Table of Contents

Table of Figures

Summary:

In this project, we developed a Human Machine Interaction HMI which implements 'touch less' mouse controlling using color detection techniques and extracting meanings from 'colored Marker' gesture. The interface allows the user to use their hands (Marker) to control the mouse as well as easily draw simple shapes. This interface is simple enough to be run using a webcam and requires little training that how to move cursor and click.

Marker-based motion-capture has been demonstrated in several interactive systems and prototypes. However, these systems require obtrusive retro-reflective markers or LEDs and expensive many camera setups. They focus on accuracy at the cost of ease of deployment and configuration. While our system cannot provide the same accuracy as high-end optical mocap, our solution is simpler, less expensive, and requires only a single camera.

Chapter One

1.1 Introduction and Background

The aim of this paper was to research and understand how computer software can be written with the objective of tracking the color. We chose this topic because our area of interest is 'computer vision' and learning how this technology will be more helpful for developing applications that are cheap and more interactive. So we decided to develop application, if properly used, could potentially increase the performance of mouse and also allow many kinds of people to use it. We have called this project "Real Time Color Tracking" because ultimately the color would be track in order to handle the mouse. So if we can detect the color, we should be able to handle it.

In this paper we proposed a Human Machine Interaction HMI which implements "Touch less" mouse controlling using color detection techniques. The interface allows the user to use their hands (Marker) to control the mouse as well as easily draw simple shapes. This interface is simple enough to be run using a webcam and requires little training that how to move cursor and click.

Image processing is one of the complex fields because it includes sometimes Artificial Intelligence or learning algorithms. We have tried to make the simpler one by using color tracking instead of learning. The use of only basic input devices such as the mouse and keyboard is limiting the creativity and capabilities of the user. Our goal with this project was to expand the ways that people are able to interact with their computers. Namely, we wanted to enable users to interact more naturally with their computer through using marker to move the mouse and perform tasks.

One task which is especially very common to perform with the mouse is clicking. This is something that many computer users do quite often - opening several applications, folders, browser windows, etc. on their screen, and then switching between them frequently. Click icons with your finger will be more easy and quick, since you already moving the cursor with hand, as touch less environment is 20 % faster than touch applications. The colored finger can be detected using simple image color detection algorithms instead of having to use complicated hand-tracking algorithms, learning and hand shape recognition techniques.

This color tracking can be recognized in real time, and once a color is recognized, measurements can be found for its movement and orientation. These measurements can then be translated into mouse events, letting the user control the cursor and easily click on desired icons.

In this fast growing world of technologies as the new things are discovering, many other matters are being explored, research on the existing issues is in progress, it's important to make awareness in people in order to get success or to further expand the knowledge. This is what the main purpose of education, seminars, workshops and other presentations. Beside this, no matter either it's a business meeting or teaching session to clear the idea or concepts of the listeners what the presenter use is a whiteboard. It's no doubt the most common and convenient way to describing the things whether it's a business model, chemical bonding, drawing or a mathematical solution. Students always try to copy all the lecture in their notebooks, employees try to keep all the key points in their minds or sticky notes, in the race of copying the material what the common behavior is observed that they usually lose their concentration or sometime their minds rattle but still they could not copy all the stuff. To resolve this problem many other solutions are being proposed like recording the sessions, multimedia presentations etc.

2. Problem Statement

The main problem is exact copy of what is written on the whiteboard in low computational cost, less expensive way and much efficient way? Because to illustrate the presentation slides again what we use is whiteboard. There are many touch technologies and surface computing as well. But the problem with them is they need to have special kind of equipments to support them. Physical equipments heat up by using. They take much time to compute for instance, when we click a mouse it generates the hardware interrupt and that take time to compute. Uses of hardware equipments generate pollution. Sometimes it needs resistance to move mouse or other hardware. Frequent use of hardware may produce the scratches over your screen. So touch less solves more less all these problems and this technology is more interactive.

There are already many applications to solve this problem but still improvement needed. Like some applications work with hand gesture recognition or different techniques. But image processing is a heavy process it occupy memory and more computational power than other applications. So we need an application that is less computationally expensive, efficient and more interactive and solve the problem of saving exact what is written on whiteboard.

Chapter Two

2.1 Literature Review

2.1.1 Computer Vision and Machine Vision

“Computer vision is the science and technology of machines that see” (Wikipedia, 2009)

“The goal of computer vision is to develop algorithms that take an image as input and produce a symbolic interpretation describing which objects are present” (Amit, 2002)

“Computer vision is the study of methods which allow computers to "understand" images, or multidimensional data in general” (NationMaster.com, 2003)

Computer Vision (CV) is a generic term, and is the foundation of more focused fields, including;

- Video tracking
- Object recognition
- Motion estimation
- Image restoration
- Image segmentation

Machine Vision (MV) is a specialization of Computer Vision (CV), typically found in industrialized environments, such as manufacturing plants. MV itself is a collection of techniques that work together to allow machines to “see”. MV is particularly good at automating tasks previously performed by humans, helping eliminate human error. MV systems can also improve automation, conduct visual inspections and perform high speed analysis of objects 24 hours a day. (MachineVision.co.uk, 2008)

MV is an implementation of some of, or all of, the above fields, in particular; Automation, Imaging, Mathematics and, to a degree, AI. The goal of MV is to "understand" an image by performing advanced analysis on that image.

MV systems perform the following tasks to "identify" objects: (MachineVision.co.uk, 2008)

2.1.2 Image Capture

Uses an optical (photosensitive) input device to capture light particles and transform them to a digital equivalent. This is the most important stage of determination. Without well lit, shadow less imagery, image processing may be fundamentally flawed and produced erroneous results. In order to achieve the best results of lighting, we need consider contrast, intensity and position of the light (Burns, 1999).

1. Proper lighting will create contrast. Good contrast will make it easier to identify:
2. Using intense light will help improve contrast. The intensity of the light will directly influence the sharpness of the image. A sharper image will have less "noise", which will result in less work needing to be done in the image pre-processing phase.
3. The angle of reflection is equal to the angle of incidence. (Photoflex Lighting School, 2009). It is important to place the light in a position where it will not cause objects to cast shadows.

Positioning multiple lights at varying angles will eliminate shadows. The light should reflect differently off the object of interest than the background. Creating a high contrast image, using multiple light sources positioned at various angles will help create an image with an easily identifiable background/foreground with "magnified" features that will help make image processing techniques work more efficiently.

1. Image Pre-processing:

This is the process of tidying up an image before performing the main algorithms on that image. This step is the process of eliminating or cleaning up elements of the image before it heads for main processing. The pre-processing phase could conduct the following processes:

1. **Grayscale**

The image could be full color dependant on the camera in use. Converting the image to grayscale could make features stand out more clearly.

2. **Binarization**

This is the process of converting a grayscale (or full color) image to a binary "black and white" image. This is well suited to template matching and blob detection algorithms. One example would be Optical Character Recognition (OCR).

3. **Pixel noise reduction** (Mathematical Morphology, Erosion) (AForge.NET, 2009)

This is the process of eliminating pixels from an image that are not surrounded by a specific number of neighboring pixels. This cleans up the image and takes away pixels that are not needed, but could be identified as "areas of interest".

2.1.3 Surface Computing

Surface computing is the term for the use of a specialized computer GUI in which traditional GUI elements is replaced by intuitive, everyday objects. Instead of a keyboard and mouse, the user interacts directly with a touch-sensitive screen. It has been said that this more closely replicates the familiar hands-on experience of everyday object manipulation.

Microsoft's Surface computer uses multi touch to let people manipulate objects more like the real world. In this light table example, photos are moved around as if they were printed on paper, but can also be resized. (multitouch)

Applications of surface computing are:

- Surface Computing / Tabletop Display Research (research.microsoft.com)
- Microsoft Surface (microsoft_surface)
- Evoluce introduces touchless multi-touch to surface computing

2.1.4 Touch Technology

Touch screen technology has the potential to replace most functions of the mouse and keyboard. The touch screen interface is being used in a wide variety of applications to improve human-computer interaction. As the technology advances, people may be able to operate computers without mice and keyboards. There are many types of touch screen technologies, but some of them are stated below:

- Wire Resistive Touch Screens (consists of a glass or acrylic panel that is coated with electrically conductive and resistive layers. The thin layers are separated by invisible separator dots.)
- Capacitive Touch Screens (consists of a glass panel with a capacitive (charge storing) material coating its surface.)
- Pen Touch Capacitive Touch Screens (a durable capacitive type touch screen with an attached pen stylus. It can be set to respond to finger input only, pen input only, or both.)
- Surface Acoustic Wave Touch Screens (one of the most advanced touch screen types. It is based on sending acoustic waves across a clear glass panel with a series of transducers and reflectors. When a finger touches the screen, the waves are absorbed, causing a touch event to be detected at that point.)

Some of the basic are Resistive, Capacitive, Infrared (IR), and Surface Acoustical Wave (SAW). Each of those designs has distinct advantages and disadvantages.

Basic working of touch screen includes these basic steps:

- Screen register touch
- Raw data is captured
- Background noise is removed
- Pressure points are measured
- Touch areas are established
- Exact coordinates are calculated

2. Working of touch technology

Think of the screen as a multilayer sandwich. On top is a thin plastic or glass sheet with a resistive coating on its bottom. The plastic or glass floats on a thin layer of nonconductive oil, which rests on a layer of glass coated with a similar resistive finish. Thin bars of silver ink line the horizontal and vertical edges of the glass. Direct current is applied alternately to each pair of bars, creating a voltage field between them. When you touch the stylus to the screen, the plastic pushes down through the gel to meet the glass (called a 'touchdown'). This causes a change in the voltage field, which is recorded by the touch screen's driver software. By sending current first through the vertical bars and then the horizontal ones, the touch screen obtains the X and Y coordinates of the touchdown point. The driver scans the touch screen thousands of times each second and sends this data to any application that needs it. In this way, a PDA or PC knows when you're tapping an on-screen icon to launch a program or gliding it across the screen to enter data. (www.funtrivia.com)

3. Problems in Touch Screens

Human touch

Whilst being used, the finger and hand obscure what is on the screen. Smudges left by fingers on the screen can decrease legibility.

Screen position

Sunlight can degrade the view ability of the display for all users. The screen should be shielded from direct or reflected sunlight or other bright light sources. The display should be viewable from the eye level of a person standing and from a person sitting in a wheelchair. People with low vision should not be prevented from getting their faces close to the screen.

Parallax problems

The conflicting requirements of tall users and short wheelchair users can lead to a significant group of users having parallax problems when lining up the controls with the displayed option. According to the Namahn report (2000), users tend to touch the sides of the screen and slightly below target areas, especially for targets near the top of the display and when the screen stands at a steep angle of between 45 and 90 degrees. (www.tiresias.org)

One of the disadvantages of having touch screen capability on your cellphone or computers is the screen of the gadgets will end up with lots of fingerprints and become greasy. The next breakthrough in technology would be looking into technology that allows you to control your gadgets and computer without having you to touch it. Guys at Norway's Elliptic Labs have achieved this. What they have is a touchless interface that lets your fingers stay a few inches or more away from your computer screen, without touching it, and you'll be able to control the movement of the 3D objects in the computer.

3. Touch less Technology

It is better than the touch technology because its computational power is 10-15 times less than that of touch technology. Gesture based interaction states many methods like:

Data and Cyber Glove:

In these cases gestures are typically recognized using pre-trained templates; however gloves can also be used to identify natural or untrained gestures. A good review of vision based gesture recognition is provided by Palovic et. al. (1995).

There are the following limitations for image based visual tracking of the hands. (Zeltzer, 1994). The resolution of video cameras is too low to both resolve the fingers easily and cover the field of view encompassed by broad hand motions. The 30- or 60- frame-per-second conventional video technology is insufficient to capture rapid hand motion.

Now latest work on agent base system which understand different body gesture and response the users. In this research paper some technologies are discussed.

Tracking Technologies

Gesture-only interfaces with syntax of many gestures typically require precise hand pose tracking. A common technique is to instrument the hand with a glove which is equipped with a number of sensors which provide information about hand position, orientation, and flex of the fingers. Once hand pose data has been captured by the gloves, gestures can be recognized using a number of different techniques. Neural network approaches or statistical template matching is commonly used to identify static hand poses, often achieving accuracy rates of better than 95%.

Natural Gesture Only Interfaces

At the simplest level, effective gesture interfaces can be developed which respond to natural gestures, especially dynamic hand motion.

Symbolic Gesture Recognition

Symbolic gesture interfaces are often used in immersive virtual environment where the user cannot see the real world to traditional input devices.

There are three major ways to perform touch less i.e

- By data glove or cyber glove
- By LEDs
- By marker tracking

Touch less technology has replaced the need of touch screen or the use of mouse. Mouse was better in a sense that it can be installed without installation software it was just plug and play. The disadvantage of mouse was, they need flat surface to be used. Or else they won't function.

There are many applications of touch less technologies and researches as well.

1. With a glove having infrared light emitters on the fingertips, the setup will allow the API to distinguish gestures such as pinching fingers, turning hands, and so forth. The goal is to use positional information and temporal trajectories to track gestures. The main goal of this design project is to replace mouse functions with hand gestures, which can be more natural and intuitive to use. To accomplish this, an API was created to recognize basic gestures. The setup includes two Wii remotes placed in parallel adjacent to each other. The user wears a glove with infra red LEDs on the index finger and thumb. The glove must be approximately one foot away from the Wii remotes. There exist more complex gestures such as drawing a circle, drawing an X to close a window, or a combination of simple movements to execute other functions. To detect such gestures, both Finite State Machines (FSM) and Neural Networks were explored. (University)

2. Another research paper presents a hand gesture recognition system to recognize real time gesture in unconstrained environments. The system consists of three modules: real time

hand tracking, training gesture and gesture recognition using pseudo two dimension hidden Markov models (P2-DHMMs). (Nguyen Dang Binh)

3. In this research Sketches are hand-drawn informal figures often created as a way of thinking about or working through a problem. Sketch-understanding systems let users interact with computers by drawing naturally, offering a freedom not available with traditional CAD systems. (Davis)

2.1.5 Previous works

Many works in this regard has been done by other researchers as well. But the common aspect in all is the error rate. Basically HMI involves the same type of root study like motion detection, color detection and gesture recognition. There is only the difference between them is technologies, approaches and mechanical devices they used along with their prior understanding.

1. **Real-time fingertip tracking and gesture recognition**

An excellent way of dealing with dynamic backgrounds and low contrast environments is to work in a different part of the electromagnetic (EM) spectrum as was done by Oka. Their use of an infrared camera made it possible to easily segment a human hand based on human body temperatures. This approach is very reliable for tracking; however, expensive hardware is needed. (Oka, 2002)

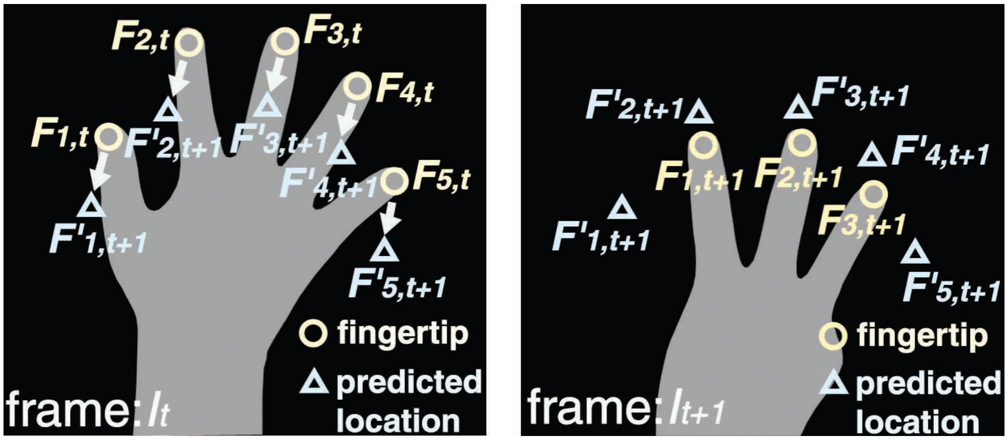

Figure 1: Taking fingertip correspondences: (a) detecting fingertips and (b) comparing detected and predicted fingertip locations to determine trajectories.

2. Understanding hand gestures using approximate graph matching

An approach presented by Miners has been successful in understanding the meaning of three to five ambiguous hand gestures. Their goal was to obtain a greater amount of understanding from gestures rather than simply recognizing them. This was not a bare-hand system; however it provided useful insight into ways to get dynamic gestures. (Miners, March 2005). They used one 18-sensor CyberGlove with an attached flock of birds is used to obtain accurate three-dimensional (3-D) representations of the hand.

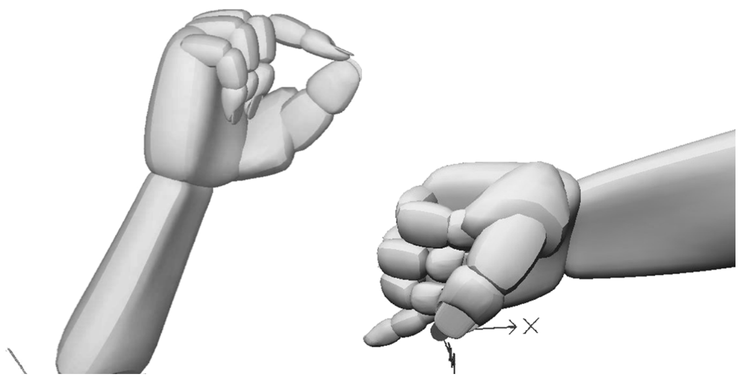

Figure 2: Visualization software illustrating part of a chicken gesture and a spaghetti gesture.

3. Robust Computer Vision-Based Detection of Pinching for One and Two-Handed Gesture Input

Another similar project was done by Andy Wilson as he controlled the cursor by a hand pinch gesture. In his project, he developed a machine-user interface which implements pinch gesture recognition using simple computer vision techniques. The interface allows the user to use their hands to control the mouse as well as easily move and resize windows that are open on their screen. (Wilson, 2006)

In his research, Wilson notes that there are several areas for improvement with his interface, including providing better cues to the user that a gesture has been recognized, such as visual or audible cues, as well as investigating better methods for implementing clicking and dragging than the current pinch-release-pinch gesture. (Fig 3)

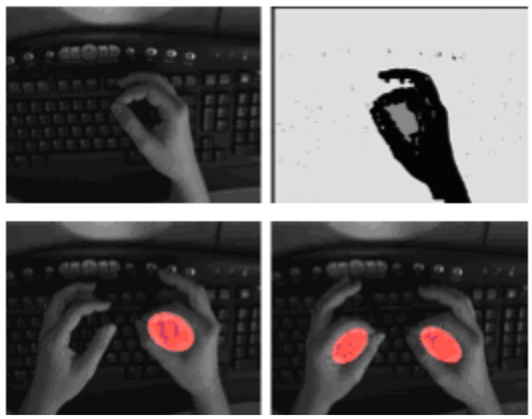

Figure 3: Upper left: Typical image of hand with thumb and forefinger together, acquired from camera above the keyboard. Upper right: Image segmentation indicates background pixels. The hole formed by the thumb and forefinger is detected as a distinct connected component (shown here darker in color). Bottom left: Ellipsoidal model of whole component for the closed hand, shown in red. Bottom right: two hands are detected.

4. FPGA-Based Real-Time Color Tracking for Robotic Formation Control

Real time color tracking has also been used for robot controlling. Here the field programmable gate array (FPGA) technology is applied in a prototypical tracking system for vehicles by using a CMOS camera to detect their color-tags. The raw image from the Bayer color pattern is used to indicate the 2-dimensional position of vehicles and encrypted infrared commands are issued to deploy them in a leader-follower formation (fig 4). (Ying-Hao Yu, 2007)

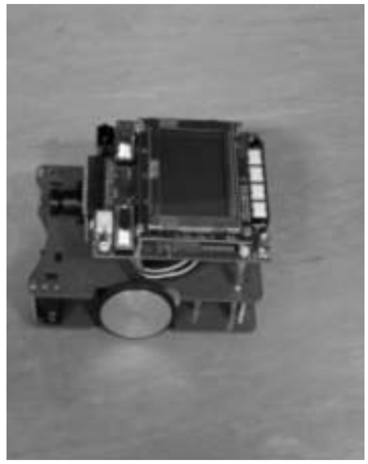

Figure 4: The performance of noise filtering

5. Real-time, Static and Dynamic Hand Gesture Recognition for HCI

Real-time, static and dynamic hand gesture recognition affords users the ability to interact with computers in more natural and intuitive ways. The hand can be used to communicate much more information by itself compared to computer mice, joysticks, etc. allowing a greater number of possibilities for computer interaction. The purpose of this work was to design, develop and study a practical framework for real-time gesture recognition that can be used in a variety of HCI (human-computer interaction) applications with the aim of developing a prototype system for controlling Microsoft PowerPoint presentations. The approach taken is to locate hands starting with the finger tips after investigating potential regions of interest that are maintained through motion detection and subsequent tracking. The fingertips are very distinguishable from many backgrounds allowing a very robust tracking system. Using techniques such as Features from Accelerated Segment Test (FAST) corner detection the tips can be found efficiently. Circles generated using Bresenham's algorithm was employed for finding corners as well as the center of the palm. Using features generated from these results along with a semi-supervised Adaptive Resonance Theory (ART) neural network, static gestures were able to be classified with an overall accuracy of 75%. Dynamic gestures on the other hand were able to be detected using the route formed by the center of the hand over a finite amount of time. Simple decision heuristics were then utilized to detect the movement of the hand. This research produced a working prototype of a robust hand tracking and gesture recognition system that can be used in numerous applications. (S.M. Hassan Ahmed, 2008). This application includes many complex algorithms and mechanisms for fingertips detection they used FAST and for the center of palm they used Bresenham's algorithm. (Fig 5)

Figure 5: Region of Interest containing a hand and this initial ROI was found from motion detection. Then the image was differenced with the reference frame in that region and binarized, giving a segmented hand image. The size of the ROI is preset during calibration.

6. An Airborne Bayesian Color Tracking System

Color tracking was also used in helicopter where only one pilot was needed to fly the helicopter and images were taken by him. The system keeps the object of interest automatically in the viewing field of the camera so that the only task left to the camera operator is to initiate the imaging process. This simple task can be additionally managed by the helicopter pilot so that a single person is sufficient to steer the helicopter and to take the images (fig 6).

Figure 6: Picture of the flying model helicopter with the mounted tracking system.

This project needed a lot of controlling without controlling this project was near to fail as stated in the experiments done with and without controlling. (Felix Woelk)

7. Real-Time Hand-Tracking with a Color Glove

In other paper, they propose a system that facilitates 3-D articulated user-input using the hands. Their approach uses a single camera to track a hand wearing an ordinary cloth glove that is imprinted with a custom pattern (fig 7). The pattern is designed to simplify the pose estimation problem, allowing us to employ a nearest-neighbor approach to track hands at interactive rates. They describe several proof-of-concept applications enabled by our system that we hope will provide a foundation for new interactions in modeling, animation control and augmented reality. (Wang)

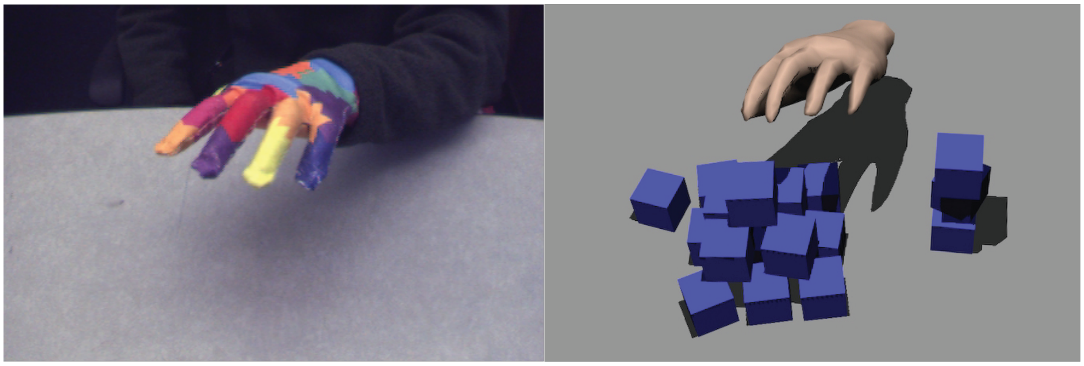

Figure 7: We describe a system that can reconstruct the pose of the hand from a single image of the hand wearing a multi-colored glove. We demonstrate our system as a user-input device for desktop virtual reality applications.

Chapter Three

3.1. Previous Attempts

In order to make this application work with maximum efficiency and less error we have done many attempts. Every experience has its own pros and cons. In this chapter we discuss about our previous attempts and their flow in next chapter we discuss about their pros and cons and then later we discuss our final attempt.

3.1.1. Laser

Our first attempt to implement HMI was with the help of laser light. The movement of cursor was controlled by the laser. The user has to have a laser, a web cam and a projector. The projector and webcam has to be in between the user and output area. As the application started it will take the image of output area, when the user pointed the laser onto it, application will scan the screen and check the brightest spot (i.e. laser will brighten the pixel it point) and as you move your laser the cursor will move according to that brightest pixel. This approach was very fast as there were no complicated algorithms involved and it had very low computation cost. Fig 8 is showing the basic flow of this approach.

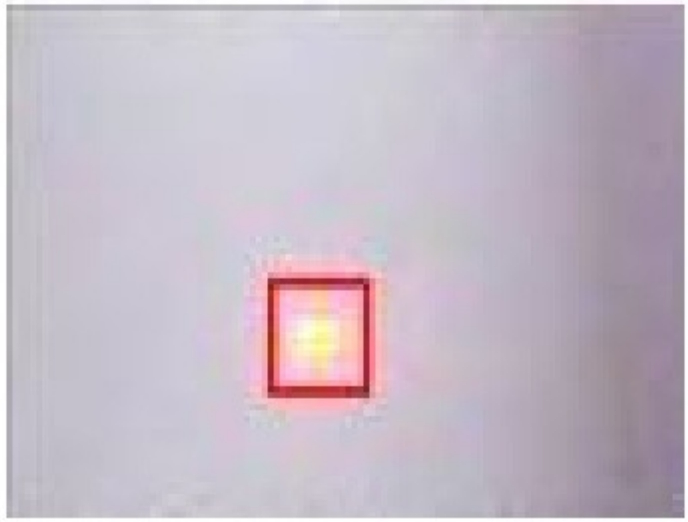

Figure 8: Brightest spot tracking

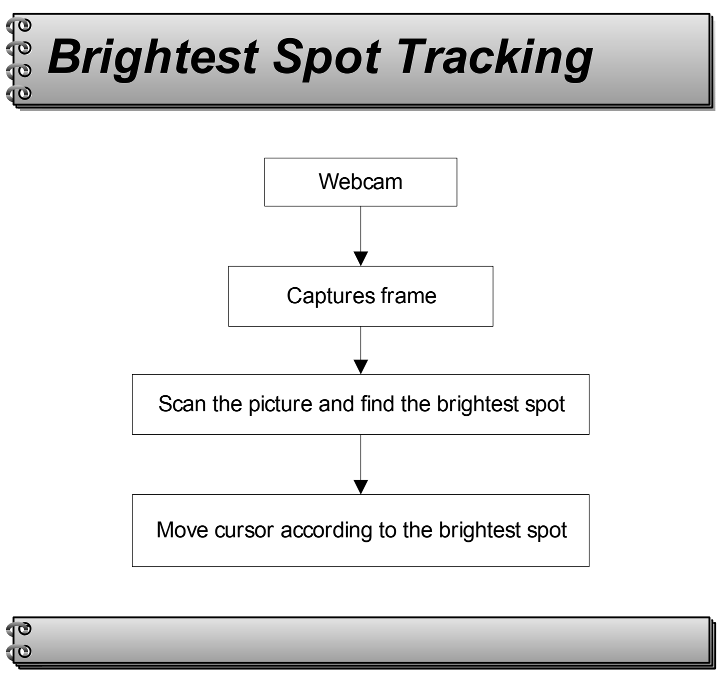

Figure 9: Color tracking with laser

3.1.2. Motion Detection

The second approach we use was motion detection. Motion Detection involves finding the areas of motion in a given scene. First we implemented it by simple color filter to detect required marker from the image. We take the motion area and then in that motion area we check required marker. You simply need a web cam that takes the image, you had to stand in front of cam with a red ball or something of red color when there motion detected in from of cam as the cam take the image of that and check for the red color in the area where motion is detected, it will start tracking it and makes the cursor move according to that red color.

3.1.2.1. Simple Background

In the other approach first we tried the Simple Background Modeling in which it compares the current image with the static background image and detects the change and if the defined color found in that area than the cursor will start moving as you move your marker.

```
Input:
Current Frame: F
Reference Frame: R
Threshold: ϕ
Output:
Binarized frame difference D

D = |F-R| ;
foreach pixel p of D do
if D(p) > ϕ then
        D(p) ⟵1 ;
end
else
        D(p)  ⟵ 0
end
end
```

Static Background Motion Algorithm 1

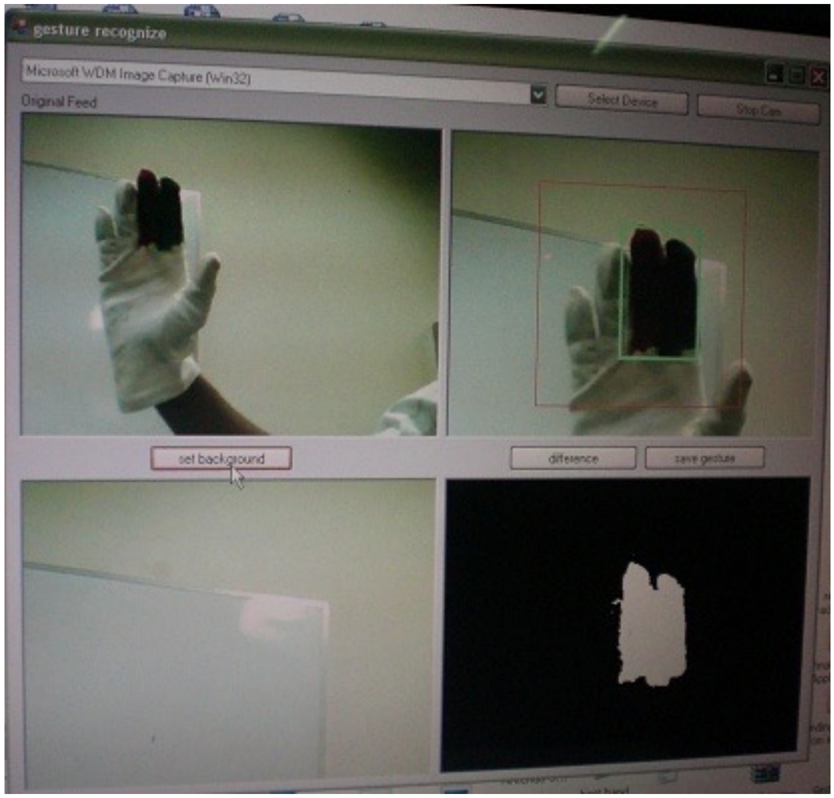

Figure 10: Motion Detection with static background

This is a simple procedure for motion detection. This grayscale image is then converted to a binarized image by examining which pixels are above and below a certain threshold. The binarized image can be processed more efficiently.

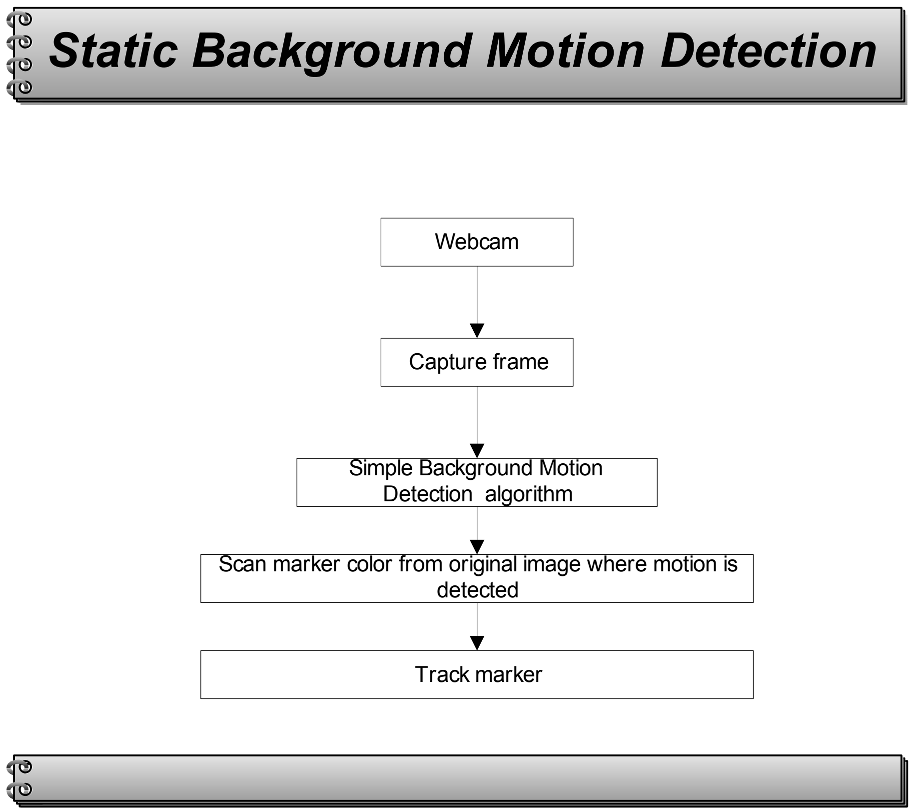

Figure 11: Motion Detection with static background

3.1.2.2. Consecutive Two Frame

Then we tried the other algorithm Two Frames Difference. In this technique it compares the current image with previous image and detects the change as it finds the motion it checks the required color into that motion area and moves the cursor according to the moving object.

Input:

Current Frame: F

Previous Frame: R

Threshold: ϕ

Output:

Binarized frame difference D

```
D = lF-Rl ;
foreach pixel p of D do
if D(p) > ϕ then
        D(p) ⟵1 ;
end
else
        D(p)  ⟵ 0
end
end
```

Dynamic Background Motion Algorithm 1

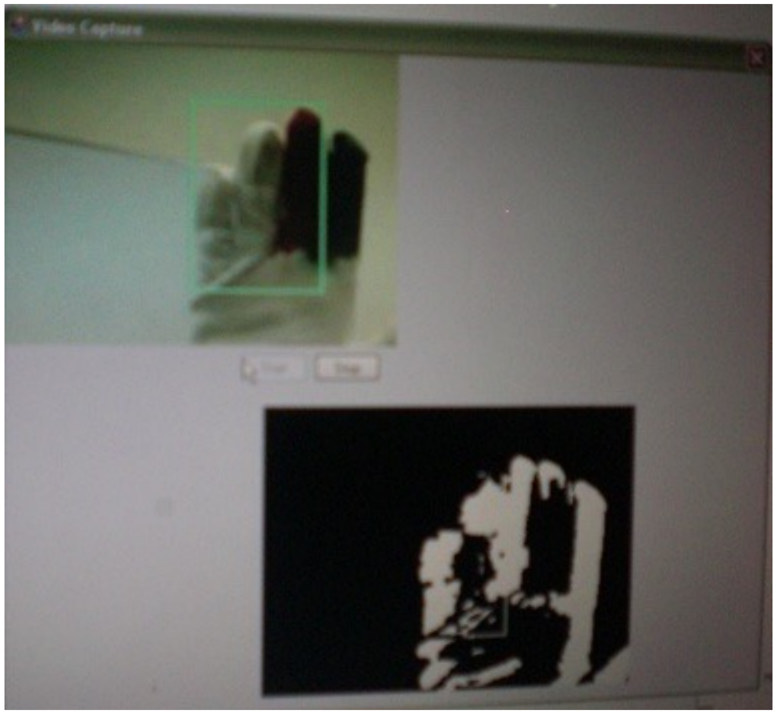

Figure 12: Motion Detection with Two frames Algorithm

Basic flow of two frames algorithm is shown in fig 13.

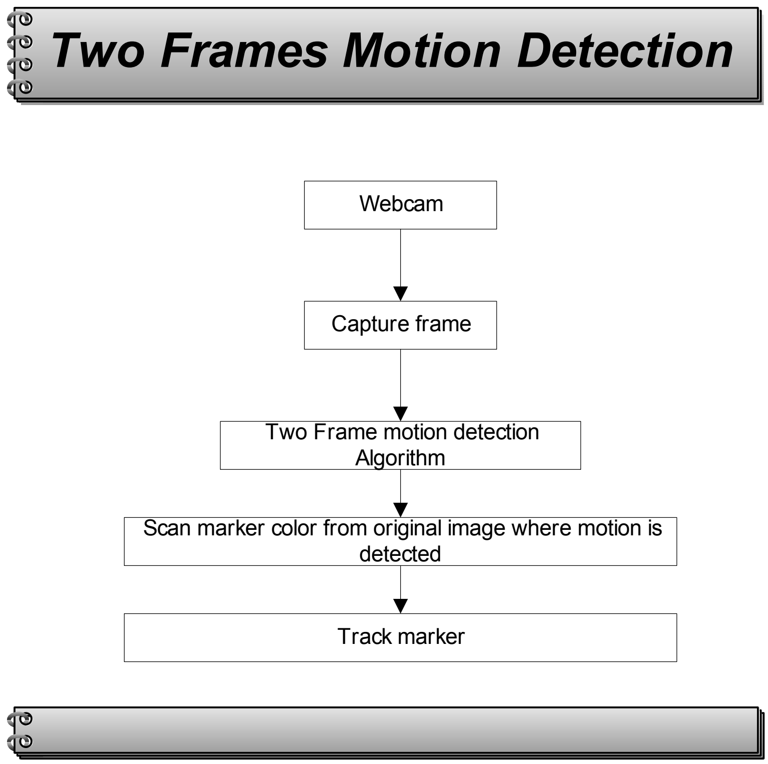

Figure 13: Motion Detection with Two frames Algorithm

3.1.3. Two Colors Tracking

Then we tried it on two colors. For instance, user is wearing a white color glove having its index and middle finger colored as red and green respectively. As the application starts up first the user has to scan both colors one for click and other for tracking. After taking the first image the algorithm starts searching for tracking color as it gets it the cursor starts moving with it. In the meanwhile it also checks the click color e.g. green whether it is close to red or not? As it detects that green is not close to red than its mean that user wants to click. (Fig 15) shows the basic flow of this approach.

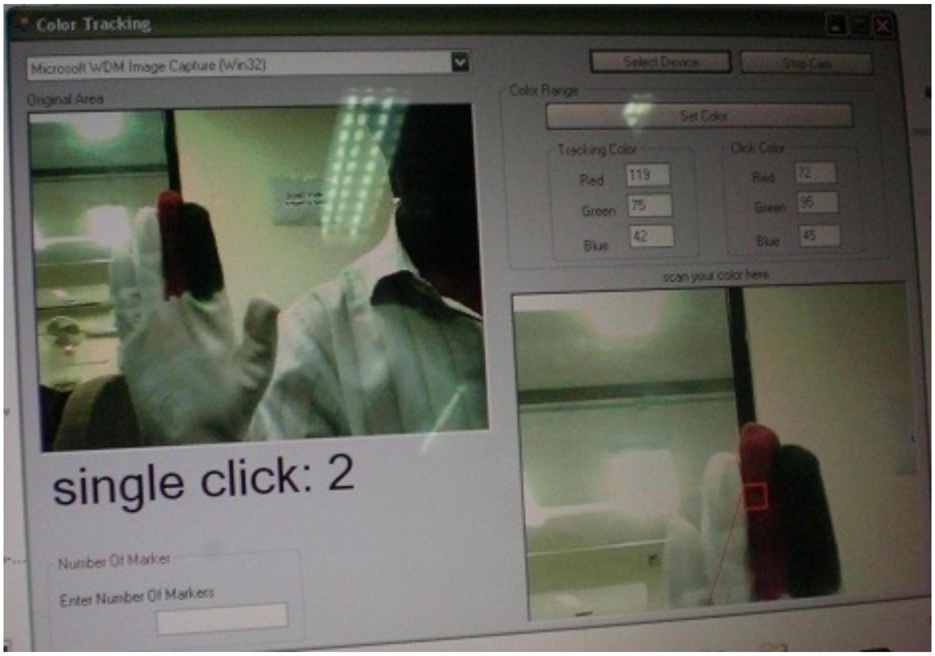

Figure 14: Two colors tracking

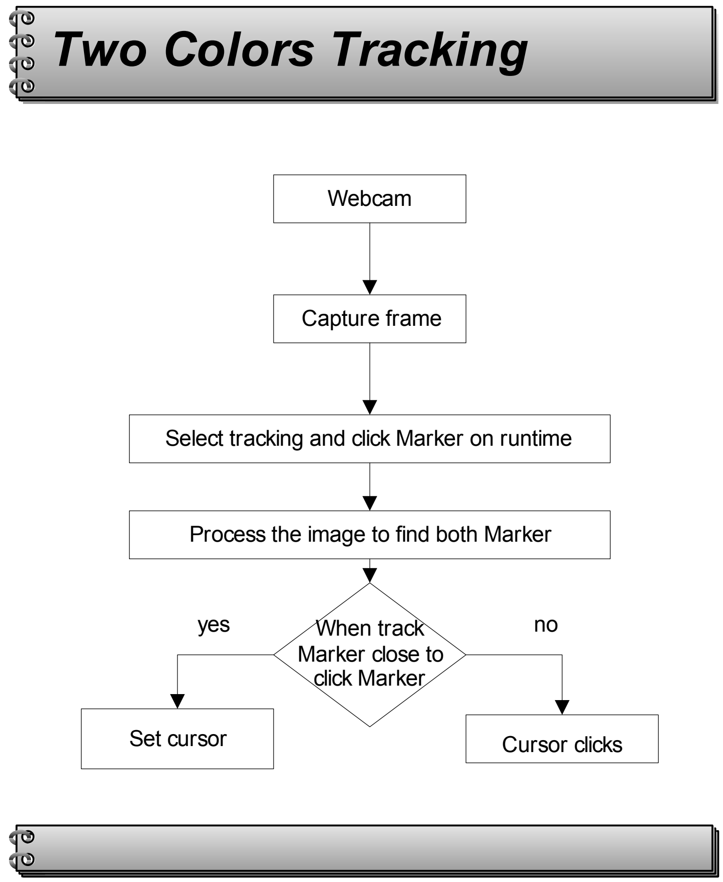

Figure 15: color tracking by red and green marker

3.1.4. Single Hand Color

Our forth trial was with the help of blob detection. Camera takes the input and filter out the required marker then to reduce the information grayscale the image then apply threshold to get the biggest blob. Cursor moves along with the biggest blob if required marker is that biggest blob, after finding coordinates of that biggest blob and calculate its mid then the cursor move according to this midpoint. In order to click the user has to move his hand in the backward position exactly half to its current position. This will be the mouse down event and then move back to its original position i.e. move hand to the forward position exactly the half distance will be the mouse up event and these events will be click event in total. For Click the cursor will remain at same position (there might be several hand gestures for click). It was faster than two colors.

Figure 16: Single Hand Color tracking

Single Hand Color

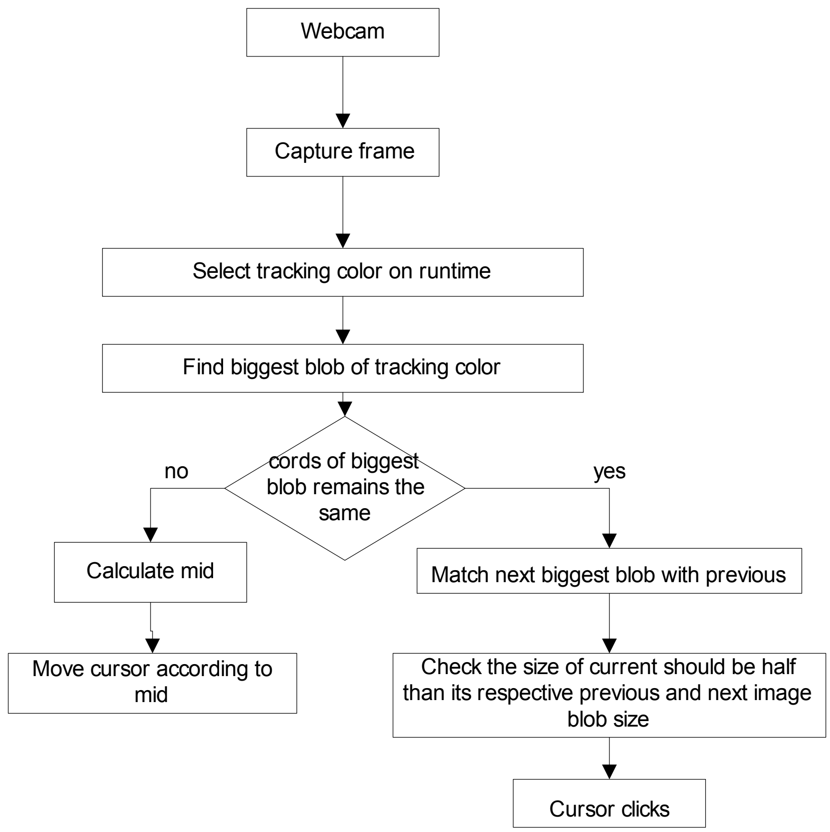

Figure 17: single hand color tracking

Chapter Four

1. Assessment and Evaluation

1.1. Laser

Although this approach was very fast and computationally it has very low cost but, it was using the mechanical device. It also restricts you to come in front of cam even if any other bright thing comes in front it application would not work properly as it will be confused which one is actual to track.

1.2. Motion Detection

First we implemented it by simple color filter to detect required marker from the image but finding required marker from a high resolution image is computationally costly. So we use motion detection algorithm to get the area of interest. Area of interest is that area where motion is detected. For motion detection there are two algorithms and we try both of them. These two approaches i.e. static background and two frame difference were fast but problem here is that we lose information of marker, like for click the size of marker will change if only a portion of marker will move other stay on same position like if you wear a glove which is required marker and if user just move finger and hand stay still this will cause ambiguous information. Simple background modeling is a very simple technique. The base image to use for comparison is obtained from the very first frame. However this technique presents some problems. The fact that just the first image is used for modeling is a big drawback since motion will be detected when an object disappears. This can be solved by adapting constantly changing the base image and find difference between two consecutive frames. This solves the problem with simple background modeling but requires a little more computer power. (Torres)

1.3. Two Color Tracking

This strategy worked well but it was comparatively slow because here we have to filter two markers from image. Two colors filtration and continuous check for click logic slower the function. Another problem was due to change in light intensity if any color looks like click marker and it is at a distance from the tracking marker then there be will click. This cause this application slow and higher error rate than other applications.

1.4. Single Color

This application is faster and has lower error rate. The efficient and fast algorithm of ‘Extraction of potential region of interest’ makes this application more efficient than all other applications. But still it has some limitations:

- Moving hand back and forth make this application complicated
- Exact required movements for click of hand some time is not possible.

- Not for click but when cursor move toward the input device and then back, demonstrate the click logic.
- Changing the distance of marker from video device change the blob size, but blob size would be small enough that its half is in range of noise.
- Change the object position change the color intensity of the object, so in color filtration the whole object might not be filtered properly.

Chapter Five

1. Method

1.1. Equipment

1.1.1. Webcam

We used a standard USB webcam at a resolution of 640x480. This viewing area proved to be not sufficient, though a larger area would have advantages as far as having more room for color tracking. This device can be place any where to get images; there is no constraint on placing the webcam at specific place.

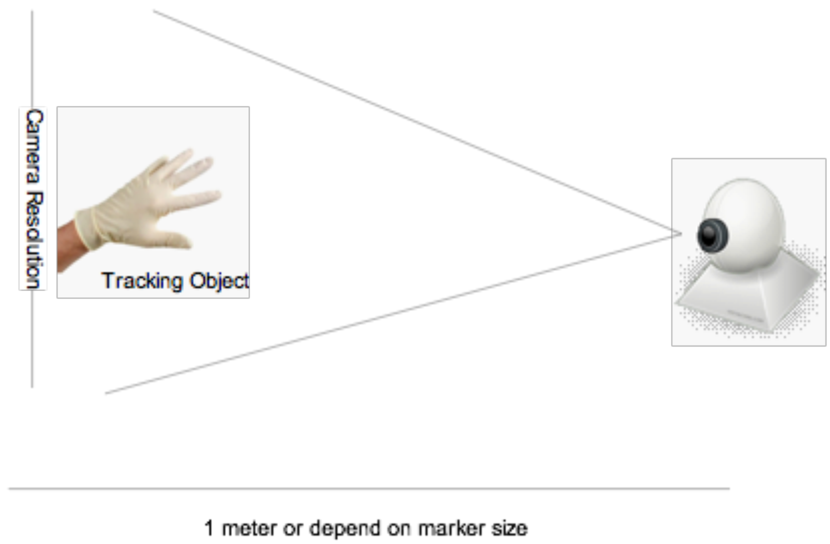

Figure 18: Example Setup

1.1.2. A Colored Marker

Any colored object can be used as the marker which can bend to half or less than half of its original size. Our use of a color glove is not inspired by any previous technique, the main purpose of using color tracking is that it requires fewer computations so it is fast as compare to gesture recognition and then track specific gesture. Change in marker size can also provide sufficient information like making different gesture.

We proposed this system that is not specific to particular design, size or color of tracker. So, fewer constraints on marker make it more interactive and real time.

Figure 19: Red color Marker on a Glove

Simple Cursor Move Gesture:

Simple marker size after sampling will be taken as cursor move gesture.

Figure 20 Cursor Move Gesture after Sampling

Click Gesture:

And half of sample tracker size will be taken as click gesture.

Figure 21 Click Gesture

1.1.3. Software

Windows: This operating system developed by Microsoft is a stable and trustworthy platform for running applications made with Microsoft Visual Studio.

Microsoft Visual Studio: It is Microsoft's software development product for programmers. It includes several programming languages such as Visual Basic, Visual C++, C# and J#. This project is developed in C#.

AForge library: it is a computer vision library developed by Microsoft. It focuses mainly on real-time image processing, and gives the user a large amount of tools that provide extensive functionality on this field.

Windows Mouse DLL is an open-source code library for mouse events.

1.1.4. Ordinary PC

Any standard PC running Windows OS is capable of running our project. The minimum hardware requirements are those of the operating system, webcam and the required software.

1.1.5. Best Environment

Best environment for this application is when the background color range doesn't lies between the ranges of marker color. And sufficient light conditions are needed for it. Proper lighting will create contrast. Good contrast will make it easier to identify required marker and its detail. The intensity of the light will directly influence the sharpness of the image. A sharper image will have less "noise", which will result in less work needing to be done in the image pre-processing. This application will also work in complex background environment.

1.2. User Interface

The interface of this application is really easy to understand, every icon explain the working behind it. There are total 5Forms in this application:

When this program is run, the user interface consists of just a window displaying the icons of monitor form, presentation form, select device form, scan tracker and control panel.

1.2.1. Main Form

This is main form of application, there are multiple icons on it some of them redirect to new form where multiple functions are available.

Figure 22: Main Form of the Application

User Actions:

1. In order to use this application first you have to select device by clicking first icon at upper side from left.

2. After selection of device select time for scanning and scan the color for tracking.
3. Now user can select system mode by clicking on the monitor icon**.**
4. Or user can select presentation mode by clicking on the icon.
5. Tracker sensitivity can be change by clicking on the mouse icon.
6. By clicking on the chat icon tooltips set to on and off.

Description:

1. There are six icons on the main form that provide different functionalities.
2. First icon upper side from left on main form is for selecting the required webcam connected to the system, there are also further operations that user can perform related of selecting the device connected to the system describe latter.
3. Monitor icon will take you to the monitor mode, where user can use system and use marker use as a mouse.
4. Mouse icon will allow you to set the sensitivity of tracker.
5. Scanner will allow you to scan the color you want to use as a marker for tracking.
6. Presentation mode icon will redirect you to presentation mode where you can write, erase, select color of the pencil, save and open the image.
7. Tool tip icon will allow you to on or off the “tool tips” for the usage of any tool.

1.2.2. Select Device

Video device is necessary for this application. This form allows multiple functions related to video devices connected to system.

Figure 23: Select Device Form of the Application

User Actions:

1. User will select the device form the drop down list.
2. Click on first icon from right to use that device.
3. Refresh the icon will recheck for the device connected to the system.
4. Third icon will stop the working device use for this application.

5. Last icon from right will close this form.

Description:

1. Name and version of all devices connected to this system are shown in the drop down list; user can select any device by just clicking on the any item in the list.
2. Clicking on ok button will connect that device to this application and video taken by that device will show in this form.
3. To stop that device just click on stop icon, third from right will stop selected device functioning.
4. Click on close button to close this form.

1.2.3. Scan Tracker

To use this application a marker is required, that marker will be selected from this from. When this form will be open video device selected in previous form will be selected again and video input will be show in this form. Required marker can be scan in multiple ways.

Figure 24: Scan Tracker Form of the Application

User Actions:

1. User need to set the timer for scanning the color. And place marker in marked area in the video.
2. User can check the scanned color by clicking on the first icon from right.
3. To restart the video device click on the refresh icon second from right.
4. User can close the window to move further towards the presentation or monitor mode.

Description:

1. A drop down list shows the timings that can be set for scanning the marker.
2. Here you scan your color you want to use as a marker.
3. After user click on time from dropdown list, use has to place the marker in marked are of the video that will scan a pixel of marker and set as required marker. The color combination of that pixel shown on the form as RGB.
4. User can also set marker by click on it from video input.
5. Or user just places the marker in marked area of video input and click on scan button. This will set the color places in the marked area as marker and also scan that particular frame and show marker.
6. RGB of the scanned color will be shown in RED, GREEN and BLUE field respectively.
7. When user select a marker, input taken from the video device will stop, by clicking on the second icon from right will restart the video input.

1.2.4. Monitor

In this mode user will be able to use system without conventional mouse, instead of using mouse user will able to track curser with marker and also able to perform click and drag and multiple functions.

Figure 25: Monitor Screen Form of the Application

User Actions:

1. In this mode user can set off the use of marker as a mouse by clicking on the mouse icon.
2. Closing this mode will take back to the main window.

Description:

1. In this mode video is taken by the selected device and selected marker will be track and user can perform different operations like click and drag etc.

2. Video taken by the video device is shown in this form and size of color marker is shown by drawing the rectangle in the form.

3. Length and width is numeric is also shown.

4. An arrow, directed down wards tells about the mouse down and up event. When marker is tracked the size of arrow is same as in start, when mouse down event happen the size of arrow will decrease and on mouse up event it came back to original position.

1.2.5. Presentation Mode

In presentation mode user can write in specified area and perform multiple operations, this white board provide many facilities as compare to writing on conventional whiteboard.

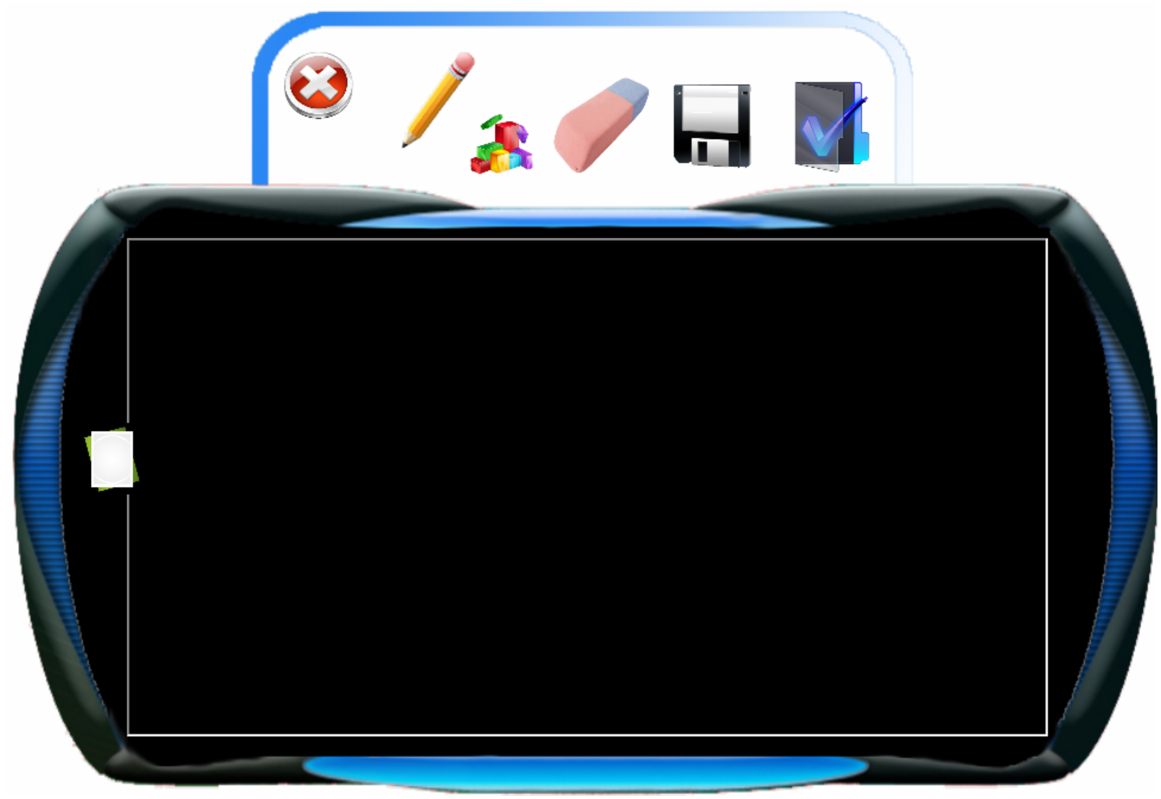

Figure 26: Presentation Mode Form of the Application

User Actions:

1. In this mode user can write on the whiteboard by selecting the pencil, first icon from left.
2. User can write in multiple colors by selecting color from color dialog box by clicking on the icon next to the pencil icon.
3. Written on the whiteboard can be eras by selecting the eraser tool next to the eraser tool.
4. What is written on the white board can be save by clicking on the save button, the 2 button from right.
5. Existing image can also be open in whiteboard using open file dialog box, first from right.
6. Whole screen can also rest to blank by clicking on the icon, on the left side of whiteboard on blank paper icon.

Description:

1. Pencil icons with multiple colors are available and user can select any color from color dialog and then write on the whiteboard with pencil.
2. Eraser tool set color of marker same as background, so that work like eraser. And whole screen can also reset by clicking on “new file” icon.
3. What is written on the whiteboard can be save using save icon and already saved image can also be open.

5.1.3 Algorithms

To make this application work following algorithms are used.

1. **Extraction of Potential Region of Interest**

In this application we have to first find area of interest, due to change in light and movement of object change the light intensity, so it is difficult to find out the region which is required and the goal of extraction of potential region of interest is to simplify the representation of an image into something that is more meaningful and easier to analyze. After analyzing that simple image we can extract meanings from them.

Algorithm for this application that extract potential region of interest from image is given below. In this algorithm we mainly use color filtration and blob manipulation algorithms. Color filtration use different algorithms that are separately specified in our color filtrating algorithm. Then by extracting that potential region we analyzing that potential region we analyze it and extract meanings from it like click and drag and cursor move and these are also separately specifies in algorithm given below.

Extraction Of Potential Region Of Interest

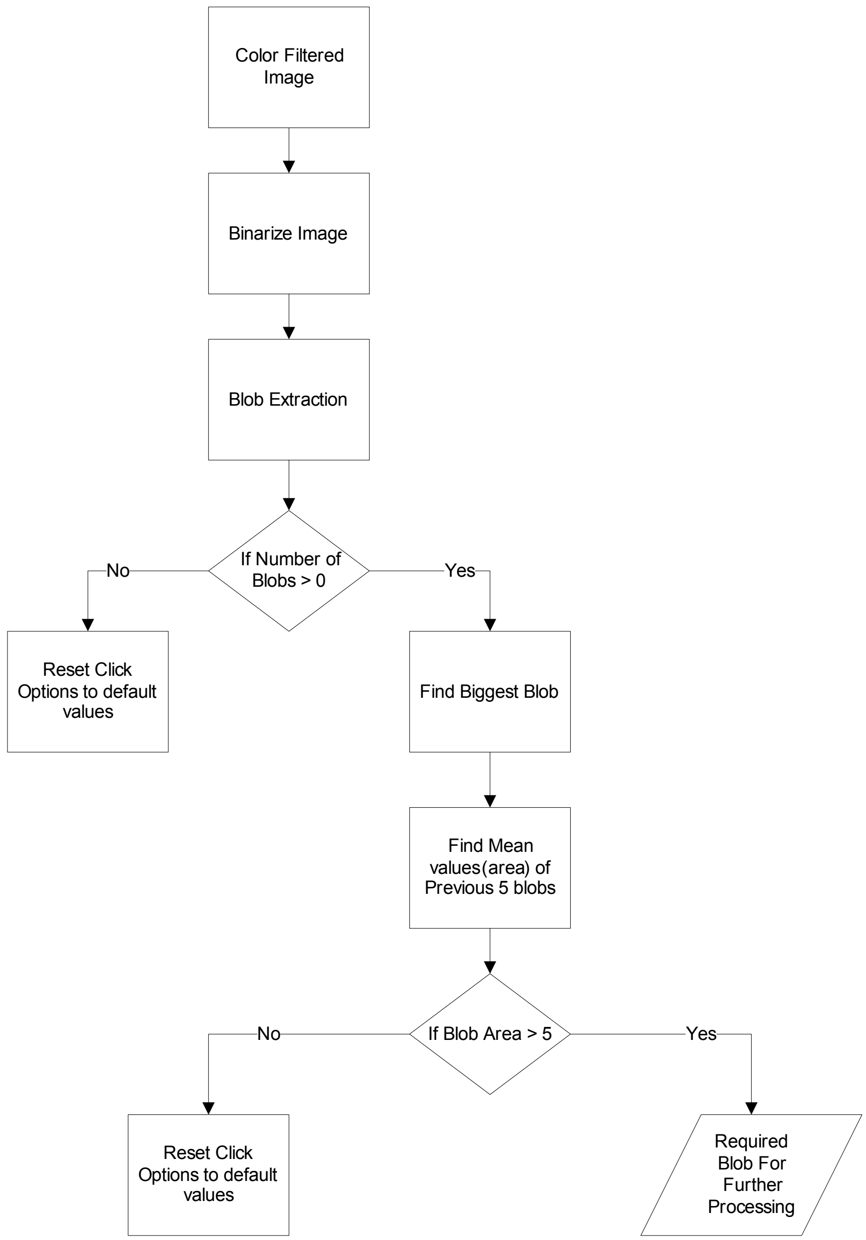

Figure 27: Flow of algorithm of Extraction of Potential Region of Interest

5.1.3.2 Color Filtration

Color filtration use following algorithms

Color filtration Algorithm: We have used color filtration algorithm of AForge library that filter out the marker color from the image. For color filtration we use RGB, Most desktop scanners, digital cameras, and video capture systems save files as RGB (cmyk_color.html, 2010).

Grayscale conversion Algorithm: We have used grayscale conversion algorithm of Aforge so that it will reduce the information of colored image. The red, green and blue color in each pixel is averaged.

Threshold Algorithm: We have used threshold algorithm of Aforge to binarize the image.

Noise filter Algorithm: Erosion filter has been used to remove the noise area from the binaries image. Noise area might be that that occur due to change in light. That noise areas are in small sizes so this filter will remove them.

Blur Function: We used blur function to combine the components of biggest blob. So that there are less variation in the blob size due to light change.

Blob Detection Algorithm: We have used the blob detection algorithm defined in Aforge library. It finds the blobs out of image. Then we sorted out the blobs according to their size and get the biggest blob.

We use these filters according to given order, using these filters in this order give approximate interested results.

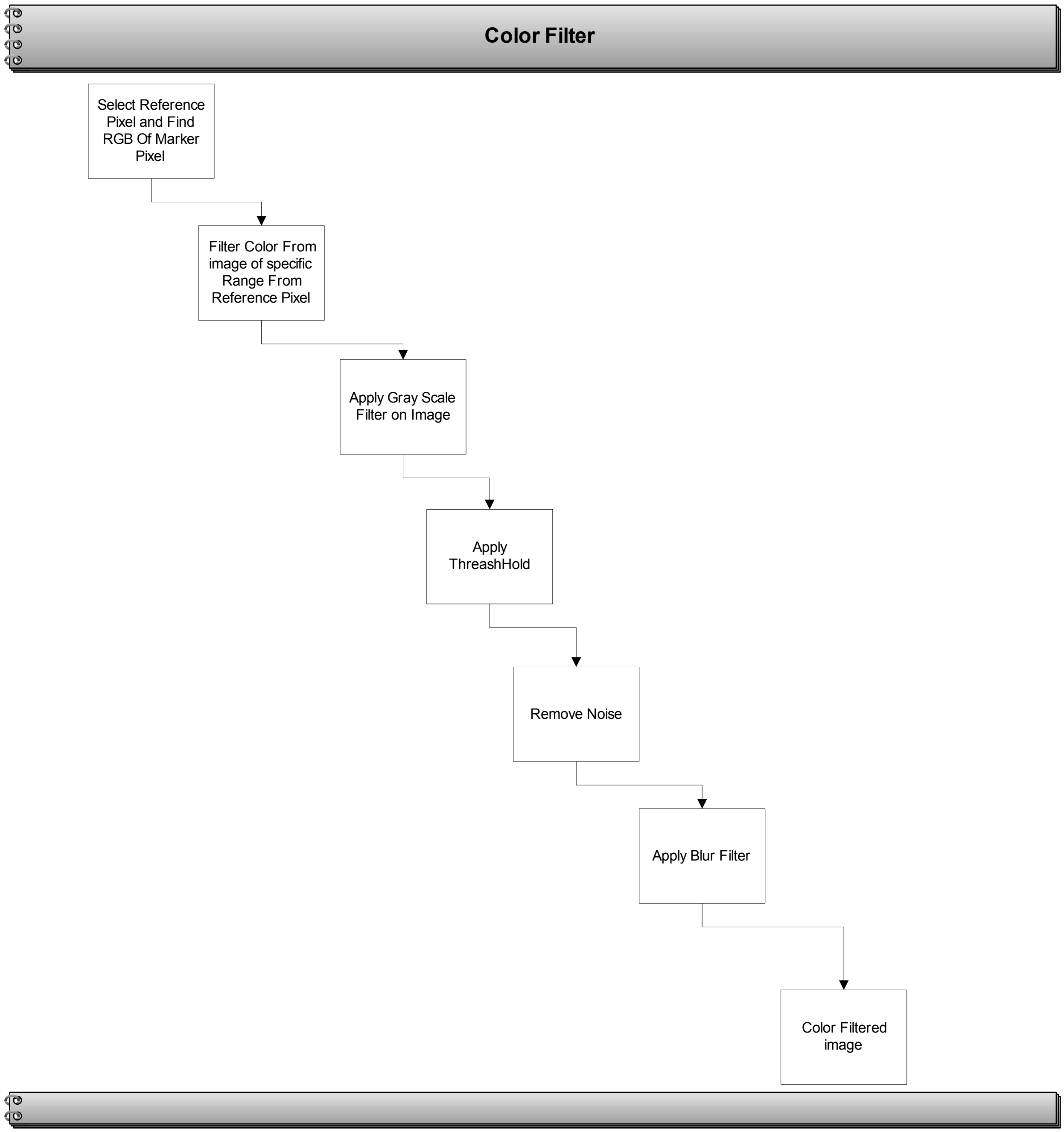

Figure 28: Flow of algorithm of Color Filtration

2. Algorithm for Click (Mouse Down and Up Event), Cursor Move and Dragging Objects

The AForge library can be used to obtain statistics about the blobs, and get the x and bottom value of the biggest blob, that blob is actually potential region of interest that analyze to obtain information for click(mouse down and up event), cursor move and dragging of objects.

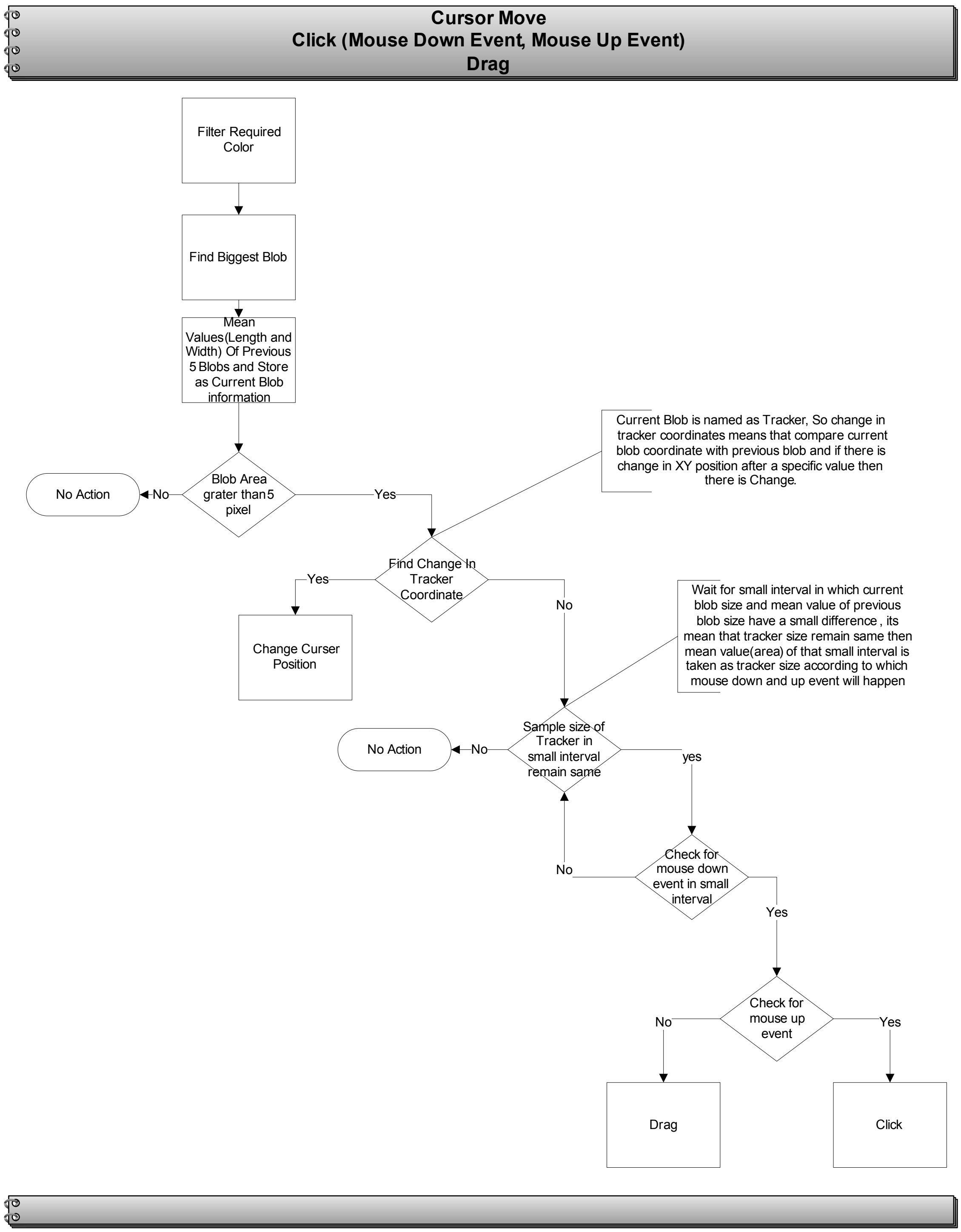

Figure 29: Flow of algorithm for click (mouse down and up event), cursor move and dragging object

4. Use Cases

Use cases of the system are described below. There detail cards are stated in Appendix A. Aggregated use case describes the system as whole whereas all functions have also been described as individual use cases.

4.1. Aggregated Use Case

This aggregated use case shows the functionality of the system as a whole. As the user of the system will run this application main form will appear. This form will be having 6 icons for Select Device, Scan Tracker, Monitor Screen, Presentation Mode, Mouse Sensitivity and ToolTip.

- **Select Device:** User will first click this icon to select the device attached to the system. All the devices will be shown in the drop down list. User has to click one of the options. Then user will start and as he/she clicks the start option the video will be visible in output area. User can also stop or refresh at any time.
- **Scan Tracker:** After the selection of device user will click the icon of scan tracker in order to scan the color of tracker or marker. This scanning can be done by setting up the timer or clicking onto the color. Color filtration is now performed and RGB values of the scanned color will be shown below the tracker area. User can refresh at anytime.
- **Monitor Screen:** When the user done with scanning then he/she will click the monitor screen icon where user can check the area which is in the input device resolution and in this resolution marker will be track. Now the system will take samples of the tracker. This sampling is done for click and drag operations. While staying on this form user can anytime switch to the mouse.

- **Presentation Mode:** Keeping the monitor screen on user can click the presentation mode. This mode is used for writing. User can click the pencil icon to write on the given area. Color of the pencil can be change at any time. User can also erase what is written on the area. The file can be saved as an image and it can be re open at any time while staying on this form. New file can be open at any time to clear the area.

- **Mouse Sensitivity:** User can set the sensitivity of mouse from the main form. The range given for this is 20 to 30.

- **Tool Tip:** This option will allow user to on or off tips for each of the tool. If the user on this tip he/she can see the tips to use each function or icon. It will make this application easy for the beginners.

Aggregated Use Case

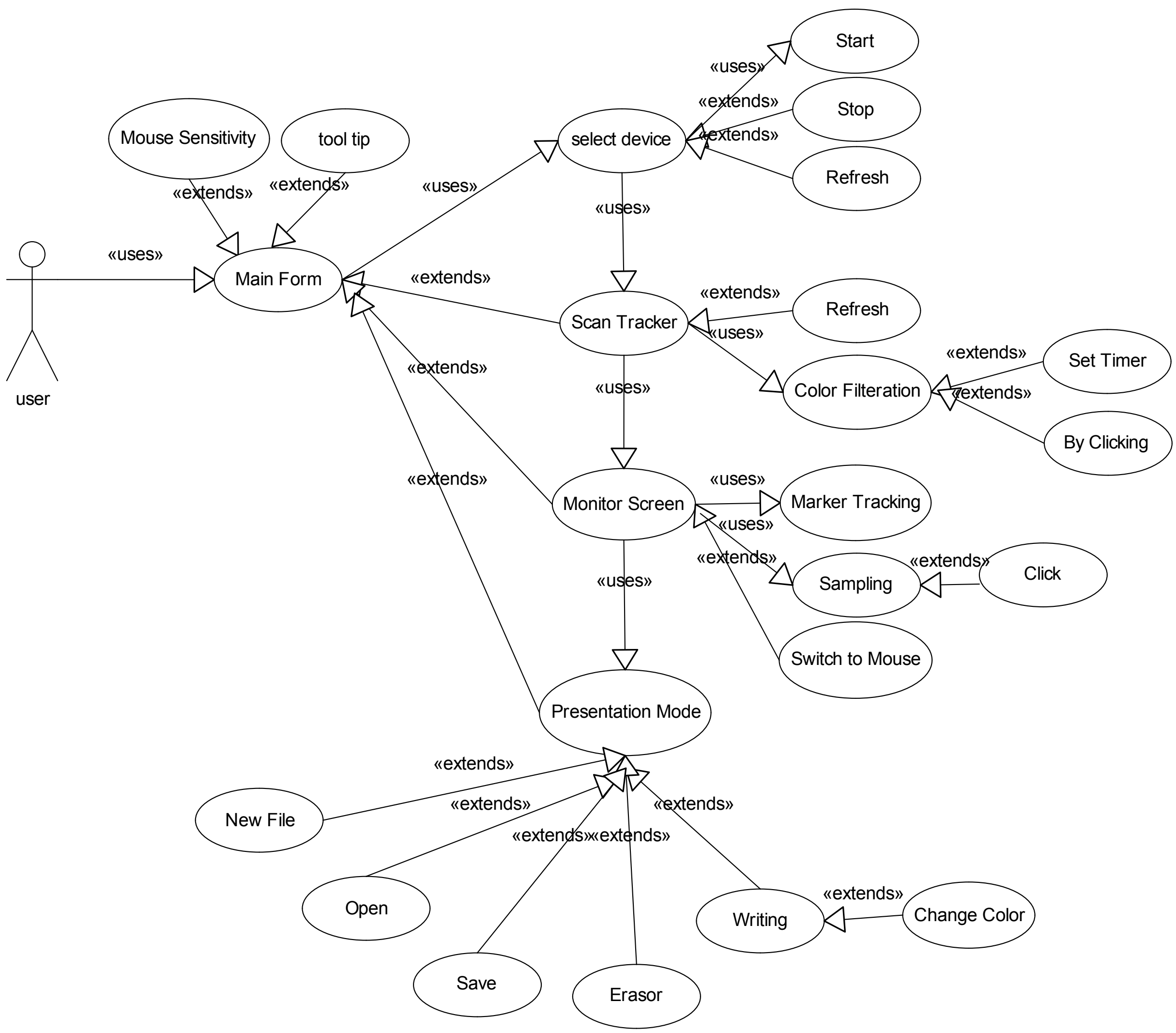

Figure 30: Aggregated Use case of the application

4.2. Main Form

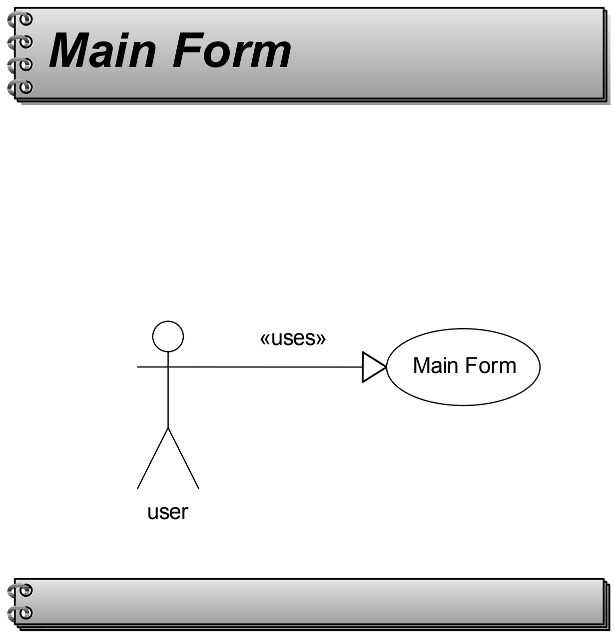

Figure 31: Use case-1 for the Main Form

Use Case Description:

Main form of this application provides access to the other modes of this application.

Use-Case for User

In order to use this application user must open the main form.

(Detailed Description of Use-Cases given in fig A.1)

4.3. Select Device

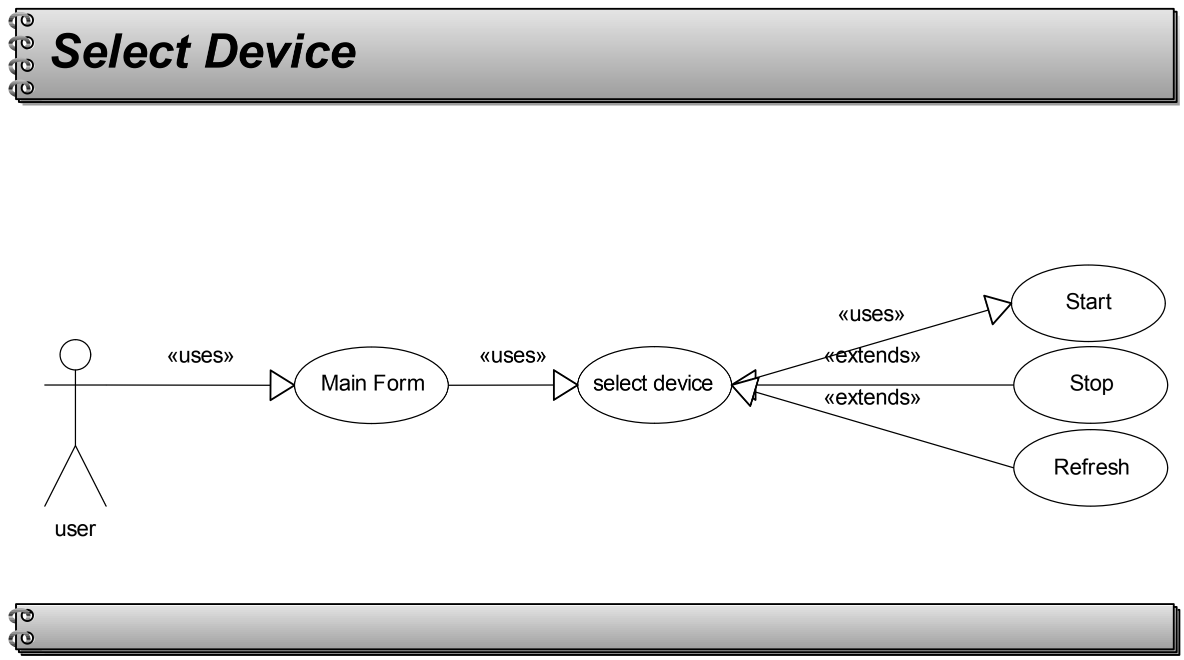

Figure 32: Use case-2 for the Select Device

Use Case Description:

Select device attached to the system.

Use-Case for User

In order start this application user first selects the device such as webcam attached to the system. Then he/she must start after selecting it. User can stop or refresh at any time while being o n this form.

(Detailed Description of Use-Cases given in A.2)

4.4. Scan Tracker

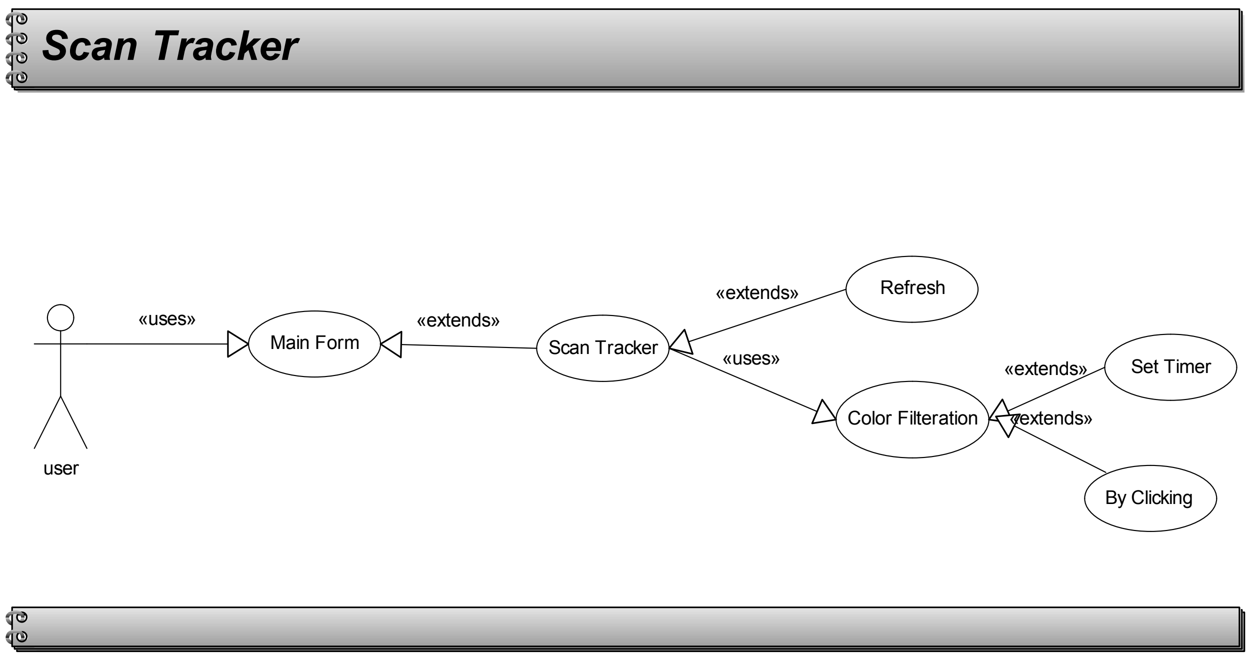

Figure 33: Use case-3 for the Scan Tracker

Use Case Description:

User of the system shall scan tracker which will be used as a marker.

Use-Case for User

After selecting the device user will scan the color of tracker which will be used as the marker. This scanning is done by click through mouse or by setting up the timer. Color filtration is used to scan the color and show its RGB values.

(Detailed Description of Use-Cases given in A.3)

4.5. Monitor Screen

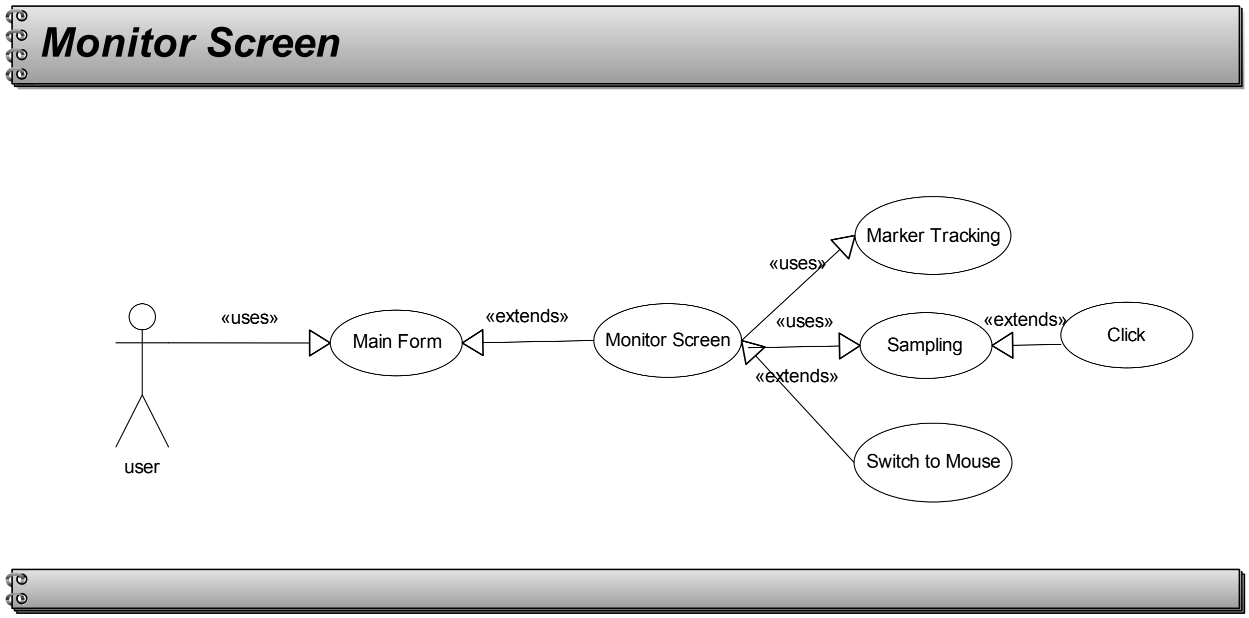

Figure 34: Use case-4 for the Monitor Screen

Use Case Description:

User of the system use monitor screen to work in touch less environment.

Use-Case for User

After scanning the tracker color user will use monitor screen to use marker for certain operations. And user can also switch off the use of marker as mouse.

(Detailed Description of Use-Cases given in A.4)

4.6. Presentation Mode

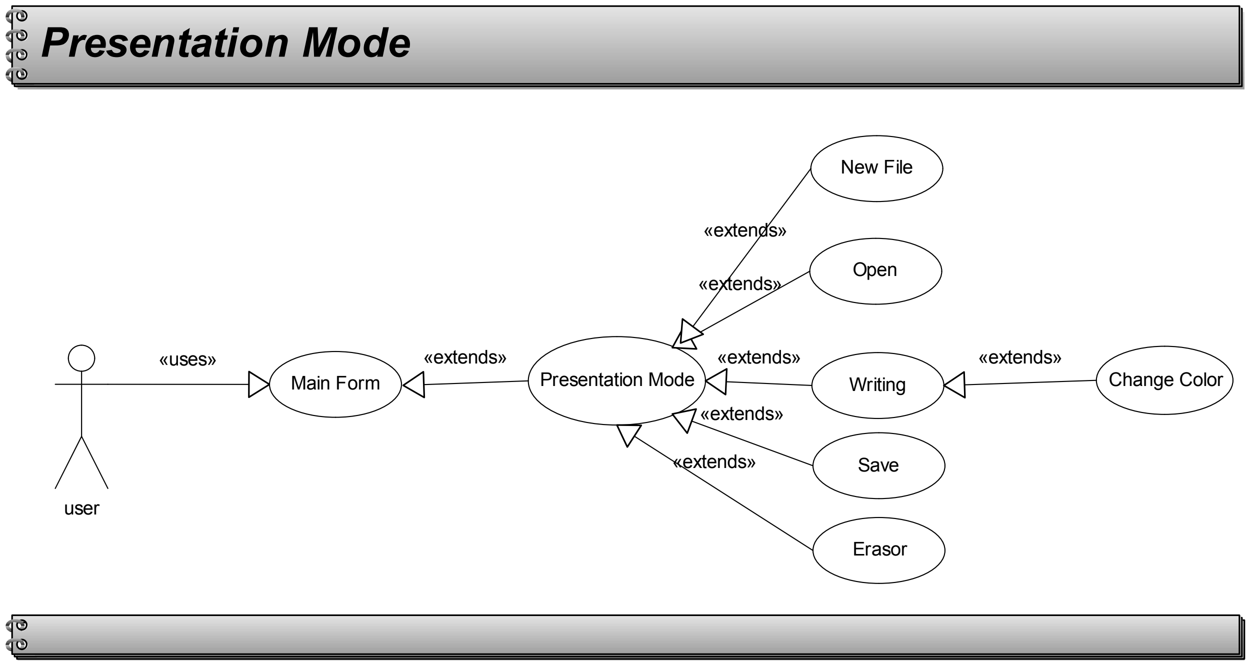

Figure 35: Use case-5 for the Presentation Mode

Use Case Description:

Users will use presentation mode to write, save, erase or can change color for writing and can also save file or reopen them.

Use-Case for User

As the user have selected the monitor screen he/she can use presentation mode where he/she can write through selecting pencil, erase the written stuff, change the color of pencil and save the written presentation into .jpg image format. This mode can be used for presentation as name suggests that.

(Detailed Description of Use-Cases given in A.5)

5. Class Diagram

This logical class diagram shows relationship (composition) between different classes. Different classes compose objects of different classes because those classes need other classes to work without the objects of other classes they can't perform their functionalities.

Device: This class is combination of different data member and member functions. This class is used here for getting information related to the devices connected to the system and their name and version. Functions are use to connect to device, and perform different operations like:

- To Initialize the device
- Stop the webcam and destroy the handle
- Shows the webcam preview in the control
- avicap32.dll to capture webcam images
- user32.dll to copy them to the clipboard

ConnectedDevice: This class contains the object of 'Device' class, this class can't exist without the Device class, because to connect to some device we have to use some operations that are specified in Device class.

PicsData: This class contains information of image taken by the device and other image on which processing is done to get the area of interest, and after processing information about the area of interest is also store in this class.

ColorRange: This class contains the object of 'PicsData' and 'TrackerColor' class and strongly associated with these classes, because reference pixel color will be taken from the original image that is taken by the device and is in 'PicsData' class, that reference color is also stored in

‘TrackerColor’ class. After getting the reference color from original images color filtration with maximum and minimum rage will be done on the clone of that image that store in ‘PicsData’ class named as ‘processedImg’, information taken from that processed image is also stored in ‘PicsData’ class in ‘BiggestBlobInformation’ object.

TrackerColor: This class contain the information of reference color taken from the original image and object of that class is contain by the ‘ColorRange’ class in which within specific range specified in ‘ColorRange’ class that filter required color from image.

ChangeInTrackerArea: This class strongly associated with the ‘ConnectedDevice’, ‘ColorRange, ‘TrackerCurosr’ and ‘Click’ classes. The image taken by the connected device through ‘ConnectedDevice’ class will be filtered by a specific color range by ‘ColorRange’ class and then information obtain from that filteration curser will be move and click will be done with the help of ‘TrackerCursor’ and ‘Click’ classes.

TrackerCursor: This class contains the information of previos and current tracker coordinate. Object of this class is in containing by the ‘ChangeInTrackerArea’ class, this class is responsible for the movement of cursor and removing the noise during the movement of cursor.

Click: This class is also link with the ‘ChangeInTrackerArea’ class, like change in cursor position is due to change in coordinates of tracker blob same click is due to change in size of tracker. This class contains information about the sampling time, mouse down event, mouse up event and click done.

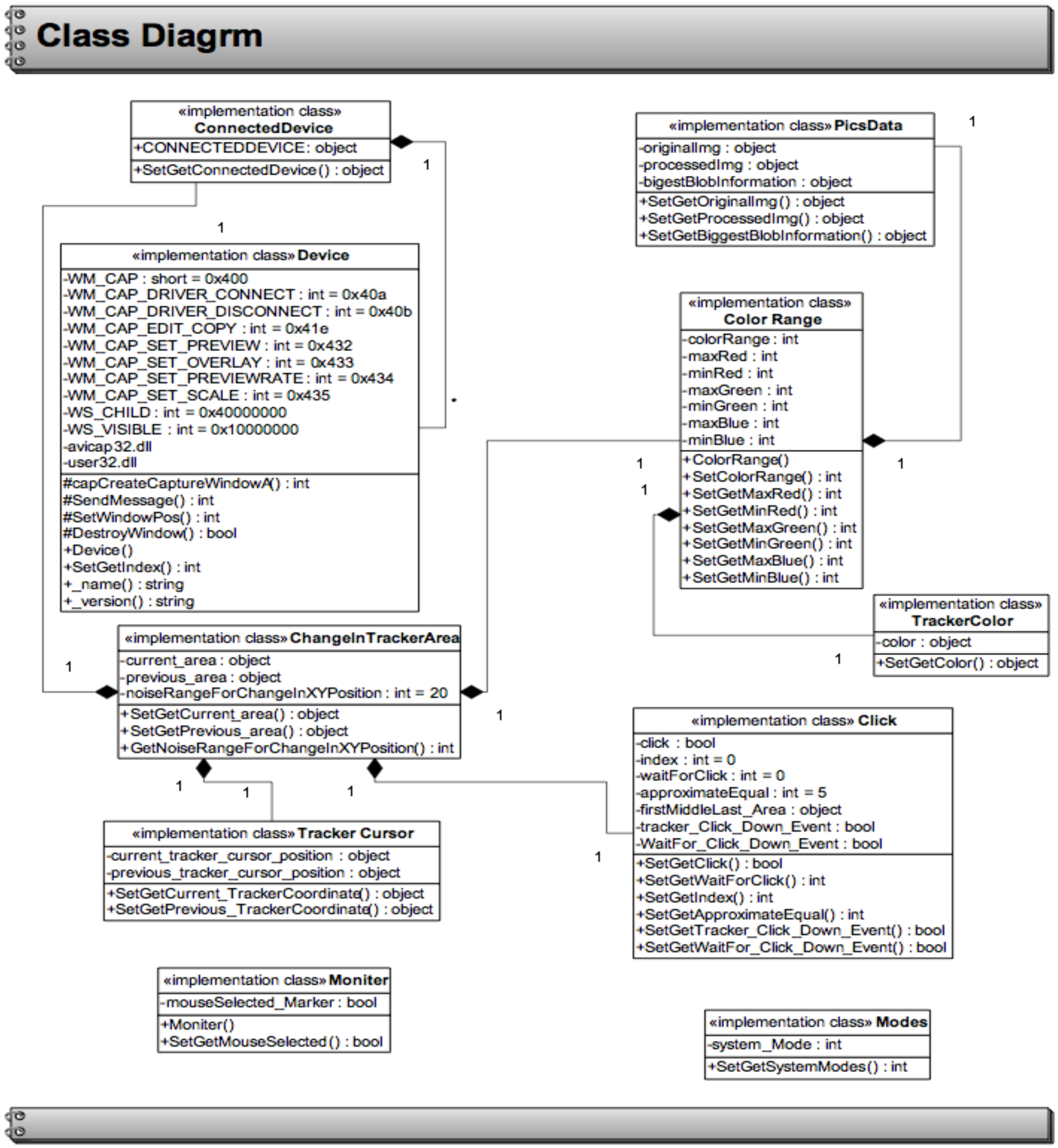

Figure 36: Logical Class Diagram of the system

6. System Architecture

This Diagram shows overall design and structure of system. There are five modules of system and three layers, this architecture describe relation of different modules and sharing of classes between the modules.

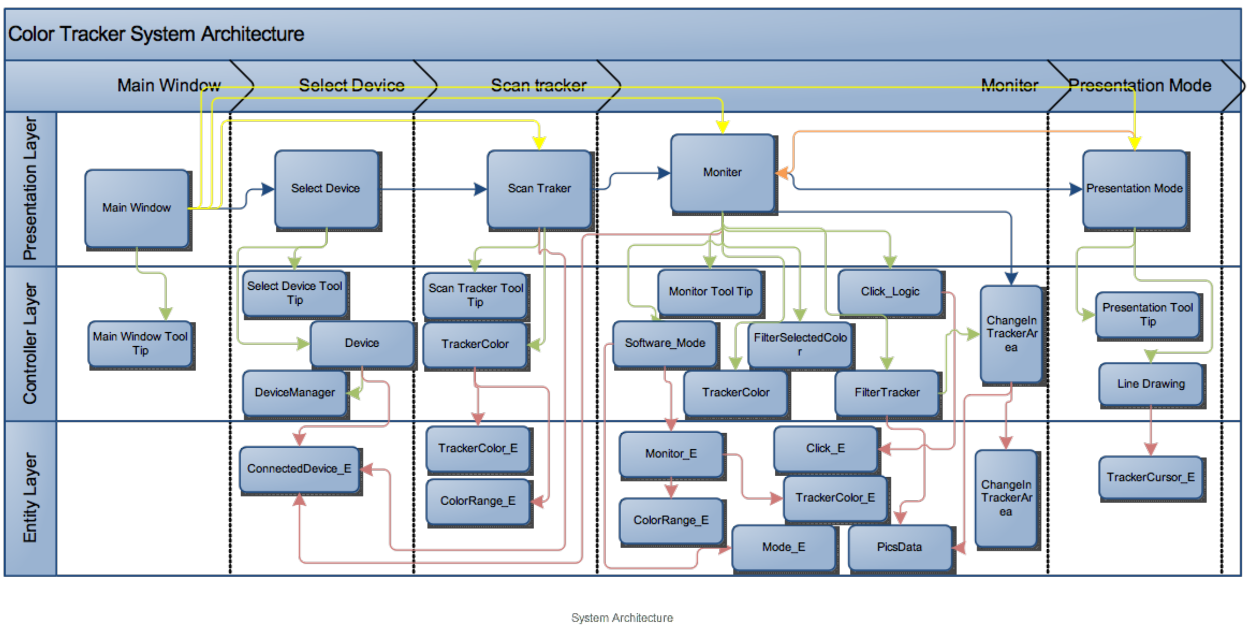

Figure 37 (System Architecture)

Chapter Six

1. Conclusion and Recommendations

In this project we explored the area of color tracking and gesture recognition for applications in human-computer interaction interfaces. This paper presents the system in which we track color marker and from the gesture of that color marker we extract the information. This efficient algorithm for color filtration and then algorithm for extracting meanings from the color marker gesture that is faster than simple hand gesture techniques and in our application there are fewer limitations than simple hand gesture recognition technique.

A number of important basic image processing tasks are analyzed in this thesis. The focus was on the vivid area of the current computer vision research called 'touch less technology' and the related applications. Our system consists of several modules such as color filtration (segmentation), extraction of potential region of interest and in same module we get information from marker gesture.

Our experience has shown us that the portion of the application that needs the most improvement is the image segmentation (color filtration). We attempted to resolve this problem by implementing marker detection in which a specified color is to be filtered. Though an improvement was noted, the results still are not perfect. Additional work could be done to fine-tune a more accurate color filtration algorithm.

This program could also be extended by implementing new gestures and functionality. There are many applications to which this technology could be applied, other than just simple cursor control. Some ideas for this implementation are that, there could be a scenario in a real life like

- In a factory on assembly line the accepted and rejected goods identified on the base of their shape and then tagged them according to their shapes with some color on the base of their accuracy rate then robots are there for filtration of the goods from assembly line and filter them on the base of color tags.
- Robots track color objects and extract instruction from the different gestures of color objects.

If appropriate means are found which could implement these applications, they would certainly be feasible to implement.

2. Future Enhancement

Enhancement in color filtration algorithm: If we change color “filtration range” from static to dynamic, we can improve the accuracy rate of this application by designing an algorithm for color filtration, currently we take first time reference pixel color and then filter the marker with a specified range, this range always remain same when object move from brighter to dark or from dark to brighter area. So every time when object move there are some areas of the object where color of the object changes and it might not be in the specified range.

So we can design an algorithm that changes color range dynamically on the base of previous color range. It could be like: finding the mean value and then check which pixel color is dominating, like in reference pixel R is dominating over RGB, so its mean that required marker is in dominating color, so on base of what thing we decide that our color filtration range? That's the area of enhancement of this application.

3. Condition for success

Our main conditions for success were that the user would be able to quickly learn how to interact with the computer and more importantly that they would enjoy themselves.

Bibliography

(n.d.). Retrieved from research.microsoft.com: http://research.microsoft.com/en-us/um/people/merrie/surface_research.html

(n.d.).

(n.d.). Retrieved from microsoft_surface: http://en.wikipedia.org/wiki/Microsoft_Surface

(n.d.). Retrieved from www.visualplanet.biz: http://www.visualplanet.biz/products/touch/?gclid=COvXo8bepKICFcpR6wod8XNoxA

(n.d.). Retrieved from inventors.about.com: http://inventors.about.com/library/inventors/bltouch.htm

(n.d.). Retrieved from www.answers.com: http://www.answers.com/topic/surface-computer-technology

(n.d.). Retrieved from www.tiresias.org: http://www.tiresias.org/research/guidelines/touch.htm

AForge.NET. (2009, 01 01). *Erosion Class*. Retrieved 01 23, 2010, from AForge.NET: http://www.aforgenet.com/framework/docs/html/90a69d73-0e5a-3e27-cc52-5864f542b53e.htm

Amit, Y. (2002). *2D Object Detection and Recognition: Models, Algorithms, and Networks.* Chicago: Cambridge, Mass. MIT Press, 2002.

Asanterabi Malima, E. Ö. (2006). A FAST ALGORITHM FOR VISION-BASED HAND GESTURE RECOGNITION FOR ROBOT CONTROL. *Faculty of Engineering and Natural Sciences, .*

Beisecker, E. (2008). Shuffleboard Scorekeeper.

Burns, R. A. (1999). *MACHINE VISION LIGHTING TECHNIQUES.* Nashua: RVSI Northeast Robotics.

cmyk_color.html. (2010). Retrieved from dx.sheridan.com: http://dx.sheridan.com/advisor/cmyk_color.html

Davis, R. (n.d.). Sketch-Understanding. *Massachusetts Institute of Technology* .

Felix Woelk, I. S. (n.d.). An Airborne Bayesian Color Tracking System. 67-72.

Green, B. (2002, 01 01). *Canny Edge Detection Tutorial*. Retrieved 01 10, 2010, from drexel.edu: http://www.pages.drexel.edu/~weg22/can_tut.html

MachineVision.co.uk. (2008, 03 01). *What is machine vision?* Retrieved 01 09, 2010, from MachineVision.co.uk: http://www.machinevision.co.uk/what_is_machine_vision.html

Malassiotis, S. a. (2008). Real-time hand posture recognition using range data. *Image Vision Computation* , 1027-1037.

Miners, B. B. (March 2005). Understanding hand gestures using approximate graph matching,. *Systems, Man and Cybernetics* , 239-248.

multitouch. (n.d.). Retrieved from www.answers.com: http://www.answers.com/topic/multitouch

NationMaster.com. (2003, 01 01). *NationMaster - Encyclopedia: Object Recognition*. Retrieved 01 01, 2010, from NationMaster.com: http://www.nationmaster.com/encyclopedia/Object-recognition

Nguyen Dang Binh, E. S. (n.d.). Real-Time Tracking and Gesture Recognition System.

Oka, K. S. (2002). Real-time fingertip tracking and gesture recognition. *Computer Graphics and Application* , 64-71.

Photoflex Lighting School. (2009, 01 01). *Angle of Relection = Angle of Incidence*. Retrieved 01 10, 2010, from Photoflex Lighting School: http://www.photoflexlightingschool.com/

Lighting_Principles/Secondary_Reflectors/Angle_of_Reflection___Angle_of_Incidence/index.html

S.M. Hassan Ahmed, T. C. (2008). Real-time, Static and Dynamic Hand Gesture Recognition for HCI. *Real-time, Static and Dynamic Hand Gesture Recognition for HCI* , 1-17.

Spann, D. M. (2006, 12 04). *Image Segmentation.* Retrieved 01 10, 2010, from Image Segmentation: http://www.eee.bham.ac.uk/spannm/Teaching%20docs/Computer%20Vision%20Course/Image%20Segmentation.ppt

Torres, G. (n.d.). Gesture Recognition Using Motion Detection.

University, M. (n.d.). Exploring Gesture Based Interfaces using Wii Remotes and IR Lights. *Electrical & Computer Engineering Department.*

Wang, R. Y. (n.d.). Real-Time Hand-Tracking with a Color Glove.

Wikipedia. (2009, 12 28). *Computer Vision*. Retrieved 12 31, 2010, from Wikipedia: http://en.wikipedia.org/wiki/Computer_vision

Wilson, A. (2006). Robust Computer Vision-Based Detection of Pinching for One and Two-Handed Gesture Input. 255-258.

Wu, Y. L. (2005). Analyzing and capturing articulated hand motion in image sequences,. *Pattern Analysis and Machine Intelligence* , 1910-1922.

www.funtrivia.com. (n.d.). Retrieved from http://www.funtrivia.com/askft/Question25285.html

Ying-Hao Yu, N. M. (2007). FPGA-Based Real-Time Color Tracking for Robotic Formation Control. *Robot Technology* .

Yoon, P. a. (2006).

Zeltzer, S. a. (1994).

7 Appendix A

Fig A.1

UC-1: Start Application
Primary Actor: User of system(Any)
Brief Description: To getting start with the application in order to use it
Pre-Condition: User must have equipments described in paper to use this application
Post-Conditions: sufficient light conditions
Basic Flow: User will run the application Main form will be open. Get start with it
Extensions (or Alternate Flows): None

Fig A.1 1

Fig A.2

UC-2: Select Device
Primary Actor: User of system(Any)
Brief Description: To select the device attached to system for tracking
Pre-Condition: At least one video device must connect to the system
Post-Conditions: During use of this application the selected device should not use for other application
Basic Flow: User will run the application Main form will be open. Click the select device icon Select device from the system Press the start button The display will start showing the video capturing by your device.
Extensions (or Alternate Flows): User can stop or refresh any time.

Fig A.1 2

Fig A.3

UC-3: Scan Tracker
Primary Actor: User of system(Any)
Brief Description: To scan the color of tracker which will be used as marker
Pre-Condition: There should be sufficient light conditions so that scanning color should be the same as in real time
Post-Conditions: The light intercity on tracker color remain same
Basic Flow: User will run the application Main form will be open. Click the scan tracker after selecting the device Set the timer and scan the color RGB values of the scanned color will be shown after which color filtration can be done
Extensions (or Alternate Flows): User can stop or refresh any time. Color can also be scanned by simple clicking through mouse.

Fig A.1 3

Fig A.4

UC-4: Monitor Screen
Primary Actor: User of system(Any)
Brief Description: To use touch less environment this mode is needed
Pre-Condition: Marker is needed for tracking
Post-Conditions: Stable light is needed for the batter use of this mode and size of tracker should not be small enough that it is hard to distinguish between the noise and the tracker
Basic Flow: User will run the application Main form will be open. Click the monitor screen after scanning the color User can use system in touch less environment
Extensions (or Alternate Flows): User can also switch to the mouse by clicking on that icon.

Fig A.1 4

Fig A.5

UC-5: Presentation Mode
Primary Actor: User of system(Any)
Brief Description: To write something user will use presentation mode.
Pre-Condition: Marker should be selected
Post-Conditions: Stable light is needed for the batter use of this mode and size of tracker should not be small enough that it is hard to distinguish between the noise and the tracker
Basic Flow: User will run the application Main form will be open. Click the presentation mode icon Select the pencil to write New file can be open by clicking the new file icon.
Extensions (or Alternate Flows): User can also change the color of pencil, erase, save or open the saved image file.

Fig A.1 5