FINGER GESTURE RECOGNITION IN DYNAMIC ENVIORMENT UNDER VARYING ILLUMINATION UPON ARBITRARY BACKGROUND

by

ARMIN MUSTAFA

(Y8104007)

DEPARTMENT OF ELECTRICAL ENGINEERING,
INDIAN INSTITUTE OF TECHNOLOGY KANPUR,
KANPUR, U.P, INDIA.
MAY 2010

FINGER GESTURE RECOGNITION IN DYNAMIC ENVIORMENT UNDER VARYING ILLUMINATION UPON ARBITRARY BACKGROUND

A thesis submitted

in the partial fulfillment of the requirements

for the degree of

Master of Technology

by

ARMIN MUSTAFA

(Y8104007)

to the

DEPARTMENT OF ELECTRICAL ENGINEERING,

INDIAN INSTITUTE OF TECHNOLOGY KANPUR,

KANPUR, U.P, INDIA.

MAY 2010

CERTIFICATE

This is to certify that the work contained in the thesis entitled “*Finger Gesture Recognition in Dynamic Environment under Varying Illumination upon Arbitrary Background* ” by *Armin Mustafa*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(K.S.VENKATESH)
Associate Professor,
Department of Electrical Engineering,
Indian Institute of Technology Kanpur

May 2010

Abstract

With the ever increasing role of computerized machines in society, the need for more ergonomic and faster Human Computer Interaction (HCI) systems has become an imperative. HCI determines the effective utilization of the available information flow of the computing, communication, and display technologies. We explore vision based interfaces in particular, and present in some detail our efforts towards developing what may be called ’accessory-free’ or, at any rate ’minimum accessory’ interfaces.

We have developed a robust method to find the fingertip point location in a dynamic changing foreground projection in varying illumination on arbitrary background. The overall performance of the system is fast, accurate, and reliable.

This dissertation basically aims at the development of sufficiently robust algorithms to detect the position of different parts of the hand by a visual band segmentation process carried out under the highly varying illumination conditions resulting from the projector output on an arbitrary background. This is a computationally efficient computer vision system for recognizing hand gestures. The system is intended to replace the mouse interface on a standard personal computer to control application software in a more intuitive manner. The system is implemented in C code with no hardware-acceleration. The main goal is to detect finger gestures without the requirement of any specified gadgets such as finger markers, colored gloves, wrist bands, or touch screens. The long term objective is to facilitate in the future graphical interaction with mobile computing devices equipped with mini projectors instead of conventional display screens. These are expected to be simultaneously communication and computing devices designed for ’anytime, anywhere use’ with no assistive tools whatever. Technologically, this requires the visual or IR band detection of the finger gestures. Our approach deals with exclusively visual detection of the shape of intrusion on the front side projected background and recognition of the trajectory of multiple salient points of the intrusion contour. Gestures can then be defined in terms of derived multi-trajectory parameters such as position, velocity acceleration, curvature, direction, etc.

Acknowledgement

I take this immense opportunity to express my sincerest gratitude to my advisor Prof. K. S. Venkatesh for his ceaseless support, guidance and encouragement through-out my Masters degree program. He has always inspired me towards innovation and his creative ideas and simple solutions have given me insight and helped me to develop a simple approach to any difficult problem. I consider myself very fortunate to have been under his supervision and get a chance to learn and grow.

I would like to thank my friends Akshata, Anima, Aparna, Bishwajit, Swarnali for learning and lending the signs, Jitendra for his creative inputs and intellectual help, Karuna to setup the initial phase of the thesis, Mr. Prem to help me work in Opencv and Bhargav for lending me his camera to perform experiments throughout my thesis and helping me in installations .I finally thank all of them for their support through-out my research and a great company for the past two years.

I am thankful to my parents, brother and sister who have always supported me in my pursuits and provided me emotional help at all points. Their contributions to my life have been immense and its course was greatly infuenced by the examples they have set in front of me. Finally, I would like to thank the Department of Electrical Engineering and ACES at IIT Kanpur for providing excellent education, congenial environment and an advanced research facility.

Dedicated

To

My Father

and

Mother

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction in HCI

Human-computer interaction (HCI) is the study of interaction between people (users) and computers. It is often regarded as the intersection of computer science, behavioural sciences, design and several other fields of study. Interaction between users and computers occurs at the user interface (or simply interface), which includes both software and hardware; for example, characters or objects displayed by software on a personal computer's monitor, input received from users via hardware peripherals such as keyboards and mice, and other user interactions with large-scale computerized systems such as aircraft and power plants.

Recently, a significant amount of effort has been dedicated in the field of HCI for the development of user-friendly interfaces employing voice, vision, gesture, and other innovative I/O channels. Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. In the past decade, studies have been widely pursued, aimed at overcoming the limitations of the conventional HCI tools such as keyboard, mouse, joystick, etc. Evolution of user interfaces shapes the change in the human computer interaction. With the rapid emergence of three dimensional (3D) applications; the need for a new type of interaction device arises. HCI in the large is an interdisciplinary area as depicted by Figure.It is emerging as a speciality concern within several disciplines, each with different emphases: computer science (application design and engineering of human interfaces), psychology (the application of theories of cognitive processes and the empirical analysis of user behaviour), sociology and anthropology (interactions between technology, work, and organization), and industrial design (interactive products).

[htbp]

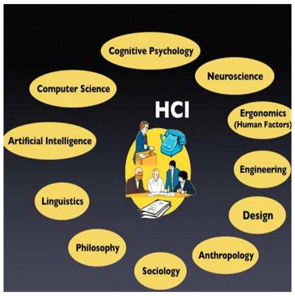


Figure 1.1: Human Computer Interaction

1.2 Evolution in HCI in General

There are certain techniques for hand and finger gesture recognition for Human Computer Interaction. These techniques have been using some gadgets or some sort of assistance tools. For example a visual way to interact with the computer using hand gestures, involved use of an Omni-directional Sensor [1], which makes the system costly and is not very easy to use. Many researchers have studied and used glove-based devices to measure hand location and shape, especially for virtual reality. In general, glove-based or wrist band- based devices [2] measure hand postures and locations with high accuracy and speed, but they aren't suitable for some applications because the cables connected to them restrict the unfettered hand motion. Some have also used hand gesture recognition in which the camera was placed a few meters away [3], but this can't be used for direct interaction with computer system in the more common modes of computer use.

[h]

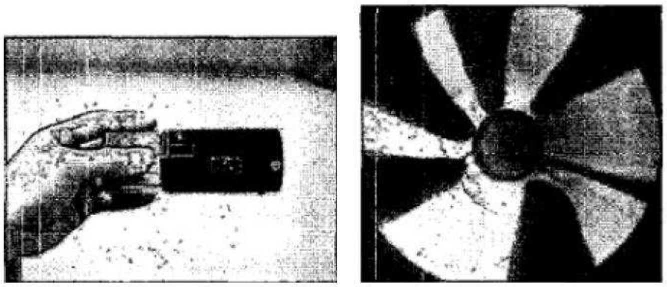


Figure 1.2: Grabbing omnidirectional sensor and capturing finger image simultaneously[1]

Later on, single- and multitouch technologies, essentially touch-based, were used for human computer interaction which used devices like touch screen (e.g., computer display, table, wall) or touchpad, as well as software that recognizes multiple simultaneous touch points. But this again required use of an externally provided Multi Touch Hardware and specific systems interfaced with it. The techniques used mostly were

[h]

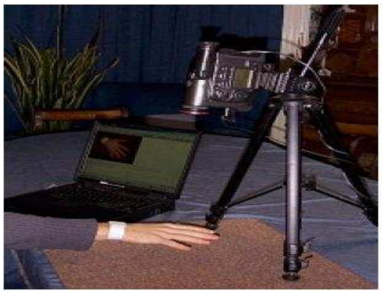


Figure 1.3: Wrist based device to measure hand location[2]

[h]

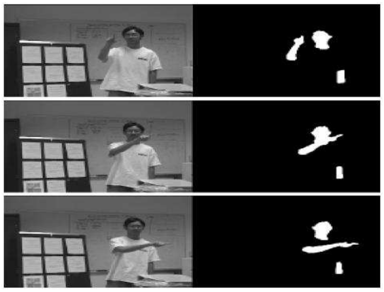


Figure 1.4: Hand gesture recognition with camera placed at few metres[3]

amongst the following: Frustrated Total Internal Reflection (FTIR), Rear Diffused Illumination (Rear DI) such as Microsoft's Surface Table, Laser Light Plan (LLP), LED-Light Plane (LED-LP) and finally Diffused Surface Illumination (DSI)[4-6].

Later certain optical or light sensing (camera) based solutions were used. The scalability, low cost and ease of setup are suggestive reasoning for the popularity of optical solutions. Overhead cameras, Frustrated Total Internal Reflection, Front and Rear Diffused Illumination, Laser Light Plane, and Diffused Surface Illumination are all examples of camera based multi-touch systems. Each of these techniques consists of an optical sensor (typically a camera), infrared light source, and visual feedback in the form of projection or LCD [7, 8].

A few techniques used Infrared Imaging for building an interface. Such techniques employ infrared cameras, infrared light source, IR LED's with few inches of acrylic sheets, baffles, compliant surfaces etc. for proper operation[10]. All these types of Multi touch devices used for HCI require complicated setups and sophisticated devices which make the system much more costly and difficult to manage. Some of the methods use some augmented desktop which involves lots of setup and is difficult to carry [20]. Similarly, infrared cameras were used to segment skin regions from background pixels in order to track two hands for interaction on a 2D

[h]

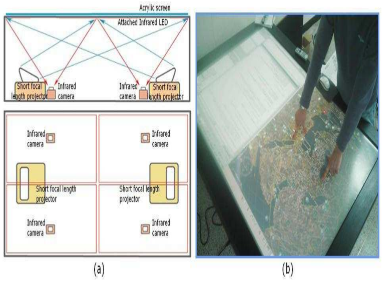


Figure 1.5: (a)The cross-sectional view of the tabletop display system, and the configuration of camera and projector. (b) The system control of our tabletop display using both hands.Implementation of Multi-touch Tabletop Display for HCI[4]

[h]

Figure 1.6: Multi-Touch Sensing through Frustrated Total Internal Reflection[5]

tabletop display. Their method then used a template matching approach in order to recognize a small set of gestures that could be interpreted as interface commands. However, no precise fingertip position information was obtained using their technique [23].

After sometime, techniques using Stereo Vision came into existence but didn't gain much popularity because of certain drawbacks like the fact that the setup requires some complex calibration and the subject needs to adjust according to the needs of the camera, which makes it difficult to use for real-life situations[11]. Some have used simple CRT/LCD displays but the capture was done with two cameras placed at two different accurate angles [9] which were not suitable for day to day applications.

Eventually, the techniques which gained popularity were vision based gesture recognition.They involved certain techniques for hand and finger gesture recognition for Human Computer Interaction. For instance, some researchers have used their tracking techniques for drawing or for 3D graphic object manipulation [12-15]. This has led to research on and adoption of computer vision techniques. One approach uses markers attached to a user's hands or fingertips to facilitate their detection [36]. While markers help in more reliably detecting hands and fingers, they present obstacles to natural interaction similar to glove-based devices. Another approach is to extract image regions corresponding to human skin by either colour segmentation or background image sub-

[h]

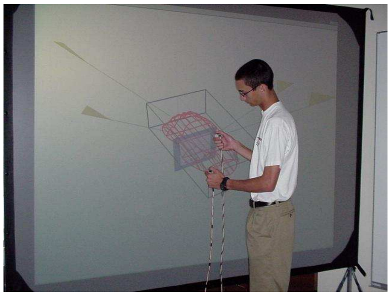


Figure 1.7: Interaction techniques for 3D modeling on large displays[6]

[h]

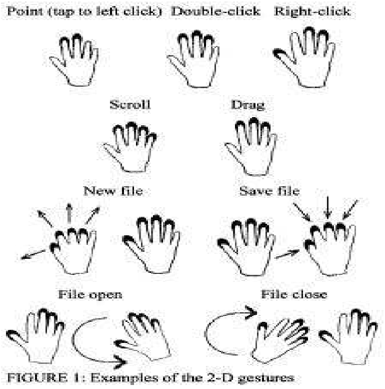


Figure 1.8: FTIR Multi touch detection on a Discrete Distributed Sensor Array[8]

traction or both. Because human skin isn't uniformly colored and changes significantly under different lighting conditions, such methods often produce unreliable segmentation of human skin regions. Methods based on background image subtraction also prove unreliable when applied to images with a complex background.

After a system identifies image regions in input images, it can analyse the regions to estimate hand posture. Researchers have developed several techniques to estimate pointing directions of one or multiple fingertips based on 2D hand or fingertip geometrical features [12, 13]. Another approach used in hand gesture analysis uses a 3D human hand model. To determine the model's posture, this approach matches the model to a hand image obtained by one or more cameras [14, 16-17] Using a 3D human hand model solves the problem of self-occlusion, but these methods don't work well for natural or intuitive interactions because they're too computationally expensive for real-time processing and require controlled environments with a relatively simple background.

A few provide a comprehensive survey of hand tracking methods and gesture analysis algorithms [18, 25]. But these are meant for whole-body gestures which are unsuitable for acting as a direct interface with the computer or any system for a seated subject. One of the original tracking systems to focus on articulated hand

[h]

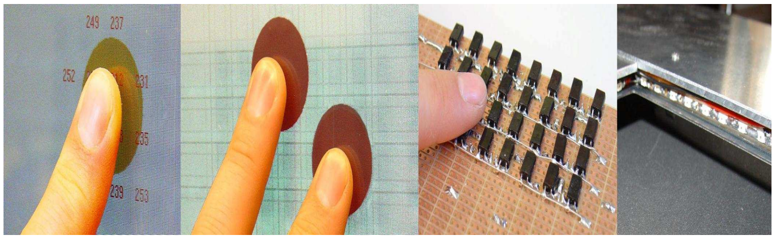


Figure 1.9: New tactile 2-D gesture interface for HCI[7]

[h]

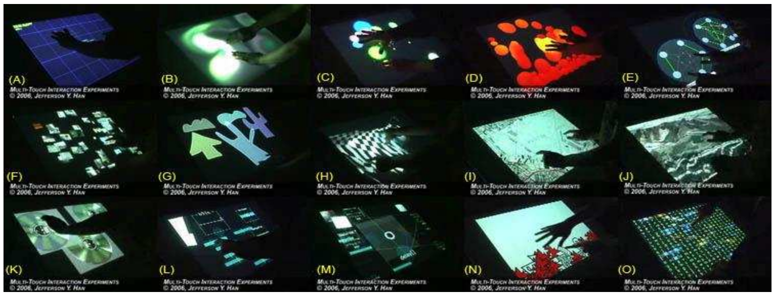


Figure 1.10: A Camera Based Multi- touch Interface Builder for Designers[10]

motion was presented in [21, 20]. In their system, a 27 degree-of-freedom hand could be tracked at 10Hz by extracting point and line features from greyscale images. However, it has difficulty tracking in the presence of occlusions and complicated backgrounds, and it requires a manual initialization step before tracking can begin. From an interaction perspective, most of the hand tracking work to date has focused on 2D interfaces. In [24], a finger was tracked across a planar region using low-cost web cameras in order to manipulate a traditional graphical interface without a mouse or keyboard. Fingertip detection was accomplished by fitting a conic to rounded features, and local tracking of the tip was performed using Kalman filtering. The procedure in [23] used pose estimation in which just pointing gesture were detected but it gave a lot of errors in terms of direction.

But when it comes to interface with computers new algorithms and methods have been found out which can be used for direct interaction with computer using background subtraction, Kalman filtering, Detection, Tracking and Recognition [26] but all these techniques based on background subtraction used a background projected system. In such types of systems we again need computers or TFT displays to recognize gestures. If we want a direct portable interface which can be carried from one place to another and makes it much more easy to use we need to do front side projection using some sort of projection device. This has in turn led to involvement of sophisticated background subtraction techniques to be used for Human Computer Interaction.

[h]

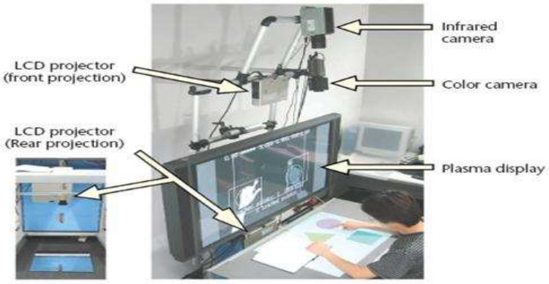

Infrared camera
LCD projector (front projection)
Color camera
LCD projector (Rear projection)
Plasma display

Figure 1.11: Real Time Finger-tip Tracking and Gesture Recognition[20]

1.3 Motivation

Today, the systems we use for computation, interaction, browsing etc. are all becoming, or will soon become, compact in size and more user friendly. In an era where people don't like to carry large gadgets, or complex setups and assistive tools or accessories with them, we need to rework our paradigm. It isn't enough to simply make the devices smaller and better: to remove the drawbacks that the user will eventually again perceive in the existing systems, we need to start with the premise that the user carries at most one device apart from his body. The HCI must retain all possible flexibility, usable anywhere, under all sorts of conditions and must provide effectively both input and output functions with a minimum of hardware. Since any computing-communicating device must have a visual output, we need some sort of display. Keyboards are still the only foolproof means of text input, so they must be accommodated somehow. So also a pointing device. This is likely to remain the case until speech recognition technology improves considerably from the present state of the art.

Another thing to keep in mind is that the interfaces should be low cost, both to make them easy to buy and less painful to lose. Easy availability often outweighs concerns of accuracy with many users. Before we proceed further with the implications of these constraints, we wish to present some of the past and ongoing work on HCI over the last few years. Our own overiding theme has been to make HCI technology 'appropriate' and low cost.

With the extensive proliferation of low cost cameras over the last few years, the automation of many tasks that require visual sensing or intervention has received a big boost. HCI has now come to get its share of benefits from the camera. The other side of a camera is the projector, which can be considered the inverse transducer to the (now) humble camera. Projectors can serve as visual information display devices, as structured light sources, and as a vehicle to virtual input devices. With prices falling in the recent past, they are on the way to becoming humble as well. Recent research in projector-camera systems has overcome many obstacles to deploying and using intelligent displays for a wide range of applications. Significant progress has been made in projected displays that utilize a camera to monitor projected imagery as well as to monitor the surface onto which it is being projected. There is an increase in resolution, brightness and contrast ratio. Blending of projected imagery with underlying surface characteristics has offered unique and profound capabilities. With the wider use of new gadgets like camera-projector equipped cell phones, the business of projector-camera systems is suddenly (or will soon be) in the mass consumption domain.

[h]

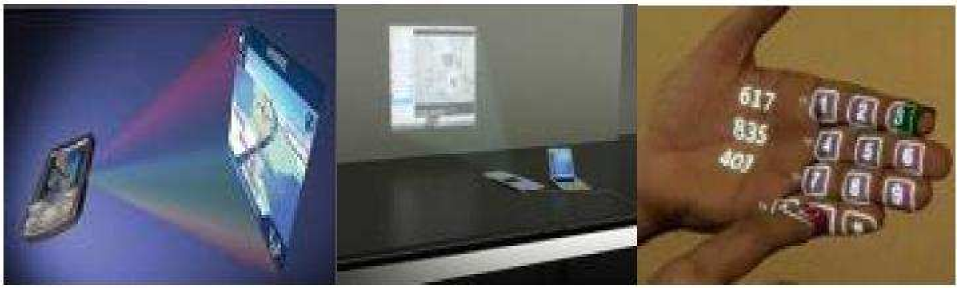


Figure 1.12: New Era systems

[h]

Figure 1.13: E- Garbage

Hence in the future, we envisage a time when personal computers will no longer use a mouse, keyboard or monitor. The monitor picture will be projected by a mini projector (already commercially available) towards any arbitrary, reasonably flat surface. The user will use his/her bare hands and fingers to interact with the objects in the projected dynamically varying image. Thus the fingers will directly serve as mouse, and by projecting a picture of a keyboard (only when required), the user can even ’type’ on the virtual keyboard. Thus, the functions of monitor, keyboard and mouse are all rolled into the single projected image thereby reducing the hardware required and hence the enormous E-Garbage which is increasing at a rapid rate day by day.

1.4 Overview of Dissertation

The present invention aims at the development of sufficiently robust algorithms to detect the position of different parts of the hand by a visual band segmentation process carried out under the highly varying illumination conditions resulting from the projector output on an arbitrary background. The main goal is to detect finger gestures without the requirement of any specified gadgets such as finger markers, colored gloves, wrist bands, or touch screens. The long term objective is to facilitate in the future graphical interaction with mobile computing devices equipped with mini projectors instead of conventional display screens. These are expected to be simultaneously communication and computing devices designed for ’anytime, anywhere use’ with no assistive tools whatever. Technologically, this requires the visual or IR band detection. Our approach deals with exclusively following two steps:

1. Dynamic background Subtraction under varying illumination upon arbitrary background using the Reflectance modeling technique that is visual detection of the shape of intrusion on the front side projected background.

2. Detecting the contour of the hand and fingers and thereby applying some basic algorithms to detect the trajectory of multiple salient points of the intrusion contour. Gestures can then be defined in terms of derived multi-trajectory parameters such as position, velocity acceleration, curvature, direction, etc.

Hence these two steps combine to give us the final output. As a special case of the above applied algorithms is the Paper Touchpad which functions as a virtual mouse for a computer, with requirement of a single webcam. The position of the finger tip detected in the second step of the above defined algorithm is mapped through homography mapping onto the remote display to simulate a cursor.

Chapter 2 deals with the the literature survey in background subtraction,Explanation of basic concepts required to develop the algorithm and Experimental setup.

Chapter 3 describes the algorithm for reliable intrusion detection in dynamic front projection under varying illumination upon arbitrary background using reflectance modelling

Chapter 4 discusses the method for Finger gesture and attribute recognition along with description of the type of gestures.

Chapter 5 deals with application of the above algorithms like paper touch pad and laser pointer mouse.

Chapter 6 explains the Results and Discussions of algorithms and applications explained in chapters before.

Chapter 7 Summarizes and concludes the dissertation with description of future scope.

Chapter 2

Basic Requirements and Literature Survey

2.1 Literature Survey-Background subtraction

Identifying foreground regions in single or multiple images is a necessary preliminary step of several computer vision applications in object tracking, motion capture or 3D modelling for instance. In particular, several 3D modelling applications optimize an initial model obtained using silhouettes extracted as foreground image regions. Traditionally, foreground regions are segmented under the assumption that the background is static and known beforehand in each image. Background subtraction methods usually assume that background pixel values are constant over time while foreground pixel values vary at some time. Based on this fact, several approaches have been proposed which take into account photometric information: greyscale, colour texture or image gradient among others, in a monocular context. For non-uniform backgrounds, statistical models are computed for pixels. In order to obtain robust results, those features that are insensitive to illumination variations should be developed for distinguishing the image changes caused by moving objects from the changes caused by illumination variations, so that background subtraction strategies can still work. Based on the feature types that are used, these methods can be classified into three categories: those using textures, those using colour, and those combining both of them. Several statistical models have been proposed to this purpose, for instance: normal distributions used in conjunction with the Mahalanobis distance [27], or mixture of Gaussian to account for multi-value pixels located on image edges or belonging to shadow regions [28,37]. In addition to these models, and to enforce smoothness constraints over image regions, graph cut methods have been widely used. After the seminal work of Boykov and Jolly[29], many derivatives have been proposed. GrabCut reduces the user interaction required for a good result by iterative optimization [30]. One of the researcher proposed a coarse to fine approach in Lazy Snapping. It provides a user interface for boundary editing [31],while some exploit the shape prior information to reduce segmentation error in the area where both the foreground and background have the similar intensities [32]. The current graph cut based methods shows good results with both static images and videos, but user interaction are often required to achieve good results.

All the aforementioned approaches assume a monocular context and do not consider multi-camera cues when available. An early attempt in that direction was to add stereo information, i.e. depth information obtained using 2 cameras, to the photometric information used for classification into background and foreground [33]. Incorporating depth information makes the process more robust, however it does not account for more than 2 camera consistencies. A method which estimates the silhouette of an object from the unknown background has also been implemented [34]. They exploit the relationship between a region of an image and the visual hull. The approach requires however good colour segmentation, since foreground regions are identified based on the regions. Sormann applied the graph cut method to multiple view segmentation problems [35]. They combine the colour and the shape prior for robust segmentation from a complex background but user interactions are still required. But all these methods consider background to be a static one; hence these approaches can't be applied to the new era systems.

2.2 Projector camera system

Projection systems can be used to implement augmented reality, as well as to create both displays and interfaces on ordinary surfaces. Ordinary surfaces have varying reflectance, color, and geometry. These variations can be accounted for by integrating a camera into the projection system and applying methods from computer vision. Projector-camera systems became popular in these years, and one of the popular purposes of them is 3D measurement. The only difference between camera and projector is the direction of its projection. 3D scene is projected onto the 2D image plane in camera; and 2D pattern in projector is projected onto 3D scene. The mathematical theory of the projective geometry is similar in camera and projector. Then, a straight-forward solution for projector calibrations is using camera calibration methods, which generally requires 3D-2D projection maps. In this dissertation we have used projector camera system to design a new era system.

2.3 Calibration using planar homographies

The planar homography is a non-singular linear relationship between points on planes. The homography between two views plays an important role in the geometry of multiple views. Images of points on a plane in one

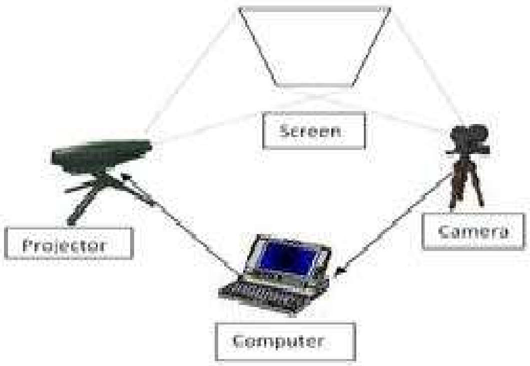


Figure 2.1: Projector camera system

view are related to corresponding image points in another view by a planar homography using a homogeneous representation. This is a projective relation since it only depends on the intersection of planes with lines. The homography transfers points from one view to the other as if they were images of points on the plane. The homography induced by a plane is unique up to a scale factor and is determined by 8 parameters or degrees of freedom. The homography depends on the intrinsic and extrinsic parameters of the cameras used for the two views and the parameters of the 3D plane. Thus, the mapping of points on a two-dimensional planar surface to the imager of our camera is an example of planar homography. The concept is shown below:

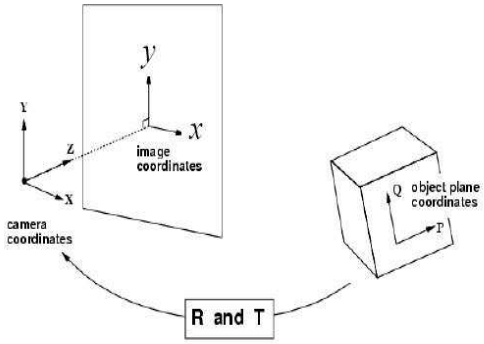


Figure 2.2: Points on planar surface

Planar homography between two views can be determined by finding sufficient constraints to fix the (up to) 8 degrees of freedom of the relation. Homography can be estimated from the matching of 4 points or lines or their combinations in general positions in two views. Each matching pair gives two constraints and fixes two degrees of freedom. It is possible to express this mapping in terms of matrix multiplication if we use homogeneous coordinates to express both the viewed point Q and the point q on the imager to which Q is mapped. If we define:

$$q = [x, y, 1]^T \; Q = [X, Y, Z, 1]^T \tag{2.1}$$

then we can express the action of the homography simply as:

$$q = sHQ \tag{2.2}$$

Here we have introduced the parameter s, which is an arbitrary scale factor (intended to make explicit that the homography is defined only up to that factor). It is conventionally factored out of H.

The algorithm applied to find out this homography matrix is shown in the diagram. There are 3 transformations required for the conversion where Mext, Mproj, Maff, Mint, M are all conversion matrices.

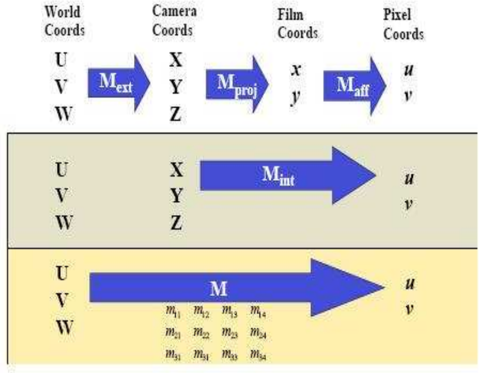
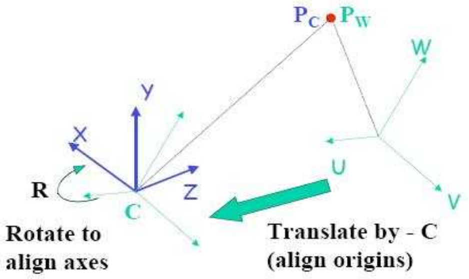


Figure 2.3: Review:Forward Projection

1. The first set of conversion that is the world to camera transformation requires rotation and translation of P_w to P_c
2. Hence the perspective matrix equation for the camera coordinates are as shown in fig.2.5:

$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z} \tag{2.3}$$

3. The second set of conversion is from film coordinates to pixel coordinates which includes two conversion matrices as shown in fig 2.6.



$$
\begin{aligned}
P_C &= R\,(P_W - C) \\
&= R\,P_W + T
\end{aligned}
$$

Figure 2.4: World to camera transformation

$$
\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
$$

$$
p = M_{\text{int}} \cdot P_{\text{c}}
$$

Figure 2.5: Perspective matrix equation(Camera coordinates)

4. Hence the projection of planar points on surface is as shown in fig 2.7 :

The final derivation of the above algorithm is described below with all the required translation, rotation compensation to get the final homography matrix H shown in fig. 2.8.

Seeing the last equation we can conclude that we need 3 sets of points in both the plane to be mapped and the target plane to find the 9 coefficients of Homography matrix that is h11 ,h12 ,h13 ,h21 ,h22 ,h23 ,h31 ,h32 and h33 and apply to this to any arbitrary point to get the final mapped point.

2.3.1 Applications

Here are some computer vision and graphics applications that employ homographies:

1. Mosaics (image processing):Involves computing homographies between pairs of input images and Employs image-image mappings

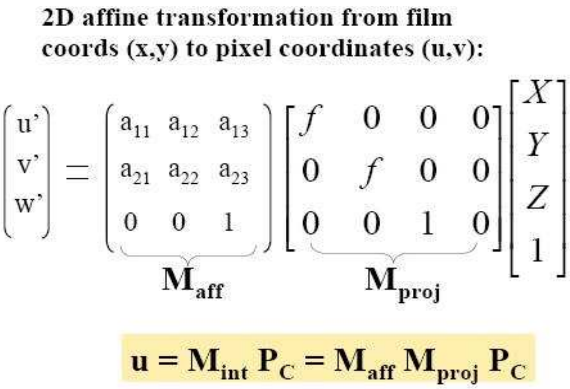

2D affine transformation from film coords (x,y) to pixel coordinates (u,v):
u' v' w'
a11 a12 a13
a21 a22 a23
0 0 1
f 0 0 0
0 f 0 0
0 0 1 0
X Y Z 1
Maff
Mproj
u = Mint PC = Maff Mproj PC

Figure 2.6: Film to pixel coordinates

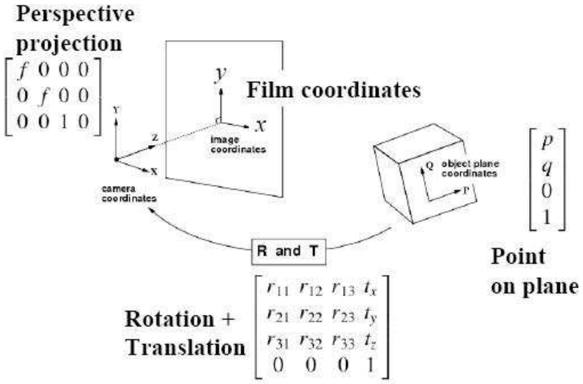


Figure 2.7: Projection of points on planar surface

2. Removing perspective distortion (computer vision):Requires computing homographies between an image and scene surfaces and Employs image-scene mappings

3. Rendering textures (computer graphics):Requires applying homographies between a planar scene surface and the image plane, having the camera as the center of projection and Employs scene-image mappings:computing planar shadows (computer graphics)

2.4 Spectral Response

Also called Spectral Reflectance. Reflectivity is the fraction of incident radiation reflected by a surface. In general it must be treated as a directional property that is a function of the reflected direction, the incident direction, and the incident wavelength

$$\rho(\lambda) = \frac{G_{refl}(\lambda)}{G_{incid}(\lambda)} \tag{2.4}$$

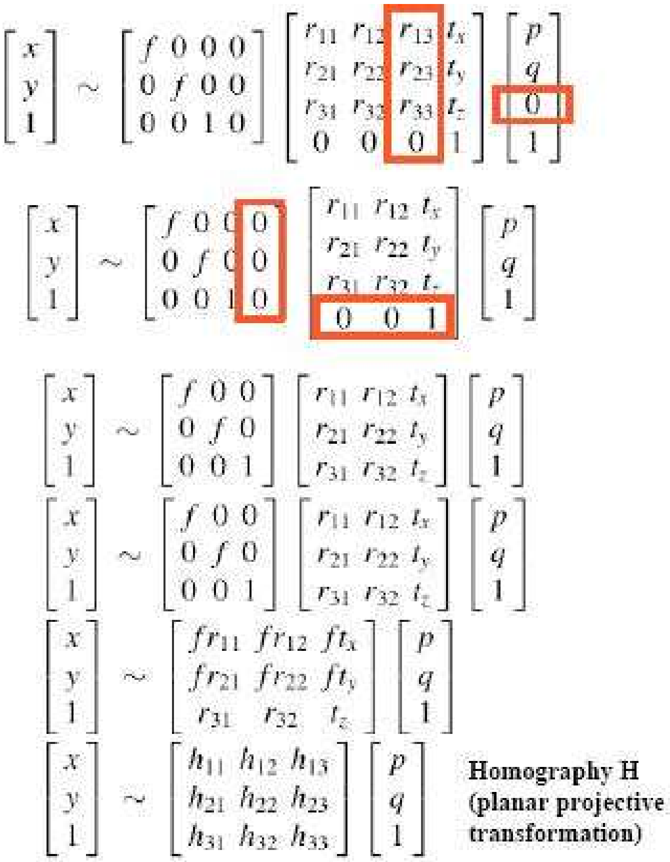


Figure 2.8: Projection of planar points

where, G_{refl} (λ) and G_{incid} (λ) are the reflected and incident spectral (per wavelength) intensity.

Reflectance refers to the fraction of incident electromagnetic power that is reflected at an interface. The reflectance is thus the square of the magnitude of the reflectivity. The reflectivity can be expressed as a complex number as determined by the Fresnel equations for a single layer, whereas the reflectance is always a positive real number.

In certain fields, reflectivity is distinguished from reflectance by the fact that reflectivity is a value that applies to thick reflecting objects. When reflection occurs from thin layers of material, internal reflection effects can cause the reflectance to vary with surface thickness. Reflectivity is the limit value of reflectance as the surface becomes thick; it is the intrinsic reflectance of the surface, hence irrespective of other parameters such as the reflectance of the rear surface.

The reflectance spectrum or spectral reflectance curve is the plot of the reflectance as a function of wavelength.

2.4.1 Surface Type and reflectance

Going back to the fact that reflectivity is a directional property, it should be noted that most surfaces can be divided into those that are specular reflection and those that are diffuse reflection.

For specular surfaces, such as glass or polished metal, reflectivity will be nearly zero at all angles except at the appropriate reflected angle.

For diffuse surfaces, such as matte white paint, reflectivity is uniform; radiation is reflected in all angles equally or near-equally. Such surfaces are said to be Lambertian.

Most real objects have some mixture of diffuse and specular reflective properties.The variable output of a light-sensitive device is based on the color of the light incident upon it. Radiant sensitivity is considered as a function of wavelength i.e. the response of a device or material to monochromatic light is a function of wavelength, also known as spectral response. It is a means of relating the physical nature of change in light to the changes in image and color spaces.

Recent computational models of color vision demonstrate that it is possible to achieve exact color constancy over a limited range of lights and surfaces described by linear models. The success of these computational models hinges on whether any sizable range of surface spectral reflectances can be described by a linear model with about three parameters. A visual system exhibits perfect or exact color constancy if its estimates of object color are unaffected by changes in ambient lighting. Human color vision is approximately color constant across certain ranges of illumination, although the degree of color constancy exhibited changes with the range of lighting examined. When the lighting and surface spectral reflectances in the scene are approximately those of the limited ranges, the color estimates are approximately correct. Analyses of two large sets of empirical surface spectral reflectances indicate that a finite-dimensional linear model with three parameters provides an essentially perfect fit.

Spectral response on the plane which projection takes place to the spectral response on the intruding object differs giving proof of intrusion. We have used the concept of reflectance modeling in our work. The reflectivities of various objects like hand, arbitrary background, the surface etc creates different models which are in turn used for foreground detection under varying illumination. Using these concepts we develop an algorithm which uses reflectance properties to detect the intrusion.

2.5 Experimental Setup

Gestures are captured by the system through the use of a single web camera facing towards the hand of the user. The videos used as input in the dissertation have been shot under the following constraints:

1. *Surface properties*: The area intended for projection should be as flat as possible and non shiny i.e..Lambertian. Its reflectance spectrum should differ sufficiently from that of human skin.
2. *Uniform Illumination*: Ambient light can be nonzero, preferably non-specular and of an intensity that does not dominate the projector illumination. Projector illumination should have an intensity that is significantly higher than the ambient. In instances of regions where both ambient and projector illumination are zero, resulting in very dark regions, intrusion detection, and hence, all the subsequent operations will fail.
3. *Bounded depth*: While capturing videos, the light intensity reflected by the fingers should be nearly constant to avoid abrupt intensity changes due to intrusions occurring too close to the camera/projector. This is ensured by keeping the hand and fingers close to the projection surface at all times. In other words, the depth variation across the projection surface during the gesturing action should be a small fraction of camera/projector distance.
4. The optics of projector and camera are kept as nearly co-axial and coincident as possible to reduce the shadow and parallax effects

In addition, we have the following constraints that are incidental to the implementation.

1. Each finger gesture video should be brief and last for no more than about 3-4 seconds. Longer gesture times will delay the identification of the gesture as identification and appropriate consequent action is only possible after each gesture performed completes.
2. We have used an image size of 640 x 480 pixels because larger sizes, while improving spatial resolution of the gestures, would increase the computational burden, and adversely affect real time performance.
3. The video sequences are stored in AVI format for pre processing.

4. At maximum, 2 fingers were used to make a proper sign. This choice varies from signer to signer and programmer to programmer. More the skin region, more is the complexity of the coding for tracking the motion of the fingers.

The experimental setup is as shown in the figure 2.9:

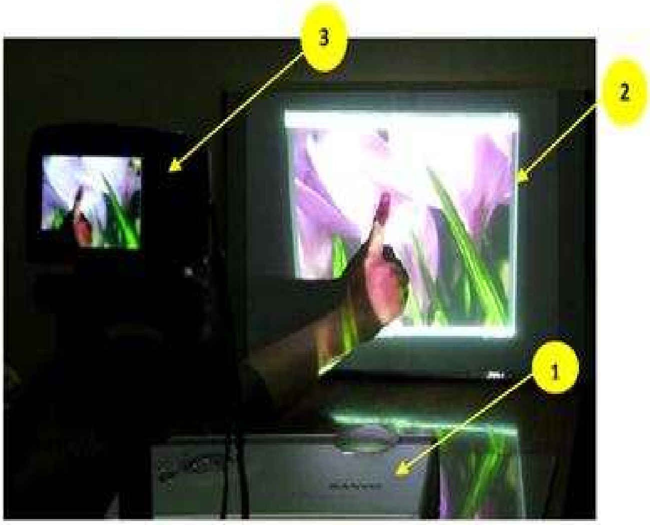


Figure 2.9: Experimental setup of the invention: 1-Projector,2-Screen on which random scenes are being projected and hand is inserted as an intrusion and 3-Camera recording the screen.

Chapter 3

Intrusion Detection in Varying Projector Illumination

3.1 The Preliminary Approach

In the process of arriving at a method that effectively achieved our goals, we first describe an approach that is more preliminary, makes more assumptions about the environment such as that no ambient illumination is present (an unrealistic assumption). Further it reallly does not constitute what may properly be termed reflectance modelling in the rigorous sense, as surface and skin reflectance models are not estimated. Thus the performance of the preliminary .approach we present in this section is markedly inferior under even compliant conditions,and places more restrictions upon the environment. On the other hand, we do choose to present it in some detail because this method was actually first implemented before the more refined approach we finally develop was realized. It also has some pedagogic value, as it directly addresses some of the most important challenges of the problem. Extracting intrusion based on color image segmentation or background subtraction often fails when the scene has a complicated background and dynamic lighting. In the case of intrusion monitoring, simple motion detection may be sufficient, such as based on color modeling. But variations in lighting conditions, constantly changing background and camera hardware settings complicate the intrusion detection problem. It is often necessary to cope with the phenomenon of illumination variations as it can falsely trigger the change detection module that detects intrusions.Further motion detection as a means of intrusion detection may also fail in the scenario we plan to work in, where the background can be dynamic, with moving entities flying across the screen at times. The information in each band of the RGB color space of the video sequences activates our pixel wise change detection algorithm in the observed input frame inspite of a continuosly changing background. This is achieved by recursively updating the background on the basis of projected information and seeking conformance to each reference frame. Ordinary surfaces can have space varying reflectance, color, and geometry. These variations can be accounted for by integrating a camera into the projection system and applying methods from computer vision. The methods currently in use are fundamentally limited since they assume the camera, projector, and scene as static.

Image sequences with dynamic backgrounds therefore often cause false classification of pixels. It is crucial to track moving intruding objects accurately in a cluttered real environment, because illumination changes are unavoidable in the real world. These changes may occur in an outdoor scene when the sun is blocked by clouds or in an indoor environment when a light is turned on or off. Today, projectors are widely used in meeting rooms, which may also affect illuminations greatly due to the changes of slides. In this dissertation we focus on situations when light intensity changes suddenly. In fact, this dissertation is primarily concerned with establishing effective intrusion detection under highly varying and dynamic, but known illumination.

In our proposed system, the videos are sent by the projector to a screen which is further captured by the camera, in the camera output we were able to detect and track moving objects even in the presence of highly varying projector illumination and continuously varying background.Thus, the projector and the camera work in a closed loop.intrusions are detected as disturnbances int he operation of the calibrated closed loop system.

Conventional intrusion detection systems involve a preliminary off-line training phase, separated from the recognition phase.

3.1.1 Training / Learning phase

1. Before we start our learning phase we need to assume that the projector screen surface has complete uniformity.

2. Pure red, green and blue colors are sent via the projector and captured by the camera for a set of *n* frames.

3. The camera output is not pure red, green or blue. Here, every pure input has all its corresponding response RGB components non zero. This is on account of an imperfect color balance match between projector and camera.

4. The mean as well as the variance for each RGB output component for every individual pure input is determined.

5. Single Gaussian surfaces of the output are formed. For our simplicity we take the mean, maximum and minimum values only.

6. Formation of color bias matrix.

3.1.2 Calibration of colors for projector camera system

Since the color values which are projected and the ones which are captured from the camera dont match (shown in fig 3.1) we carry out color calibration.

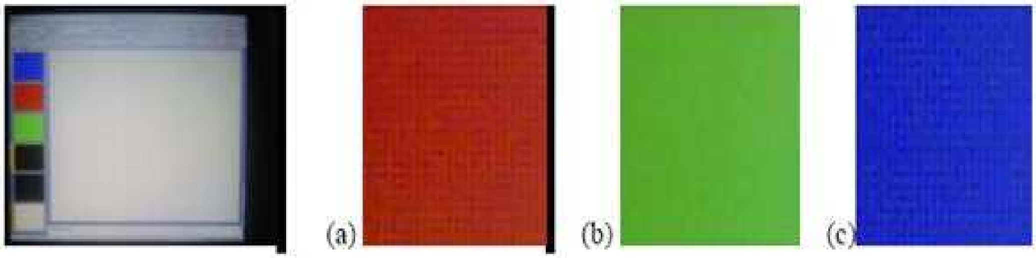


Figure 3.1: Projected output for RGB alongwith camera output for (a)Red (b)Green (c)Blue

Over *n* frames

Considering the red input only:

1. Find the mean red, mean green and mean blue of the output for the n frames.
2. Find the maximum and minimum for each red, green and blue output from the n frames.
3. Find the difference between maximum and the mean value for every RGB output component for the red input which gives the deviation.
4. Follow the same procedure for green as well as blue input for n frames.

The projected RGB values are represented by R^P, G^P and B^P . These values when projected and captured by camera are represented by new values as shown below in matrices respectively for the red, green and blue input where, $R_c^r(t)$,$R_c^g(t)$ and $R_c^b(t)$ are the red, green and blue output as seen from the camera.

Now for detecting the intrusion blob we need to calculate the mean and maximum values for each input RGB component

For red:

$R\mu_r$, $R\mu_g$ and $R\mu_b$ gives mean values for rgb output of red input.

$$R\mu_r = \frac{\sum_{k=0}^{n} R_c^r(t)}{n * framewidth * frameheight} \tag{3.1}$$

$$\underset{R^p}{\begin{pmatrix}255\\0\\0\end{pmatrix}}\underset{G^p}{\begin{pmatrix}0\\255\\0\end{pmatrix}}\underset{B^p}{\begin{pmatrix}0\\0\\255\end{pmatrix}}\begin{pmatrix}R^r{}_c(t)\\R^g{}_c(t)\\R^b{}_c(t)\end{pmatrix}\quad\begin{pmatrix}G^r{}_c(t)\\G^g{}_c(t)\\G^b{}_c(t)\end{pmatrix}\quad\begin{pmatrix}B^r{}_c(t)\\B^g{}_c(t)\\B^b{}_c(t)\end{pmatrix}$$

Figure 3.2: Matrices depicting the various values

$$R\mu_g = \frac{\sum_{k=0}^{n} R_c^g(t)}{n * framewidth * frameheight} \tag{3.2}$$

$$R\mu_b = \frac{\sum_{k=0}^{n} R_c^b(t)}{n * framewidth * frameheight} \tag{3.3}$$

x_r, x_b, x_g gives difference between maximum to mean value for red, green and blue components.

$$x_r = R_{max}^r - R\mu_r \tag{3.4}$$

$$x_g = R_{max}^g - R\mu_g \tag{3.5}$$

$$x_b = R_{max}^b - R\mu_b \tag{3.6}$$

where R_{max}^r ,R_{max}^g,R_{max}^b are the maximum red green and blue components for the red input.

For green:

$G\mu_r$, $G\mu_g$ and $G\mu_b$ gives mean values for rgb output of green input.

$$G\mu_r = \frac{\sum_{k=0}^{n} G_c^r(t)}{n * framewidth * frameheight} \tag{3.7}$$

$$G\mu_g = \frac{\sum_{k=0}^{n} G_c^g(t)}{n * framewidth * frameheight} \tag{3.8}$$

$$G\mu_b = \frac{\sum_{k=0}^{n} G_c^b(t)}{n * framewidth * frameheight} \tag{3.9}$$

y_r, y_b, y_g gives difference between maximum to mean value for red, green and blue components.

$$y_r = G_{max}^r - G\mu_r \tag{3.10}$$

$$y_g = G_{max}^g - G\mu_g \tag{3.11}$$

$$y_b = G_{max}^b - G\mu_b \tag{3.12}$$

For blue:

$B\mu_r$, $B\mu_g$ and $B\mu_b$ gives mean values for rgb output of blue input.

$$B\mu_r = \frac{\sum_{k=0}^{n} B_c^r(t)}{n * framewidth * frameheight} \tag{3.13}$$

$$B\mu_g = \frac{\sum_{k=0}^{n} B_c^g(t)}{n * framewidth * frameheight} \tag{3.14}$$

$$B\mu_b = \frac{\sum_{k=0}^{n} B_c^b(t)}{n * framewidth * frameheight} \tag{3.15}$$

z_r, z_b, z_g gives difference between maximum to mean value for red, green and blue components.

$$z_r = B_{max}^r - B\mu_r \tag{3.16}$$

$$z_g = B_{max}^g - B\mu_g \tag{3.17}$$

$$z_b = B_{max}^b - B\mu_b \tag{3.18}$$

Now we have the mean values and deviation for each component of red, green and blue input and hence we can formulate the color bias matrix.

3.1.3 Color Bias Matrix

This matrix is formed by the mean values and deviations in each of the red, green and blue inputs and outputs. The matrix is as shown in fig 3.3:

This matrix is used to calculate the expected values by performing matrix multiplication with the known input.

$$\left(\begin{matrix} R\mu_r \pm x_r & R\mu_g \pm x_g & R\mu_b \pm x_b \\ G\mu_r \pm y_r & G\mu_g \pm y_g & G\mu_b \pm y_b \\ B\mu_r \pm z_r & B\mu_g \pm z_g & B\mu_b \pm z_b \end{matrix}\right) \begin{matrix} \longleftarrow \text{output for red input} \\ \longleftarrow \text{output for green input} \\ \longleftarrow \text{output for blue input} \end{matrix}$$

Figure 3.3: Color Bias Matrix

3.1.4 Total maximum deviations in RGB

The total deviation for each component is the sum of deviation or variance at each input. To find these values we need to follow the equation given below:

Var(R) =deviation due to red input + deviation due to green input+ deviation due to blue input

$$var(R) = \sigma(R) = x_r + y_r + z_r \tag{3.19}$$

Similarly,

$$var(B) = \sigma(B) = x_b + y_b + z_b \tag{3.20}$$

$$var(G) = \sigma(G) = x_g + y_g + z_g \tag{3.21}$$

Each red green and blue will have their individual Gaussian models that can be represented as shown in the figures 3.4-3.6:

The equation and graph of of the Gaussian model is as depicted in fig 3.7:

According to the Statistical Gaussian models obtained above we can do background subtraction by defining a range of around 2σ around the mean which constitutes the background and the values obtained outside that range is considered to be intrusion(fig 3.8).

Now we take each red, green and blue component of the observed value of each pixel and apply these equations 1 and 2 on it to detect the intrusion where k is a constant which is obtained by trial and error and σ is the variance and expected values are exp_red, exp_green and exp_blue of the respective RGB component.

$$Observedvalue \leq (expectedvalue(k * \sigma))thanitisbackground \tag{3.22}$$

$$Observedvalue \geq (expectedvalue(k * \sigma))thanitisintrusion. \tag{3.23}$$

3.1.5 Matching frames of the projected and captured videos

In the experiment conducted we fixed the no of frames in both captured and projected video and hence calibrated and matched the captured and projected videos.

- Projected video has 100 frames between two black frames
- Captured video has 500 frames between two nearest black frames
- Result:1 black frame of projector was equal to 5 black frame of captured video

3.1.6 Finding expected values

- Form a video with manually inserted black frames after every 100 frames.
- Project the video
- Convert it into number of frames
- Every pixel of every single frame is now decomposed into its RGB components
- These RGB values are then normalized by dividing each by 255
- Now we multiply this normalized RGB with the color bias matrix to get the expected values

For any single pixel p(i,j) of the projected video,let the value of RGB components be given by $[R,G,B]^T$.To calculate the expected value in the absence of intrusion, we need to do matrix multiplication of the pixels RGB values and the color bias matrix. Let the final expected values for the red, green and blue be exp_red, exp_green, exp_blue then the equation becomes as in fig 3.9: This expected value is sent for 5 frames of the camera output frames.

The RGB values of every pixel of the captured frames are now taken and compared with the expected values as given before in eqns 3.22 and 3.23. the values of k can be checked by hit and trial methods. The value which gives best result can be used for thresholding. After the detection of intrusions, the pixels with intrusions are given the values $[255,255,255]^T$ and those where intrusions are not detected are given values$[0,0,0]^T$,resulting in the formation of an intrusion blob in a binary image.

3.1.7 Steps for finding the observed values

- Interpolate and resize the captured video to projected video for pixel matching.
- Convert the captured videos to frames
- Every pixel of every single frame is now decomposed into its RGB components
- Every expected values per frame found earlier is sent to 5 frames of the captured video
- Intrusion detection is done according to the equations 1 and 2.
- Equations are derived which relate the image coordinates in the camera to the external coordinate system.

3.2 The Reflectance Modelling Method

Reflectance modeling represents the more refined approach to the problem of intrusion detection in highly varying and dynamic illuminationin the presence of near-constant non-dominant ambient illumination. We now launch into a discussion of this method in a systematic manner. The main aim of the problem was detection of events that differ from what is considered normal. The normal in this case, is, arguably, the possibly highly dynamic scene projected on the user specified surface by the computer through the mini projector. We aim to detect the intrusion through a novel process of reflectance modeling. The session begins with a few seconds of calibration which includes generating models of the hand, the surface, and the ambient illumination. Subsequently, we proceed to detect the hand in constantly changing background caused by the mixture of relatively unchanging ambient illumination and the highly varying projector illumination under front projection. This kind of detection requires carefully recording the camera output with certain constraints followed by the learning phase and projector-camera co-calibration to match the no of frames per second and number of pixels per frame. This is then executed with the steps explained below:

3.2.1 Calculation of expected RGB values and detecting intrusion at initial stages under controlled projector illumination

This includes the following set of steps:

1. Recording and modeling surface under ambient (ambient lighting is on and projector is off). This defines a model say S_A , which is surface under ambient lighting and is true for any sort of arbitrary surface.

2. Now hand is introduced on the surface illuminated by the ambient lighting and a model for hand is obtained say H_A , which is hand/skin under ambient light. This is done through the following steps: first the region occupied by the hand is segmented by subtraction, and a common Gaussian mixture model for all the sample pixels of the hand available over the space of the foreground and over all the frames of the exposure.

3. Hand is removed from the visibility of camera and the projector is switched on with just white light. This is followed by observing and modeling of the surface in ambient light in addition to the white light of projector, which can be represented by, S_{AP} .

4. Now surface in projector light is found out by differencing S_{AP} and S_A. The equation is as follows:

   $$S_P = \left[S_P^R, S_P^G, S_P^B\right]^T \tag{3.24}$$

   This specifies the green, red and blue component of the surface under projection

   $$S_P^R = S_{AP}^R - S_A^R \tag{3.25}$$

   $$S_P^G = S_{AP}^G - S_A^G \tag{3.26}$$

   $$S_P^B = S_{AP}^B - S_A^B \tag{3.27}$$

5. Hand is introduced inside this scene that is when ambient light is on and projector is displaying white light. This is new model of hand which is H_{AP} captured in ambient light and projector white light.

6. Hence we get the model of the hand in projected white light,H_P which is obtained in the same way as S_P

   $$H_P = \left[H_P^R, H_P^G, H_P^B\right]^T \tag{3.28}$$

   This specifies the green, red and blue component of the surface under projection

   $$H_P^R = H_{AP}^R - H_A^R \tag{3.29}$$

   $$H_P^G = H_{AP}^G - H_A^G \tag{3.30}$$

   $$H_P^B = H_{AP}^B - H_A^B \tag{3.31}$$

   It must be note that while all the other models of the surface under different illumination conditions are functions of position, i.e., pixel-wise models, only H_A and H_P are position invariant

7. Now project the known changing data on the surface under observation by camera. Let us assume the data which is being projected is D[n]. But camera receives the sum of the reflections of the data being projected and the ambient lighting from the surface.

8. Normalization of the models H_P and S_P is done to obtain values which are less than or equal to one by dividing each Red, Green and Blue component by 255, which is the maximum value that each component can reach.

9. Now we find out the expected values of the dynamic background being projected, seen through the camera by performing matrix multiplication of D[n] and S_P followed by addition of S_A .

   $$S_{new} = D[n] * S_P + S_A \tag{3.32}$$

   where,

   $$S_{new} = \left[S_{new}^R, S_{new}^G, S_{new}^B\right]^T \tag{3.33}$$

   The RGB values of every pixel of the captured frames are now taken and compared with the expected values as given before in eqns 3.22. and 3.23 specified above.

3.2.2 Luminance compensation

The method estimates the illumination conditions of the observed image and normalizes the brightness after carrying out background subtraction. This was done by color space transformation.

The RGB color space does not provide sufficient information about the illumination conditions and effect of such conditions on any surface. The color space also has the issue of the luminance and chrominance properties not been separated. A transformation is performed on the RGB values by applying a transform matrix onto the image set. The transformation changes the representation of the image to an YCbCr color space. The Y component threshold was then applied to enhance the segmentation further by using the intensity properties of the image. Threshold segmentation was implemented as the first step to decrease the details in the image set greatly for efficient processing. Hence we calculate luminance at each pixel and then calculate the new value for k the deflection coefficient at each pixel according to the value of luminance. This was done by developing a linear relationship between luminance and k :

$$k_{new1} = (slope * L) + (.82) - (slope * L_{min}) \tag{3.34}$$

where k_{new1} is the factor by which the old value of k must be multiplied.All numerical values used were arrived at by hit and trial method.

$$slope = \frac{0.06}{(L_{max} - L_{min})} \tag{3.35}$$

L-Luminance L_{min} - Minimum Luminance for all the pixels in the frame L_{max} - Maximum luminance for all the pixels in the frame Hence,

$$k_{new1} = k * k_{new1} \tag{3.36}$$

3.2.3 Dominant color compensation

After the luminance compensation another type of adjustment is performed.This compensates for different white balance settings in the camera and the projector and for possible inbuilt white balance adaptation by the camera. The value of k was adjusted according to the dominant color so as to increase the sensitivity according to the color whose value is maximum.

$$k_{new2} = \frac{R + G + B}{(3 * Dom\_color) + 0.9} \tag{3.37}$$

where, Dom_color-Dominant color for that particular pixel

Hence final value of constant k is:

$$k_{final} = \frac{k_{new1}}{k_{new2}} \tag{3.38}$$

3.2.4 Intrusion detection using the skin reflectance model as well as the surface reflectance model in tandem

Skin detection is an important step in hand detection. One of the primary problems in skin detection is color constancy. Ambient light, bright lights, and shadows change the apparent color of an image. Different cameras affect the color values as well. Movement of an object can cause blurring of colors. Finally, skin tones vary dramatically within and across individuals. In the past, different color spaces have been used in skin segmentation.

Although skin colors of different people appear to vary over a wide range, they differ much less in color than in brightness. In other words, skin colors of different people are very close, but they differ mainly in intensities and this variation across individuals and samples can be as much due to illumination variations as due to skin tone differences .

But all these models cannot be applied in a dynamic background; so, we have performed modeling of the skin by matrix multiplication of the normalized RGB values in the model H_P with D[n], the data being projected, followed by addition of H_A .

$$H_{new} = D[n] * H_P + H_A \tag{3.39}$$

where,

$$H_{new} = \left[H_{new}^R, H_{new}^G, H_{new}^B\right]^T \tag{3.40}$$

The net outcome of the above calculation are the values expected in the region of the hand skin pixels during intrusion in the combination of ambient lighting and foreground projection on the hand. Now these values can be used to detect the blobs for the fingers of the hand entering the frames by detecting skin regions manipulated by the models obtained earlier.This completes the discussion of method of reflectance modeling and the algorithm is shown in figure 3.10.

The various models are also depicted using figures 3.11-3.16:

3.2.5 Shadow Removal and other Processing

Shadows are often a problem in background subtraction because they can show up as a foreground object under segmentation by change detection. A shadow may be viewed as a geometrically distorted version of the pattern that together with the pattern produces an overall distorted figure. Shadows can greatly hinder the performance level of pattern detection and classification systems.

There are a number of possible methods for the detection and removal of image shadows. In our method we employ the concept that the point where shadows are cast has the same ratio between the RGB components expected in the absence of intrusion to those observed in its presence. Hence the red, green and blue component ratios are calculated at each point in the area where intrusion is detected and this ratio is used to determine shadow regions where these ratios is consistent across R, G, B.

After removing the shadow, Noise removal algorithm is applied on the image to remove both salt and pepper and Gaussian noise using a 4×4 median filter and gaussian filter respectively.

This is then followed by application of connected component technique by performing foreground cleanup in a raw segmented image.This form of analysis returns the required contour of hand removing the other disturbances and extra contours.

3.2.6 Applications

This algorithm can be applied in numerous ways.Certain conditions may be relaxed to get attractive applications:

- When the front projection is absent ie.. when no dynamic or white light is being projected on to the screen.In this case we can design systems like paper touchpad, virtual keyboard, virtual piano etc. These applications just have arbitrary background.
- Considering a case of back lit projection where dynamic data is being projected at the back allows us to design a system where we can directly interact with the monitor or screen.

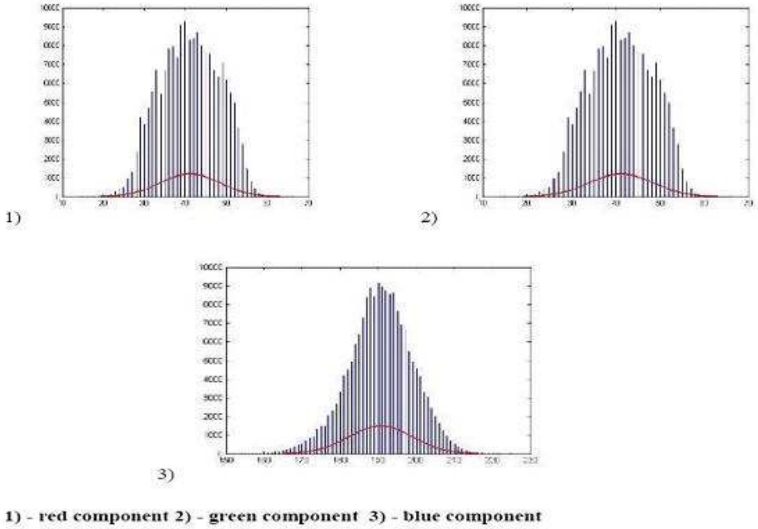


Figure 3.4: Gaussian graphs for blue input

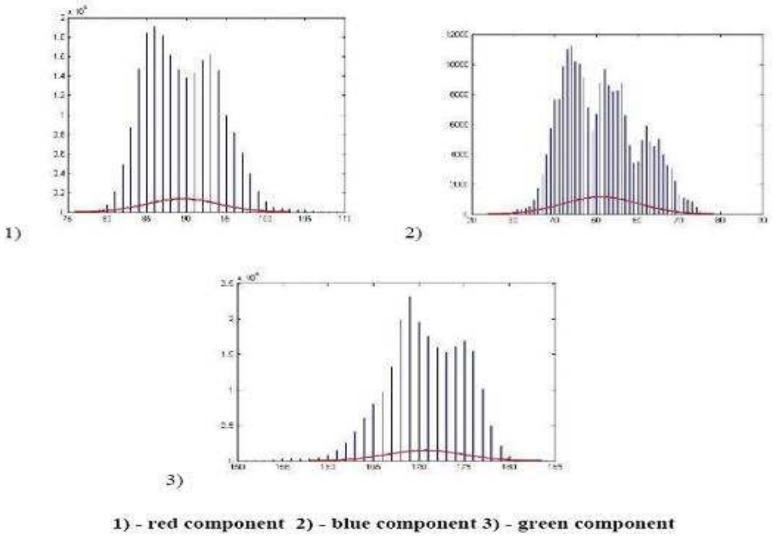


Figure 3.5: Gaussian graphs for green input

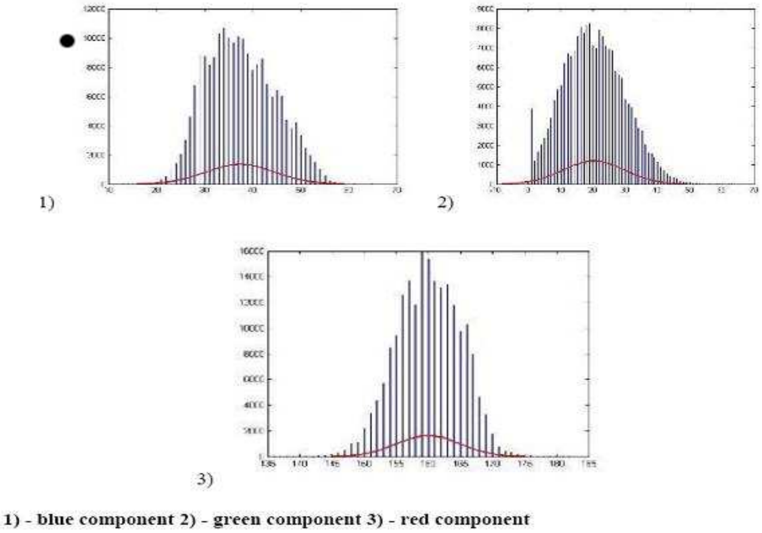


Figure 3.6: Gaussian graphs for red input

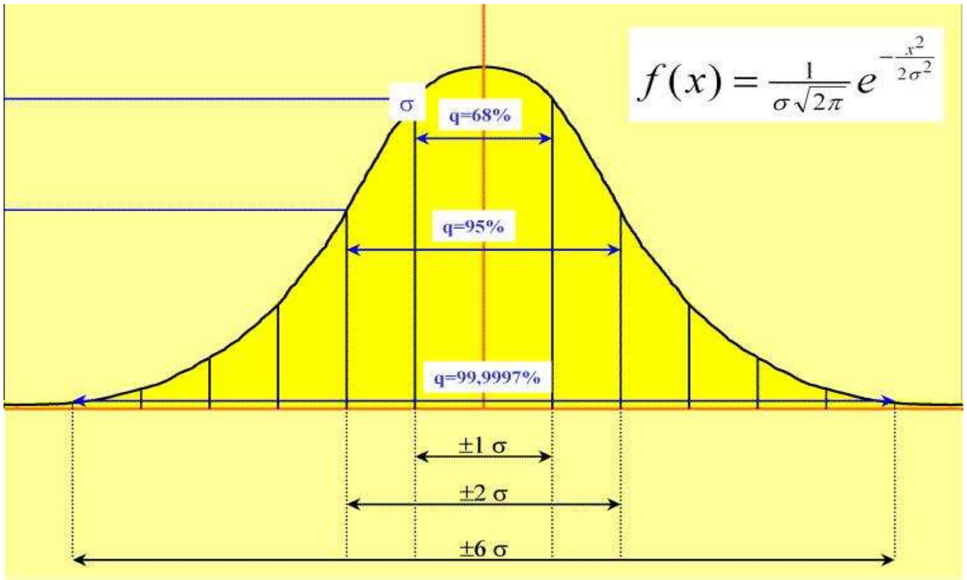


Figure 3.7: Gaussian model

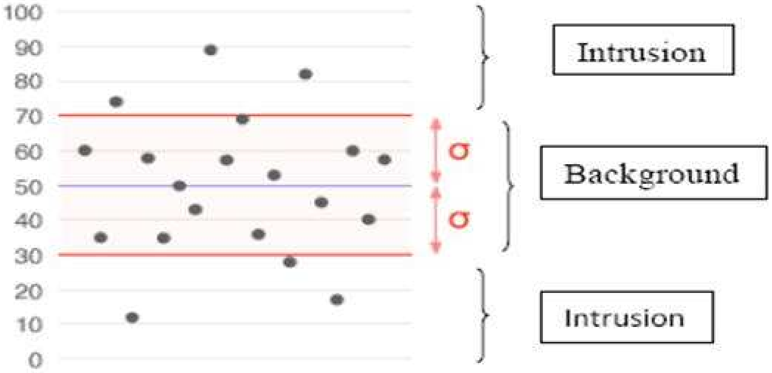


Figure 3.8: Intrusion Plot

$$\begin{pmatrix} R\mu_r & R\mu_g & R\mu_b \\ G\mu_r & G\mu_g & G\mu_b \\ B\mu_r & B\mu_g & B\mu_b \end{pmatrix} * \begin{pmatrix} R \div 255 \\ G \div 255 \\ B \div 255 \end{pmatrix} = \begin{pmatrix} exp\_red \\ exp\_green \\ exp\_blue \end{pmatrix}$$

Figure 3.9: Expected values

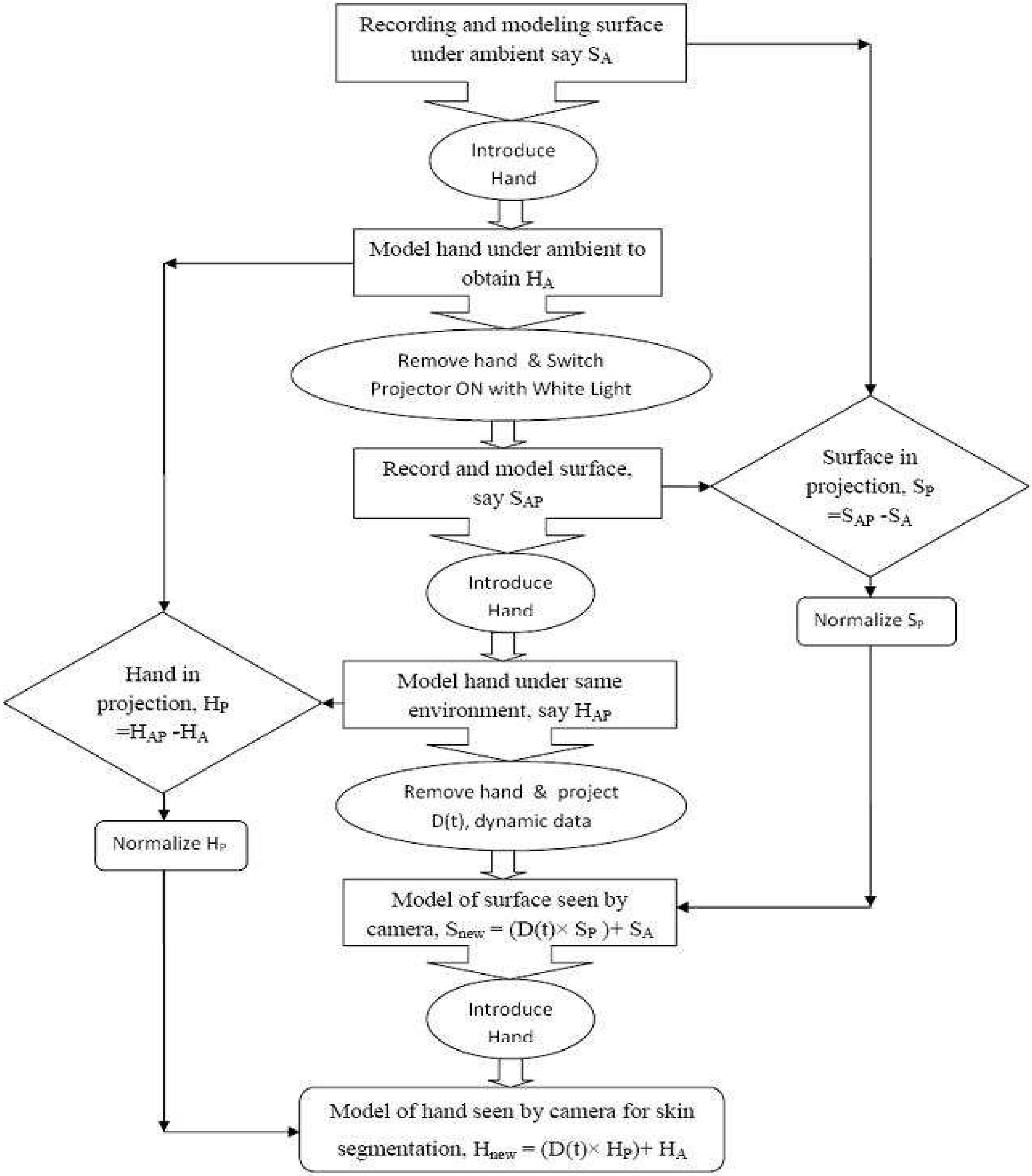

Recording and modeling surface under ambient say S_A
Introduce Hand
Model hand under ambient to obtain H_A
Remove hand & Switch Projector ON with White Light
Record and model surface, say S_{AP}
Surface in projection, S_P $=S_{AP} - S_A$
Normalize S_P
Introduce Hand
Model hand under same environment, say H_{AP}
Hand in projection, H_P $=H_{AP} - H_A$
Normalize H_P
Remove hand & project D(t), dynamic data
Model of surface seen by camera, $S_{new} = (D(t) \times S_P) + S_A$
Introduce Hand
Model of hand seen by camera for skin segmentation, $H_{new} = (D(t) \times H_P) + H_A$

Figure 3.10: Algorithm to detect intrusion using reflectance modelling

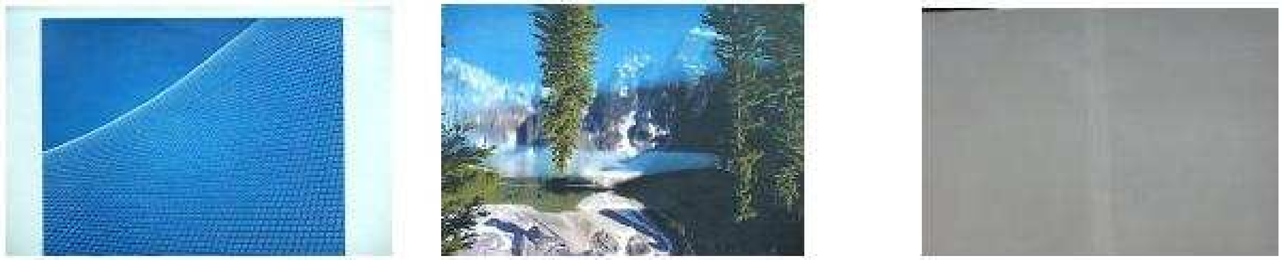


Figure 3.11: Plane and arbitrary surface in ambient

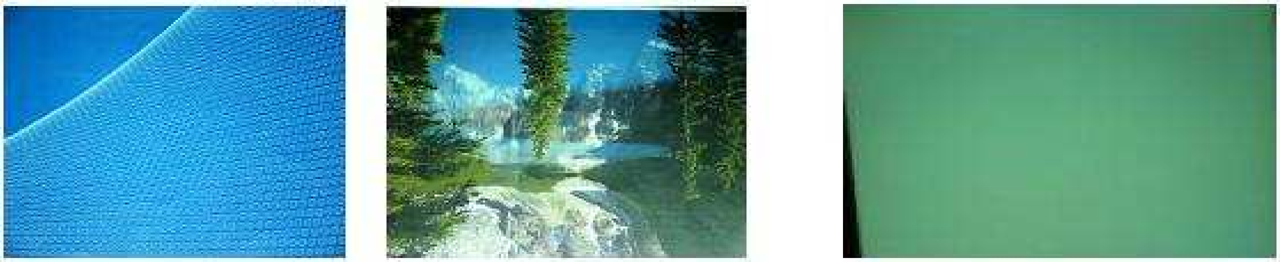


Figure 3.12: Plane and arbitrary surface in ambient and white projected light

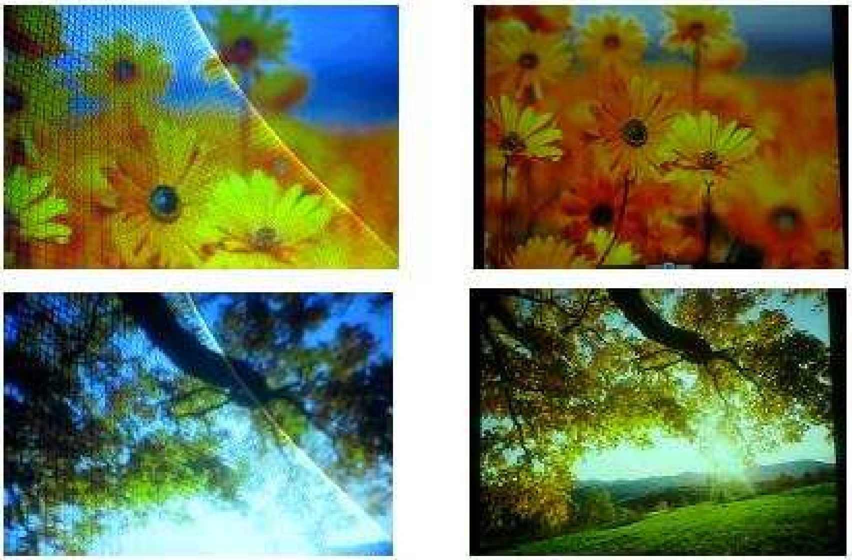


Figure 3.13: Plane and arbitrary surface in ambient and dynamic projection

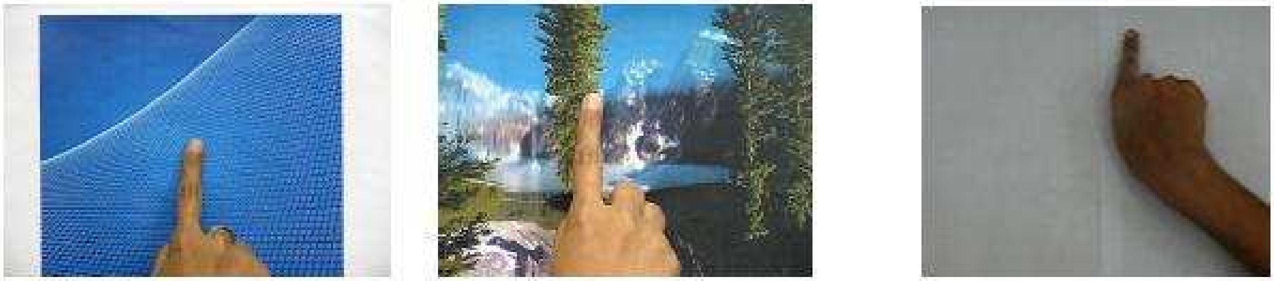


Figure 3.14: Hand on plane and arbitrary surface in ambient

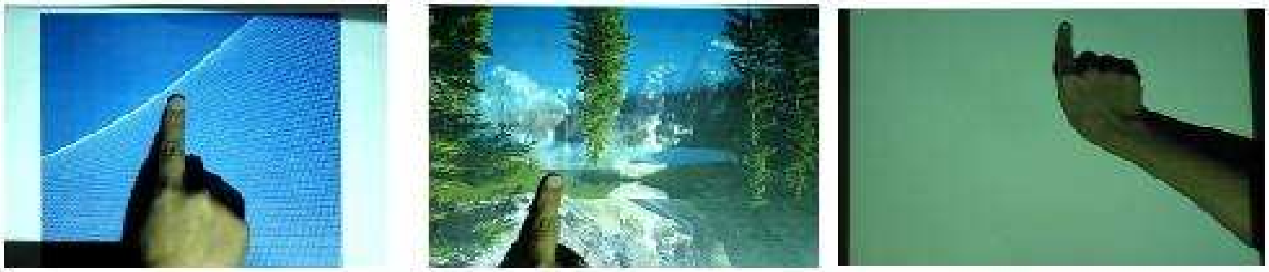


Figure 3.15: Hand on Plane and arbitrary surface in ambient and white projected light

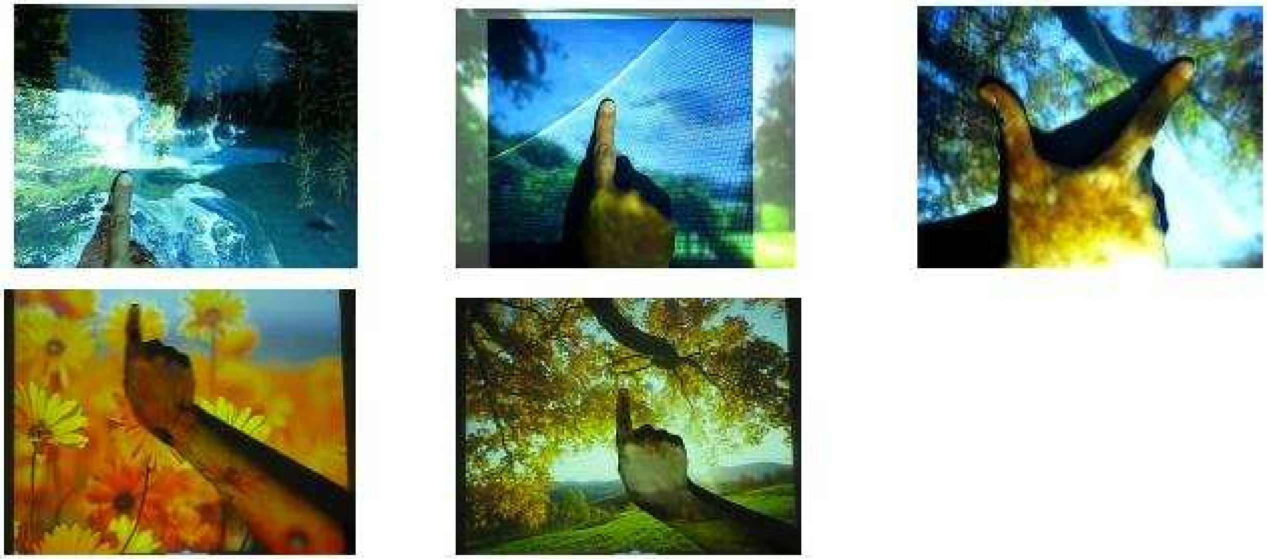


Figure 3.16: Hand on plane and arbitrary surface in ambient and dynamic projection

Chapter 4

Finger Gesture Recognition

After detection of the binary images by techniques outlined in the previous chapters, we need to detect the finger tips and the type and attributes of the gestures. The aim of this project is to propose a video based approach to recognize gestures (one or more fingers). The algorithm includes the following steps and is shown in Figure 4.1.

4.1 Contour Detection

Contour detection in real images is a fundamental problem in many computer vision tasks. Contours are distinguished from edges as follows. Edges are variations in intensity level in a gray level image whereas contours are salient coarse edges that belong to objects and region boundaries in the image. By salient is meant that the contour map drawn by human observers include these edges as they are considered to be salient. In edge detection we extract the contours of the detected skin regions in the binary image obtained after noise removal. The boundaries are extracted by removing the interior pixels. A pixel is set to 0 if all its 4-connnected neighbours are 1, thus leaving only the boundary pixels off. There are six different edge-finding methods:

- The Sobel method finds edges using the Sobel approximation to the derivative. It returns edges at those points where the gradient of I is maximum.
- The Prewitt method finds edges using the Prewitt approximation to the derivative. It returns edges at those points where the gradient of I is maximum.
- The Roberts method finds edges using the Roberts approximation to the derivative. It returns edges at those points where the gradient of I is maximum.

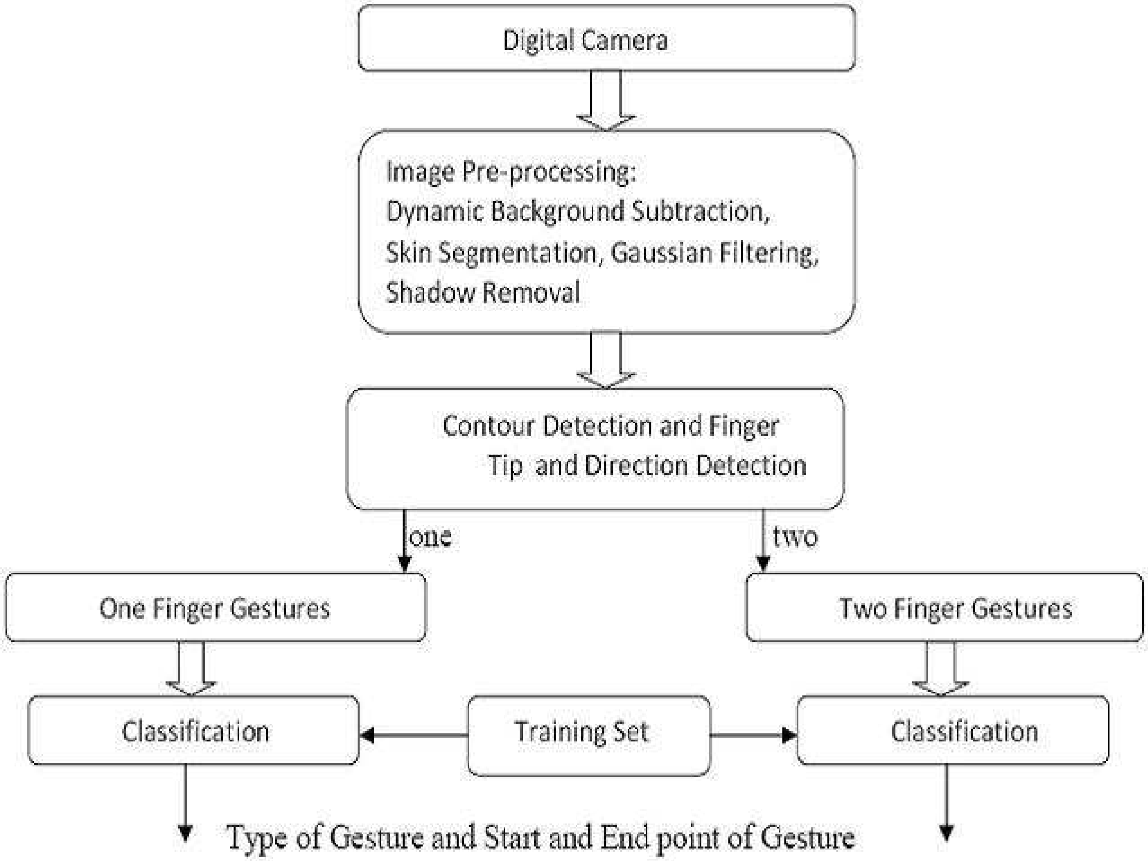

Digital Camera
Image Pre-processing:
Dynamic Background Subtraction,
Skin Segmentation, Gaussian Filtering,
Shadow Removal
Contour Detection and Finger
Tip and Direction Detection
one
two
One Finger Gestures
Two Finger Gestures
Classification
Training Set
Classification
Type of Gesture and Start and End point of Gesture

Figure 4.1: Algorithm to detect finger gesture after intrusion detection

- The Laplacian of Gaussian method finds edges by looking for zero crossings after filtering I with a Laplacian of Gaussian filter.

- The zero-cross method finds edges by looking for zero crossings after filtering I with a filter you specify.

- The Canny method finds edges by looking for local maxima of the gradient of I. The gradient is calculated using the derivative of a Gaussian filter.

Although algorithms like the Canny edge detector can be used to find the edge pixels that separate different segments in an image, they do not tell you anything about those edges as entities in themselves. The next step is to be able to assemble those edge pixels into contours. A contour is a list of points that represent, in one way or another, a curve in an image. This representation can be different depending on the circumstance at hand. There are many ways to represent a curve. Contours can also be represented by sequences in which every entry in the sequence encodes information about the location of the next point on the curve. Contours are sequences of points defining a line/curve in an image.Application of contour detection algorithm in Opencv leads to detection of the boundary of hand and fingers.

4.2 Curvature Mapping

Curvature is the amount by which a geometric object deviates from being flat, or straight in the case of a line, but this is defined in different ways depending on the context. This may be of two types:
(a)Extrinsic curvature and (b) Intrinsic curvature
There is a key distinction between extrinsic curvature, which is defined for objects embedded in another space (usually a Euclidean space) in a way that relates to the radius of curvature of circles that touch the object, and intrinsic curvature, which is defined at each point in a Riemannian manifold.

The primordial example of extrinsic curvature is that of a circle, which has curvature equal to the inverse of its radius everywhere. Smaller circles bend more sharply, and hence have higher curvature.
The curvature of a smooth curve is defined as the curvature of its osculating circle at each point.Curvature may either be negative or positive.The standard surface geometries of constant curvature are elliptic geometry (or spherical geometry) which has positive curvature, Euclidean geometry which has zero curvature, and hyperbolic geometry (pseudosphere geometry) which has negative curvature.The exmaple of positive and negative curvature is shown Fig 4.2:

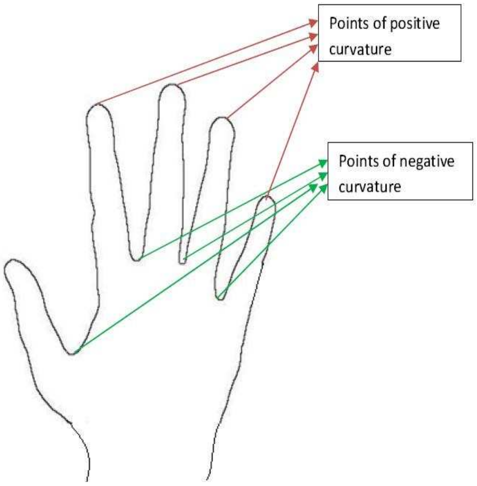


Figure 4.2: Positive and negative curvatures being shown in hand

Applying these concepts we calculate curvature at each point in the contour by applying the usual formula for signed curvature(k) calculation:

$$k=\frac{x^{'}y^{"}-y^{'}x^{"}}{(x'^{2}+y'^{2})^{\frac{3}{2}}} \tag{4.1}$$

where $x^{'}$ and $y^{'}$ gives the first derivative in horizontal and vertical direction. $y^{"}$ and $x^{"}$ are the second derivatives in the horizontal and vertical direction.

4.3 Positive Curvature extrema extraction

Determining the highest positive corner points.This is done by two methods:
One method finds out the maximum positive peaks of the signed curvature calculated in the step above and other method finds the corner points by computing second derivatives which is explained in the paragraph below. In case of more than one positive curvature points of almost equivalent magnitude of curvature, we classify the gesture to be multiple finger.

Along with detection of corner points by curvature calculation which are one of the most important traceable features, we also by find them by computing second derivatives. The most commonly used definition of a corner was provided by Harris. The Harris corner detector is based on the local auto-correlation function of a signal;where the local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions.This definition relies on the matrix of the second-order derivatives of the image intensities. We can think of the second-order derivatives evaluated at each point of an image as forming new second-derivative image. Second derivatives are useful because they do not respond to uniform gradients. The Harris corner definition has the further advantage that, when we consider only the eigenvalues of the autocorrelation matrix, we are considering quantities that are invariant also to rotation, which is important because objects that we are tracking might rotate as well as move. This terminology comes from the Hessian matrix around a point, which in two dimensions, evaluates the autocorrelation at the point for the Harris corner, we consider the autocorrelation matrix of the second derivative images over a small window around each point. Both of these matrices are shown in below:

$$H(p)=\begin{bmatrix}\frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x\partial y}\\ \frac{\partial^2 I}{\partial y\partial x} & \frac{\partial^2 I}{\partial y^2}\end{bmatrix} \tag{4.2}$$

$$M(x,y)=\begin{bmatrix}\sum_{\substack{i\geq -K\\ j\leq K}} w_{i,j}I_x^2(x+i,y+j) & \sum_{\substack{i\geq -K\\ j\leq K}} w_{i,j}I_x(x+i,y+j)I_y(x+i,y+j)\\ \sum_{\substack{i\geq -K\\ j\leq K}} w_{i,j}I_x(x+i,y+j)I_y(x+i,y+j) & \sum_{\substack{i\geq -K\\ j\leq K}} w_{i,j}I_y^2(x+i,y+j)\end{bmatrix} \tag{4.3}$$

(Here H is the Hessian matrix, M is the sutocorrelation matrix and $w_{i,j}$ is a weighting term that can be uniform but is oft en used to create a circular window or Gaussian weighting.) Corners, by Harriss definition, are places in the image where the autocorrelation matrix of the second derivatives has two large eigenvalues. It was later found by Shi and Tomasi that good corners resulted as long as the smaller of the two eigenvalues was greater than a minimum threshold. Shi and Tomasis method was not only sufficient but in many cases gave more satisfactory results than Harris' method. Hence using this definition we have used a function which conveniently computes the second derivatives (using the Sobel operators) that are needed and from those in turn, computes the needed eigenvalues. It then returns a list of the corner points.

The two methods defined above, mainly curvature maximization and corner detection are applied jointly upon each frame, because it was found that corner detection alone produced many false positives.

4.4 Segregating the gesture into single or multiple finger

If two corner points are acquired then we put it in the category of two finger gestures whereas if it is just one then it falls in the category of one finger gestures.

The algorithm easily scales to handle tracking of multiple fingers. The gestures presently include single finger gestures like click, frame, pan and rotate as shown in Fig 4.3 and two finger gestures like drag and zoom as shown in Fig 4.4.

4.5 Frame to frame fingertip tracking using motion model

Finger tips are then tracked through the frames to trace the tip trajectory, finger pointing direction evolution, start and end points of each finger in the gesture performed. This is done by determining the corner or high curvature point in every frame on the retrieved contour and applying motion model to check if the point detected lies in the range defined by calculation of movement in the preceding frames. Tracking motion feedback is used to handle momentary errors.

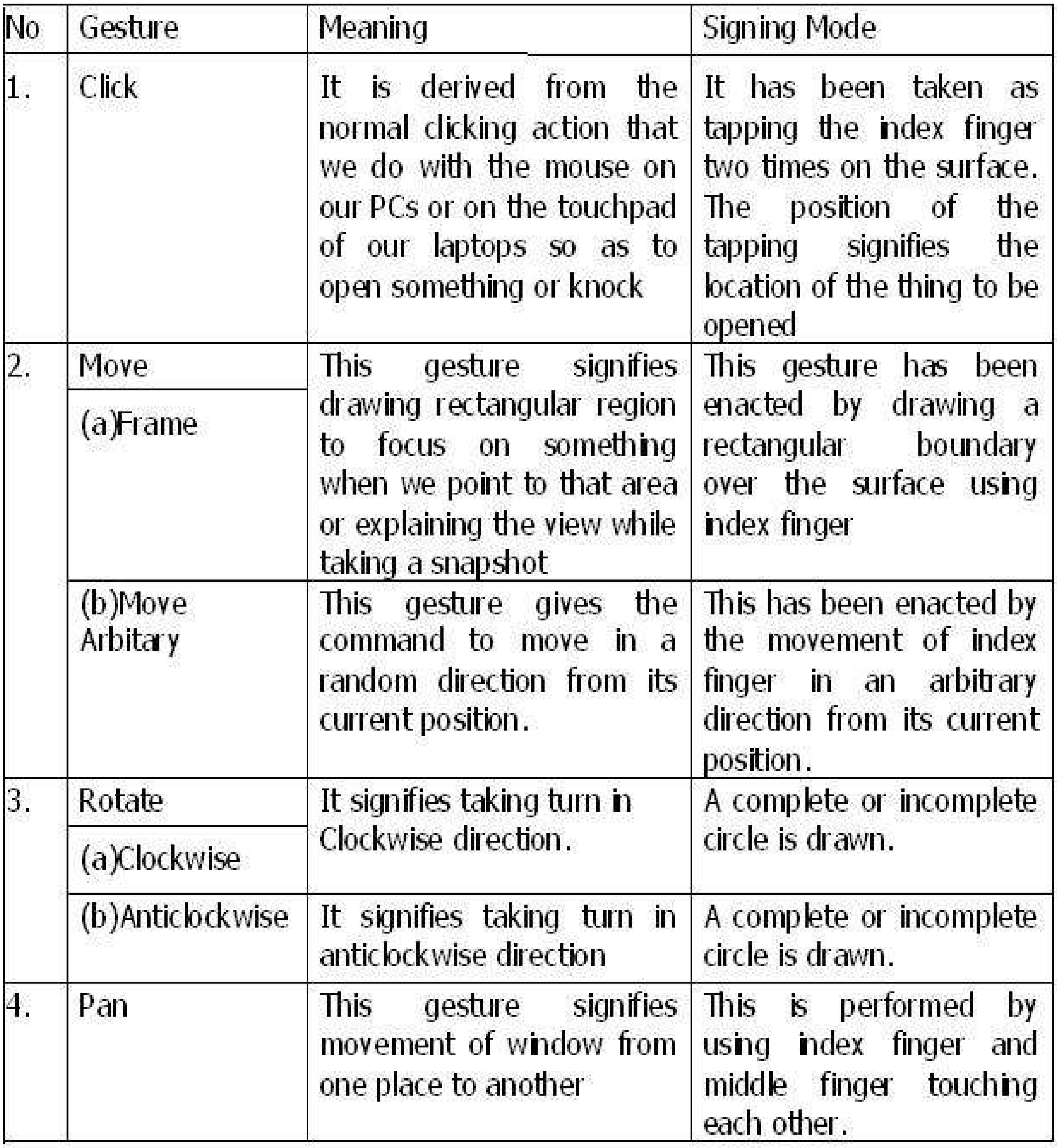
Let at $t = 0$, the position coordinates of the corner or finger tip may be (x_0,y_0)

At $t = 1$, the position coordinates of the corner or finger tip may be (x_1,y_1)

Then at $t = 2$, or in the next frame, the position coordinates of the same finger tip becomes (x_2,y_2)

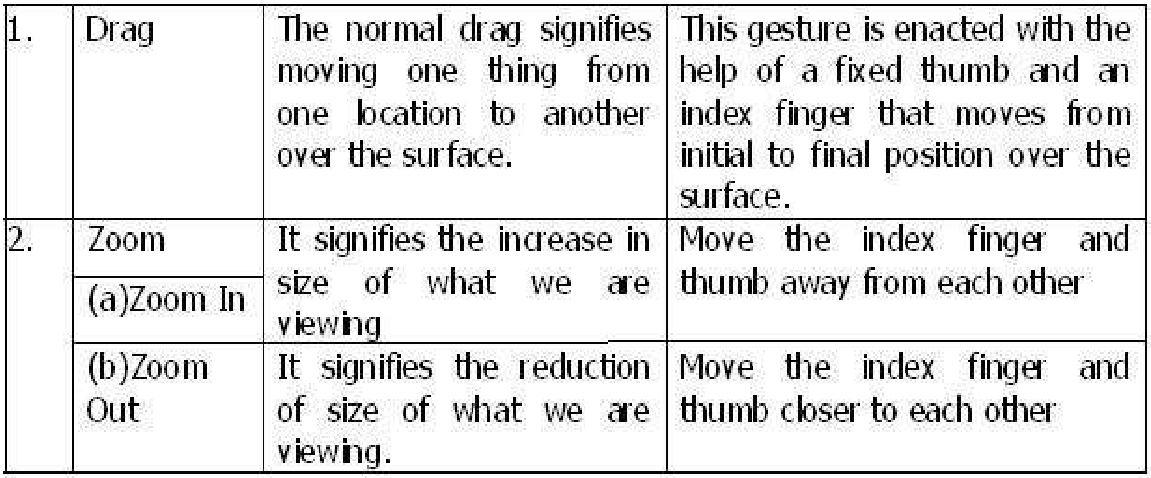
Vertical velocity, $y_2^{'}$can be defined as:

$$y_2^{'} = y_2 - y_1 \tag{4.4}$$



No	Gesture	Meaning	Signing Mode
1.	Click	It is derived from the normal clicking action that we do with the mouse on our PCs or on the touchpad of our laptops so as to open something or knock	It has been taken as tapping the index finger two times on the surface. The position of the tapping signifies the location of the thing to be opened
2.	Move (a)Frame	This gesture signifies drawing rectangular region to focus on something when we point to that area or explaining the view while taking a snapshot	This gesture has been enacted by drawing a rectangular boundary over the surface using index finger
	(b)Move Arbitary	This gesture gives the command to move in a random direction from its current position.	This has been enacted by the movement of index finger in an arbitrary direction from its current position.
3.	Rotate (a)Clockwise	It signifies taking turn in Clockwise direction.	A complete or incomplete circle is drawn.
	(b)Anticlockwise	It signifies taking turn in anticlockwise direction	A complete or incomplete circle is drawn.
4.	Pan	This gesture signifies movement of window from one place to another	This is performed by using index finger and middle finger touching each other.

Figure 4.3: Table to depict type of single finger gestures



1.	Drag	The normal drag signifies moving one thing from one location to another over the surface.	This gesture is enacted with the help of a fixed thumb and an index finger that moves from initial to final position over the surface.
2.	Zoom	It signifies the increase in size of what we are viewing	Move the index finger and thumb away from each other
	(a)Zoom In		
	(b)Zoom Out	It signifies the reduction of size of what we are viewing.	Move the index finger and thumb closer to each other

Figure 4.4: Table to depict type of two finger gestures

Similarly the horizontal velocity$x_2^{'}$can be defined as:

$$x_2^{'} = x_2 - x_1 \tag{4.5}$$

Now vertical acceleration $y_2^{''}$, can be defined as:

$$y_2^{''} = y_2^{'} - y_1^{'} \tag{4.6}$$

Similarly horizontal acceleration can be defined as:

$$x_2^{''} = x_2^{'} - x_1^{'} \tag{4.7}$$

Hence by applying the model above we can predict the corner in the subsequent frame. Say the corner now is (x,y) Then since we know the velocities and acceleration the new corner in the subsequent frame can be predicted to be:

$$x_{new} = x_2^{''} + x_1^{'} + x_1 \tag{4.8}$$

$$y_{new} = y_2^{''} + y_1^{'} + y_1 \tag{4.9}$$

4.6 Gesture Classification

The classification and subsequent gesture quantification is performed on the basis of this data. Currently we are working on a set of 9 gestures for arbitrary backgrounds and dynamic projection under highly varying

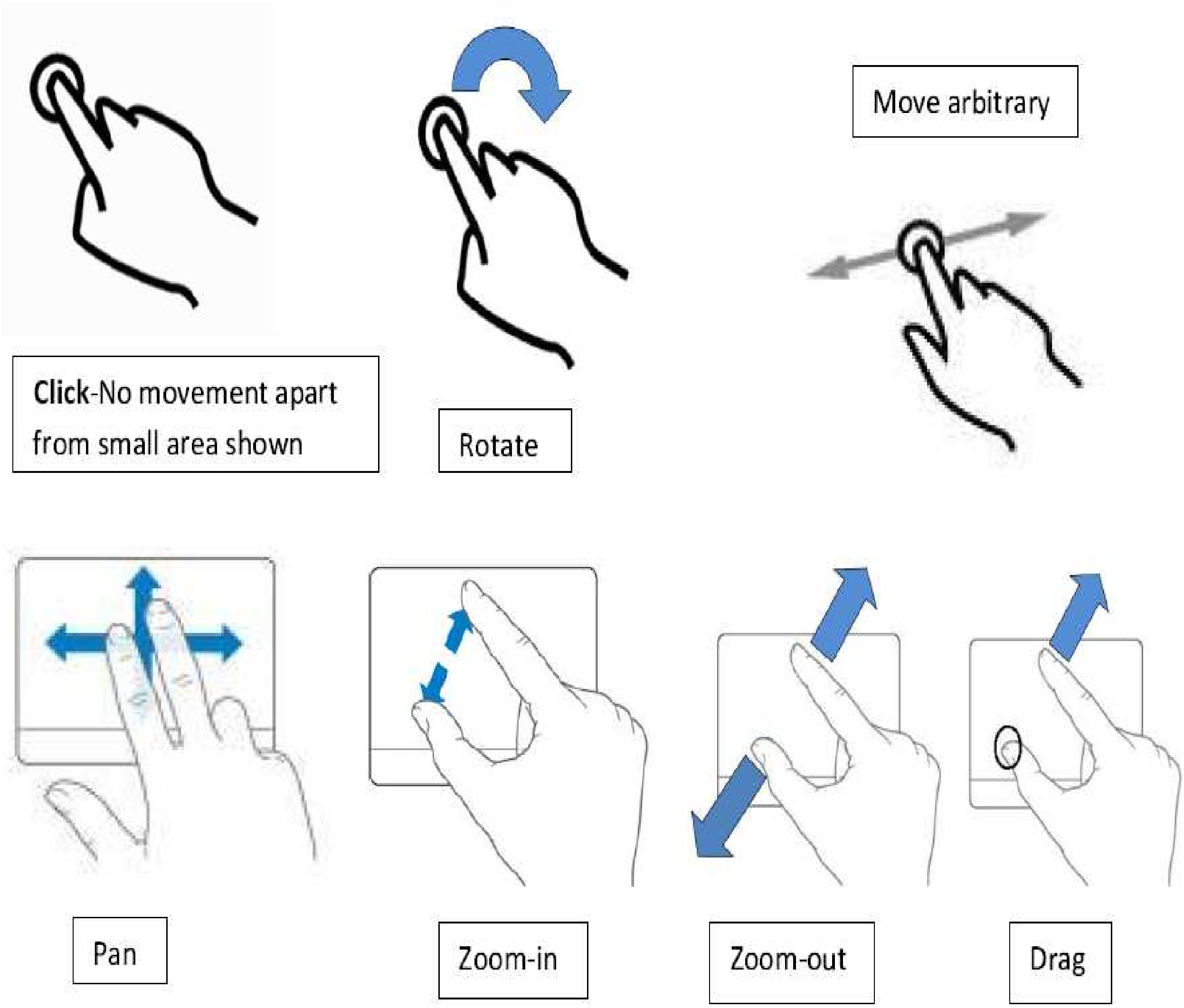


Figure 4.5: Detailed drawing of gestures

illumination. The gestures are detected on basis of following points and depicted in Fig. 4.5:

Single finger gestures:

Click: When there is no significant movement in the finger tip.

Pan: When the comparative thickness of the contour is above some threshold.

Move: When there is significant movement in the finger tip in any direction.

Rotate: For this slope is calculated at each point and the and following equations are implemented:

Let at some time t the coordinates of finger tip are (x,y)

Then at some time $t+k$ the coordinates are (x',y')

$$a = \frac{y' - y}{x' - x} \tag{4.10}$$

$$b = \frac{x' - x}{y' - y} \tag{4.11}$$

Now when the gesture ends find out how many times both *a* and *b* becomes zero and whats the sum of two. And by checking these two concepts we find out whether our gesture is rotate or not.

Two finger gestures:

Drag: When one of the finger tip stays constant and other finger tip moves.

Zoom out- When the Euclidean distance between the two finger tips decrease gradually.

Zoom-in- When the Euclidean distance between two finger tips increase gradually.

Chapter 5

Applications of the Algorithms

5.1 Paper Touchpad

In many intelligent environments, instead of using conventional mice, keyboards and joysticks, people look for an intuitive, immersive and cost-efficient interaction device. Here we design and describ a vision-based interface systems that is a paper touchpad. This is a type of virtual mouse for a desktop computer or a laptop and requires just a camera and a piece of paper with a drawing of a touchpad on it.

We can find many applications where this type of vision-based interfaces is desired. For an instance, in a smart room, the user wants to control a remote and large display or play a game, but he/she is in a sofa instead of in front of a computer, and therefore the mouse and keyboard or joystick may not be accessible. Then, what could he/she do? He/she may pick up an arbitrary piece of paper at hand and move his fingers or pens on the paper to drive a cursor or to type some text, or move the paper to control the game. Certainly, such an interaction is made possible by having a camera to look at the user and analyzing the movement of the paper and the user.Or maybe we would require different devices at different times like a piano, calculator, keyboard, mouse etc. What we need for all this is a single webcam and drawings of the respective instruments. This is the basic conceptual model and is as shown in the figure 5.1.

The earlier analysis of finger gesture recognition and intrusion detection is of the most general case. Relaxing one or more of the conditions will still yield situations of considerable interest such as the paper touchpad desribed above. Here we simply set the dynamic illumination component to zero in our equations and solve them in the presence of ambient lighting and arbitrary background.

The design of a paper touchpad involves application of the above concept and hence what we need is just that the camera be repositioned in such a way that it can view the paper touchpad.

Figure 5.1: Conceptual model

5.1.1 Setup

1. A monitor where mouse operations can be shown
2. A piece of paper with drawing of mouse or touchpad on it.
3. A USB camera: A simple USB web-cam, used for sensing the movement of hand on the paper touchpad.
4. A processor for receiving the sensed image from the camera, determining the location of the fingertip, converting it to monitor coordinates, tracing the region coordinate belongs to and finally communicating it to the pointer routine of the operating system.

The setup is shown in Fig 5.2

5.1.2 Algorithm

The algorithm is shown in the flowchart in Figure 5.3 and it includes the implementation of the following steps:

- Accurate image screen calibration: if the screen is flat, the plane perspectivity from the screen plane and its 2D projection on the image plane is described by a homography, a 3 $\times$ 3 matrix defined up to a scale factor. This matrix can be easily determined from 4 pairs of image-screen point correspondences. The correspondences are not difficult to obtain because we know the screen coordinates of four screen corners, and their corresponding image points can either be detected automatically or specified by the user.

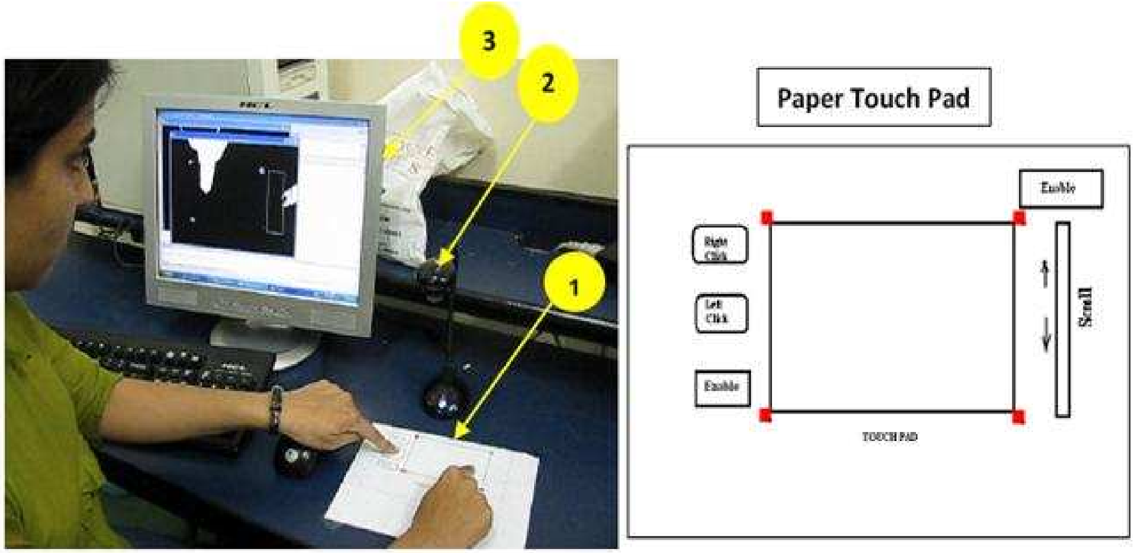


Figure 5.2: Paper touch pad

- Background subtraction and hand detection: Here we use the concept of background subtraction by differencing two frames one captured at the earlier(training) stages and the other captured when the finger enters, followed by application of adaptive thresholding.After this, we define an optimum threshold value of the intensity above which the pixels are considered as to constitute the intrusion. This is then followed by skin segmentation to obtain the blobs for the fingers entering the frames by detecting skin and non skin regions. A colour subspace strategy is adopted to take advantage of its stability under varying illumination conditions. In this method, the hue and saturation colour subspace (HS) is used. The main reason is that the HSI colour space closely corresponds to the human perception of colour and it has exhibited more accuracy in distinguishing shadows. Moreover, the influence of intensity can be decreased effectively. Hence we take the range of hues in which human skin lies and detect the skin regions accordingly.

- Robust finger tip location: This is done by contour detection followed by curvature calculation(corner point detection is found unnecessary for these relaxed conditions) at each point and detecting the positive extrema which in turn gives the tip of the finger as explained in Chapter 4.

- Mapping of regions for Left Click , Right Click, Enable and Scroll in the touchpad to perform the obvious actions. Enable is used for activating the basic part of touchpad through which the mouse pointer is moved. When this is disabled we can use the buttons like left and right click and scroll. Now these button are mapped to functions of mouse events in the windows library to do the required action. This step is depicted in figure 5.4:

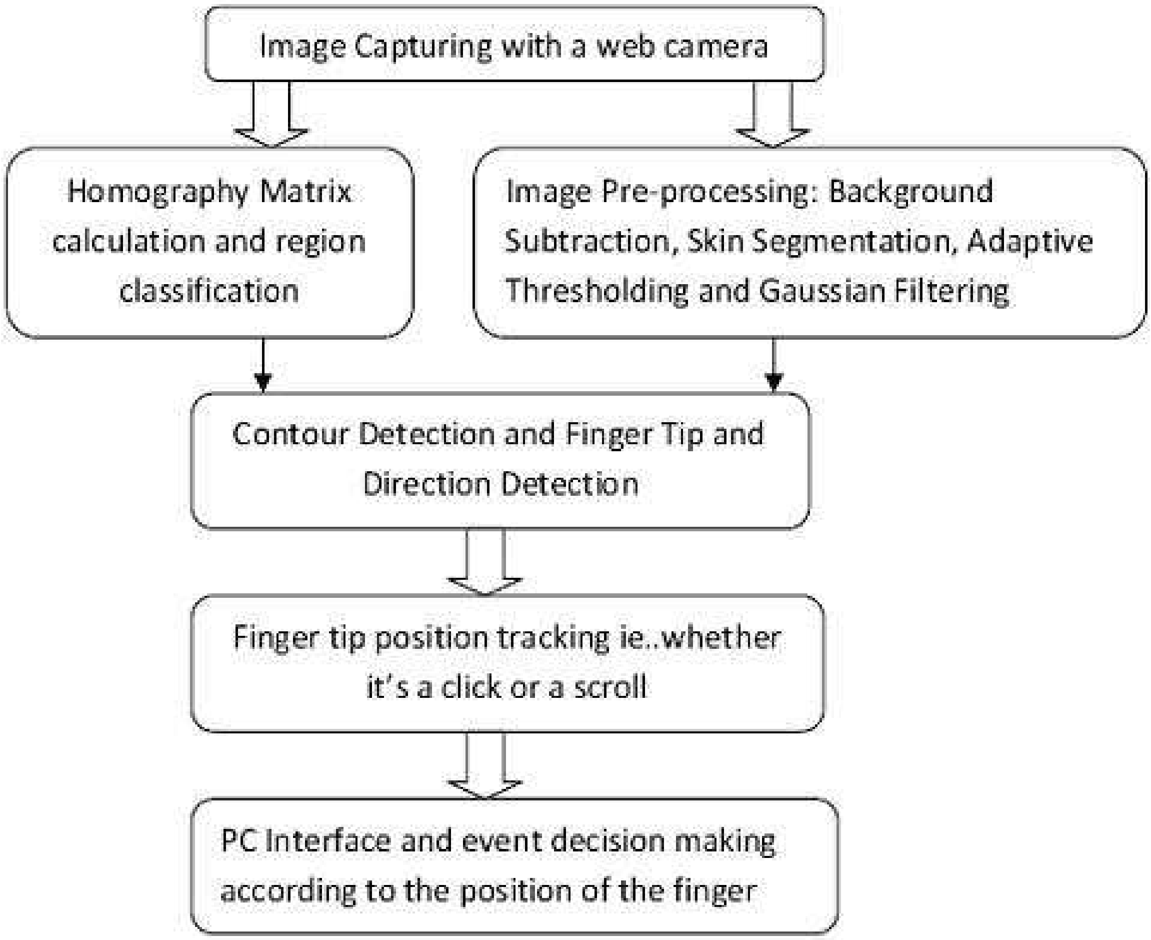


Image Capturing with a web camera
Homography Matrix calculation and region classification
Image Pre-processing: Background Subtraction, Skin Segmentation, Adaptive Thresholding and Gaussian Filtering
Contour Detection and Finger Tip and Direction Detection
Finger tip position tracking ie..whether it's a click or a scroll
PC Interface and event decision making according to the position of the finger

Figure 5.3: Implementation of paper touchpad

5.2 Laser Pointer Mouse

When making a large screen presentation from an PC before an audience presentation, a wireless remote controller is usually used if a speaker is away from the PC. However, the operability of the wireless remote controller is inferior to that of a mouse because it is difficult to repositon the mouse cursor quickly and precisely with the scroll buttons on the remote. This problem can be solved by managing the mouse movement through a laser pointer adapted to function as a pointing device and clicking can be done using the buttons of a wireless mouse.This setup is far more convenient to use better than a mouse, and can be applied to human-computer interaction applications such as games.

We present a novel LASER-Pointer tracking system for use in interactive presentations. The LASER-Pointers red dot is meant to draw the audiences attention to a specific place in a slide. Our system enables presenters to use the LASER-Pointer as they would a regular mouse cursor. The system detects the red dot on the screen

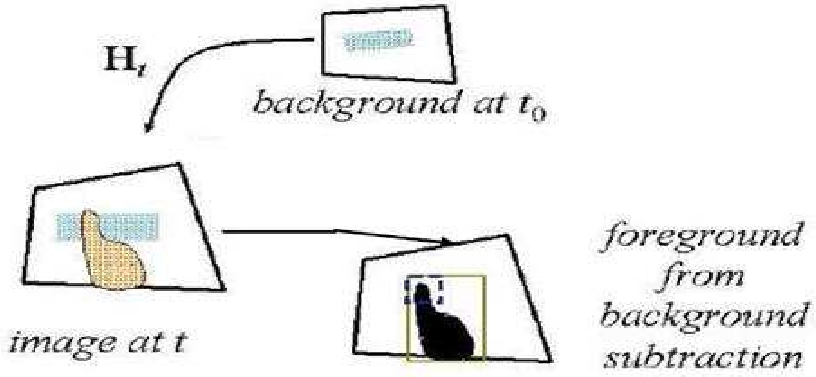


Figure 5.4: Foreground extraction

through a camera and automatically brings the cursor to it. To accomplish the position determination accurately, a calibration routine can be executed prior to each session of use. This feature paves the way for highly interactive and dynamic presentations.

5.2.1 Setup

1. A monitor or a projector: A large screen projector must be connected to the computer.
2. A laser pointer: Presenters usually use them to draw the audiences' attention to important issues on the screen.
3. A flat surface where information can be projected
4. A USB camera: A simple web-cam must be connected to the computer using a USB interface. It is used for sensing the image on screen and sending it for processing.
5. A processor for receiving the sensed image from the camera, determining the location of the laser spot with respect to the sensed image, and communicating it to the pointer routine of the operating system.

The apparatus setup is shown in the figure 5.5, 5.6 and 5.7.

5.2.2 Algorithm

- Calibration: In order to move the mouse cursor to the position of the laser spot which was detected from captured image, the coordinates of the captured image must be transformed to the coordinates of the desktop (computer screen) or the projected screen. The calibration procedures are as follows.

  a) A black square with small green triangular corners is displayed on the screen. Then, the image of the

Figure 5.5: Laser pointer system

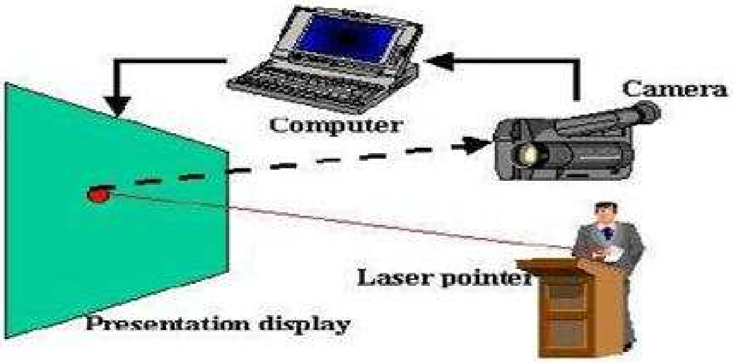


Figure 5.6: Laser projector system setup

screen is captured using the image sensor with normal camera mode.

b) Now by applying image processing techniques, the four corners are detected in that image and are calibrated to the four known corners of desktop by homography matrix calculation.

- Spot detection: We must not only reliably determine the spot position but also reliably detect whether or not it is present. The spot recognition software can sometimes lead to delays of greater than 200 milliseconds. Much slower sampling rates make the movement of the cursor appear jerky. Detecting laser on and laser off is somewhat problematic when using cheap cameras with automatic brightness control and low resolution. The automatic brightness controls continually shift the brightness levels as various room lighting and interactive displays change. This causes the detection algorithm to occasionally deliver a false off. Overlooking all these problems the brightest red spot is detected using image processing techniques within a radius of 10 pixels and to avoid jerks in the mouse pointer we apply some stabling techniques.

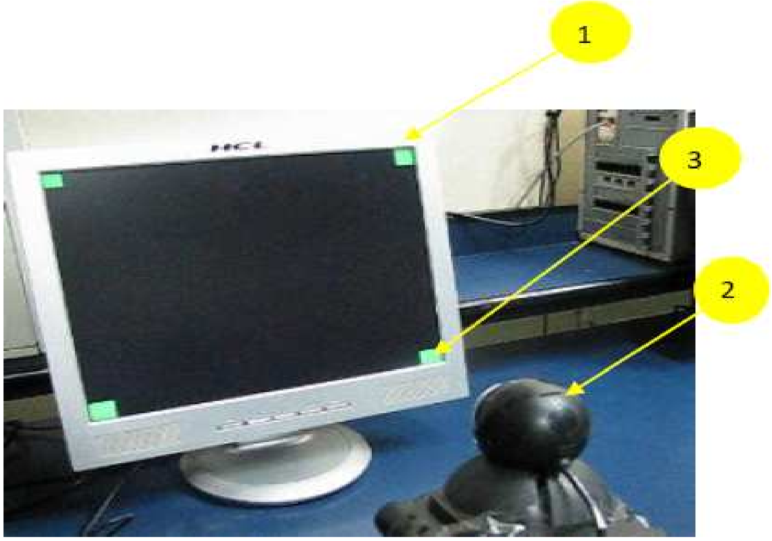


Figure 5.7: Laser pointer mouse.1 is the image which can be displayed on the projector and 2 is the webcam capturing the image 3 represent green corners for calibration

- Coordinate conversion: The point where red dot or laser pointer is detected is then converted to coordinates of the screen using the homography matrix calculated earlier

- Interface with mouse pointer: This coordinate is then interfaced with windows mouse operating functions to move the mouse pointer immediately to the converted coordinate.

Hence a function to automatically move the mouse cursor to the laser spot in higher illumination environment is realized. It is also shown that the laser spot can be easily and robustly detected even if non-uniform pattern like a desktop of Windows is used as a background.

Chapter 6

Results and Discussions

6.1 Results

We aim to design an accessory free system which is completely portable.In view of the above, as the first step we detect the intrusion successfully using two methods, One of them uses directly the statistical modelling whereas other uses Reflectance modelling followed by finger gesture recognition.

We subsequently have also designed two applications based on the system, the paper touch pad and laser pointer mouse. The results of each of the above are shown below.

6.1.1 Results with the direct statistical method for intrusion detection

This method was applied on two sets of data as given below:

1. With full human intrusion: Here body intrusion was detected in dynamic projection over plane background under varying projector illumination and near zero ambient light. Due to the color of clothes the detection was not so proper. The value of k used here is 0.85. It was observed that with greater values of k, the intrusions were not detected with ease. And lesser values of k led to more misdetections. The intrusions on the screen were detected and blobs were computed. The above was coded in OpenCV.The results are shown in fig 6.1

2. With hand intrusion on plane background: Here just the hand was introduced in the dynamic projection and by applying our algorithm we detect the intrusion, as shown in Fig. 6.2.

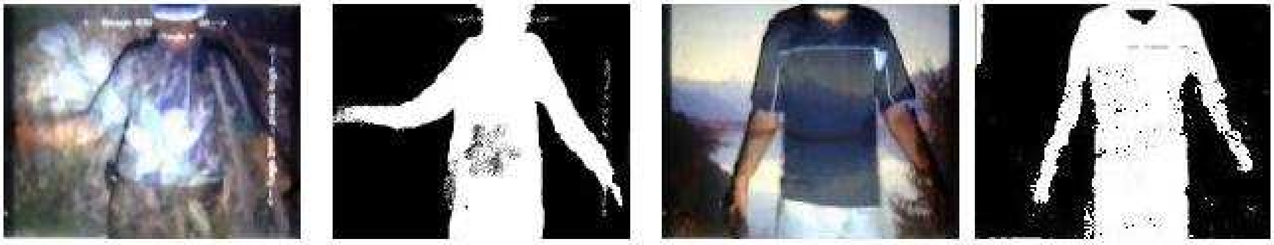


Figure 6.1: Foreground segmentation with full human intrusion using statistical model

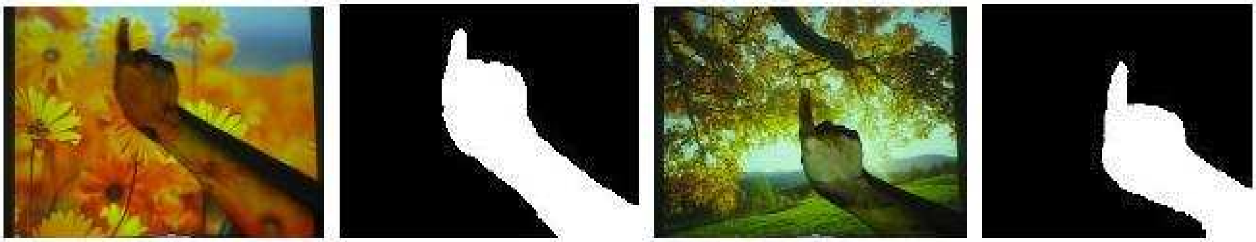


Figure 6.2: Foreground segmentation with hand intrusion using statistical model

6.1.2 Results with Reflectance modeling method of intrusion detection

This method was applied on the following sets of data:

1. Hand intrusion in a dynamic projection upon plane background: Figure 6.3

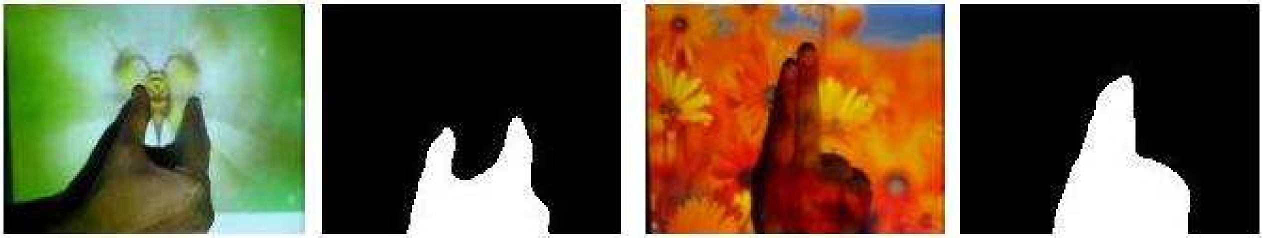


Figure 6.3: Foreground segmentation with hand intrusion using reflectance modelling on plane background

2. Hand intrusion in a dynamic projection upon arbitrary background: Figure 6.4

The reflectance modelling method gives better results as compared to the direct statistical method since we model not just the surface but also the hand in the ambient and front dynamic projection. The results for plain and arbitrary background are both equally accurate since change of background does not affect the output in case of Reflectance modelling. Not much effect was seen in the two because our algorithm is invariant to

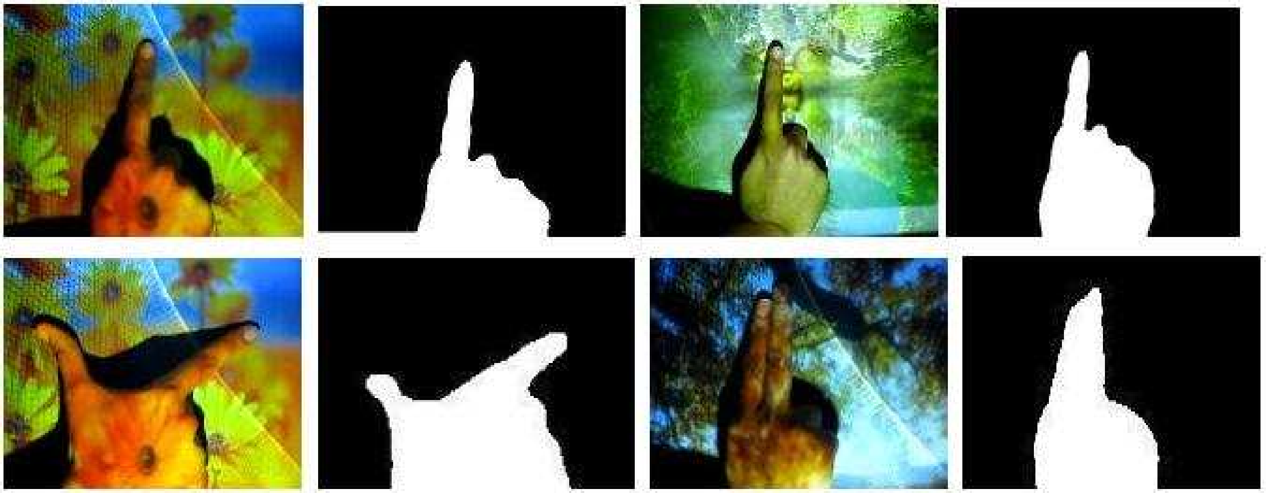


Figure 6.4: Foreground segmentation with hand intrusion using reflectance modelling on arbitrary background

arbitrary backgrounds and skin detection algorithm worked successfully in each case as can be seen from the results.

6.1.3 Finger gesture and attribute recognition

In our algorithm of gesture detection, we have achieved optimal results. We found that the primary hurdle in developing a successful finger tracker was the clean segmentation of the hand. Once a clean binary image of the hand is produced using the method of reflectance modelling preferably, finger detection can be achieved easily by applying the algorithm explained in chapter 4. Specifically, the system can track the tip positions of the thumb and fingers effectively thereby detecting the gesture and its attributes like direction, trajectory, velocity, orientation, etc.

1. With plain background:After intrusion detection, we detect the finger tips in the intrusion. Depending upon the number of tips we classify the gesture into the appropriate category:
   (a) For one finger gestures: The results are shown in figure 6.5

   (b)For multiple finger gestures: Here the various poses of two finger gestures are shown in fig 6.6.

2. With arbitrary background: Figure 6.7

6.1.4 Paper touchpad

The paper touchpad is a kind of a virtual mouse used for providing mouse cursor and its functions in any computer system using an ordinary sheet of paper with a few markings on it. The red dots on the corner of the

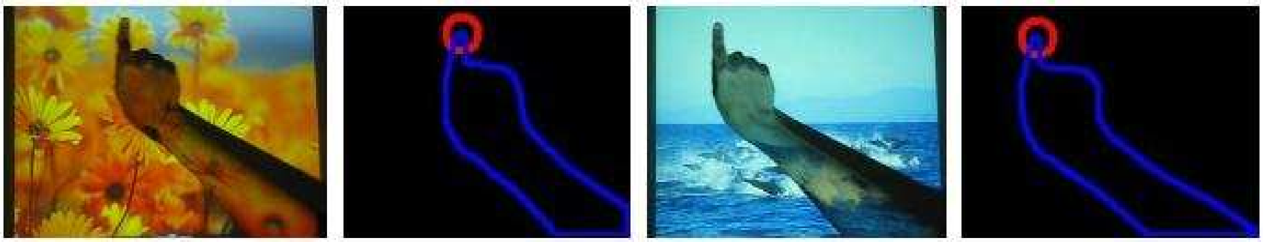


Figure 6.5: Fingertip and type of gesture detection for one finger gestures on plane background

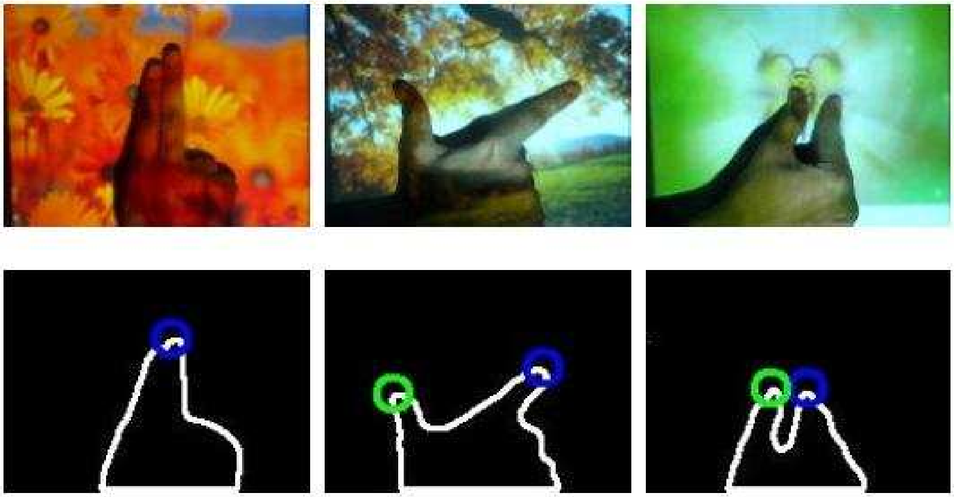


Figure 6.6: Fingertip and type of gesture detection for multiple finger gestures on plane background

printout of the touchpad are used for homographic mapping. The figure 6.8 shows the movement of the cursor and leftclick operation of the mouse. In the first figure we left click on ’My Pictures’.In the figure besides it, the window of My Pictures is opened on the display screen as a result.

6.1.5 Laser pointer mouse

This kind of system allows the movement of the mouse pointer on any projector system alongwith the movement of the laser dot on the projection screen. This system tracks the dot precisely in each frame and any abrupt change like switching on and off of laser and even abrupt jumps between two far off positions are easily and accurately. This also requires an initial calibration phase to map the screen to the window where laser pointer moves to get the smooth and accurate movement of the mouse cursor.Its working is shown in the Fig 6.9 and 6.10.

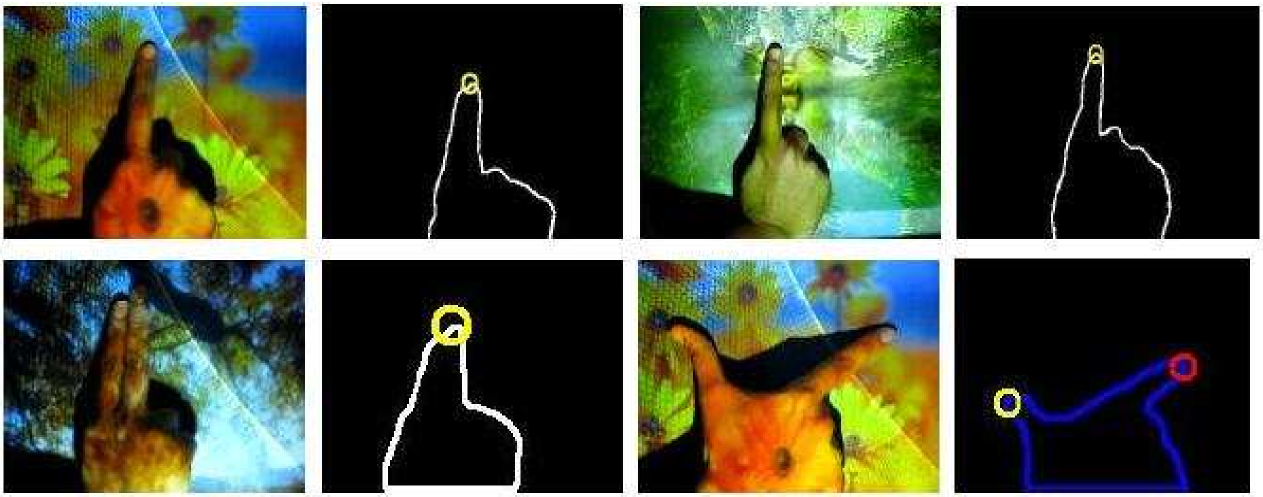


Figure 6.7: Fingertip and type of gesture detection for single and multiple finger gestures on arbitrary background

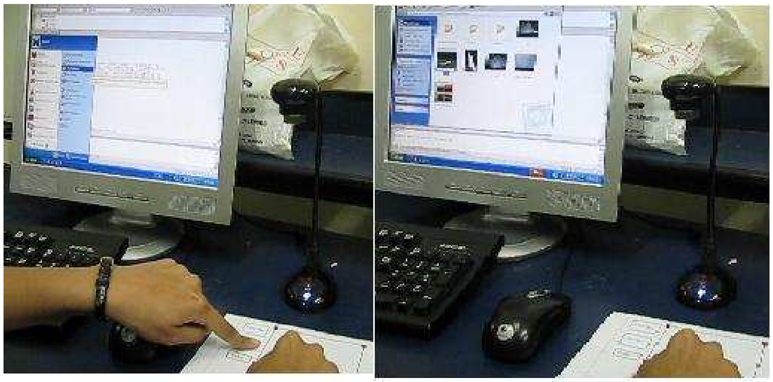


Figure 6.8: Paper touch pad showing operation of left click

6.2 Discussion

6.2.1 Requirements

The following are the requirements for getting best results for the algorithms described in chapters above:

1. Good computing power
2. Relatively light colored videos to be projected.
3. Camera, preferably without AGC and white balance adaptation.
4. Camouflage (same color on foreground and background) must be avoided in case of human intrusion because a lot of detection problems may arise.

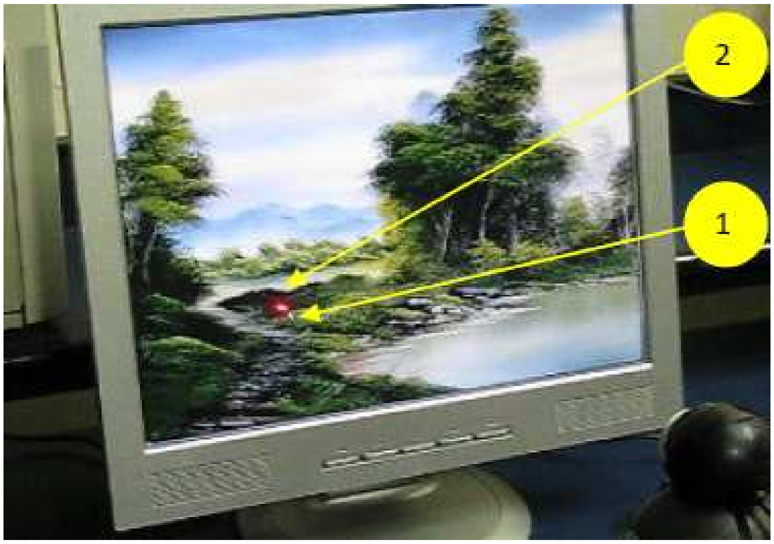


Figure 6.9: Laser pointer mouse showing the recording webcam 1-Mouse cursor and 2-red dot of laser pointer

6.2.2 Limitation of projector camera system

With the development of projectors, the display quality an issue of concern. The luminance uniformity over the area of projection is a common problem in projection displays. We recommend a co-axial projector-camera system whose geometric correspondence is thus independent of changes in the environment. To handle photometric changes, our method uses the errors between the desired and measured appearance of the projected image and compensates for them using reflectance modelling. A key novel aspect of our algorithm is that we combine a physics-based model with dynamic feedback to achieve real time adaptation to the changing environment. The camera can also assist projection by color correcting a homogeneous colored surface or by correcting for spatially varying color and texture. A fundamental assumption we make is that the surface is Lambertian and if the image looks correct to the camera, we will assume it looks correct to a viewer.

6.2.3 Limitations due to camera

1. Automatic gain control is abbreviated AGC. It is a feature where the amount of increase is adjusted automatically based upon the strength of the incoming signal. Weaker signals receive more gain; stronger signals receive less gain or none at all.It is an adaptive system found in many electronic devices. The average output signal level is fed back to adjust the gain to an appropriate level for a range of input signal levels. The desired output signal remains essentially constant despite variations in input signal strength.Our algorithm needs to take special care to cancel the effects of AGC.Thus matters would be simpler if AGC could be disabled.

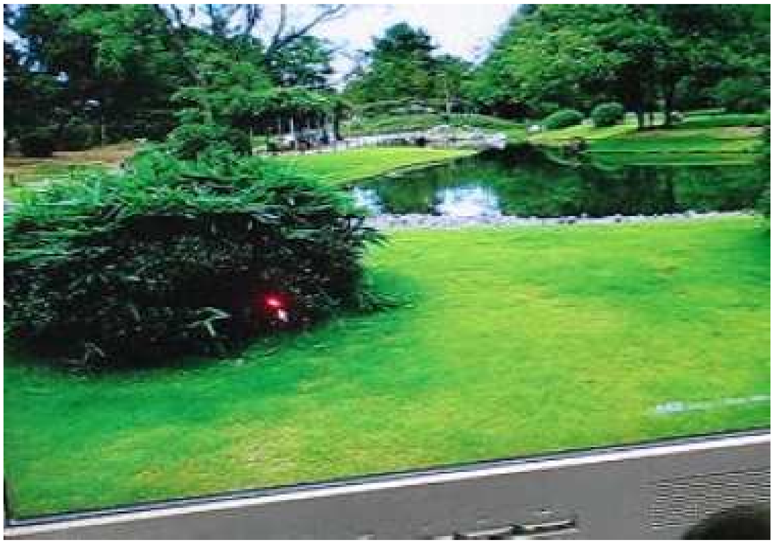


Figure 6.10: Laser pointer mouse showing movement of cursor with red dot

2. Intrusions should not be too close to the camera. Blooming will occur as it will reflect almost all the light into the camera causing the camera to go blind(saturated).

6.2.4 Improvements

- We avoid the correspondence problems of projector camera system altogether, by making the optics of the projector and the camera coaxial or as nearly coaxial as possible. This configuration ensures that all surfaces visible to the camera can also be projected upon; there is no possibility of occlusion and no parallax.
- Camera without automatic gain control and white balance adaptation preferable should be used.
- Intrusions should be introduced near the projector screen and not near the camera.

Chapter 7

Conclusion and Future Possibilities

7.1 Conclusion and Summary

In this dissertation, we have developed a vision-based human-computer interaction system implemented in OpenCV

By application of our algorithms for both plain and arbitrary backgrounds, we detect the intrusion successfully. This method is accurate and robust and works over a wide range of ambient lighting and varying illumination conditions.The few key points are as follows:

- Since background learning is not required, intrusions can be detected even with the help of difference in reflectivity from the screen surface and the intrusions.
- The update of model parameters has to be carried out pixel wise or block wise for both the projected video as well as the camera captured videos. A one to one correspondence between the pixels is then taken into consideration. We can thus now apply foreground extraction technique to figure out the pixels containing intrusions.
- Blending the surface reflectance characteristics and the use of hue modelling for skin detection gives good results

Secondly we aimed at detection of the finger tips and also finding out the type and attributes of the gestures performed.We can robustly and accurately track the fingertip in the gestures thereby detecting the trajectories and pointing direction which in turn helps in classification of the gesture precisely into the categories mentioned in chapter 4. Some key points are:

- Finding the number of maximum positive comparable highest curvature points in the gesture is followed by tracking of the fingertips by application of a motion model to improve robustness.
- The update occurs frame to frame and the values are verified by the two models applied for detection.

Based on our main algorithm we have also implemented paper touchpad as explained in chapter 5 .We set the dynamic illumination component to be zero in our model to design this system as there is no variation in the background with time.Whenever the paper touch pad is put to use we need an initial calibration phase which ensures proper working.The delay between the finger movement on the paper touchpad and the cursor on the screen is in milliseconds. This aims at replacing the hardware mouse which just contributes to E garbage. The following implementation have clearly shown high robustness, accuracy and flexibility. Many other applications are possible like controlling a calculator, painting with fingers, virtual keyboard, virtual piano and controlling the display of 3D objects.

Alongwith the above algorithms and applications this dissertation also discusses an implementation of a laser pointer mouse which can be used in any projector camera system. Laser may be used to control the mouse movement on the projected data.

7.2 Applications

This work finds many applications in day to day life for new era systems which can act as both mobile and computers. The best application is in the making of a human computer interface (HCI) where the interfacing devices like keyboard, mouse, calculator, piano etc would become obsolete. It will help in creating a new era system consisting of a projector-camera combined with a processor which can be used as a computing device much smaller then any of the existing systems.

There are several factors that make creating applications in HCI difficult. They can be listed as:

- The information is very complex and inconsistent
- Intrusion detection techniques should be highly effective
- Developers must understand exactly what it is that the end user of their computer system will be doing.

7.3 Future Possibilities

- This may be further extended for whole body gestures which may be used for sign language recognition or for robotic and other applications.

- We may also use an infra-red laser or flood illumination as an invisible 4th channel for detecting more details of gestures performed and to remove the effects of the visible band varying illumination even further.
- Extract more information like speed and acceleration from the gesture performed and allowing the user to communicate through these parameters as well.
- As the end result, we aim to design a robust real time system which can be embedded into a mobile device that can be used without accessories anywhere a flat surface and some shade is available. The single unit would substitute for the computer/communicator, the display, keyboard and pointing device which may require a projector, camera, processor and memory.
- We can move on to develop vision techniques to recognize sequences, instead of just individual gestures as well as more complicated finger gestures, which can be a great help in understanding sign language better

Bibliography

[1] Motonorihi , Shinji Ueda and Kohei Akiyama, Human Interface based on finger Gesture Recognition using Omni-Directional Sensor

[2] Ray Lockton,Oxford University,Hand Gesture Recognition using special glove and wrist band.

[3] Andrew Wu, Mubarak Shah and N Da Vitoria Lobo 3D Finger Tracking using Single Camera

[4] Song-Gook Kim, Jang-Woon Kim and Chil-Woo Lee, Implementation of Multi-touch Tabletop Display for HCI (Human Computer Interaction)

[5] Han, J.Y.: Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, pp. 115118.

[6] Grossman, T., Balakrishnan, R., Kurtenbach, G., Fitzmaurice, G., Khan, A., Buxton, B.: Interaction techniques for 3D modeling on large displays. In: Proceedings of the 2001 symposium on Interactive 3D graphics, pp. 1723 (2001)

[7] Wayne Westerman, John G. Elias and Alan Hedge, A Multi touch :A new tactile 2-D gesture interface for Human Computer Interaction

[8] Ramon Hofer, Daniel Naeff and Andreas Kunz, FLATIR:FTIR Multi touch detection on a Discrete Distributed Sensor Array

[9] Xing Jian-guo, Wang Wen-long, Zhao Wen-min, Huang Jing, ”A Novel Multi-Touch Human-Computer-Interface Based on Binocular Stereo Vision,” iuce, pp.319-323, International Symposium on Intelligent Ubiquitous Computing and Education, 2009

[10] Han-Hong Lin and Teng-Wen Chang, A Camera Based Multi- touch Interface Builder for Designers

[11] Ross Eldridge and Heiko Rudolph,Stereo Vision for Unrestricted Human Computer Interaction

[12] M. Fukumoto, Y. Suenaga, and K. Mase, Finger-pointer: Pointing Interface by Image Processing, Computer and Graphics, vol. 18, no. 5, 1994, pp. 633-642.

[13] J. Segan and S. Kumar, Shadow Gestures: 3D Hand Pose Estimation Using a Single Camera, Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 99), IEEE Press, Piscataway, N.J., 1999, pp. 479-485.

[14] A. Utsumi and J. Ohya, Multiple-Hand-Gesture Tracking Using Multiple Cameras, Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 99), IEEE Press, Piscataway, N.J., 1999, pp. 473- 478.

[15] J. Crowley, F. Berard, and J. Coutaz, Finger Tracking as an Input Device for Augmented Reality, Proc. IEEE Intl Workshop Automatic Face and Gesture Recognition (FG 95), IEEE Press, Piscataway, N.J., 1995, pp. 195-200.

[16] N. Shimada et al., Hand Gesture Estimation and Model Refinement Using Monocular Camera-Ambiguity Limitation by Inequality Constraints, Proc. 3rd IEEE Intl Conf. Automatic Face and Gesture Recognition (FG 98), IEEE Press, Piscataway, N.J., 1998, pp. 268-273.

[17] Y. Wu, J. Lin, and T. Huang, Capturing Natural Hand Articulation, Proc. IEEE Intl Conf. Computer Vision (ICCV 01), vol. 2, IEEE Press, Piscataway, N.J., 2001, pp. 426-432.

[18] V. Pavlovic, R. Sharma, and T. Huang, Visual Interpretation of Hand Gestures for Human-Computer Interaction: IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, , pp. 677-695.

[19] Kenji Yoka, Yoichi Sato and Hideki Koike, Real Time Finger-tip Tracking and Gesture Recognition.

[20] M. Jones, J. Rehg. Statistical color models with application to skin detection. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1999. Vol. 1, pp. 274-280.

[21] J. Rehg, T. Kanade. DigitEyes: Vision-Based Human Hand-Tracking. School of Computer Science Technical Report CMU-CS-93-220, Carnegie Mellon University, December 1993.

[22] Y. Sato, Y. Kobayashi, H. Koike. Fast tracking of hands and fingertips in infrared images for augmented desk interface. In Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition (FG), 2000. pp. 462-467.

[23] J. Segen, S. Kumar. Shadow gestures: 3D hand pose estimation using a single camera. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1999. Vol. 1, pp. 479-485.

[24] Z. Zhang, Y. Wu, Y. Shan, S. Shafer. Visual panel: Virtual mouse keyboard and 3d controller with an ordinary piece of paper. In Proceedings of Perceptual User Interfaces, 2001.

[25] F. Dadgostar and A. Sarrafzadeh, A component based architecture for vision based Gesture recognition

[26] Daniel Heckenberg and Brian C. Lovell, MIME: A Gesture driven computer Interface.

[27] Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-Time Tracking of the Human Body. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7)(1997)780785

[28] Rowe, S., Blake, A.: Statistical mosaics for tracking. IVC 14 (1996) 549564

[29] Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: International Conference on Computer Vision.Volume1.(2001)105112

[30] Rother, C., Kolmogorov, V., Blake, A.: Grabcut-interactive goreground extraction using iterated graph cuts. In: ACM SIGGRAPH. Volume 24. (2004) 309314

[31] Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. In: ACM SIGGRAPH.Volume23.(2004)303308

[32] Freedman, D., Zhang, T.: Interactive graph cut based segmentation with shape priors. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition.Volume1.(June2005)755762

[33] Gordon, G., Darrell, T., Harville, M., Woodfill, J.: Background Estimation and Removal Based on Range and Color. (1999) 459464

[34] Zeng, G., Quan, L.: Silhouette extraction from multiple images of an unknown background. In Hong, K.S., Zhang, Z., eds.: Asian Conference on Computer Vision.Vol 2., Asian Federation of Computer Vision Societies (2004) 628633

[35] Sormann, M., Zach, C., Karner, K.: Graph cut based multiple view segmentation for 3D reconstruction. In: The 3rd International Symposium on 3D Data Processing,Visualization and Transmission. (2006)

[36] Pranav Mistry, Pattie Maes and Liyan Chang, WUW - Wear Ur World - A Wearable Gestural Interface

[37] Friedman, N., Russell, S.: Image Segmentation in Video Sequences: A Probabilistic Approach. In: Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence.(1997)

[38] Daniel hackenberg and Brian C lovell: A gesture driven computer interface.