

HAND GESTURE RECOGNITION SYSTEM USING SIXTH SENSE TECHNOLOGY

By

RITIKA TYAGI- 101413

SUBMITTED TO

PROF. DR. SATYA PRAKASH GHRERA



Submitted in partial fulfillment of the Degree of Bachelor of Technology

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND
INFORMATION TECHNOLOGY**

JAYPEE UNIVERSITY OF INFORMATION

TECHNOLOGY-WAKNAGHAT

MAY-2014

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION.....	1
1.1 What is Sixth Sense?.....	2
1.2 Sixth Sense Technology.....	3
1.3 Applications of Sixth Sense Technology.....	4
1.3.1 Taking Pictures.....	4
1.3.2 Media Player Controller.....	4
1.3.3 Drawing Application.....	5
1.3.4 Making Calls.....	5
1.3.5 Interacting with Physical Objects.....	6
1.4 Advantages.....	6
CHAPTER 2 LITERATURE REVIEW.....	8
2.1 Sixth Sense Prototype.....	8
2.2 Working of a Sixth Sense Device.....	9
2.3 Augmented Reality.....	10
2.4 Gesture Recognition.....	11
2.5 Computer Vision based Algorithm.....	13
2.6 Technologies that use Sixth Sense as a Platform.....	14
2.6.1 Radio Frequency Identification.....	15
2.6.2 Sixth Sense Washing Machine.....	16
2.7 Color Models.....	17
2.7.1 RGB Color Model.....	17
2.7.2 HSV Color Model.....	18
2.7.3 CMYK Color Model.....	20

CHAPTER 3 ANALYSIS DESIGN AND IMPLEMENTATION.....	21
3.1 Overall Description of the Project.....	21
3.2 Components.....	22
3.2.1 Web Camera.....	22
3.2.2 Color Markers.....	23
3.2.3 Laptop.....	24
3.3 Specific Requirements.....	24
3.3.1 External Interfaces.....	24
3.3.2 Functions.....	24
3.3.3 Performance Requirements.....	25
3.3.4 Design Constraints.....	25
3.4 Design.....	25
CHAPTER 4 CODE ANALYSIS AND IMPLEMENTATION.....	28
4.1 Detecting the Finger Tip with Colored Markers.....	28
4.2 Media Player Controller.....	31
4.3 Mouse Controller.....	35
4.4 Power point Controller.....	39
4.5 Taking Snapshot.....	45
4.6 Implementation.....	46
CONCLUSION.....	49
APPENDIX A: TOOL DESCRIPTION- MATLAB.....	50
REFERENCES.....	52

CERTIFICATE

This is to certify that the work entitled, “**Hand Gesture Recognition System using Sixth Sense Technology**” submitted by **Ritika Tyagi(101413)** in partial fulfillment for the award of degree of Bachelor of Technology in Information Technology of Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Prof. Dr. Satya Prakash Ghrera

**(Head of Department of Computer Science Engineering
And Information Technology)**

ACKNOWLEDGEMENT

I owe a great many thanks to people who have been helping and supporting me during this project. My deepest thanks to **Prof. Dr. Satya Prakash Ghrera** , the Guide of the project for guiding and correcting me at every step of my work with attention and care. He has taken pain to go through the project and make necessary correction as and when needed.

I would also take privilege to record our heartiest thanks to the other faculty members of the department for their suggestions and help.

I would fail in my duty if I do not acknowledge the assistance extended to me by the laboratory staffs in letting me work in the laboratory even during odd hours and for making available all the resources required.

I would also like to thank my dearest colleagues whose constant suggestions regarding my project report was unbounded source of inspiration for me.

Ritika Tyagi- 101413

SUMMARY

Objective

The goal is to manage computers and other devices with gestures rather than pointing and clicking a mouse or touching a display directly.

Description of the work

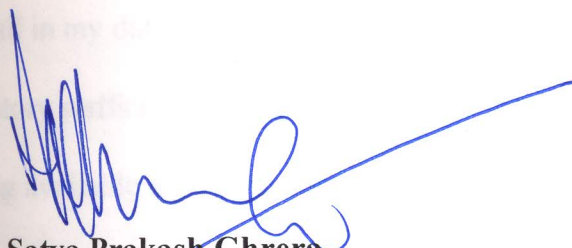
Here, I have used a new approach for controlling mouse movement using a real-time camera. Most existing approaches involve changing mouse parts such as adding more buttons or changing the position of the tracking ball. Instead, I propose to change the hardware design. My method is to use a camera, colored finger markers and sixth sense technology, such as image capturing and gesture recognition to control mouse tasks and I show how it can perform everything current mouse devices can. So, I intend to build this mouse control system using sixth sense technology.

Why I choose this topic?

Humans take decisions after acquiring inputs from the senses. But the information we collect aren't enough to result in the right decisions. But the information which could help making a good decision is largely available on internet. Although the information can be gathered by connecting devices like computers and mobiles but they are restricted to the screen and there is no direct interaction between the tangible physical world and intangible digital world. This sixth sense technology provides us with the freedom of interacting with the digital world with hand gestures. This technology has a wide application in the field of artificial intelligence. This methodology can aid in synthesis of bots that will be able to interact with humans.

CERTIFICATE

This is to certify that the work entitled, "**Hand Gesture Recognition System using Sixth Sense Technology**" submitted by **Ritika Tyagi(101413)** in partial fulfillment for the award of degree of Bachelor of Technology in Information Technology of Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.



Prof. Dr. Satya Prakash Ghrera

**(Head of Department of Computer Science Engineering
And Information Technology)**

CHAPTER 1

INTRODUCTION

We've evolved over millions of years to sense the world around us. When we encounter something, someone or some place, we use our five natural senses which include eye, ear, nose, tongue mind and body to perceive information about it; that information helps us make decisions and chose the right actions to take. But arguably the most useful information that can help us make the right decision is not naturally perceivable with our five senses, namely the data, information and knowledge that mankind has accumulated about everything and which is increasingly all available online.

Although the miniaturization of computing devices allows us to carry computers in our pockets, keeping us continually connected to the digital world, there is no link between our digital devices and our interactions with the physical world. Information is confined traditionally on paper or digitally on a screen. SixthSense bridges this gap, bringing intangible, digital information out into the tangible world, and allowing us to interact with this information via natural hand gestures. 'SixthSense' frees information from its confines by seamlessly integrating it with reality, and thus making the entire world your computer.

"Sixth Sense Technology", it is the newest jargon that has proclaimed its presence in the technical arena. This technology has emerged, which has its relation to the power of these six senses. Our ordinary computers will soon be able to sense the different feelings accumulated in the surroundings and it is all a gift of the "Sixth Sense Technology" newly introduced.

SixthSense is a wearable "gesture based" device that augments the physical world with digital information and lets people use natural hand gestures to interact with that information.

It was developed by Pranav Mistry, a PhD student in the Fluid Interfaces Group at the MIT Media Lab. A grad student with the Fluid Interfaces Group at MIT, he caused a storm with his

creation of SixthSense. He says that the movies “Robocop” and “Minority Report” gave him the inspiration to create his view of a world not dominated by computers, digital information and human robots, but one where computers and other digital devices enhance people’s enjoyment of the physical world.

Right now, we use our “devices” (computers, mobile phones, tablets, etc.) to go into the internet and get information that we want. With SixthSense we will use a device no bigger than current cell phones and probably eventually as small as a button on our shirts to bring the internet to us in order to interact with our world!

SixthSense will allow us to interact with our world like never before. We can get information on anything we want from anywhere within a few moments! We will not only be able to interact with things on a whole new level but also with people! One great part of the device is its ability to scan objects or even people and project out information regarding what you are looking at.

1.1 What is sixth sense?



Figure 1.1 Sixth Sense

Sixth Sense in scientific (or non-scientific) terms is defined as Extra Sensory Perception or in short ESP. It involves the reception of information not gained through any of the five senses. Nor

is it taken from any experiences from the past or known. Sixth Sense aims to more seamlessly integrate online information and tech into everyday life. By making available information needed for decision-making beyond what we have access to with our five senses, it effectively gives users a sixth sense.

1.2 Sixth sense Technology

Sixth Sense is a wearable gestural interface that enhances the physical world around us with digital information and lets us use natural hand gestures to interact with that information. It is based on the concepts of augmented reality and has well implemented the perceptions of it. Sixth sense technology has integrated the real world objects with digital world. The fabulous **sixth sense technology** is a blend of many exquisite technologies. The thing which makes it magnificent is the marvelous integration of all those technologies and presents it into a single portable and economical product. It associates technologies like hand gesture recognition, image capturing, processing, and manipulation, etc. It superimposes the digital world on the real world.

Sixth sense technology is a perception of augmented reality concept. Like senses enable us to perceive information about the environment in different ways it also aims at perceiving information. Sixth sense is in fact, about comprehending information more than our available senses. And today there is not just this physical world from where we get information but also the digital world which has become a part of our life. This digital world is now as important to us as this physical world. And with the internet the digital world can be expanded many times the physical world. God hasn't given us sense to interact with the digital world so we have created them like smart phones, tablets, computers, laptops, net books, PDAs, music players, and others gadgets. These gadgets enable us to communicate with the digital world around us.

But we're humans and our physical body isn't meant for digital world so we can't interact directly to the digital world. For instance we press keys to dial a number; we type text to search it and so on. This means for an individual to communicate with the digital world he/she must learn

it. We don't communicate directly and efficiently to the digital world as we do with the real world. The sixth sense technology is all about interacting to the digital world in most efficient and direct way. Hence, it wouldn't be wrong to conclude sixth sense technology as gateway between digital and real world.

1.3 Applications of Sixth Sense technology

1.3.1 Taking Pictures

Application of sixth sense devices is the implementation of a gestural camera. The camera takes the photo by detecting the framing gesture. After taking the desired number of photos we can also project them onto any surfaces and then use gestures to sort through those photos and organize and resize them.



Figure 1.2 Taking Snapshot

1.3.2 Media Player Controller

With the help of gestures, we can control the media player such as we can lower the volume or can increase the volume; we can also pause the player and can replay it.

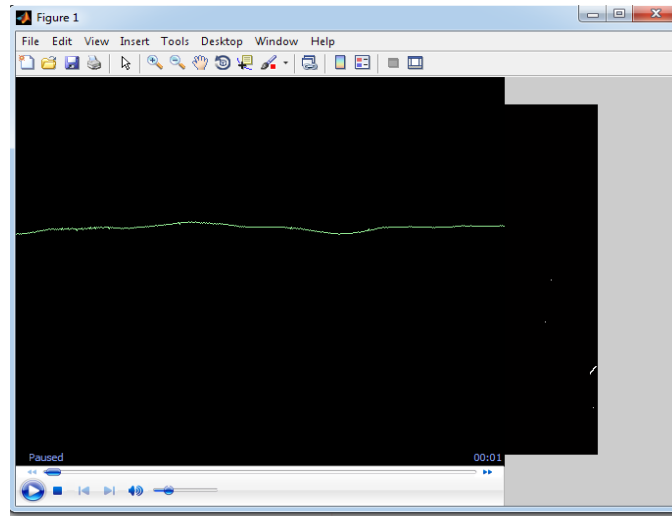


Figure 1.3 Media Player Controller

1.3.3 Drawing Application

The drawing application allows the user you to draw on any surface by tracking the fingertip movements of the user's index finger. The pictures that are drawn by the user can be stored and replaced on any other surface. The user can also shuffle through these various pictures and can draw by using the hand gesture movements.



Figure 1.4 Drawing Application

1.3.4 Making calls

We can make calls with the help of Sixth Sense technology. The Sixth Sense device is used to protect the keyboard into your palm and using that virtual keypad we can make calls to anyone.

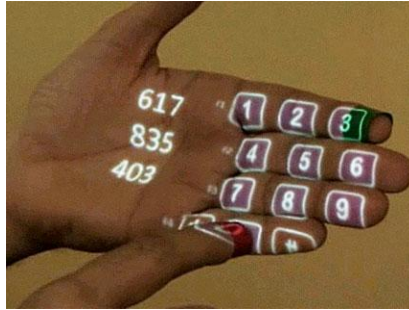


Figure 1.5 Making Call

1.3.5 Interacting with physical objects

The Sixth Sense system also helps to interact with physical objects. It augments physical objects by projecting more information about these objects projected on them. For example, a gesture of drawing a circle on the user's wrist projects a watch on the user's hand. Similarly a newspaper can show live video news or dynamic information can be provided on a regular piece of paper.



Figure 1.6 Interacting with Physical Object

1.4 Advantages

1. Sixth Sense is a user friendly interface which integrates digital information into the physical world and its objects, making the entire world as computer.
2. Sixth Sense does not change human habits but causes computer and other machines to adapt to human needs.
3. It allows users to use hand gestures to interact with the digital world and also allows multi-user touch and multi-user interaction.

4. It is gesture-controlled wearable computing device that feeds our relevant information and turns any surface into an interactive display.
5. It is portable and easy to carry.
6. The device could be used by anyone without even a basic knowledge of a keyboard or mouse.

CHAPTER 2

LITERATURE REVIEW

2.1 Sixth Sense Prototype

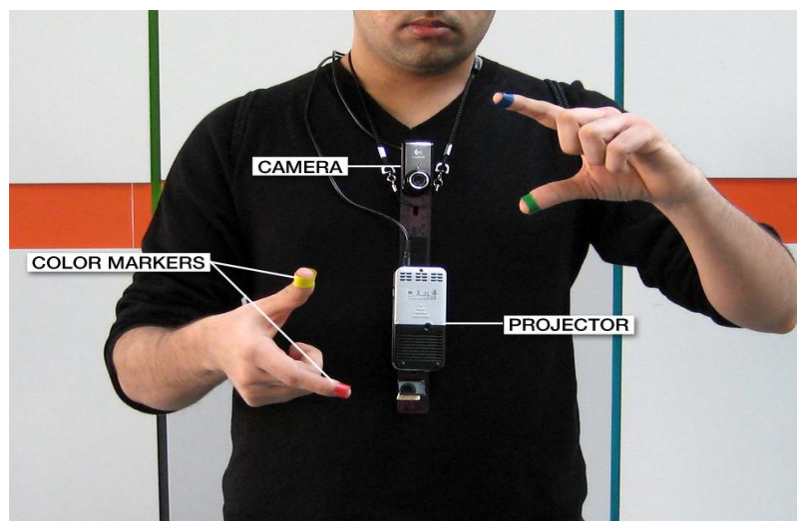


Figure 2.1 Prototype of sixth sense

The Sixth Sense prototype is composed of a pocket projector, a mirror and a camera. The hardware components are coupled in a pendant-like mobile wearable device. Both the projector and the camera are connected to the mobile computing device in the user's pocket.

We can very well consider the Sixth Sense Technology as a blend of the computer and the cell phone. It works as the device associated to it is hung around the neck of a person and thus the projection starts by means of the micro projector attached to the device. Therefore, in course, you turn out to be a moving computer in yourself and the fingers act like a mouse and a keyboard.

The prototype was built from an ordinary webcam and a battery-powered 3M projector, with an attached mirror — all connected to an internet-enabled mobile phone. The setup allows the user

to project information from the phone onto any surface walls, the body of another person or even your hand.

The device has to be worn around a neck and colored Marker caps on four fingers (red, blue, green and yellow) helped the camera distinguish the four fingers and recognize hand gestures with software created.

2.2 Working of a Sixth Sense Device

The Sixth Sense Technology works as follows:

1. It captures the image of the object in view and tracks the users hand gestures.
2. There is color markers placed at the tip of user fingers. Marking the user's fingers with red, yellow, green and blue colored tape helps the webcam to recognize the hand gestures. The movements and arrangement of these markers are interpreted into gestures that act as an interaction instruction for the projected application interfaces.
3. The Smartphone searches the web and interprets the hand gestures with the help of the colored markers placed at the finger tips.
4. The information that is interpreted through the Smartphone can be projected into any surface.
5. The mirror reflects the image on to a desired surface.

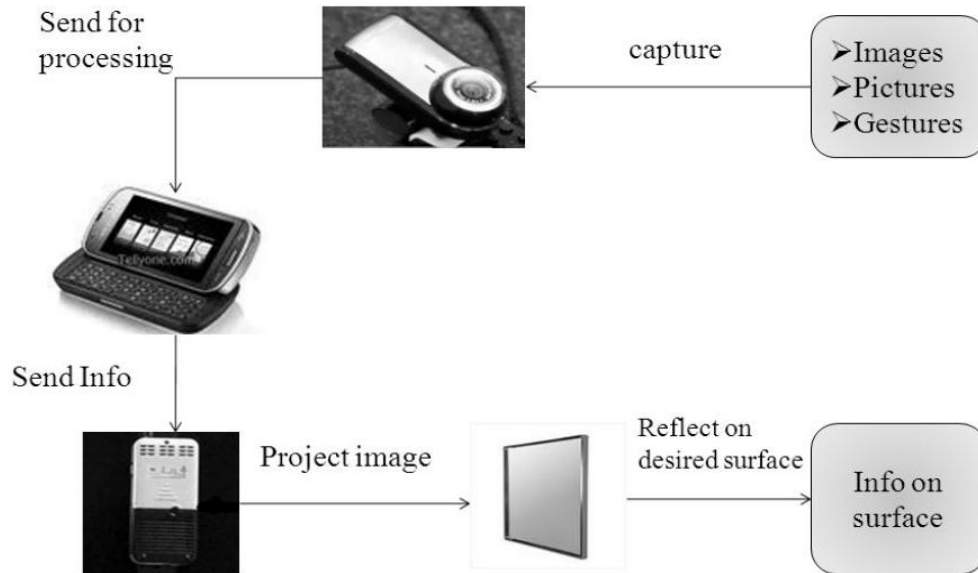


Figure 2.2 Working of a Sixth Sense Tecnology

2.3 Augmented reality

Augmented reality (AR) is a term for a live direct or indirect view of a physical real-world environment whose elements are augmented by virtual computer-generated imagery. It is related to a more general concept called mediated reality in which a view of reality is modified (possibly even diminished rather than augmented) by a computer. The augmentation is conventionally in real-time and in semantic context with environmental elements.

Sixth sense technology which uses Augmented Reality concept to super imposes digital information on the physical world. With the help of advanced AR technology (e.g. adding computer vision and object recognition) the information about the surrounding real world of the user becomes interactive and digitally usable. Artificial information about the environment and the objects in it can be stored and retrieved as an information layer on top of the real world view.

The main hardware components for augmented reality are: display, tracking, input devices and computer. Combination of powerful CPU, camera, accelerometers, GPS and solid state compass are often present in modern Smartphone, which make them prospective platforms. There are three major display techniques for Augmented Reality:

- Head Mounted Displays
- Handheld Displays
- Spatial Displays

Head Mounted Displays

A Head Mounted Display (HMD) places images of both the physical world and registered virtual graphical objects over the user's view of the world. The HMD's are either optical see-through or video see-through in nature.

Handheld Displays

Handheld Augmented Reality employs a small computing device with a display that fits in a user's hand. All handheld AR solutions to date have employed video see-through techniques to overlay the graphical information to the physical world. Initially handheld AR employed sensors such as digital compasses and GPS units for its six degree of freedom tracking sensors.

Spatial Display

Spatial Augmented Reality augments real world objects and scenes without the use of special displays such as monitors, head mounted displays or hand-held devices. SAR makes use of digital projectors to display graphical information onto physical objects. The key difference in SAR is that the display is separated from the users of the system. Because the displays are not associated with each user, SAR scales naturally up to groups of users, thus allowing for collocated collaboration between users.

Examples include shader lamps, mobile projectors, virtual tables, and smart projectors.

2.4 Gesture Recognition

Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current focuses in the field

include emotion recognition from the face and hand gesture recognition. Many approaches have been made using cameras and computer vision algorithms to interpret sign language.

Gestures can exist in isolation or involve external objects. Free of any object, we wave, beckon, fend off, and to a greater or lesser degree (depending on training) make use of more formal sign languages. With respect to objects, we have a broad range of gestures that are almost universal, including pointing at objects, touching or moving objects, changing object shape, activating objects such as controls, or handing objects to others.

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 humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse. Gesture recognition enables humans to interface with the machine (HMI) and interact naturally without any mechanical devices.

Gestures can be used to communicate with a computer so we will be mostly concerned with empty handed semiotic gestures. These can further be categorized according to their functionality.

Symbolic gestures

These are gestures that, within each culture, have come to a single meaning. An Emblem such as the “OK” gesture is one such example; however American Sign Language gestures also fall into this category.

Deictic gestures

These are the types of gestures most generally seen in HCI and are the gestures of pointing, or otherwise directing the listeners attention to specific event or objects in the environment.

Iconic gestures

As the name suggests, these gestures are used to convey information about the size, shape or orientation of the object of discourse. They are the gestures made when someone says “The plane

flew like this”, while moving their hand through the air like the flight path of the aircraft.

Pantomimic gestures

These are the gestures typically used in showing the use of movement of some invisible tool or object in the speaker’s hand. When a speaker says “I turned the steering wheel hard to the left”, while mimicking the action of turning a wheel with both hands, they are making a pantomimic gesture.

Using the concept of gesture recognition, it 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-screens redundant. Gesture recognition can be conducted with techniques from computer vision and image processing.

The literature includes ongoing work in the computer vision field on capturing gestures or more general human pose and movements by cameras connected to a computer.

2.5 Computer vision based algorithm

Computer vision is the science and technology of machines that see. As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner.

Computer vision, on the other hand, studies and describes the processes implemented in software and hardware behind artificial vision systems. The software tracks the user’s gestures using computer-vision based algorithms. Computer vision is, in some ways, the inverse of computer graphics. While computer graphics produces image data from 3D models, computer vision often produces 3D models from image data. There is also a trend towards a combination of the two disciplines, e.g., as explored in augmented reality.

The fields most closely related to computer vision are image processing, image analysis and machine vision. Image processing and image analysis tend to focus on 2D images, how to transform one image to another. This characterization implies that image processing/analysis neither require assumptions nor produce interpretations about the image content. Computer vision tends to focus on the 3D scene projected onto one or several images, e.g., how to reconstruct structure or other information about the 3D scene from one or several images.

Machine vision tends to focus on applications, mainly in manufacturing, e.g., vision based autonomous robots and systems for vision based inspection or measurement.

The Recognition algorithms

The computer vision system for tracking and recognizing the hand postures that control the menus is based on a combination of multi-scale color feature detection; view based hierarchical hand models and particle filtering. The hand postures or states are represented in terms of hierarchies of multi-scale color image features at different scales, with qualitative inter-relations in terms of scale, position and orientation. In each image, detection of multi scale color features is performed. The hand postures are then simultaneously detected and tracked using particle filtering, with an extension of layered sampling referred to as hierarchical layered sampling. To improve the performance of the system, a prior on skin color is included in the particle filtering.

2.6 Technologies that uses Sixth Sense as Platform

SixthSense technology takes a different approach to computing and tries to make the digital aspect of our lives more intuitive, interactive and, above all, more natural. When you bring in connectivity, you can get instant, relevant visual information projected on any object you pick up or interact with. So, pick up a box of cereal and your device will project whether it suits your

preferences. Some of the technologies that uses this are Radio Frequency identification, gesture gaming, washing machine.

2.6.1 Radio Frequency Identification

SixthSense is a platform for Radio Frequency Identification based enterprise intelligence that combines Radio Frequency Identification events with information from other enterprise systems and sensors to automatically make inferences about people, objects, workspaces, and their interactions.

Radio Frequency Identification is basically an electronic tagging technology that allows the detection and tracking of tags and consequently the objects that they are affixed to. This ability to do remote detection and tracking coupled with the low cost of passive tags has led to the widespread adoption of RFID in supply chains worldwide.

Pranav Mistry, a researcher at the media lab of the Massachusetts Institute Technology, has developed a 'sixth sense' device – a gadget worn on the wrist that can function as a 'touch screen' device for many modern applications. The gadget is capable of selecting a product either by image recognition or radio frequency identification (RFID) tags and project information, like an Amazon rating.

The idea of SixthSense is to use Radio Frequency Identification technology in conjunction with a bunch of other enterprise systems such as the calendar system or online presence that can track user activity. Here, we consider an enterprise setting of the future where people (or rather their employee badges) and their personal objects such as books, laptops, and mobile phones are tagged with cheap, passive RFID tags, and there is good coverage of RFID readers in the workplace.

SixthSense incorporates algorithms that start with a mass of undifferentiated tags and automatically infer a range of information based on an accumulation of observations. The technology is able to automatically differentiate between people tags and object tags, learn the

identities of people, infer the ownership of objects by people, learn the nature of different zones in a workspace (e.g., private office versus conference room), and perform other such inferences. By combining information from these diverse sources, SixthSense records all tag-level events in a raw database. The inference algorithms consume these raw events to infer events at the level of people, objects, and workspace zones, which are then recorded in a separate processed database.

Applications can either poll these databases by running SQL queries or set up triggers to be notified of specific events of interest.

SixthSense infers when a user has interacted with an object, for example, when you pick up your mobile phone. It is a platform in that its programming model makes the inferences made automatically available to applications via a rich set of APIs. To demonstrate the capabilities of the platform, the researchers have prototyped a few applications using these APIs, including a misplaced object alert service, an enhanced calendar service, and rich annotation of video with physical events.

2.6.2 Sixth Sense Washing Machine

Whirlpool AWOE 8758 White Washing Machine is a remarkable front loader that incorporates the unparalleled Sixth Sense technology. Whirlpool's 2009 range of washing machines comes integrated with enhanced 6th sense technology that gives more optimization of resources and also increased saving in terms of energy, water and time.

Ideal washing machine for thorough washing that requires sixth sense to detect stubborn stains and adjust wash impact. It is a feature packed washing ally with Sixth Sense Technology and several customized programs to enhance the washing performance and dexterously assist you in heavy washing loads.

The New Generation 6th Sense appliances from Whirlpool are helping to protect the environment and to reduce your energy bills. Whirlpool 6th Sense appliances are designed to be intelligent and energy efficient appliances that adapt their performance to better suit your needs.

All Whirlpool appliances with intelligent 6th Sense technology work on three key principles; Sense, Adaption and Control, to ensure that they achieve optimal performance each and every time that they are used.

Whirlpool 6th Sense washing machines can save you up to 50% less water, energy and time during the cycle. These intelligent machines sense the size of the load and adjust and control the cycle dependent on the load inside in order to optimize the use of water, energy and time. Some models also contain a detergent overdosing monitor to make sure that you do not use too much washing detergent. Tumble dryers use 6th Sense technology to minimize energy and time wastage by monitoring the humidity inside your laundry and adjusting the drying time accordingly.

2.7 Color Models

A **color model** is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components. When this model is associated with a precise description of how the components are to be interpreted (viewing conditions, etc.), the resulting set of colors is called color space. This section describes ways in which human color vision can be modeled. Basically, there are three color models:

- RGB Color Model
- HSV Color Model
- CMYK Color Model

2.7.1 RGB Color Model

The **RGB color model** is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.

RGB is a *device-dependent* color model: different devices detect or reproduce a given RGB value differently, since the color elements (such as phosphors or dyes) and their response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. Thus an RGB value does not define the same *color* across devices without some kind of color management.

Typical RGB input devices are color TV and video cameras, image scanners, and digital cameras.

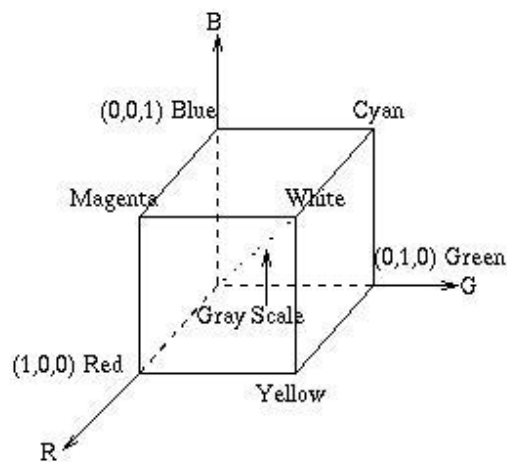


Figure2.3 RGB Color Model

2.7.2 HSV Color Model

Hue, Saturation, Value or **HSV** is a color model that describes colors(hue or tint) in terms of their shade (saturation or amount of gray) and their brightness (value or luminance).

The HSV color wheel may be depicted as a cone or cylinder. Instead of Value, the color model may use Brightness, making it HSB (Photoshop uses HSB).

- **Hue** is expressed as a number from 0 to 360 degrees representing hues of red (starts at 0), yellow (starts at 60), green (starts at 120), cyan (starts at 180), blue (starts at 240), and magenta (starts at 300).
- **Saturation** is the amount of gray (0% to 100%) in the color.
- **Value** (or **Brightness**) works in conjunction with saturation and describes the brightness or intensity of the color from 0% to 100%.

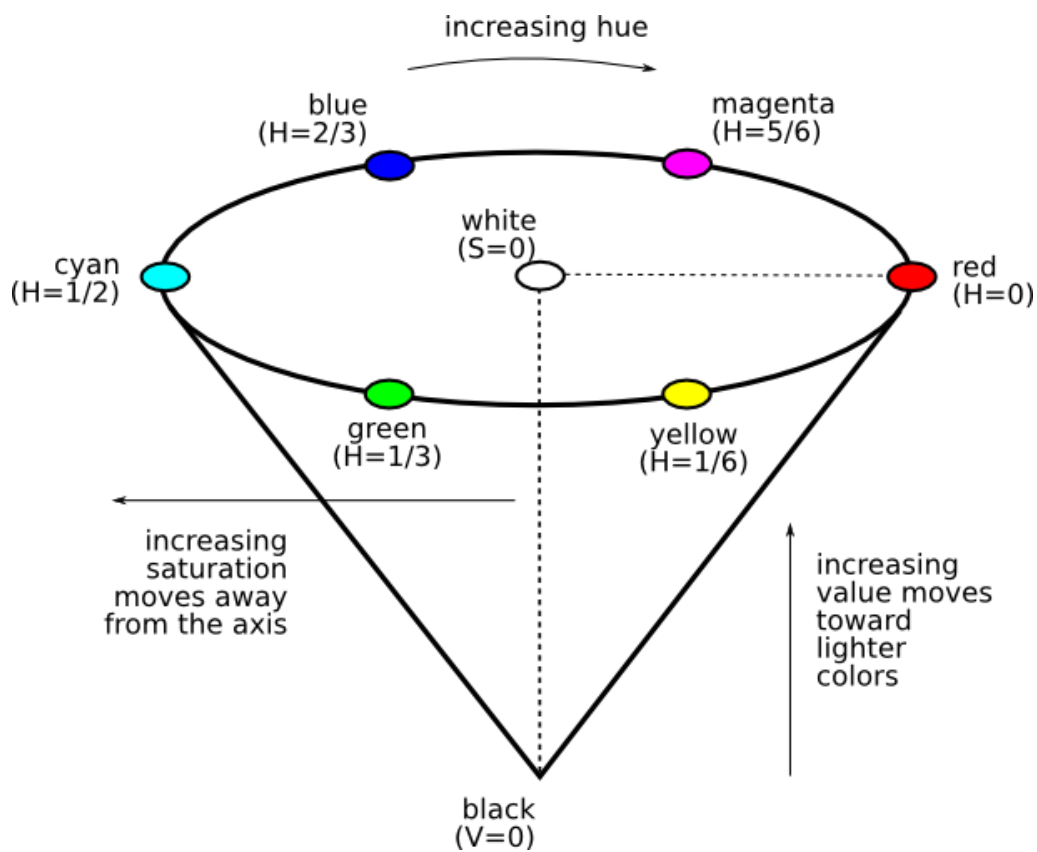


Figure2.4 HSV Color Model

2.7.3 CMYK Color Model

The **CMYK color model (process color, four color)** is a subtractive color model, used in color printing, and is also used to describe the printing process itself. **CMYK** refers to the four inks used in some color printing: **cyan, magenta, yellow, and key (black)**. Though it varies by print house, press operator, press manufacturer, and press run, ink is typically applied in the order of the abbreviation.

The "K" in CMYK stands for *key* because in four-color printing, cyan, magenta, and yellow printing plates are carefully *keyed*, or aligned, with the *key* of the black *key plate*. Some sources suggest that the "K" in CMYK comes from the last letter in "black" and was chosen because B already means blue. However, this explanation, although useful as a mnemonic, is incorrect.

The CMYK model works by partially or entirely masking colors on a lighter, usually white, background. The ink reduces the light that would otherwise be reflected. Such a model is called *subtractive* because inks "subtract" brightness from white.

In additive color models such as RGB, white is the "additive" combination of all primary colored lights, while black is the absence of light. In the CMYK model, it is the opposite: white is the natural color of the paper or other background, while black results from a full combination of colored inks. To save money on ink, and to produce deeper black tones, unsaturated and dark colors are produced by using black ink instead of the combination of cyan, magenta and yellow.

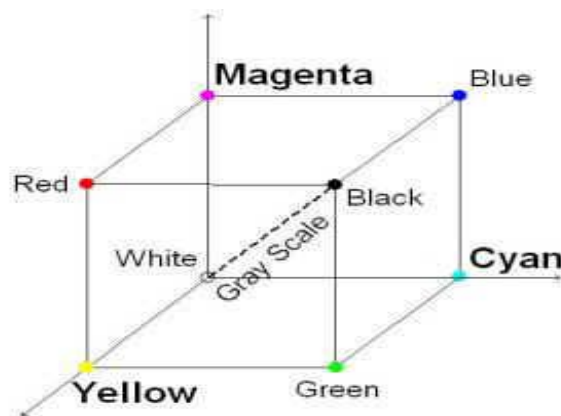


Figure 2.5 CMY Color Model

CHAPTER 3

ANALYSIS DESIGN AND IMPLEMENTATION

3.1 Overall Description of the project

a) Product Perspective

The project is an attempt to manage computers and other devices with gestures rather than pointing and clicking a mouse or touching a display directly. For this purpose I have implemented various algorithms in matlab and experimented with various matlab functions and java libraries.

b) Product Functions

- Taking video of the hand using web camera.
- Converting the video into frames of image.
- Recognizing the finger caps.
- Selection of multiple pixels on a finger cap and then calculation of threshold value.
- Then the movement of mouse cursor.

c) User Characteristics

The project does not have any constraints on who can use it; it can be used by anyone and everyone, from researcher, teacher and student to developer. The user can be any human body facing the camera with its hand in front of the camera view. The background should be preferably white.

d) Design and implementation constraints

- All the programming code is written in Matlab.
- A decent web camera is required and should not be used in darkness.
- The project is supposed to run in windows.

e) Assumptions and dependencies

- The background should be preferably white.
- The finger caps of red or yellow or blue color must be used.

3.2 COMPONENTS

The hardware components that are used are:

- Web camera
- Color Markers
- Laptop

3.2.1 Web Camera



Figure 3.1 Web Camera

It captures the image of the object in view and tracks the user's hand gesture. The camera recognizes individuals, images, pictures, gestures that user makes with his hand. It also tracks the movements of the thumbs and index fingers of both of the user's hands. The camera then sends this data to hand gesture recognition software for processing. Basically the camera acts as a digital eye, which connects to the world of digital information.

3.2.2 Color Markers

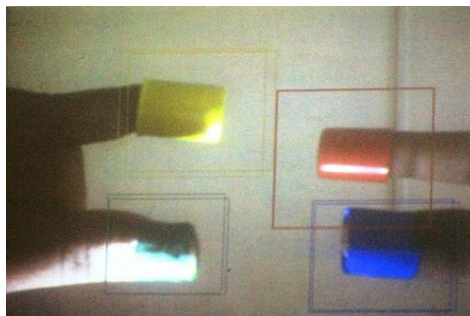


Figure 3.2 Color Marker

There is color markers placed at the tip of user's finger. Marking the user's fingers with red, yellow green and blue colored tape helps the web camera to recognize the hand gestures. The movements and arrangement of these markers are interpreted into gestures inside the hand gesture recognition software.

3.2.3 Laptop



Figure 3.3 Laptop

The Sixth Sense device consists of a laptop which process the data send by the camera. The laptop interprets the hand gestures with help of the colored markers placed at the fingertips. It then processes the information and then perform the functions accordingly.

3.3 Specific requirements

3.3.1 External Interfaces

Hardware Interface: There is a web camera + computer interaction the only hardware used in webcam and the finger caps.

Software Interface: We have used Matlab for running the application.

Communication Interface: Web camera is the only communication interface.

3.3.2 Functions

The main program: takes in live feed from web camera and detects the finger caps, then accordingly respond to hand gestures and performs the respective functions such as:

- Mouse operations i.e. right click and left click
- Taking a snapshot with hand gesture
- Media player controller
- PowerPoint controller

3.3.3 Performance Requirements

The basic requirement for running the application

- Web camera
- Matlab installed

3.3.4 Design constraints

- The results are better on more efficient computers with a good quality of web camera (especially with a kinect camera).
- Constantly (the faster the better) the hand should be segmented from the background.

3.4 DESIGN

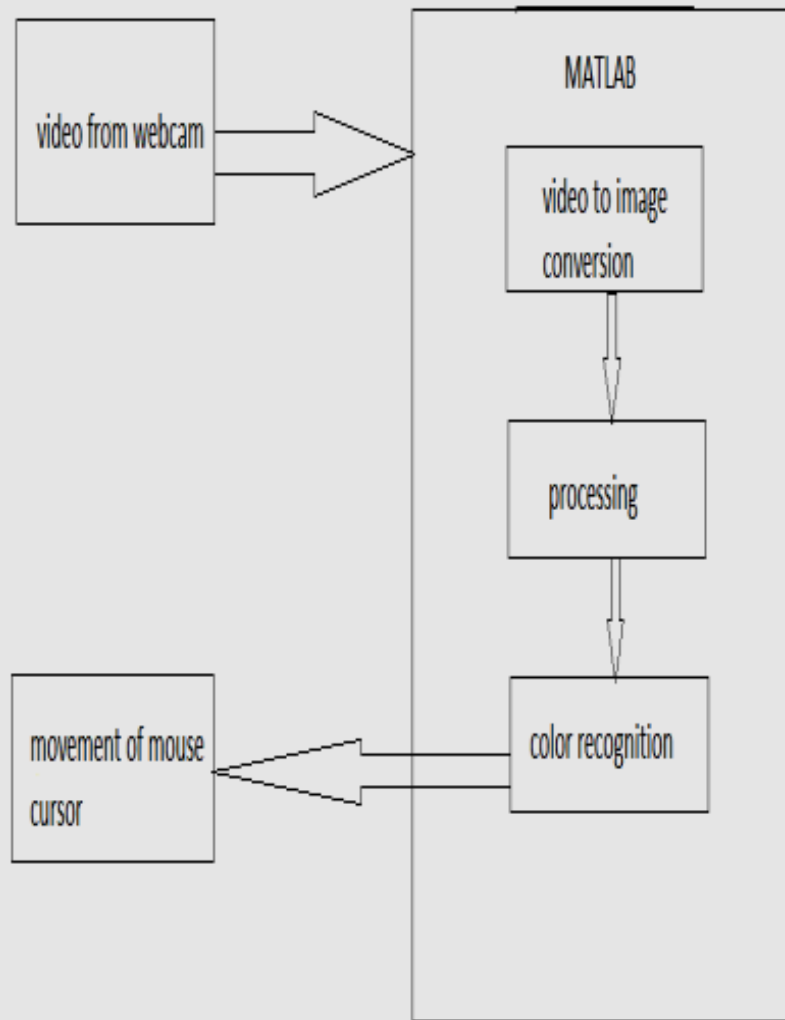


Figure3.4 System Design

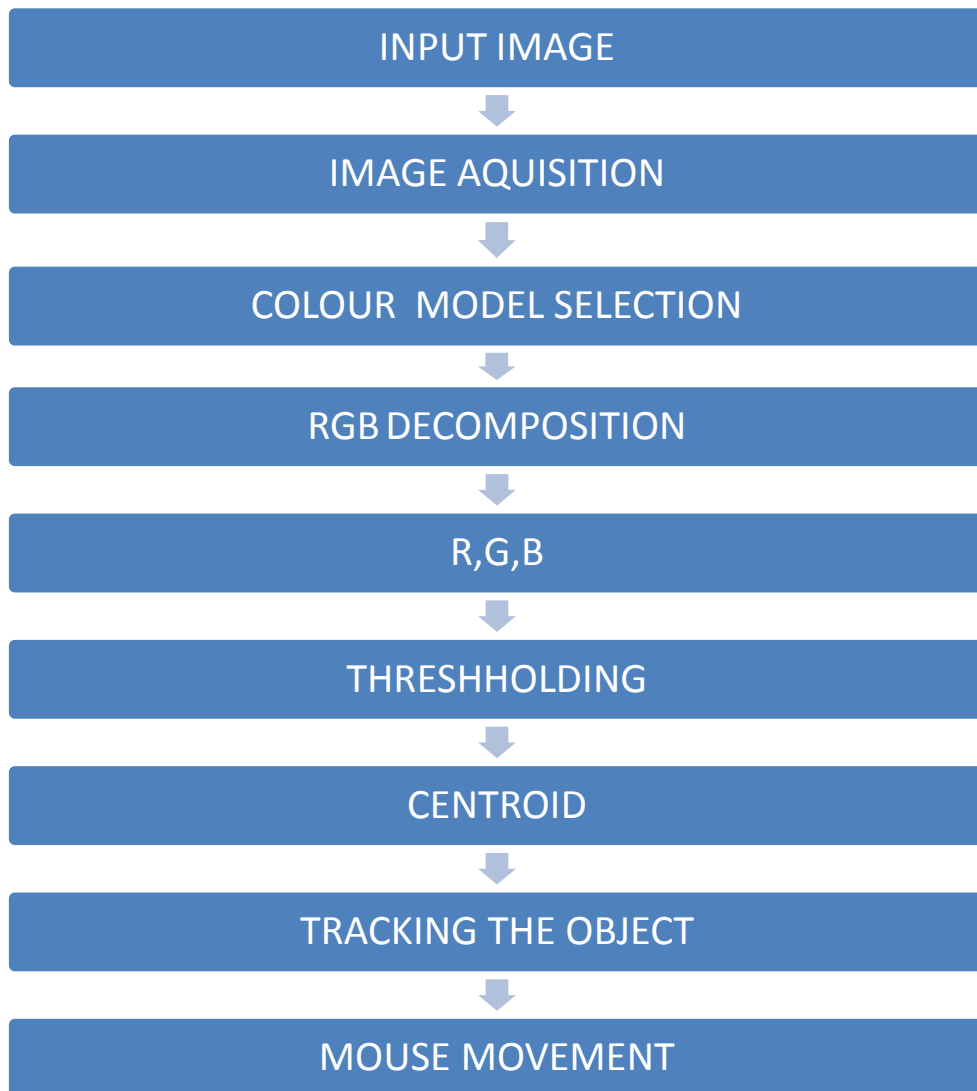


Figure 3.5 Design Flow

CHAPTER 4

CODE ANALYSIS AND IMPLEMENTATION

4.1 Detecting the finger tip with the colored markers

Main.m

```
clc                                % clear the command prompt
fprintf('wait please');            % to print a string in command prompt
imaqreset                           % disconnect and delete all image acquisition objects
clear all                            % clear variable and functions from memory
vidobj = videoinput('winvideo',1);   % creates a video input object
set(vidobj,'FramesPerTrigger',1);    % set the property of a object
set(vidobj,'TriggerRepeat',inf);
triggerconfig(vidobj,'manual');     % configure video input object trigger setting
start(vidobj);                       % Start timer(s) running.

while(1)
    trigger(vidobj);                 % Manually initiate logging/sending for running object.
    frame=getdata(vidobj);           % Return acquired data samples from engine to MATLAB
workspace.
    image_ycbcr=rgb2ycbcr(frame);    % Convert RGB values to YCBCR color space.
    image_y=image_ycbcr(:,1);
    image_cb=image_ycbcr(:,2);
    image_cr=image_ycbcr(:,3);
    [r c d]=size(image_ycbcr);
```

```

output_image_r=zeros(r,c);    %image red
output_image_g=zeros(r,c);    %image green
output_image_y=zeros(r,c);    %image Yellow
A=0;

%Detect Red mark/tape
for i1=1:r
    for i2=1:c

if((image_cb(i1,i2)>160)&&(image_cb(i1,i2)<175)&&(image_cr(i1,i2)>105)&&(image_cr(i1,i2)
)<140))
        output_image_r(i1,i2)=1;
    else
        output_image_r(i1,i2)=0;
    end
    end
end
total_pix_r=sum(sum(output_image_r));
if (total_pix_r>100)
    A=A+1;
end
%Detect Green mark/tape
for i1=1:r
    for i2=1:c

if((image_cb(i1,i2)>95)&&(image_cb(i1,i2)<120)&&(image_cr(i1,i2)>110)&&(image_cr(i1,i2)
<135)&&(image_y(i1,i2)<120))
        output_image_g(i1,i2)=1;
    else
        output_image_g(i1,i2)=0;
    end
    end
end

```

```

        end
    end
end
total_pix_g=sum(sum(output_image_g));
if (total_pix_g>100)
    A=A+1;
end
%Detect Yellow mark/tape
for i1=1:r
    for i2=1:c

if((image_cb(i1,i2)>125)&&(image_cb(i1,i2)<150)&&(image_cr(i1,i2)>170)&&(image_cr(i1,i2)
)<200)&&(image_y(i1,i2)<120))
        output_image_y(i1,i2)=1;
    else
        output_image_y(i1,i2)=0;
    end
end
end
imshow(output_image_y+output_image_g+output_image_r);
total_pix_y=sum(sum(output_image_y));
if (total_pix_y>100)
    A=A+1;
end

if(A==1)
    disp('one');
elseif(A==2)
    disp('two');
elseif(A==3)

```

```

    disp('three');
else
    disp('No finger');
end
clear A;
end

```

Delay.m

```

function []=delay(x)
for i=1:500
    for j=1:500
        for k=1:x %if u give x=420 u will get a delay of 1 second.
            end
        end
    end
end
end

```

4.2 Media Player Controller

Main.m

```

clc                % clear the command prompt
close all
fprintf('wait please\n');    % to print a string in command prompt
imaqreset          % disconnect and delete all image aquisition objects
clear all          % clear variable and functions from memory
vidobj = videoinput('winvideo',1);    % creates a video input object
set(vidobj,'ReturnedColorSpace','RGB'); % setting the properties of object
set(vidobj,'FramesPerTrigger',1);
set(vidobj,'TriggerRepeat',inf);
triggerconfig(vidobj,'manual');

```

```

start(vidobj);
range=20;
fprintf('Bring Markers....\n');      % to print a string in command prompt
delay(2000);
fprintf('Thresolding for GREEN Marker\n');
im = getsnapshot(vidobj);      % taking a snapshot
fprintf('Resolution is ');
size(im)          % to print the resolution of image in command prompt
[GR_M_R GR_M_G GR_M_B] = get_MY_THRESHOLDING(im);
delay(300);
close all;
%=====Opening media player=====
m=figure;
h=activexcontrol('WMPlayer.OCX.7', [0 0 500 550], m);
[filename pathname] = uigetfile('*..*','Please select a file');
h.URL=[pathname filename];
h.controls.play;
i=0;j=0;
set(h.settings,'volume',i);  % Setting volume of media player
figure;
i=0;
while(1)
    trigger(vidobj);          % Manually initiate logging/sending for running object.
    im=getdata(vidobj);      % Return acquired data samples from engine to MATLAB
workspace.
    [r c d]=size(im);
output_image_g=zeros(r,c);  %image red
    %To Detect red mark/tape
    for i1=1:r
        for i2=1:c

```



```

        if( (im(i1,i2,1)>GR_M_R-range) && (im(i1,i2,1)<GR_M_R+range) &&
(im(i1,i2,2)>GR_M_G-range) && (im(i1,i2,2)<GR_M_G+range) && (im(i1,i2,3)>GR_M_B-
range) && (im(i1,i2,3)<GR_M_B+range) )
            output_image_g(i1,i2)=1;
        end
    end
end
imshow(output_image_g);
[r_cent_g c_cent_g]=centroid1(output_image_g);    % to find centre of marker
total_pix=sum(sum(output_image_g));            % to find total no. of pixels of marker
if (total_pix>500)
    h.controls.pause;
    disp('Media Paused');
elseif (c_cent_g>(240)) % My camera resolution = 120*160
    disp('Volume Up');
    i=i+3;
    h.controls.play;
elseif (c_cent_g<(220)) % My camera resolution = 120*160
    disp('Volume Down');
    i=i-3;
    h.controls.play;
else
    h.controls.play;
    disp('Playing');
end
if(i<2)        % to restrict min. volume as 2
    i=2;
end
if(i>100)      % to restrict max. volume as 100
    i=100;

```

```

    end
    set(h.settings,'volume',i); % Depending on i, Volume will increase or decrease
end

```

get_MY_THRESHOLDING.m

```

function [red grn blu]=get_MY_THRESHOLDING(im)
% This function gives the rgb values to be used
% for thresholding of a MARKER/COLOR
p = impixel(im); % Lets user select multiple pixels on the marker
[r c]=size(p);
red = mean(p(:,1));
grn = mean(p(:,2));
blu = mean(p(:,3));
[r c d]=size(im);
output_image=zeros(r,c);
    % Creating output image based on selected pixels on the MARKER/COLOR.
range = 20;
for i1=1:r
    for i2=1:c
        if( (im(i1,i2,1)>red-range) && (im(i1,i2,1)<red+range) && (im(i1,i2,2)>grn-range) &&
(im(i1,i2,2)<grn+range) && (im(i1,i2,3)>blu-range) && (im(i1,i2,3)<blu+range) )
            output_image(i1,i2)=1;
        end
    end
end
figure();
imshow(output_image);
end

```

Centroid1.m

```
function [r_cent c_cent]=centroid1(arg)
[r_index c_index]=find(arg);
r_cent=mean(r_index);
c_cent=mean(c_index);
```

Delay.m

```
function []=delay(x)
for i=1:500
    for j=1:500
        for k=1:x %if u give x=420 u will get a delay of 1 second.
            end
        end
    end
end
```

4.3 Mouse Controller

Main.m

```
clc % clear the command prompt
close all
fprintf('wait please\n'); % to print a string in command prompt
imaqreset % disconnect and delete all image aquisition objects
clear all % clear variable and functions from memory
vidobj = videoinput('winvideo',1); % creates a video input object
set(vidobj,'ReturnedColorSpace','RGB'); % setting the properties of object
set(vidobj,'FramesPerTrigger',1);
set(vidobj,'TriggerRepeat',inf);
triggerconfig(vidobj,'manual');
start(vidobj);
```

```

range=20;
fprintf('Bring Markers....\n');
delay(2000);
fprintf('Thresolding for RED Marker\n');
im = getsnapshot(vidobj);
[RD_M_R RD_M_G RD_M_B] = get_MY_THRESHOLDING(im);
delay(300);
fprintf('Thresolding for YELLOW Marker\n');
%im = getsnapshot(vidobj);
[YL_M_R YL_M_G YL_M_B] = get_MY_THRESHOLDING(im);
delay(300);
close all

import java.awt.Robot;
import java.awt.event.*;
mouse = Robot;
while(1)
    trigger(vidobj);          % Manually initiate logging/sending for running object.
    im=getdata(vidobj);      % Return acquired data samples from engine to MATLAB
workspace.
    [r c d]=size(im);
    output_image_r=zeros(r,c);    %image red
    % output_image_g=zeros(r,c);    %image green
    output_image_y=zeros(r,c);    %image Yellow
    A=0;
%To Detect Red mark/tape
    for i1=1:r
        for i2=1:c
            if( (im(i1,i2,1)>RD_M_R-range) && (im(i1,i2,1)<RD_M_R+range) &&
(im(i1,i2,2)>RD_M_G-range) && (im(i1,i2,2)<RD_M_G+range) && (im(i1,i2,3)>RD_M_B-
range) && (im(i1,i2,3)<RD_M_B+range) )

```

```

        output_image_r(i1,i2)=1;
    end
end
end
imshow(output_image_r); % To see the marker Position for Debugging
[r_cent_r c_cent_r]=centroid1(output_image_r);
disp([r_cent_r c_cent_r]);
%To Detect Yellow mark/tape
for i1=1:r
    for i2=1:c
        if( (im(i1,i2,1)>YL_M_R-range) && (im(i1,i2,1)<YL_M_R+range) &&
(im(i1,i2,2)>YL_M_G-range) && (im(i1,i2,2)<YL_M_G+range) && (im(i1,i2,3)>YL_M_B-
range) && (im(i1,i2,3)<YL_M_B+range) )
            output_image_y(i1,i2)=1;
        end
    end
end
[r_cent_y c_cent_y]=centroid1(output_image_y);
disp([r_cent_y c_cent_y]);
r = sqrt( ((r_cent_r)-(r_cent_y)).^2 + ((c_cent_r)-(c_cent_y)).^2 ); %To Find Distance
Between Two Centroid
disp(r);
mouse.mouseMove(0, 0);
mouse.mouseMove(((8.2)*(160 - c_cent_r)),(6.7*(r_cent_r))); %To Expand The Area Of
Cursor Movement
if(r<50)
    mouse.mousePress(InputEvent.BUTTON3_MASK); %Right Click
    mouse.mouseRelease(InputEvent.BUTTON3_MASK);
    delay(500)
end

```

end

get_MY_THRESHOLDING.m

```
function [red grn blu]=get_MY_THRESHOLDING(im)
% This function gives the rgb values to be used
% for thresholding of a MARKER/COLOR
p = impixel(im); %select multiple pixels on the marker
[r c]=size(p);
red = mean(p(:,1));
grn = mean(p(:,2));
blu = mean(p(:,3));
[r c d]=size(im);
output_image=zeros(r,c);
    % Creating output image based on selected pixels on the MARKER/COLOR.
range = 20;
for i1=1:r
    for i2=1:c
        if( (im(i1,i2,1)>red-range) && (im(i1,i2,1)<red+range) && (im(i1,i2,2)>grn-range) &&
(im(i1,i2,2)<grn+range) && (im(i1,i2,3)>blu-range) && (im(i1,i2,3)<blu+range) )
            output_image(i1,i2)=1;
        end
    end
end
figure();
imshow(output_image);
end
```

Centroid1.m

```
function [r_cent c_cent]=centroid1(arg)
```

```
[r_index c_index]=find(arg);
r_cent=mean(r_index);
c_cent=mean(c_index);
```

Delay.m

```
function []=delay(x)
for i=1:500
    for j=1:500
        for k=1:x %if u give x=420 u will get a delay of 1 second.
            end
        end
    end
end
```

4.4 PowerPoint Controller

Main.m

```
clc % clear the command prompt
close all
fprintf('wait please\n'); % to print a string in command prompt
imaqreset % disconnect and delete all image aquisition objects
clear all % clear variable and functions from memory
vidobj = videoinput('winvideo',1); % creates a video input object
set(vidobj,'ReturnedColorSpace','RGB'); % setting the properties of object
set(vidobj,'FramesPerTrigger',1);
set(vidobj,'TriggerRepeat',inf);
triggerconfig(vidobj,'manual');
start(vidobj);
range=40;
fprintf('Bring Markers....\n');
```

```

delay(2000);
fprintf('Thresolding for RED Marker\n');
im = getsnapshot(vidobj);
[RD_M_R RD_M_G RD_M_B] = get_MY_THRESHOLDING(im);
delay(1000);
close all
import java.awt.Robot;
import java.awt.event.*;
mouse = Robot;
while(1)
    trigger(vidobj);          % Manually initiate logging/sending for running object.
    im=getdata(vidobj);      % Return acquired data samples from engine to MATLAB
workspace.
    [r c d]=size(im);
    output_image_r=zeros(r,c); %image red
    %To Detect Red mark/tape
    for i1=1:r
        for i2=1:c
            if( (im(i1,i2,1)>RD_M_R-range) && (im(i1,i2,1)<RD_M_R+range) &&
(im(i1,i2,2)>RD_M_G-range) && (im(i1,i2,2)<RD_M_G+range) && (im(i1,i2,3)>RD_M_B-
range) && (im(i1,i2,3)<RD_M_B+range) )
                output_image_r(i1,i2)=1;
            end
        end
    end
    imshow(output_image_r); % See the marker Position for Debugging
    [r_cent_r c_cent_r]=centroid1(output_image_r);
    disp([r_cent_r c_cent_r]);
    mouse.mouseMove((8.7*(160 - c_cent_r)),(6.2*r_cent_r));
    if(c_cent_r<10) % for my laptop (120x160)

```



```

mouse.mouseWheel(1); %Jump To The Next Slide
delay(500);
elseif(c_cent_r>150) % for my laptop (120x160)
mouse.mouseWheel(-1); %Jump To The Previous Slide
delay(500);
end
end
end

```

get_MY_THRESHOLDING.m

```

function [red grn blu]=get_MY_THRESHOLDING(im)
% This function gives the rgb values to be used
% for thresholding of a MARKER/COLOR
p = impixel(im); % Lets user select multiple pixels on the marker
[r c]=size(p);
red = mean(p(:,1));
grn = mean(p(:,2));
blu = mean(p(:,3));
[r c d]=size(im);
output_image=zeros(r,c);
% Creating output image based on selected pixels on the MARKER/COLOR.
range = 20;
for i1=1:r
for i2=1:c
if( (im(i1,i2,1)>red-range) && (im(i1,i2,1)<red+range) && (im(i1,i2,2)>grn-range) &&
(im(i1,i2,2)<grn+range) && (im(i1,i2,3)>blu-range) && (im(i1,i2,3)<blu+range) )
output_image(i1,i2)=1;
end
end
end
end
figure();

```

```
imshow(output_image);  
end
```

Centroid1.m

```
function [r_cent c_cent]=centroid1(arg)  
[r_index c_index]=find(arg);  
r_cent=mean(r_index);  
c_cent=mean(c_index);
```

Delay.m

```
function []=delay(x)  
for i=1:500  
    for j=1:500  
        for k=1:x %if u give x=420 u will get a delay of 1 second.  
            end  
        end  
    end  
end
```

4.5 Taking Snapshot

Main.m

```
clc % clear the command prompt  
close all  
fprintf('wait please\n'); % to print a string in command prompt  
imaqreset % disconnect and delete all image aquisition objects  
clear all % clear variable and functions from memory  
vidobj = videoinput('winvideo',1); % creates a video input object  
set(vidobj,'ReturnedColorSpace','RGB'); % setting the properties of object  
set(vidobj,'FramesPerTrigger',1);
```

```

set(vidobj,'TriggerRepeat',inf);
triggerconfig(vidobj,'manual');
start(vidobj);
range=20;
fprintf('Bring Markers....\n');
delay(2000);
fprintf('Thresolding for RED Marker\n');
im = getsnapshot(vidobj);
[RD_M_R RD_M_G RD_M_B] = get_MY_THRESHOLDING(im);
delay(2000);
fprintf('Thresolding for YELLOW Marker\n');
%im = getsnapshot(vidobj);
[YL_M_R YL_M_G YL_M_B] = get_MY_THRESHOLDING(im);
delay(300);
close all
while(1)
    trigger(vidobj);          % Manually initiate logging/sending for running object.
    im=getdata(vidobj);      % Return acquired data samples from engine to MATLAB
workspace.
    [r c d]=size(im);
    output_image_r=zeros(r,c);    %image red
    output_image_y=zeros(r,c);    %image Yellow
    A=0;
%To Detect Red mark/tape
    for i1=1:r
        for i2=1:c
            if( (im(i1,i2,1)>RD_M_R-range) && (im(i1,i2,1)<RD_M_R+range) &&
(im(i1,i2,2)>RD_M_G-range) && (im(i1,i2,2)<RD_M_G+range) && (im(i1,i2,3)>RD_M_B-
range) && (im(i1,i2,3)<RD_M_B+range) )
                output_image_r(i1,i2)=1;

```

```

    end
end
end
figure(2),imshow(output_image_r);
    title('Red');
[r_cent_r c_cent_r]=centroid1(output_image_r);
disp([r_cent_r c_cent_r]);
%To Detect Yellow mark/tape
for i1=1:r
    for i2=1:c
        if( (im(i1,i2,1)>YL_M_R-range) && (im(i1,i2,1)<YL_M_R+range) &&
(im(i1,i2,2)>YL_M_G-range) && (im(i1,i2,2)<YL_M_G+range) && (im(i1,i2,3)>YL_M_B-
range) && (im(i1,i2,3)<YL_M_B+range) )
            output_image_y(i1,i2)=1;
        end
    end
end
figure(3),imshow(output_image_y);
    title('Yellow');
[r_cent_y c_cent_y]=centroid1(output_image_y);
disp([r_cent_y c_cent_y]);
r = sqrt( ((r_cent_r) - (r_cent_y)).^2 + ((c_cent_r) - (c_cent_y)).^2 )
if(r<40)
    delay(1500);
    Image=getsnapshot(vidobj);
    figure(4),imshow(Image);
end
end

```

get_MY_THRESHOLDING.m

```

function [red grn blu]=get_MY_THRESHOLDING(im)
% This function gives the rgb values to be used
% for thresholding of a MARKER/COLOR
p = impixel(im); % Lets user select multiple pixels on the marker
[r c]=size(p);
red = mean(p(:,1));
grn = mean(p(:,2));
blu = mean(p(:,3));
[r c d]=size(im);
output_image=zeros(r,c);
    % Creating output image based on selected pixels on the MARKER/COLOR.
range = 20;
for i1=1:r
    for i2=1:c
        if( (im(i1,i2,1)>red-range) && (im(i1,i2,1)<red+range) && (im(i1,i2,2)>grn-range) &&
(im(i1,i2,2)<grn+range) && (im(i1,i2,3)>blu-range) && (im(i1,i2,3)<blu+range) )
            output_image(i1,i2)=1;
        end
    end
end
figure();
imshow(output_image);
end

```

Centroid1.m

```

function [r_cent c_cent]=centroid1(arg)
[r_index c_index]=find(arg);
r_cent=mean(r_index);
c_cent=mean(c_index);

```

Delay.m

```
function []=delay(x)
for i=1:500
    for j=1:500
        for k=1:x %if u give x=420 u will get a delay of 1 second.
            end
        end
    end
end
end
```

4.6 IMPLEMENTATION

- The first step is taking a video input object. The video input object should have the user's hand with finger caps preferably with a white background. Take a snapshot of it.

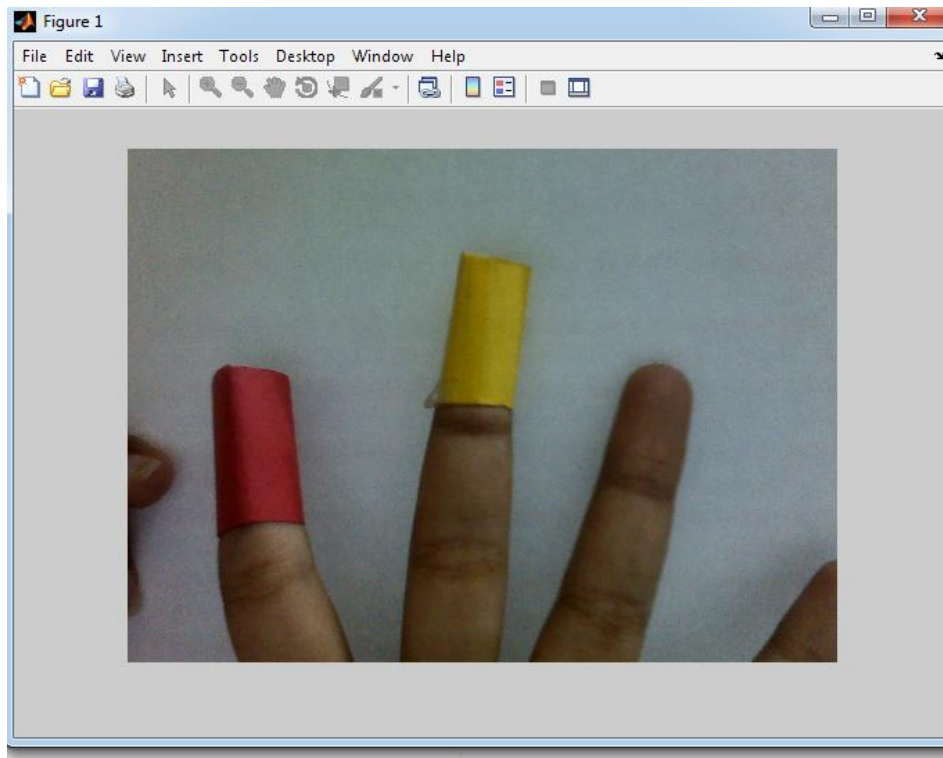


Figure 4.1 Detecting Finger caps

- The second step is selecting multiple pixels on a finger cap.

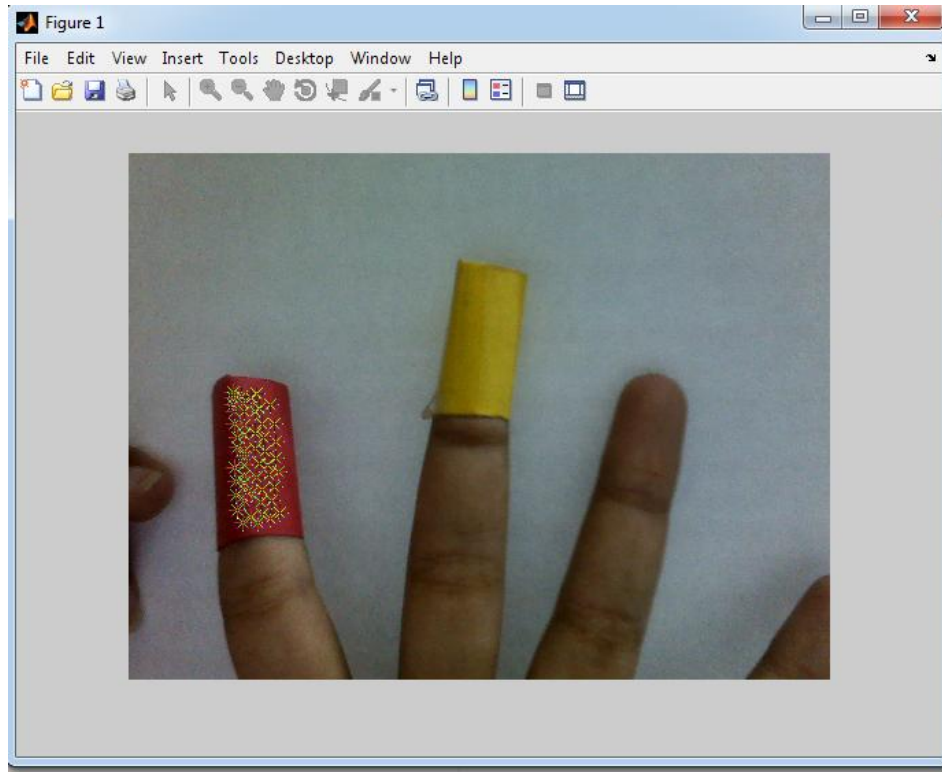


Figure 4.2 Select multiple pixels

- Then a threshold value for the colored marker will be calculated. Based on that threshold value, the finger cap will be detected while performing the gestures.

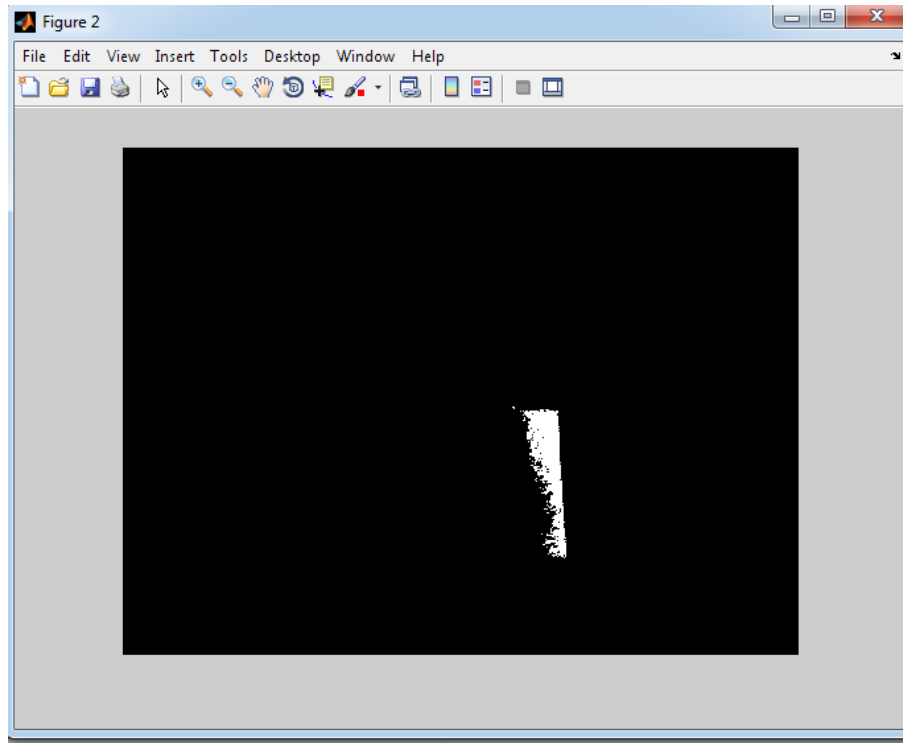


Figure 4.3 Detection of colored marker

- Then based on the user's gesture operations will be performed.

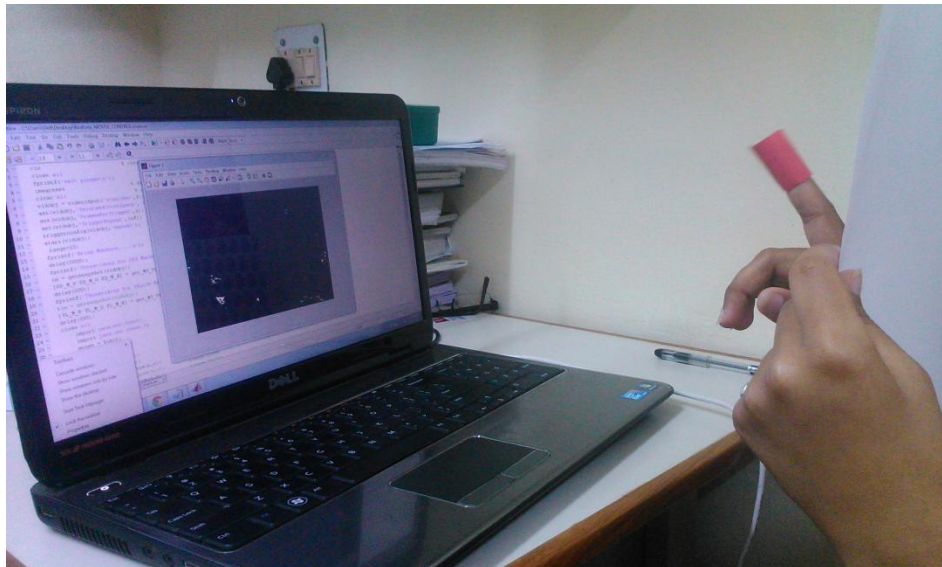


Figure 4.4 Performing gestures

CONCLUSION

1. Sixth Sense is a wearable device that enables us to have the whole world of digital information at our fingertips.
2. This sets information free and the whole world is our screen or personal computer.
3. It has been classified under the category of wearable computing.
4. The true power of Sixth Sense lies on its potential to connect the real world with the Internet, and overlaying the information on the world itself.
5. The key here is that Sixth Sense recognizes the objects around us, displaying information automatically and letting us access it in any way we want, in the simplest way possible.
6. Clearly, this has the potential of becoming the ultimate "transparent" user interface for accessing information about everything around us. If the colored finger caps are removed and it ever goes beyond the initial development phase. But as it is now, it may change the way we interact with the real world and truly give everyone complete awareness of the environment around us.

APPENDIX A

Tool description

Matlab

MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

The MATLAB application is built around the MATLAB language, and most use of MATLAB involves typing MATLAB code into the Command Window (as an interactive mathematical shell), or executing text files containing MATLAB codes, including scripts and/or functions.

We can use MATLAB for a range of applications, including signal processing and communications, image and video processing, image processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

Advantages of using matlab

- A very large (and growing) database of built-in algorithms for image processing and computer vision applications
- MATLAB allows you to test algorithms immediately without recompilation. You can type something at the command line or execute a section in the editor and immediately see the results, greatly facilitating algorithm development.
- The MATLAB Desktop environment, which allows you to work interactively with your data, helps you to keep track of files and variables, and simplifies common programming/debugging tasks
- The ability to read in a wide variety of both common and domain-specific image formats.
- The ability to call external libraries, such as OpenCV
- Clearly written documentation with many examples, as well as online resources such as web seminars ("webinars").
- Bi-annual updates with new algorithms, features, and performance enhancements

- If you are already using MATLAB for other purposes, such as simulation, optimization, statistics, or data analysis, then there is a very quick learning curve for using it in image processing.
- The ability to process both still images and video.
- Technical support from a well-staffed, professional organization (assuming your maintenance is up-to-date)
- A large user community with lots of free code and knowledge sharing
- The ability to auto-generate C code, using MATLAB Coder, for a large (and growing) subset of image processing and mathematical functions, which you could then use in other environments, such as embedded systems or as a component in other software.

REFERENCES

1. International Journal on Recent and Innovation Trends in Computing and Communication, April 2013
2. Pranav Mistry, The Thrilling Potential of Sixth Sense Technology, TEDIndia 2009
3. Pranav Mistry, Integrating Information with the real world, MIT Media Lab
4. Monika Arora, Basic Principles of Sixth Sense Technology, VSRD International Journal of Computer Science and Information Technology, 2012, vol. 2(8)
5. R. Silva, J. C. Oliveira, G. A. Giraldi, Introduction to Augmented Reality, National Laboratory for Scientific Computation, Av. Getulio Vargas, 333 - Quitandinha - Petropolis-RJ, Brazil

SUMMARY

Objective

The goal is to manage computers and other devices with gestures rather than pointing and clicking a mouse or touching a display directly.

Description of the work

Here, I have used a new approach for controlling mouse movement using a real-time camera. Most existing approaches involve changing mouse parts such as adding more buttons or changing the position of the tracking ball. Instead, I propose to change the hardware design. My method is to use a camera, colored finger markers and sixth sense technology, such as image capturing and gesture recognition to control mouse tasks and I show how it can perform everything current mouse devices can. So, I intend to build this mouse control system using sixth sense technology.

Why I choose this topic?

Humans take decisions after acquiring inputs from the senses. But the information we collect aren't enough to result in the right decisions. But the information which could help making a good decision is largely available on internet. Although the information can be gathered by connecting devices like computers and mobiles but they are restricted to the screen and there is no direct interaction between the tangible physical world and intangible digital world. This sixth sense technology provides us with the freedom of interacting with the digital world with hand gestures. This technology has a wide application in the field of artificial intelligence. This methodology can aid in synthesis of bots that will be able to interact with humans.