold, often 60 degrees, the angle is marked as being of interest. This results of this process with the outline of a hand as an input is illustrated below.



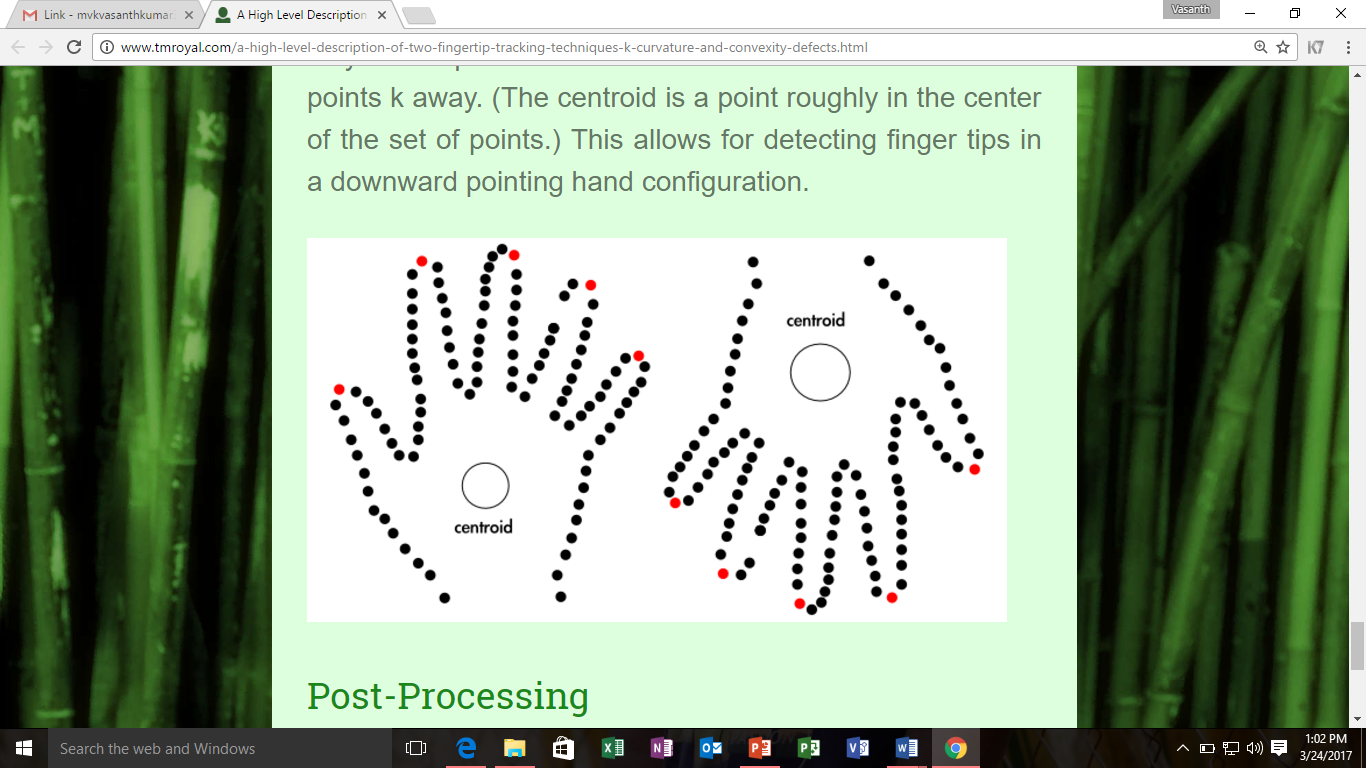
**Fig.5.7 Illustration of the raw results of k-curvature**

At this point, the algorithm has not distinguished between the space between fingers and the fingertips themselves. Often, implementations look for those places in which the points k away are lower in height than the points under consideration. This works for upward pointing hands, but not downward pointing hands.

****

**Fig.5.8 Illustration of k-curvature**

Another way of distinguishing between finger tips is to keep only those points that are further from the centroid than points k away. (The centroid is a point roughly in the centre of the set of points.) This allows for detecting finger tips in a downward pointing hand configuration.



**Fig.5.9 Illustration of Centroid**

**Post-Processing**

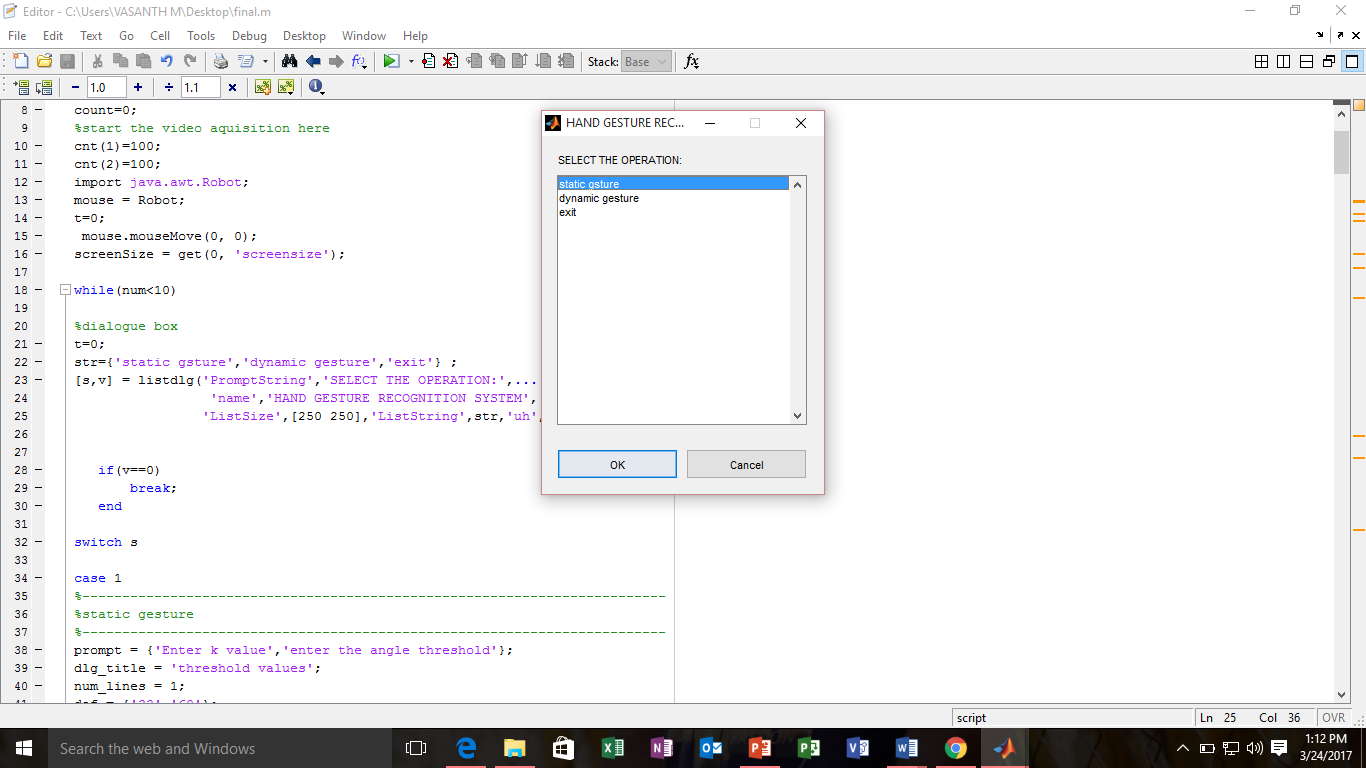
These techniques are far from perfect. The hand easily succumbs to occlusion, meaning that it is easy to hide any given fingertip behind any other part of the hand. Further, these techniques are two dimensional and unable to detect, for example, a finger pointing directly at the camera. Some of these imperfections may be able to be remedied by using machine learning techniques such as particle filters. Others may be remedied through 3d cameras. Ultimately, fingertip tracking using cameras has not been perfected.

**RESULTS**

**6. RESULTS**

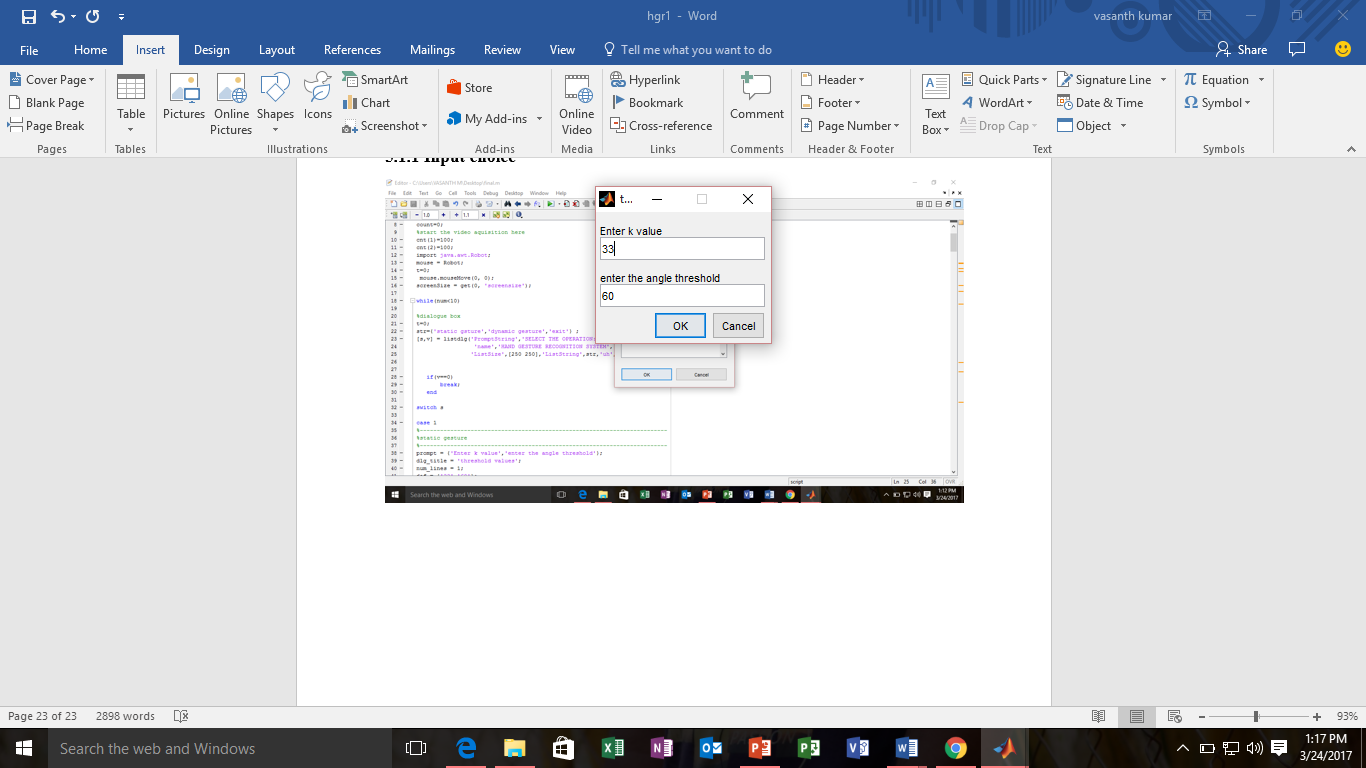
**6.1 Snapshots**

**6.1.1 Input choice**

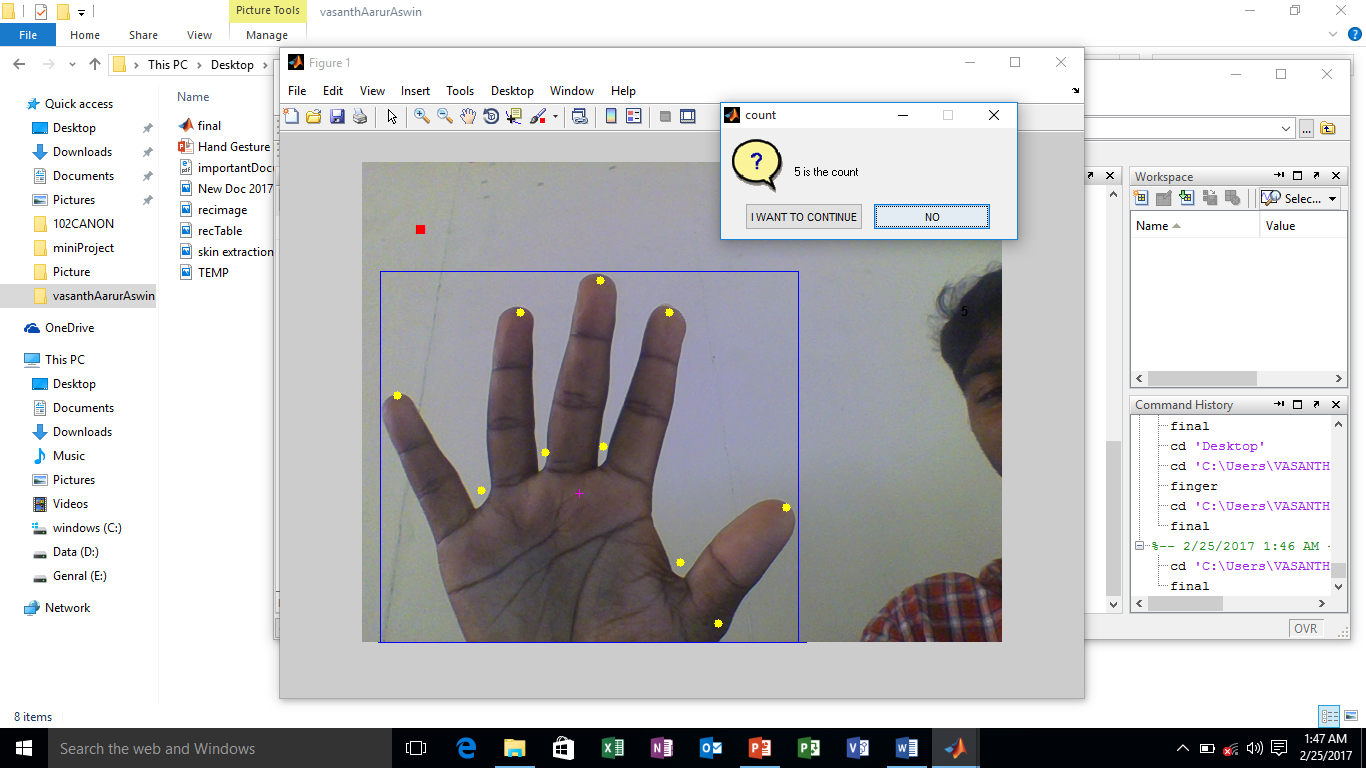


**Fig 6.1 Illustration of output**

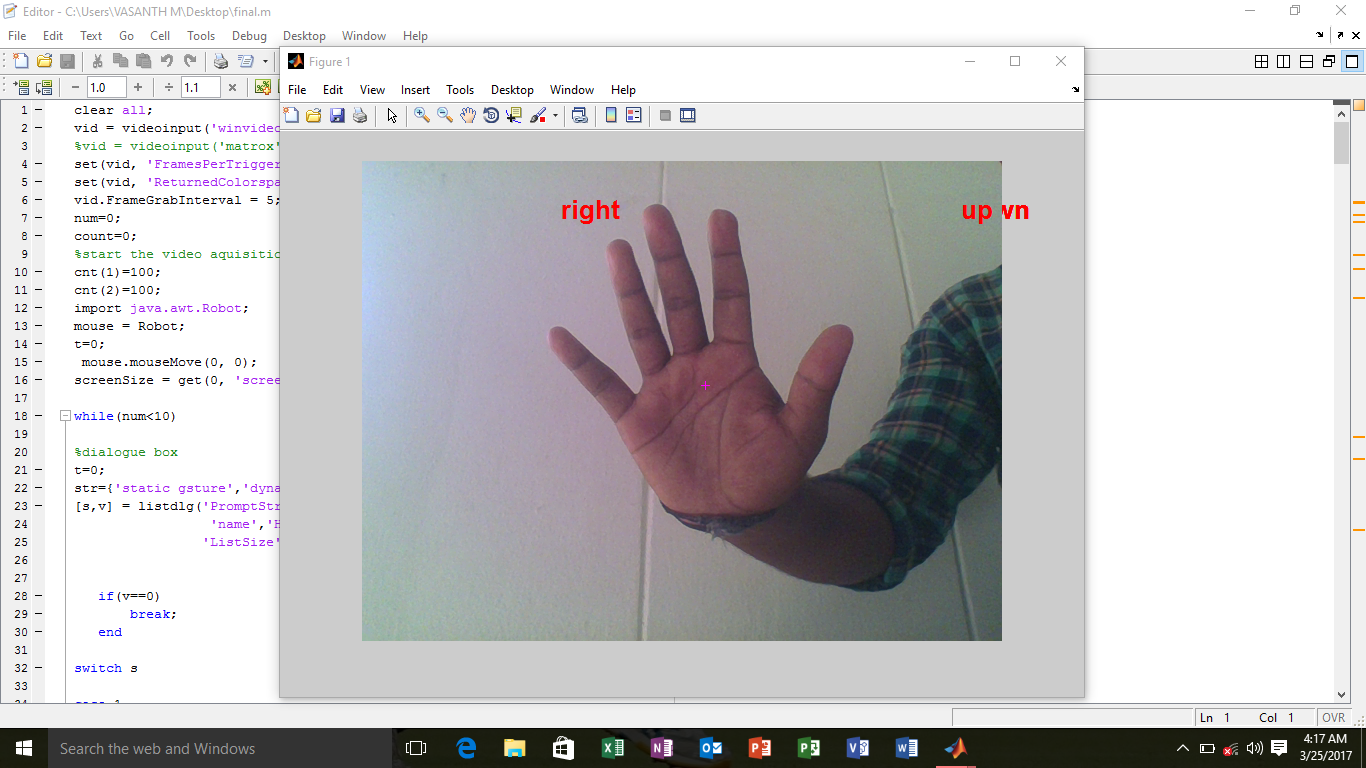
**6.1.2 Input values**

 **Fig 6.2 Providing input values**

**6.1.3 Static Recognition Output**

****

**Fig 6.3 Illustration of Static recognition**

**6.1.4 Dynamic Recognition output**

**Fig 6.1 Illustration of Dynamic recognition**

**CONCLUSION AND FUTURE SCOPE**

**7. CONCLUSION AND FUTURE SCOPE**

The work is completely done by using MATLAB. The system consists of two independent units Static and dynamic that is the virtual mouse. The static hand gesture recognition system uses K curvature algorithm and other system uses the centroid measuring and tracking. The performance of the designed system can be verified using hands of different people.

The major extension to this work can be done to make system able to work at much complex background and compatible with different light conditions. It can be made as an effective user interface and which can include all mouse functionalities.

**REFERENCES**

**8. REFERENCES**

[1]. Nikhil Thakur Nikhil Devle Nitish Singh “Hand Gesture Recognition System (Static and Dynamic)”, *IJSRD - International Journal for Scientific Research & Development| Vol. 2, Issue 2, 2014 | ISSN (online): 2321-0613.*

[2]. C. Swapna, Shabnam S. Shaikh “Literature Survey on Hand Gesture Recognition”, *International Journal of Advanced Research in Computer Science and Software Engineering | Volume 4, Issue 11, November 2014 | ISSN: 2277 128X*

[3]. Canny, J., “A Computational Approach to Edge Detection*”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8:679-714, November 1986.*

[4]. BurakOzer, Tiehan Lu, and Wayne Wolf, “Design of a Real-Time Gesture Recognition System,” *IEEE Signal Processing Magazine, 2005.*

[5]. Chen-Chiung Hsieh and Dung-Hua Liou, “A Real Time Hand Gesture Recognition System Using Motion History Image, “*2nd International Conference on Signal Processing Systems (ICSPS), 2010.*

[6]. D. H. Liou, “A real-time hand gesture recognition system by adaptive skin- color detection and motion history image*,” Master Thesis of the Dept. of Computer Science and Engineering, Tatung University, Taipei, Taiwan,2009.*

[5]. Kwang-Ho Seok, Chang-Mug Lee, Oh-Young Kwon, Yoon Sang Kim, A Robot Motion Authoring using Finger-Robot Interaction”, *Fourth International Conference on Computer Sciences and Convergence Information Technology,2009.*

[6]. Nguyen Dang Binh, Enokida Shuichi, Toshiaki Ejima,” Real-Time Hand Tracking and Gesture Recognition System”, *GVIP 05 Conference, CICC, Cairo, Egypt, 19-21 December, 2005.*

[7]. S.M.Hassan Ahmeda, Todd C.Alexanderb, and Georgios C. Anagnostopoulosb, ”Real-time, Static and Dynamic Hand Gesture Recognition for Human- Computer Interaction*”, IEEE, 2006.*

[8]. T. Maung, “Real-time hand tracking and gesture recognition system using neural networks,” *Proc. Of World Academy of Science, Engineering and Technology, vol. 38, pp. 470-474, Feb. 2009.*

[9]. Vladimir I. Pavlovic, Rajeev Sharma, Thomas S Huang, “Visual Interpretation of Hand Gestures for Human Computer Interaction: A review” *IEEE Transactions of pattern analysis and machine intelligence, Vol 19, NO 7, July,1997.*