***“Towards Global Technical Excellence”***

A Project Report on

**" Virtual Mouse Using Hand Gesture Recognition"**

**For the Degree of Bachelor of Technology**

By

**Mr. Samyak V. Moon**

(16005046)

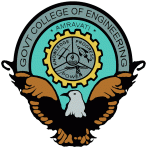
&

**Mr. Shubham P. Surjuse**

(16005047)

Under the Guidance of

**Prof. P. M. Chouragade**



Department of Computer Science & Engineering

Government College of Engineering

Amravati.

(An Autonomous Institute of Government of Maharashtra)

**MARCH 2018-2019**

**CERTIFICATE**

This is to certify that the minor project entitled, **" Virtual mouse using hand gesture recognition ''** which is being submitted and completed successfully by **Mr. Samyak V. Moon** and **Mr. Shubham P. Surjuse** under my supervision and guidance within the four walls of the institute and the same has not been submitted elsewhere for the award of any degree.

|  |  |
| --- | --- |
| **Prof. P. M. Chouragade** | **Prof. A. V. Deorankar** |
| (Project Guide) | (Head of Department) |
| (Dept. of Computer Science & Engineering) | (Dept. of Computer Science & Engineering) |

**Date:**

**Place: Amravati.**

**DECLARATION**

We hereby declare that the project entitled, " **Virtual Mouse Using Hand Gesture Recognition** " was carried out and written by us under the guidance of **Prof. P. M. Chouragade** Department of Computer Science and Engineering, Government College of Engineering, Amravati. This work has not been previously formed the basis for the award of any degree of diploma or certificate nor has been submitted elsewhere for the award of any degree or diploma.

Place: Amravati Samyak V. Moon Shubham P. Surjuse

Date: (16005046) (16005047)

**ACKNOWLEDGEMENT**

We feel great pleasure in expressing our great sense of gratitude and sincere thanks to our guide **Prof. P. M. Chouragade**, for their valuable guidance during the project work, without which it would have been very difficult task for us. We have no words to express our sincere thanks, for valuable guidance, extreme assistance and cooperation extended to all the staff members of our department.

We are also thankful to our Head of Department **Prof. A. V. Deorankar** for providing us all facilities during this project work.

We would thank **Prof. R. S. Dalu**, Principal, Government College of Engineering, Amravati for providing all the facilities at the right period of time.

We would like to thank all the teaching, non-teaching staff members of our department and our college who helped us directly or indirectly for completion of this work successfully.

Expressing our gratitude.

Place: Amravati Samyak V. Moon Shubham P. Surjuse

Date: (16005046) (16005047)

**ABSTRACT**

As the computer technology is growing up, the importance of human computer interaction is rapidly increasing. Most devices use touch screen technology which cannot be affordable to all the applications. A virtual human computer interactive module such as virtual mouse, can be an alternative way for the traditional touch screen. The objective is to create a hand tracking application to interact with system, and develop a virtual human computer interaction module. This module can be used for creating an interface between computer and human using hand gestures. This module is expected to gain abundance of attention because of its applications for human machine interface and its most futuristic design.

***Keywords*** *- Virtual mouse, Human-computer Interaction (HCI), Hand gesture, Image Processing*

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **TITLE** | **PAGE NO.** |
|  | **List of Figures** | 7 |
| 1 | **INTRODUCTION** | 8 |
|  |  |  |
| 2 | **LITERATURE** | 9 |
| 3 | **PACKAGES USED**  3.1 OpenCV  3.2 NumPy  3.3 wxPython  3.4 Pynput | 10  10  11  12  12 |
| 4 | **IMPLEMENTATION**  4.1Introduction to the system  4.2System description  4.2.1Capturing real time video using Web-Camera  4.2.2Flipping of each image frame.  4.2.3Finding the region of the image and calculating its centroid.  4.2.4Tracking the mouse pointer  4.2.5Simulating the left click and the right click events of the mouse  4.3Problems and drawbacks | 13  13  14  14  14  15  15  17  19 |
| 5 | **ALGORITHM DESIGN** | 20 |
| 6 | **CONCLUSION**  6.1Conclusion | 22 |
| 7 | **REFERENCES** | 23 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 4.2.1 | The Basic Block Diagram of the System | 14 |
| 4.2.1.1 | Capturing the Video | 15 |
| 4.2.3.1 | Flipping of an Image | 16 |
| 4.2.3.2 | Flipping of a Real Time Image | 17 |
| 4.2.3.1 | Finding the centroid of an object | 17 |
| 4.2.3.2 | Code for finding centroid | 18 |
| 4.2.3.3 | Code for tracking the mouse pointer | 18 |
| 4.2.3.4  4.2.3.5 | Code for left click  Code for right click | 19  19 |
| 5.1 | HSV Model | 21 |
|  |  |  |

**1. INTRODUCTION**

As the technology is improving day by day, the importance of human computer interaction is rapidly increasing. But this technology is still in its nascent stage to be used on systems of different applications. A virtual mouse using hand gesture recognition is a system that allows users to give mouse inputs to computer without using an actual mouse hardware. To the extreme, it can also be called as hardware because it uses a camera for tracking hands. A virtual mouse can usually be operated with multiple input devices, which may include an actual mouse or a computer keyboard. Many people had tried to define gestures but their actual meaning is still arbitrary. For a successful communication, a sender and a receiver must have the same set of information for a particular gesture. As per the context of our project we calculate the movements of hands to define the movement of mouse pointer and use gesture recognition to perform various mouse events. The primary step in gesture recognition systems is the detection of hands and the segmentation of the corresponding image regions. This segmentation is crucial because it isolates the task-relevant data from the image background, Computer Vision Techniques for Hand Gesture Recognition before passing them to the subsequent tracking and recognition stages. Various methods have been proposed in the literature that utilize a several types of visual features and, in many cases, their combination. Such features are skin colour, shape, motion and anatomical models of hand.

* 1. **Aim: -**

To create a Virtual Mouse using Hand Gesture in Real Time

**1.2 Objective: -**

* Taking Color as Input from the Webcam
* Recognizing the Color From the Input
* Detecting the Contours
* Applying Morphology for Smoothening Image
* Creating a Pointer Using Mathematical Calculation

**2. LITERATURE SURVEY**

|  |  |  |  |
| --- | --- | --- | --- |
| SR. NO. | AUTHORS | DESCRIPTION | DRAWBACKS |
| 1 | Angel, Neethu.P. S | Real-Time Static and Dynamic Hand Gestures Recognition | The hand tracking has to be specifically adapted for each user. This system was implemented only in a restricted to the indoor environment. This system is prone to noise and sensitive to the change of the illumination |
| 2 | J.L. Raheja,  A. Chaudhary,  K. Singal | Proposed using hsv algorithm but this uses special sensor kinect to capture image and processes it | User has to spend more money for the sensor. |
| 3 | Abhik Banerjee, Abhirup Ghosh | Mouse Control using a Web Camera based on Colour Detection | The presence of other coloured objects in the background might cause the system to give an erroneous response. If the resolution of the camera is too high then the system might run slow |

**3. TECHNOLOGIES (Packages used)**

**3.1 OpenCV:**

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

**OpenCV's application areas: -**

* 2D and 3D feature toolkits
* Ego motion estimation
* Facial recognition
* Human-computer interaction (HCL)
* Mobile robotics
* Motion understanding
* Object identification
* Segmentation and recognition
* Stereopsis stereo vision: depth perception from 2 cameras
* Structure from motion (SFM)
* Motion tracking

**3.2 NumPy: -**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.  
Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**Installation:**

* **Mac**and **Linux**users can install NumPy via pip command:

pip install numpy

* **Windows** does not have any package manager analogous to that in Linux or mac.Please download the pre-built windows installer for NumPy from official site of Windows (according to your system configuration and Python version).  
  And then install the packages manually.

**3.3 wxPython: -**

Wx Python is a wrapper for the cross-platform GUI API (often referred to as a "toolkit") Wx Widgets (which is written in C++) for the Python programming language. It is one of the alternatives to Tkinter, which is bundled with Python. It is implemented as a Python extension module. With Wx Python software developers can create truly native user interfaces for their Python applications, that run with little or no modifications on Windows, Macs and Linux or other Unix-like systems.

**Installation:**

Installing wxPython4 (Phoenix) on Windows and OSX Binary wheels for these platforms are available on PyPI so you can install with this simpler command if build is available for your target Python:

pip install -U wxPython

**3.4 Pynput**

This library allows you to control and monitor input devices. It contains subpackages for each type of input device supported: pynput.mouse Contains classes for controlling and monitoring a mouse or trackpad. pynput.keyboard Contains classes for controlling and monitoring the keyboard.

**4. IMPLEMENTATION**

Here, the details about each the project and different sections are explained.

**4.1 INTRODUCTION TO THE SYSTEM**

In our work, we have tried to control mouse cursor movement and click events using a camera based on colour detection technique. Here real time video has been captured using a Web-Camera. The user wears coloured tapes to provide information to the system. Individual frames of the video are separately processed. The processing techniques involve an image subtraction algorithm to detect colours. Once the colours are detected the system performs various operations to track the cursor and performs control actions, the details of which are provided below.

No additional hardware is required by the system other than the standard webcam which is provided in every laptop computer

**4.2 SYSTEM DESCRIPTION**

Following are the steps in our approach:

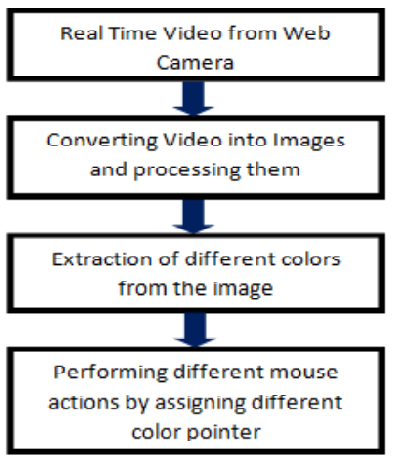
(I)Capturing real time video using Web-Camera.

(ii)Flipping of each image frame.

(iii)Finding the region of the image and calculating its centroid.

(iv)Tracking the mouse pointer using the coordinates obtained from the centroid.

(v)Simulating the left click and the right click events of the mouse by assigning different colour pointers.

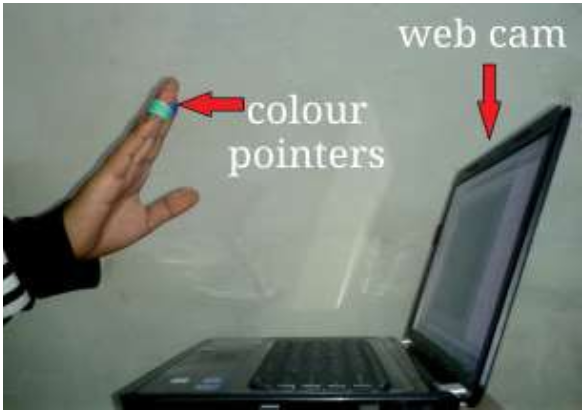


**4.2.1 The Basic Block Diagram of the System**

**4.2.1 Capturing the real time video**

For the system to work we need sensor to detect the hand movements of the user. The webcam of the computer is used as a sensor. The webcam captures the real time video at a fixed frame rate and resolution which is determined by the hardware of the camera. The frame rate and resolution can be changed in the system if required.

* Computer Webcam is used to capture the Real Time Video
* Video is divided into Image frames based on the FPS (Frames per second) of the camera
* Processing of individual Frames

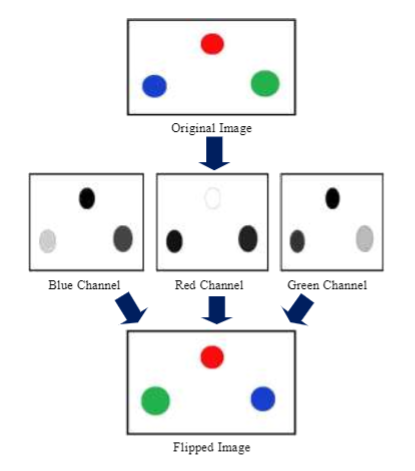


**4.2.1.1 Capturing the Video**

**4.2.2 Flipping of Images**

When the camera captures an image, it is inverted. This means that if we move the colour pointer towards the left, the image of the pointer moves towards the right and vice-versa. It’s similar to an image obtained when we stand in front of a mirror (Left is detected as right and right is detected as left). To avoid this problem, we need to vertically flip the image. The image captured is an RGB image and flipping actions cannot be directly performed on it. So, the individual colour channels of the image are separated and then they are flipped individually. After flipping the red, blue and green coloured channels individually, they are concatenated and a flipped RGB image is obtained.

**4.2.3.1 Flipping of an Image**



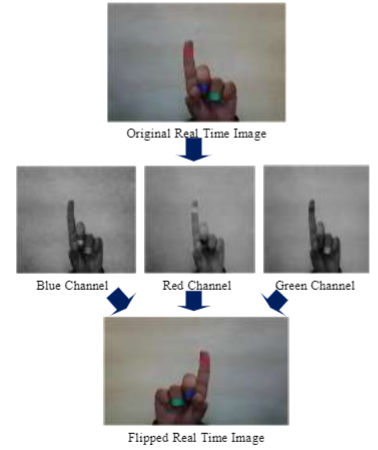
**4.2.3.1 fig. Flipping of an Image**

Mouse.position = (sx-(mouseLoc[0]\*sx/camx),mouseLoc[1]\*sy/camy)

Sx, sy= mouse coordinates

Camx, camy= mouse coordinates on camera

MouseLoc= current mouse location



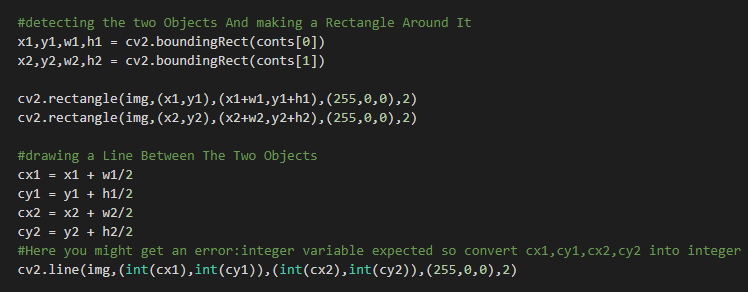
**4.2.3.2 Flipping of a Real time Image**

**4.2.3 Finding Centroid of an object and plotting**

For the user to control the mouse pointer it is necessary to determine a point whose coordinates can be sent to the cursor. With these coordinates, the system can control the cursor movement. The output of function is a matrix consisting of the X(horizontal)and Y(vertical)coordinates of the centroid. These coordinates change with time as the object moves across the screen. Centroid of the image is detected Its co-ordinates are located and stored in a variable

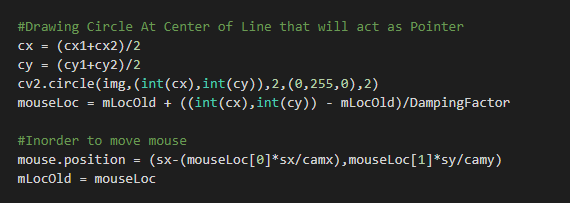
|  |  |
| --- | --- |
|  |  |

**4.2.3.1 Finding the centroid of object**



**4.2.3.2 Code for finding centroid**

**4.2.4Tracking the Mouse pointer**

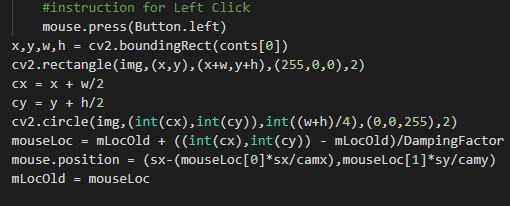
****

**4.2.3.3 Code for tracking the mouse pointer**

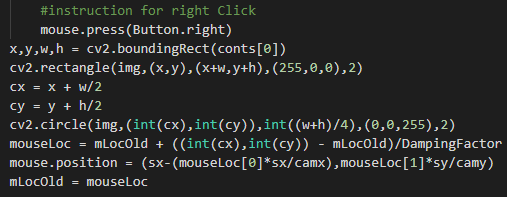
**4.2.5 Performing Clicking Actions:**

The control actions of the mouse are performed by controlling the flags associated with the mouse buttons. The user has to perform hand gestures in order to create the control actions. Due to the use of colour pointers, the computation time required is reduced. Furthermore, the system becomes resistant to background noise and low illumination conditions. The detection of green and blue colours follows the same procedure discussed above.

Clicking action is based on simultaneous detection of two colours. If Red along with Green colour is detected, Left clicking action is performed If Red along with Blue colour is detected, Right clicking action is performed



**4.2.3.4 Code for left click**



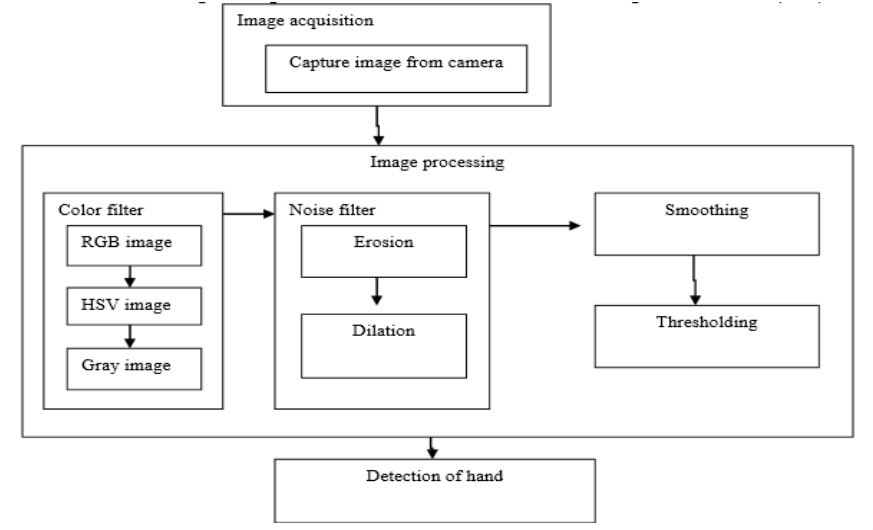
**4.2.3.5Code for right click**

**4.3 PROBLEMS AND DRAWBACKS**

Since the system is based on image capture through a webcam, it is dependent on illumination to a certain extent. Furthermore, the presence of other coloured objects in the background might cause the system to give an erroneous response. Although by configuring the threshold values and other parameters of the system this problem can be reduced but still it is advised that the operating background be light and no bright coloured objects be present. The system might run slower on certain computers with low computational capabilities because it involves a lot of complex calculations in a very small amount of time. However, a standard pc or laptop has the required computational power for optimum performance of the system. Another fact is that if the resolution of the camera is too high then the system might run slow. However, this problem can be solved by reducing the resolution of the image by making changes in the system.

**5. ALGORITHM DESIGN**

Algorithm is HSV which stands for hue, saturation, and value, it is also often called as HSB (B for brightness). HSV algorithm is simple transformations of device-dependent RGB models, the physical primary colors they define depend on the colors of the red, green, and blue primaries of the device or of the particular RGB space, and on the gamma, correction used to represent the amounts of those primaries. At the end, each unique RGB object has unique HSL and HSV absolute color spaces to accompany it (just as it has unique RGB absolute color space to accompany it), and the same numerical HSL or HSV values (just as numerical RGB values) may be displayed differently by different devices. The HSB or HSV, model describes colors in terms of hue, saturation, and value (brightness). Hue corresponds directly to the concept of hue in the Color Basics section. The advantages of using hue are the relationship between tones around the color circle is easily identified. Shades, tints, and tones can be generated easily without affecting the hue. Saturation corresponds directly to the concept of tint in the Color Basics section, except that full saturation produces no tint, while zero saturation produces white, a shade of gray, or black. Value corresponds directly to the concept of intensity in the Color Basics section. Pure colors are produced by specifying a hue with full saturation and value. Shades are produced by specifying a hue with full saturation and partial value. Tints are produced by specifying a hue with partial saturation and full value. Tones are produced by specifying a hue and partial saturation and value. White is produced by specifying zero saturation and full value, regardless of hue. Black is produced by specifying zero value, regardless of hue or saturation. Shades of gray are produced by specifying zero saturation and partial value. One of the biggest advantages of HSV is that each of its attributes corresponds directly to the basic concepts of color, which makes it simple



**5.1 HSV Model**

The various steps involved in the HSV model are explain as follows

**5.1 Image acquisition**

A single web camera is used to capture a series of images. Image acquisition is the creation of photographic images, such as of a physical scene or of the structure of an object. The term is often assumed to imply or include the processing, compression, storage, and display of such images.

**5.2 Color filter**

An improved color based segmentation technique was applied to segment the skin areas in a picture and use of skin-based segmentation in face recognition. An essential part is skin-colour pixels assumed in finding appearances in colored images and chromaticity estimations of distinctive colour spaces could be productively utilized for the information picture or image. The pictures that are taken form the camera is in the format of RGB that are converted in to HSV values and then other operations are performed. Segmentation of the color image was done into skin and non-skin areas are the first stage of face detection. They divided image segmentation in to four kinds pixel, edge, region and model based on specific application and working environment. The diversity of color spaces provides the ability to select the proper color space that can be utilized well under different environment conditions.

**5.3 Image processing**

Signal processing is another form of image processing for which the data is an image, such as photos, frames of video etc. The result of image processing can be either an image or a set of characteristics or parameters related to the image. Many of algorithms for image-processing techniques involve using the image as a two-dimensional signal and applying standard signal processing techniques to it. Image processing usually refers to DIP digital image processing, but optical and analog image processing are also possible. In this project image processing refers to splitting each and every pixel of the image into RGB components. Erosion will remove the unwanted details or background and the dilation will load the defects to get interested area (hand).

**5.4 Thresholding**

After extracting the region that is moving object, the thresholding on the frame difference can be applied for the extraction of the possible moving region in complex background. Traditional thresholding methods, such as Ostu thresholding are not suitable for the case of detecting motion difference. Instead, a simple thresholding technique is used to extract moving regions.

**6. CONCLUSION**

In this project, an object tracking based virtual mouse application has been developed and implemented using a webcam. The system has been implemented in environment using Image Processing Toolbox. This technology has wide applications in the fields of augmented reality, computer graphics, computer gaming, prosthetics, and biomedical instrumentation. Furthermore, a similar technology can be applied to create applications like a digital canvas which is gaining popularity among artists. This technology can be used to help patients who don’t have control of their limbs. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a person’s motions are tracked and interpreted as commands. Most of the applications require additional hardware which is often very costly. Our motive was to create this technology in the cheapest possible way and also to create it under a standardized operating system. Various application programs can be written exclusively for this technology to create a wide range of applications with the minimum requirement of resources.

**7. REFERENCES**

[1] Abhik Banerjee, Abhirup Ghosh ,2015, “Mouse Control using a Web Camera based on Colour Detection” International Journal of Electrical Electronics & Computer Science Engineering Volume 2, Issue 4.J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[2] Angel, Neethu.P. S, 2013, Real-Time Static and Dynamic Hand Gesture Recognition International Journal of Scientific & Engineering Research Volume 4

[3] J.L. Raheja, A. Chaudhary, K. Singal, 2011, “Tracking of Fingertips and Centre of Palm using KINECT”, In proceedings of the 3rd IEEE International Conference on Computational Intelligence, Modelling and Simulation.

[4] Hojoon Park, “A Method for Controlling the Mouse Movement using a Real Time Camera”, 2008, Brown University, Providence, RI, USA, Department of computer science.

[5] Chu-Feng Lien, “Portable Vision-Based HCI – A Real-time Hand Mouse System on Handheld Devices”, National Taiwan University, Computer Science and Information Engineering Department.