BACKGROUND REFLECTANCE MODELING FOR ROBUST FINGER GESTURE DETECTION IN HIGHLY DYNAMIC ILLUMINATION

Armin Mustafa[1] and K.S.Venkatesh[1]

[1]Indian Institute of Technology, Kanpur
arminmustafa@gmail.com, venkats@iitk.ac.in

Abstract: We aim to develop an ’accessory-free’ or ’minimum accessory’ interface used for communication and computation without the requirement of any specified gadgets such as finger markers, colored gloves, wrist bands, or touch screens. We detect various types of gestures, by finding fingertip point locations in a dynamic changing foreground projection with varying illumination on an arbitrary background using visual segmentation by reflectance modeling as opposite to recent approaches which use IR(invisible) channel to do so. The overall performance of the system was found to be adequately fast, accurate, and reliable. The objective is to facilitate in the future, a direct graphical interaction with mobile computing devices equipped with mini projectors instead of conventional displays. We term this a dynamic illumination environment as the projected light is liable to change continuously both in time and space and also varies with the content displayed on colored or white surface.

Keywords: Computer vision, HCI(Human Computer Interaction), Reflectance Modeling, Gesture detection

1. Introduction

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.

Our approach operates exclusively with visual detection and applies the principle of reflectance modeling to achieve this: it includes the following two steps:

1. Dynamic background subtraction under varying illumination upon arbitrary background using a reflectance modeling technique that carries out visual detection of the shape of intrusion on the front side projected background.
2. Detecting the gestures and quantifying them: this is achieved by detection of contour trajectory of the intruding hand through time and tracking multiple salient points of the intrusion contour. Gestures can then be classified and subsequently quantified in terms of the extracted multi trajectory parameters such as position, velocity, acceleration, curvature, direction, etc.

A special case of the above general approach is the demonstrated Paper Touchpad which functions as a virtual mouse for a computer, operating under conditions of stable (non-dynamic) illumination on arbitrary backgrounds, with the requirement of a single webcam and a piece of paper upon which ‘touchpad’ is printed. This is an interactive device easy to use anywhere, anytime and requires homographic mapping between screen and piece of paper. The paper touchpad does not obviate the display.

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 (from excessively bright light such as direct sunlight) is available. The single unit would substitute for the computer or communicator, the display, keyboard, mouse, piano, calculator etc. and pointing device which may require a projector, camera, processor and memory.

2. Related Work

The techniques available hitherto have usually been using some gadgets or other sort of assistive tools. For example visual ways to interact with the computer using hand gestures involved use of glove-based or wrist band based devices [9]. Later on, single- and multitouch technologies, essentially touch-based, used Multi Touch surfaces [2,4,10] and specific systems interfaced with it. 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 [8,14]. Infrared Imaging for building an interface [3] and augmented desktops [6] also came into picture. One approach uses markers attached to a user’s hands or fingertips to facilitate their detection [7]. A few provide a comprehensive survey of hand tracking methods and gesture analysis algorithms [12]. For an efficient algorithm we need that the dynamic background subtraction must give good results. Normal distributions used in conjunction with the Mahalanobis distance[15], or mixture of Gaussian to account for multi-value pixels located on image edges, or depth information obtained using 2 cameras was used for classification into background and foreground[1].

But all these either assume background to be static one or use infrared light for visual segmentation. [11] shows detection of hand gestures for the replacement of mouse but that is working for static background and we need a separate monitor for operation. Our system is generic and aims to replace monitor, keyboard, piano, mouse etc. [13] and [5] detect hand gestures but not in dynamic background and highly changing lighting conditions and does not eliminate the use of monitor, keyboard etc. Also in [5], hand gestures are detected to interact with sound/music system which is specific whereas our system is much more generic.

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. Our system attempts to be a 'minimum accessory interface' which uses visual segmentation techniques for foreground segmentation on dynamic backgrounds. With reducing prices of cameras and projectors; it becomes low cost also and replace hardware like items mouse, keyboard, monitor etc.

3. Techniques

3.1 Projector Camera System and its Calibration

Projection systems can be used 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. Our system uses projector camera system to design a new era 'accessory free' system. The system is shown in Fig. 1. And the calibration is done as follows:

1. 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.
2. Pure red, green and blue colors are sent via the projector and captured by the camera for a set of n frames. 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. The variance for each RGB output component for every individual pure input is then determined.

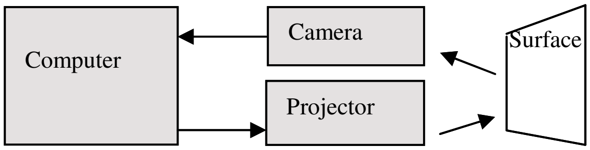

Figure 1: Projector camera system

3.2 Assumptions and Calibration

We expect the surface of the system (consisting of the computing device, its projector and its camera) meet some general criteria of near-flatness, Lambertian (non-specular) reflectivity uniform over the projection surface, and with a reflectance coefficient that is not too low at any wavelength in the visible band. We allow the surface to possess some space-varying texture, subject to meeting the criteria set down above at each point individually. We further allow ambient illumination to be present, and to have any spectral bias, so long as its intensity allows the projector output to sufficiently dominate and so long as it is constant over time.

The camera-projector combination is however assumed to be co-axial and fixed in space relative to each other (by sharing a common chassis, for example) and fixed in space relative to the projection surface during an interaction session. The hand and fingers are held close to the projection surface to ensure bounded depth i.e… uniform reflection. Under the abovementioned set of assumptions, the system's operation during a session is initiated with a session calibration phase whose process consists of the following:

1. Calibration to ambient illumination.
2. Calibration to skin color under ambient illumination.
3. Surface texture calibration under projector illumination.
4. Skin color calibration under projector illumination.
5. Camera-projector co-calibration for white balance.

Apart from all these session-specific parameter settings, the system has to be one-time factory calibrated to map camera and projector spatial and temporal resolutions to one other. The experimental setup is shown in Fig. 2

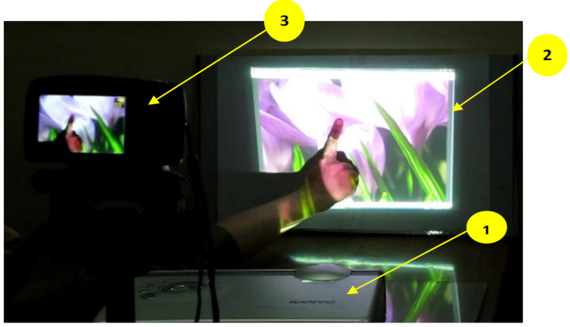

Figure 2: Experimental setup of the invention: 1-Projector,2-Screen on which random scenes are being projected and hand is inserted as an intrusion and 3-Camera recording the screen.

3.3 Requirements:

1. Relatively light colored videos to be projected
2. Camera, preferably without AGC and white balance adaptation
3. Camouflage (same color on foreground and background) must be avoided in case of human intrusion.
4. Each finger gesture video should last for no more than about 3-4 seconds. Longer gesture times will delay the identification of the gesture as it processes once gestures are complete
5. At a 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.

4. Proposed Approach using Reflectance Modeling

According to the properties of the surface, Spectral response on the plane upon which projection takes place will differ with the spectral response of the intruding object, thus giving evidence of intrusion. We have used the concept of reflectance modeling in our work. The reflectances of various objects like hand, arbitrary background, the surface etc creates different models which are in turn used for foreground detection under varying illumination. Since it is not the appearance of the surface that is being modeled, but its reflectance, intrusion detection becomes possible over a wide range of even spatially and temporally varying illumination conditions. Using these concepts we develop an algorithm which uses reflectance properties to detect the intrusion.

4.1 Intrusion Detection

Reflectance modeling: Reflectance modeling represents the more refined approach to the problem of intrusion detection in highly varying and dynamic illumination in the presence of near-constant non-dominant ambient illumination. The main aim of the problem was detection of events in possibly highly dynamic scene projected on the user specified surface by the computer through the mini projector. 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:

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

a) Recording and modeling surface under ambient lighting (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 texture plane surface.

b) Now, the 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.

c) Hand is removed from the visibility of camera and the projector is switched on with just white light (projector RGB = 255,255,255). 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}.

d) Now, a surface in projector light (**S_P**) is determined by differencing S_{AP} and S_A. The relationship of the parameters is as follows: **$S_P = [S_P^R, S_P^G, S_P^B]^T$.** This specifies the green, red and blue component of the surface under projection.

$$S_P^R = S_{AP}^R - S_A^R;\ \ S_P^G = S_{AP}^G - S_A^G;\ S_P^B = S_{AP}^B - S_A^B; \tag{1}$$

e) The hand is introduced under a scenario when the ambient light is on, and the projector is displaying white light. This is a new model of hand which is H_{AP} captured under combination of ambient light and projector white light.

f) Hence, the model of the hand in projected white light is obtained, H_P which is obtained in the same way as **S_P.**

$$H_P = [H_P^R, H_P^G, H_P^B]^T\ H_P^R = H_{AP}^R - H_A^R;\ H_P^G = H_{AP}^G - H_A^G; H_P^B = H_{AP}^B - H_A^B; \tag{2}$$

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

h) 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.

i) Now the expected values of the dynamic background being projected (**S_{new}**) is obtained, seen through the camera by performing a matrix multiplication of the D(t) and the S_P followed by addition of the S_A .

$$S_{new} = (D[n]\times S_P) + S_A;\ \ S_{new} = [S_{new}^R, S_{new}^G, S_{new}^B]^T \tag{3}$$

We build individual statistical Gaussian models for each red, green and blue colors. According to the models 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 as shown below in Fig. 3.

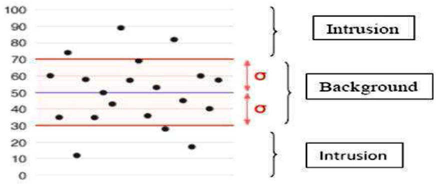

Figure 3: Relative distribution of pixels showing intrusion and background

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 is represented by Fig. 4.

Here:

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

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

And hence other symbols are self explanatory.

The expected values for red, green and blue component can be calculated using matrix multiplication of the means and normalized RGB value as shown in Fig. 5

$$\begin{pmatrix} R\mu_r \pm x_r & R\mu_g \pm x_g & R\mu_b \pm x_b \\ G\mu_r \pm y_r & G\mu_g \pm y_g & G\mu_b \pm y_b \\ B\mu_r \pm z_r & B\mu_g \pm z_g & B\mu_b \pm z_b \end{pmatrix} \begin{array}{l} \leftarrow \text{output for red input} \\ \leftarrow \text{output for green input} \\ \leftarrow \text{output for blue input} \end{array}$$

Figure 4: Color Bias Matrix

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

Figure 5: Matrix multiplication to calculate the expected values

The RGB values of every pixel of the captured frames can be put and compared with the expected values as expressed herein below. If;

$\textit{Observed value} > (\textit{Expected value} \pm (k*\sigma))$ --Then it is Intrusion (4)

$\textit{Observed value} \leq (\textit{Expected value} \pm (k*\sigma))$ --Then it is Background (5)

where, ‘k’ is a constant, can be checked by hit and trial methods. Best match is used for thresholding.

‘σ’ is the variance calculated during the projector camera calibration phase for each rgb color and is defined by:

For red: $var(R) = \sigma(R) = x_r + y_r + z_r$ (6)

For green: $var(G) = \sigma(G) = x_g + y_g + z_g$ (7)

For Blue: $var(B) = \sigma(B) = x_b + y_b + z_b$ (8)

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. So we transform in YC_bC_r space and then apply threshold to Y component to enhance the segmentation 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 is done by developing a linear relationship between luminance and ‘k’:

$$\mathbf{k_{new1} = (slope*L)+(.82)-(slope*L_{min})} \qquad (9)$$

where, k_{new1} -The factor by which the old value of ‘k’ must be multiplied

$$\mathbf{slope = (0.06/(L_{max} - L_{min}));} \qquad (10)$$

L_{min}, L_{max} -Minimum and Maximum Luminance for pixels in the frame respectively. Hence,

$$\mathbf{k_{new1} = \quad k * k_{new1}} \qquad (11)$$

3. ***Dominant color compensation:*** Compensates for a different white balance settings in the camera and the projector and for possible inbuilt white balance adaptation by the camera. The value of ‘k’ is adjusted according to the dominant color so as to increase the sensitivity according to the color whose value is maximum

$$\mathbf{k_{new2} = ((R+G+B) \div (3 * Dom\_color))+0.9} \tag{12}$$

where, Dom_color - Dominant color for that particular pixel and 'k_{new2}' is a new constant for interpreting 'k'
Hence, the final value of constant 'k' is:

$$\mathbf{k_{final} = k_{new1} \div k_{new2}} \tag{13}$$

After this Dominant color and lighting compensation we replace **k** with $\mathbf{k_{final}}$ in (4) and (5) and detect intrusion.

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. We have performed modeling of the skin by matrix multiplication of the normalized RGB values in the model $\mathbf{H_P}$ with **D[n]**, the data which is getting projected again, followed by an addition of the model of hand in ambient $(\mathbf{H_A})$. This gives us a check point in tandem to the Gaussian model method.

$$\mathbf{H_{new} = (D[n])\times H_P)+ H_A\ ;\ \ H_{new} = [H_{new}^{\ R}, H_{new}^{\ G}, H_{new}^{\ B}]^T} \tag{14}$$

The average result of the net outcome of the above calculation and the gaussian model method is 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.

5. ***Shadow Removal and other Processing:*** 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.

The algorithmic chart is shown in Fig. 6

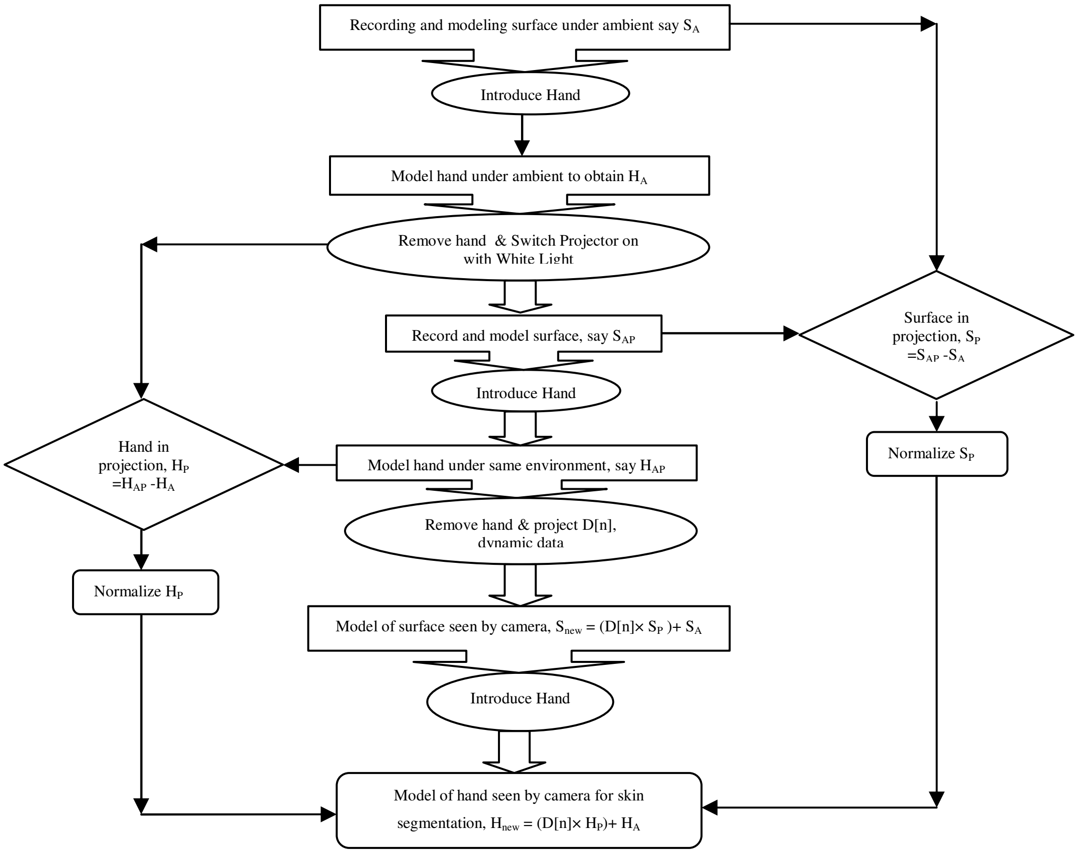
Recording and modeling surface under ambient say S_A
Introduce Hand
Model hand under ambient to obtain H_A
Remove hand & Switch Projector on with White Light
Record and model surface, say S_{AP}
Surface in projection, $S_P = S_{AP} - S_A$
Introduce Hand
Hand in projection, $H_P = H_{AP} - H_A$
Model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

Figure 6: Algorithmic representation of Intrusion detection

The output of intrusion detection is shown in Fig. 7:

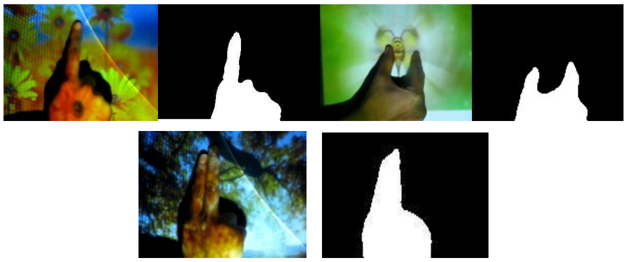

Figure 7: Output of Intrusion detection

4.2 Gesture Detection

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 Fig. 8

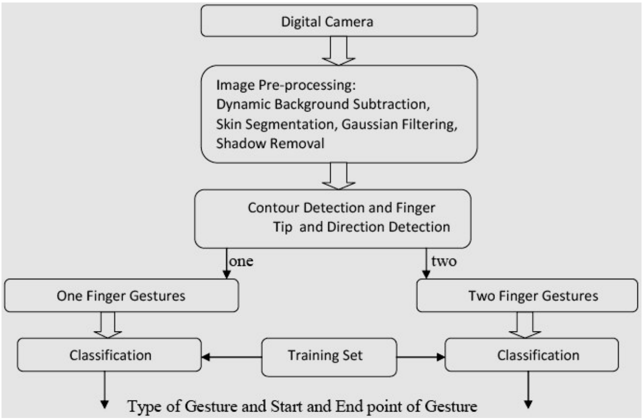
Digital Camera
Image Pre-processing:
Dynamic Background Subtraction,
Skin Segmentation, Gaussian Filtering,
Shadow Removal
Contour Detection and Finger
Tip and Direction Detection
one
two
One Finger Gestures
Two Finger Gestures
Classification
Training Set
Classification
Type of Gesture and Start and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

Figure 8: Flowchart representation of Reflectance Modeling method

i) Contour detection of hand which is represented by sequences in which every entry in the sequence encodes information about the location of the next point on the curve

ii) Curvature mapping: The curvature of a smooth curve is defined as the curvature of its osculating circle at each point. Calculation of curvature at each point in the contour by applying the usual formula, along with detection of corner points by computing second derivatives using Sobel operators and finding eigen values from the autocorrelation function obtained. Using the first method of curvature, we apply the usual formula for signed curvature(k):

$$k = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}}. \tag{15}$$

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

iii) Positive curvature extrema extraction on contour or Determining the highest positive corner points: This is done by two methods:
After finding out the maximum positive peaks we find the corner points by computing second derivatives. In case of more than one positive curvature points of almost equivalent magnitude of curvature, we classify the gesture to be multiple fingers. The two methods are applied jointly upon each frame, because it was found that corner detection alone produced many false positives. Segregating the gestures into Single finger gestures like Click, Rotate (Clockwise and Anticlockwise), Move arbitrary and Pan and Multiple finger gestures: Zoom (Zoom-In and Zoom-Out), Drag

iv) Frame to frame fingertip tracking by using motion model estimation: The trajectory, direction evolution, starts and end points of each finger in the gesture performed is traced through the frames. 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 detected point lies in the range defined by calculation of movement in preceding frames. Tracking motion feedback is used to handle errors.
Let at $\boldsymbol{t = 0}$, the coordinate of the corner or finger tip may be $\boldsymbol{(x_0, y_0)}$ ***and*** at $\boldsymbol{t = 1}$, the position coordinate be $\boldsymbol{(x_1, y_1)}$
Then at $\boldsymbol{t = 2,}$ or in the next frame, the coordinate of the same finger tip becomes $\boldsymbol{(x_2,y_2)}$
Vertical and Horizontal velocity,

$$\boldsymbol{y_2' = y_2 - y_1,\ x_2' = x_2 - x_1} \tag{16}$$

Vertical and Horizontal acceleration,

$$\boldsymbol{y_2'' = y_2' - y_1',\ x_2'' = x_2' - x_1'} \tag{17}$$

Hence by applying the model above we can predict the corner in the subsequent frame can be predicted to be:

$$\mathbf{x_{predicted} = x_2'' + x_1' + x_1;\ y_{predicted} = y_2'' + y_1' + y_1} \tag{18}$$

v) Gesture classification and quantification: The final classification and subsequent gesture quantification is performed on the basis of the following data which is represented diagrammatically in Fig. 9.
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 $\boldsymbol{t}$ the coordinates of finger tip are $\boldsymbol{(x,y)}$ and Then at some time $\boldsymbol{t + k}$ the coordinates are $\boldsymbol{(x',y')}$

$$a = \frac{y' - y}{x' - x} \quad b = \frac{x' - x}{y' - y} \tag{19}$$

where ‘a’ and ‘b’ represents the slope and reverse slope respectively

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

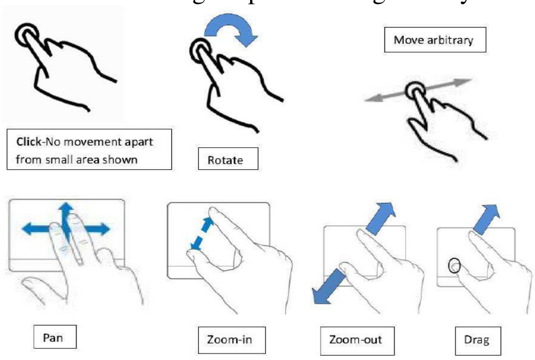

Figure 9: Gesture Classification Criteria

The detailed description of each gesture is given in the tables 1 and 2:

Table 1: Single finger gestures

No	Gesture	Meaning	Signing Mode
1	Click	It is derived from normal clicking action as we do on mouse of PC’s or laptop so as to open something	Tapping index finger on the surface. The position on thing specifies the action location
2	Move (a)Frame	Drawing rectangular region to focus on something or defining view for snapshot	Drawing rectangle on the surface with index finger
	(b)Move	Move in random directions from current position	Move index finger in arbitrary direction on the surface
3	Rotate (a)Anti-Clockwise	Rotating object in Anti-clockwise direction like taking turn	Complete or incomplete circle is drawn with index finger in Clockwise and Anti clockwise direction
	(b)Clockwise	Rotating an object in clockwise direction	
4	Pan	Movement of object or window from one place to another	Index and middle finger together moving in arbitrary direction

Table 2: Multiple finger gestures

No.	Gesture	Meaning	Signing mode
1	Drag	It signifies movement of window or object in one direction	Enacted by fixed thumb and arbitrary movement of index finger
2	Zoom (a)Zoom-in	Increase in size of window or object	Move index finger and thumb away from each other
	(b)Zoom-out	Decrease in size of window or object	Move index finger and thumb away from each other

5. RESULTS

First a clean Binary image of hand is obtained using the method of reflectance modeling, then gesture detection can be achieved easily by applying the algorithm explained above. Specifically, the system can track the tip positions of the fingers to classify the gestures and find out their attributes. The figure shows the detection of contour of hand and tip of finger(s) in dynamic projection on arbitrary background which is followed by tracking the trajectories, velocities and direction of the movement thereby classifying the gestures. These positions depict the commonly held positions of hand, common to all gestures.

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 reflectance between the naked 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 modeling for skin detection gives good results

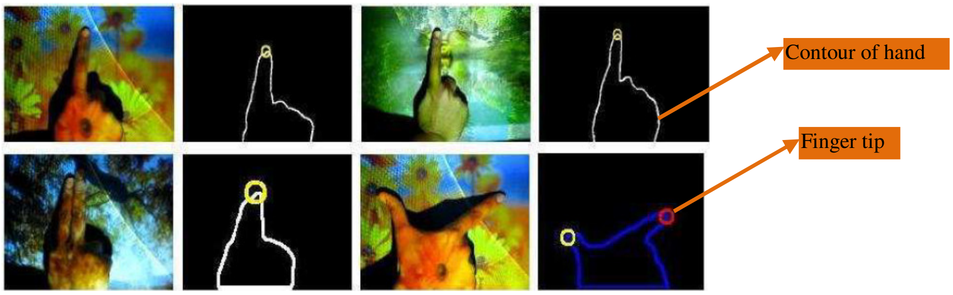

Figure 10: Shows detection of contour and finger tip for single and multiple finger gesture on arbitrary backgound

6. CONCLUSION

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 and cheaper than any of the existing systems which requires hardware markers and huge and costly setups.

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, 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.

One of the key applications is ***Paper Touch-pad*****:** 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 setup and the layout of paper touch pad is shown in Fig. 11 along with the left click operation. The red dots on the corner of the printout of the touchpad are used for homographic mapping.

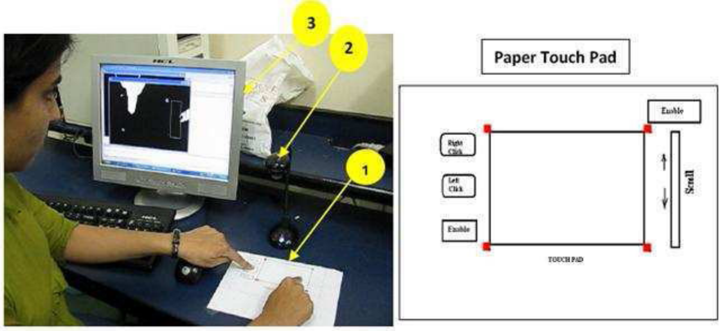

Figure 11: Paper touch pad setup on left and the printed touchpad on sheet of paper on right. 1.Paper touchpad, 2. Webcamera just above the paper touchpad and 3. Monitor which is mapped to the touchpad

7. REFRENCES

1. Gordon G., Darrell T., Harville M., Woodfill(1999).: Background Estimation and Removal Based on Range and Color.
2. Grossman T., Balakrishnan R., Kurtenbach G. Fitzmaurice, G. Khan, A. Buxton (2001): Interaction techniques for 3D modeling on large displays. In: Proceedings of the symposium on Interactive 3D graphics, pp. 1723
3. Han-Hong L. and Teng-Wen C.: A Camera Based Multitouch Interface Builder for Designers
4. Han, J.Y.: Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In: Proceedings of the 18thAnnual ACM Symposium on User Interface Software and Technology
5. Helman. S, Juan. W. and Leonel .V, Aderito M.(2007) Proceedings of GW - 7th International Workshop on Gesture in Human-Computer Interaction and Simulation
6. M. Jones, J. Rehg. (1999) Statistical color models with application to skin detection. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1
7. Pranav M., Pattie M. and Liyan C.,(2007) WUW - Wear Ur World - A Wearable Gestural Interface
8. Ramon H., Daniel N. and Andreas K.: FLATIR: FTIR Multi touch detection on a Discrete Distributed sensor Array
9. Ray Lockton, Oxford University,Hand Gesture Recognition using special glove and wrist band.
10. Song-Gook Kim, Jang-Woon Kim and Chil-Woo Lee, Implementation of Multi-touch Tabletop display for HCI
11. Thomas M.(1994). Finger Mouse: A freehand computer pointing interface. Doctoral thesis- The University of Illionois
12. V. Pavlovic, R. Sharma, and T. Huang, Visual Interpretation of Hand Gestures for HCI: IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, , pp. 677-695.
13.Vladimir. I, Rajeev. S, Thomas. S(1993).Visual Interpretation of Hand gestures for HCI: A review, University of Illionois
14.Wayne Westerman, John G. Elias and Alan Hedge, A Multi touch :A new tactile 2-D gesture interface for HCI
15.Wren, C., Azarbayejani, A., Darrell, T. (1997): Pfinder: Real-Time Tracking of the Human Body. IEEE Transactions on Pattern Analysis and Machine Intelligence.