

Gestures in scheming cursors for camera and projector in touch screens

¹Swapan Debbarma, ²Agnivesh Pandey,

^{1,2}Assistant Professor,

¹Computer Science & Engineering Department,

¹NIT Agartala, India,

²Department of information technology, School of studies in engineering and technology,

²Guru Ghasidas Vishwavidyalaya, Koni bilaspur, Chhattisgarh, India

Abstract: Virtual Touch Screen, a device that has an interactive display projected into the air or onto an ordinary solid surface, is a system that provides us a graphical work-field for controlling specific kind of operations. To develop the augmented reality with the portable environment, human-computer interaction interface satisfying the mobility, ease and accuracy is required. Unlike a physical touch screen, which requires pressure to activate its sensors, the virtual touch screen relies only on monitoring the position of the hands and fingers on some projected area. For the formation of a Virtual Touch Screen (VTS), an ordinary screen can be transformed using a webcam which takes the picture of a pointer (finger tips with LEDs). The touch screen then can be simulated by locating the pointer's tip position on the image to the position on the screen. According to the pointer's tip movement on the image, the cursors then moves on the projection surface and as indicated by the gesture of the pointer, different cursors events take place. Therefore, this paper focuses on preparing a model which, I) detects the gestures formed by the indicator movement and II) establishes communication with the operating system to control cursors event. Simulation results', using the above proposed model, verifies the suitability of choosing such kind of gesture detection and implementation in this paper.

Index Terms – Virtual Touch Screen; augmented reality; human computer interaction; gestures; cursors event.

I. INTRODUCTION

VTS is an unconventional low cost solution for the real touch screen and an achievement of Augmented Reality [1]. This leads to intensive research in the area of human computer interaction (HCI). In the proposed paper, VTS system comprises of a PC, Web Camera, LED lights (3 no's) and a Projector. The projector will project the computer screen on a wall or a regular screen, where the user will perform the pre-defined gestures. The camera will capture the picture [2] of the regular screen on which gestures are made. This captured moving picture will feed into the computer. The computer will receive the captured picture as frame by frame, analyze those frames, and if gestures are found to be valid, then the corresponding actions will be performed in the computer [3]. In this paper, we have defined four gestures to control five cursors events, by which different operations in computer will be done. The cursors events are:

- (i) Cursors Move
- (ii) Right Click
- (iii) Single Left Click
- (iv) Double Left Click
- (v) Cursors Drag & Drop

II. BACKGROUND

From education to corporate field and from training rooms to broadcasting studios, Interactive Whiteboards are massively used instead of normal white or black boards. An interactive whiteboard (IWB) is a large display that is connected to a computer and projector but to facilitate it we need a lot of sensors mounted on the white board, which may be very expensive, depending on the size of the board. Therefore, a low cost answer to this problem is VTS. In this system, by capturing the image of the projected regular screen, one can get all the facilities of an IWB in a cost effective way [4]. The projected surface can be controlled by LED's attached with the hand. The webcam used in the setup helps to recognize the different hand gestures as input.

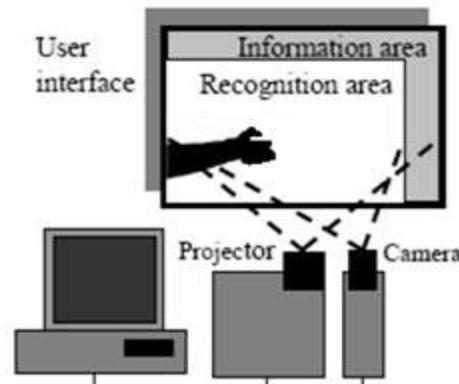


Fig. 1: System Overview

III. OBJECTIVE OF THE SYSTEM

The proposed system focuses on two things. They are

- (i) Gesture detection and
- (ii) Establishing Communication with the OS to control cursors events.

In this paper, four different gestures^[5] and one mixed gesture (combination of two gestures), total five gestures is predefined. Anything else except those will be treated as invalid. These gestures actually denote five different cursors events. So, detecting the validity of the gestures^[6] is an important task. After mapping the coordinates of cursors pointer, we need to detect the gesture in order to know what cursors event should be called at the required position of the cursors pointer. It is only possible by detecting the valid gesture.

After the valid gesture detection, interaction with the OS through system functions is done to control the cursors movement and different cursors events.

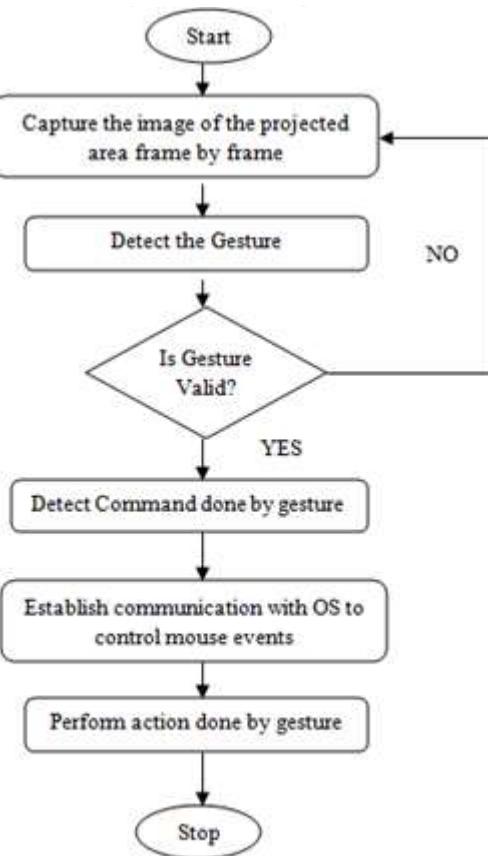


Fig. 2: Flow Chart of the System

IV. CAPTURING FRAMES

For detecting the gestures we need to capture the image of the projected area frame by frame and have to analyze those frames. The maximum frame rate of used 2MP camera is up to 30 fps but the processing speed of our program is up to 11 fps. So we need to cut short the capturing of frames until the processing of current frame is done. This halt process is shown in the diagram:

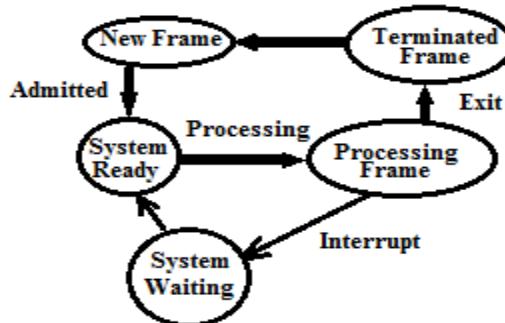


Fig. 3: Frame Capturing and Analysis

V. GESTURE RECOGNITION

A. What is Gesture?

A **gesture** is a form of non-verbal communication in which visible bodily actions communicate particular messages, either in place of speech or together and in parallel with spoken words. Gestures include movement of the hands, face or other parts of the body. Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs, which still limit the majority of input to keyboard and cursors. Gesture recognition can be conducted with techniques from computer vision and image processing. Here we have conducted the gesture recognition from image processing.

B. Define gesture:

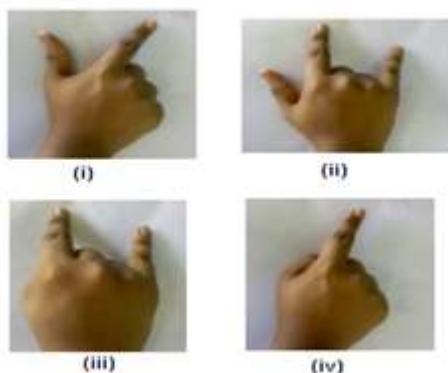


Fig. 4: Different Gestures

Here gesture (i) denotes single left cursors click, (ii) denotes double left cursors click, (iii) denotes single right cursors click, (iv) denotes cursors move. Cursors drag & drop event will be a combination of gesture (i) & (iv). As we are using only 3 numbers of LEDs so only these four gestures are required to depict the desired five cursors events.

C. Implementation of defined gestures

For implementing the defined gestures [7], we have to identify the three fingers in our hand used in the gestures defined. For this we have a few options like,
 (i) Using color bands in fingers,
 (ii) Using skin color detection,
 (iii) Using LED lights.

Now, in case of using the color bands in fingers, there may be ambiguity in detecting the finger from the background screen. As the background will be a projection of a computer screen, there is a possibility that in some cases the color of the projected screen matches with the color of the bands in fingers; so it will be difficult to locate the fingers in the screen.

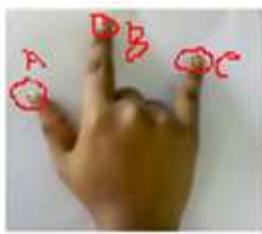
For skin color detection, as the color of human hands has a wide range, we have to maintain a database for that color and we have to search the entire screen for human hands on the basis of color and again we can find the problem, faced in color band method.

We can use normal white LED lights because our experimental result shows that using colorful lights we cannot achieve the brightest point i.e. a point with highest luminous value (255). Therefore, we have selected the LEDs for detecting fingers on screen.

D. Detecting Valid gestures

As brightest point method is used, we have to fix any of the fingers with respect to which we will identify the rest two and the valid gestures. In our case, we have fixed the index finger as it is common in all the four gestures. At the beginning user should give any gesture like gesture (ii) which will be recorded in the program for further use. When other fingers (e.g. thumb or small finger) are used, while capturing the video of the projected screen frame by the co-ordinate analysis with different frames, index finger can be found out.

Now after identifying the index finger in gesture, we can use the x-y co-ordinate analysis to detect which other finger has been used along with index finger and finally the gesture can be detected by following steps:

**Fig. 5: Finger Positions**

- (i) Let A (x_1, y_1), B (x_2, y_2), C (x_3, y_3) are having the coordinates.
- (ii) In our screen x increases in right hand side from top left corner. So x_1, x_2, x_3 are related by:

$$x_1 < x_2 < x_3$$

- (iii) So by comparing the x- coordinate we can distinguish the fingers.

As index finger is present in all gestures it will represent the cursors cursors and the cursors movement will be done according to the movement of the index finger. So gesture (iv) will represent cursors move. Gesture (ii) and (iii) will represent double left click and right click. Cursors drag and drop and single left click events will be done using the combination of gesture (i) and (iv).

VI. COMMUNICATION WITH THE OS

How OS handles cursors events

In Operating system, right click occurs when right cursors button down and up events occur. The time between them is not considered. For left cursors click it is same as right cursors click but the difference is here we have to press left cursors button. In left double click three time gaps are counted. The time gap between first cursors down and first cursors up, the time between first cursors up and second cursors down, the time between second cursors down and second cursors up. Here all three time intervals are predefined in operating system.

For cursors drag it is the combination of cursors move and left cursors button down.

Finding the relative position of cursors

After the valid gesture is detected, action should be performed by the detected valid gesture. We need to communicate with the operating system to control the cursors cursors and its different events. So we necessitate finding the corresponding point of cursors cursors in the computer screen with respect to the index finger in the projection screen.

For this purpose, we have used bilinear transformations [8]. Suppose, the point in which index finger is present on the projected screen at any frame is (x, y) and the corresponding position of cursors cursors in the computer screen is (x_1, y_1), then these two points are related by,

$$x_1 = a_0 + a_1x + a_2y + a_3xy$$

$$y_1 = b_0 + b_1x + b_2y + b_3xy$$

Here $a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$ are all constants, but they vary from system to system. Now we can find the values of these eight constants by following steps:

Step 1:

Display a window formed with 4 red shapes drawn at the 4 corners of the form covering the whole screen.

Step 2:

Capture the frame.

Step 3:

After scanning the whole image detect the 4 corners by color detection method. These are the (x, y) values.

Step 4:

For all 4 pair of co-ordinates there will be total 8 equations. Solving them we will get the coordinate values of $a_0, a_1 \dots b_3$.

Step 5:

Using Cramer's rule [3] we can find out the values of the constants.

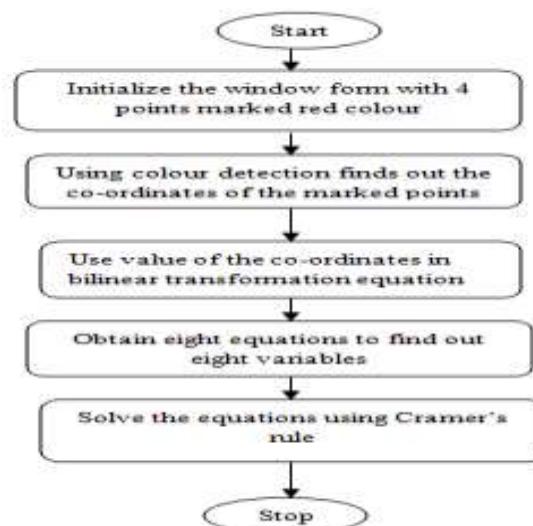


Fig. 6: Flow Chart for finding Constants

After having the values, we can detect the valid gestures on the projected screen and also we can find the corresponding coordinates in computer screen. After that interfacing with OS [9], we can move the cursors in the computer in accordance to the hand gestures in projection area.

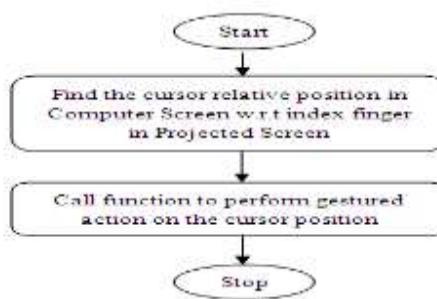


Fig. 7: Flow Chart for OS handling

VII. SIMULATION RESULTS

The following simulation results show the validity of the proposed model. The simulations in the given figures are generated after running the total proposed model using visual basic software.

Below in Fig. 8, four Red shapes are made in four corners of the screen to cover the whole screen, projected by the projector and captured by the webcam.



Fig. 8: Before Corner Detection

Below in Fig. 9, screen shot is taken after corner detection. After scanning the whole image , corners are detected by color detection method.

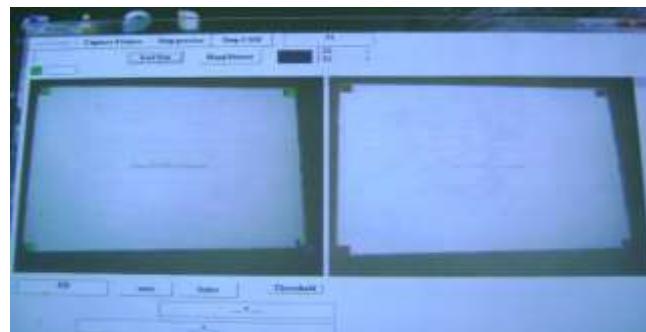


Fig. 9: Corner Detected

Below in Fig. 10, screen shot is taken after a gesture detection. From the projected screen, the gesture for left double click cursors event is detected.

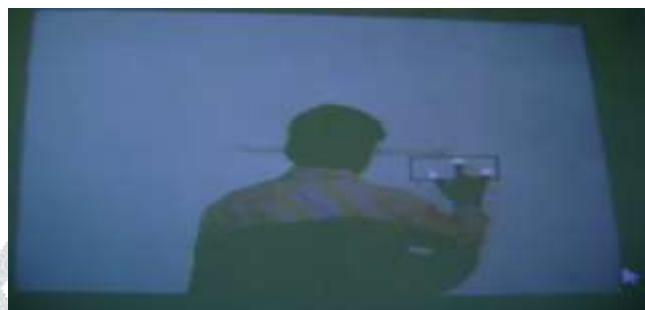


Fig. 10: Gesture Detected



Fig. 11: Cursors Move

In Fig. 11, screen shot is taken for the cursors move gesture. It is done using only the index finger having a LED attached with the finger.

VIII. CONCLUSION

Virtual Touch Screen can provide a new, user-friendly, fast, accurate and cost effective multipurpose presentation tool. It is cost effective because it uses a normal web camera and a projector which is cheaper than the sophisticated sensors mounted in a touch screen. Two most important contributions of the proposed paper are given as:

Firstly, Identification of different types of valid Gestures has given the system an ease which can be managed via capturing the moving image of the finger tips by a simple web camera.

Secondly, realization of the gestures to manage the Cursors Events by interfacing with the OS for implementation of VTS offers the natural and comfortable HCI interface by not using any uncomfortable devices for interacting. Because the system recognizes the color, shape and movement of the hand, the users do not need any additional devices.

Our current systems overcome all the short-comings of all the former systems and give the user a new way to manipulate and experience the operating system on the projected surface. The simulation results also show the efficiency and correctness of the system.

References

- H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual object manipulation on a table-top AR environment," *Augmented Reality, ISAR 2000*, pp. 111-119, 2000.
- Z. Zhang, "Vision-based Interaction with Fingers and Papers," *Proc. International Symposium on the CREST Digital Archiving Project*, pp. 83-106, May. 2003.
- Eunjin Koh, Jongho Won, Changseok Bae, "Vision-based Virtual Touch Screen Interface", *Consumer Electronics, 2008. ICCE 2008*.
- Somkiat Wangsiripitak, "Cursors Position Estimation Model for Virtual Touch Screen Using Camera", *World Academy of Science, Engineering & Technology 5*, 2005

- C. Maggioni and B. Kammerer, "Gesture computer – history, design and applications, in Ed. R. Cipolla and A. Pentland, editors," *Computer Vision for Human-Machine Interaction*, Cambridge University Press, 1998.
- Anala Pandit et al, "A Simple Wearable Hand Gesture Recognition Device using iMEMS", 2009 International Conference of Soft Computing and Pattern Recognition
- Andrew D. Wilson, "TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction", *ICMI'04*, , State College, Pennsylvania, USA, October 13–15, 2004
- M. S. Floater, "one to one piecewise linear mapping over triangulations", mathematics of computations, vol 72, No. 242, Pg – 685 – 696, October, 2002.
- J. M. S. Dins, P. Santos, and P. Nande, "In your hand computing: tangible interfaces for mixed reality," IEEE International Conference on Augmented Reality Toolkit Workshop, pp. 29-31, 2003.

