# SmartTouch: A Cost-effective Infrared based Imaging Touch Screen

Shubhansh Varshney, Niharika Jain, Sandeep Singh
CSE and IT Department
Jaypee Institute of Information Technology
Noida, India
shubhansh12@gmail.com

***Abstract*— Current wireless human-computer interaction devices such as wireless mice and touch screens, by-and large, incorporate a sophisticated electronic architecture. The sophistication achieves wireless capabilities but carries over a cost overhead. In this paper, foundation is laid for developing a novel human-computer interaction device with reduced hardware sophistication as well as reduced cost. SmartTouch is an infrared based touch screen device for desktops that uses a webcam and infrared LEDs. SmartTouch transforms an otherwise normal LCD computer screen into an input/output surface suitable for gesture-based interaction wherein users can naturally use their fingers or stylus to issue commands and type text. To increase its utility beyond simple touch it has also been empowered with multi-touch interaction.**

***Keywords—Computer vision; gesture recognition; computer human interaction; virtual touch screen; multi-touch interaction;***

## I. Introduction

Virtual Touch Screens are call for the day and are continuously becoming popular. There are varieties of approaches proposed in literature to design touch screens. These approaches incorporate a sophisticated electronic architecture. The sophistication achieves wireless capabilities but carries over a cost overhead. One of the existing approaches named as Diamond Touch System [1], sense the capacitive coupled touch of the user's hand using horizontal and vertical rows of electrodes. The hardware cost of Diamond Touch System's is very high and it also requires separate table for implementing touch functionality. Similarly, HoloWall[2], Designer's Outpost [3] and MetaDesk [4] use video cameras and computer vision techniques to compute a touch image. These systems permit simultaneous video projection and surface sensing by using a diffusing screen material. The limitation with these approaches is that they only resolve those objects that are on or very near the diffusing surface. Further, in these proposed techniques [2-4], cameras need to be placed behind the screen which makes them unsuitable for desktop use. Designer's Outpost identifies sticky notes and its movements along with text written on it through high resolution video camera. MetaDesk identifies physical objects and their movements on Desk using video camera and infra-red camera. In these applications, restriction is to place cameras on the rear side of the screen and uses diffuser coating to prevent capturing images of objects at a distance more than 10 cms. Both these approaches work on large screens/walls on which images are projected using projector. Another approach by name uTouch [5], uses two cameras and skin detection algorithm to locate point of touch. This approach fails when background contains color similar to skin color.

All these limitations in the related approaches as well as the objective to reduce cost of hardware has motivated us to develop SmartTouch in order to construct cost-effective and computationally efficient virtual desktop based touch screen that overcome shortcomings of already proposed related techniques.

SmartTouch is for LCD screens. In SmartTouch, unlike above approaches, no coating is used and camera is placed in-front of the screen. It aims at transforming normal LCD screens (output devices) to touch screens (Input/Output device). This technique uses infrared webcam and an array of infrared Light Emitting Diodes (LEDs) to implement the proposed design. Array of LEDs emits infrared rays which are used to detect touch on screen through infrared camera. A normal webcam has been transformed to infrared webcam in order to provide a cost effective solution. SmartTouch is scalable, versatile and affordable.

SmartTouch System analyses images from infrared camera, finds the point of touch, calculates the pixels corresponding to the touch on screen area, recognizes gesture and sends system commands to the Operating System (OS). Simple hardware design along with efficient algorithm to detect gestures has made SmartTouch computationally efficient. Users can naturally use their fingers or other tip pointers to issue commands and type texts. SmartTouch is also enabled to recognize multiple gesture functionalities like rotate, zoom, scroll up and down, etc. This has further enhanced the degree of interaction.

The rest of the paper is organized as follows: section II describes the hardware setup required to build the SmartTouch, section III describes the entire functioning and design of SmartTouch and finally paper concludes with applications of SmartTouch, its comparative advantages with other designs and its scope for further improvement.

## II. Hardware Setup

SmartTouch setup is designed to achieve maximum efficiency at minimum hardware requirements, thus making its approach cost effective. The technique consists of an infra-red optical sensor (typically a camera), infrared light source and visual feedback in the form of desktop LCD screen.

*A. Array of Infrared LEDs*

Infrared light is used to distinguish between a visual image on the touch surface and the object(s)/finger(s) being tracked. Infrared LEDs are efficient and effective at providing infrared lights. Wavelength of IR LEDs ranges from 780 nm to 940 nm and hence light from IR-LEDs are easily seen by most IR- web cameras. Array of seven Infrared LEDs are equidistantly arranged on the cardboard with parallel connection and an eighteen ohm, two watt resistor connected in series. The array of LEDs is connected to a switch to enable it to be switched on/off. To avoid addition of any other power source, LEDs are USB powered. Array of LEDs are placed below the screen area to emit infrared rays parallel to screen.

*B. Transformation of Webcam to Infra-red Webcam*

'Near Infrared' (NIR) is the lower end of the infrared light spectrum and typically consider wavelengths between 700 nm to 1000 nm. Most digital web camera sensors are also sensitive to at least NIR and are often fitted with a filter to remove that part of the spectrum so they only capture the visible light spectrum. A normal webcam has been transformed to infrared webcam in order to provide a cost effective solution. In the transformation, the infrared filter is removed from the normal webcam and replaced with exposed photographic film. The film removes the visible light instead of infrared light. This transforms a normal web camera to an infrared web camera that only captures infrared light. This modified infrared web camera is mounted in front of a computer screen such that camera can see the entire computer screen. It has no positioning constraints as described in [1], as far as it can see the entire screen. The web camera used in our approach has the following technical specifications:

**Resolution:** The resolution of the camera is very important. The higher the resolution the more pixels are available to detect finger or objects in the camera image and hence higher is the accuracy. While designing this approach, screen resolution was 1024x768 and camera resolution was 640x480. Application is designed to handle different camera and screen resolutions. Screen resolution is obtained using APIs whereas camera resolution is obtained using dimension of images captured.

**Frame rate:** The frame rate is the number of frames a camera can take within one second. More snapshots means more data of what happened in a specific timestamp. In order to cope with fast movements and responsiveness of the system a camera with at least a frame rate of 30 frames per second (FPS) is recommended. Frame rate used for testing the SmartTouch device is between 10 - 30 fps.

## III. SmartTouch Functioning

In this section functioning of SmartTouch System is described. Activity diagram describing the entire functionality of SmartTouch is shown in Figure 7. Details of each major activity are described below.

*A. Selection of screen area*

Selection of screen area requires camera to be placed at a point from where the entire screen is visible. While selecting screen area, care is taken that IR LEDs are switched off initially. Screen area is selected by first preprocessing the images from IR webcam. Preprocessing of images includes conversion of images to grayscale and finding contours. Contours are the set of points with color intensity equal or more than a threshold value. Threshold is adjusted manually using track-bar as shown in Figure 1.

LCDs have the property that they emit IR rays of very low intensity. These IR rays from the LCD screen are captured by the IR webcam. User only needs to set threshold and precision values through the track-bar so that complete screen area is covered by the contours. Contours are displayed on the screen through red lines and corners are marked green. Figure 1 shows the selection of screen area.

In our case, threshold value is the value of color intensity (in the range from 0 to 255 for gray scale image) such that pixels with color intensity greater or equal to threshold value are included in formation of contours. Boundary of points forming contour is then drawn on screen to show contours. Precision is "level of accuracy". While finding contours, if precision is high (say maximum value = 100), then contour will be formed with the set of points with color intensity exactly equal to threshold value, while if precision is low(say minimum value = 0), then contour will include nearly all points in a contour. So it is required to fix the approximate value close to which complete screen is captured in a single rectangular contour. Threshold value in application can be adjusted manually using track-bar as can be seen in screen-shot image Figure 1.

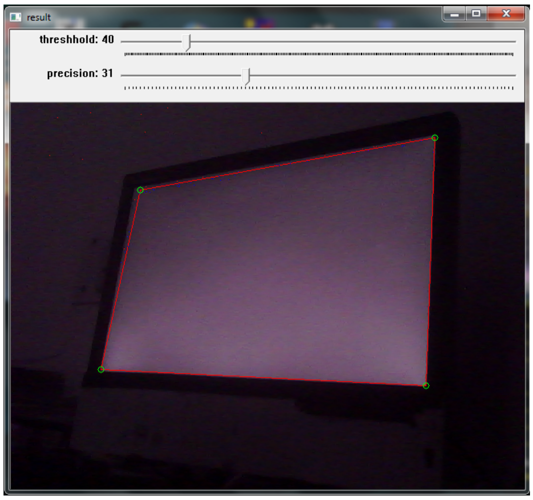


**Figure 1 Selection of screen area**

*B. Analysis of images from infrared webcam*

Homography is used to extract the plane of interest (screen area) as an image. Figure 2 shows homography applied on a webcam image. Threshold for further processing of contours is set automatically by the system. This is done by initializing the threshold to 100 units and making increment of 5 units per iteration until contour is encountered on the screen. In case of no contour on the screen with given threshold value, iteration terminates and threshold is increased by marginal value of 23. IR LEDs are switched on during this process.

Images captured from the IR webcam are converted to grayscale and analyzed for the presence of stylus (or finger) close to screen. Presence of stylus (or finger) is observed in the system by the presence of contour with given threshold on the screen. Contour is formed when stylus (or finger) is brought close to the screen due to IR illumination which crosses the threshold value.

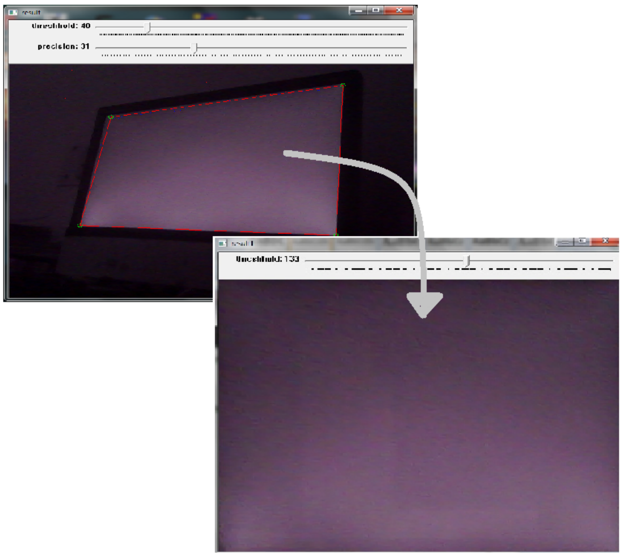

**Figure 2 Homography applied area of interest**

*C. Finding point of touch*

On finding contour in the area of interest, i.e. screen area, right most top point is chosen from the set of points in the contour because (in experiments) camera is placed to the right of the screen below screen level. If there is stylus (or finger) close to the screen, it signals the SmartTouch device that user will perform normal mouse functionalities. Normal mouse functionalities include left click, right click, drag & drop, scroll up, scroll down. Figure 3 shows illumination of stylus when brought close to the screen area. By calculating the point of touch further calculations are carried out.

If there are multiple fingers close to the screen, they also illuminate and separate contours will be observed in the image from IR webcam. Multiple contours signal the SmartTouch device that user will perform multi-gesture functionality. Thus, processing related to multi-gestures is performed. Multi-gesture functionality includes that of rotate clockwise, rotate anti-clockwise, zoom in and zoom out.

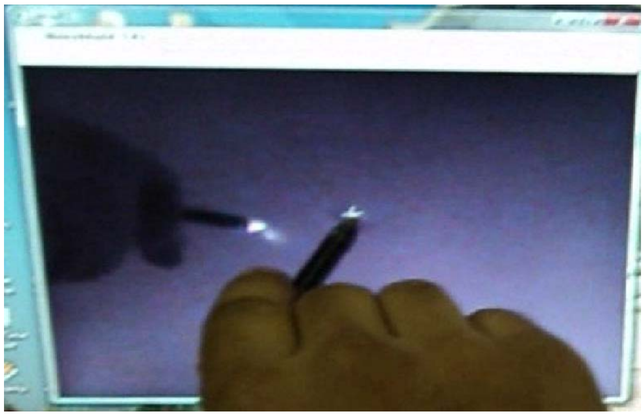

**Figure 3 Stylus illumination when brought close to screen**

*D. Calculation of pixels corresponding to touch on screen area*

The Resolution of images from webcam is different from that of screen, so pixels corresponding to screen needs to be mapped. Formulae given in (1) and (2) are applied to map image coordinates to screen coordinates:

$$screen_x = point_x * screen_x / image_x \quad (1)$$

$$screen_y = point_y * screen_y / image_y \quad (2)$$

where $screen_x$ & $screen_y$ are screen's x-y coordinates, $point_x$ & $point_y$ are touch coordinates with respect to image from IR webcam, imagex & imagey are image x-y resolution and $screen_x$ & $screen_y$ are screen's x-y resolution.

*E. Gesture recognition in the SmartTouch and its Testing*

**Performing mouse functionalities:** Normal mouse functionalities like right/left click, double click, scroll up, scroll down and drag & drop can be easily performed using SmartTouch. Above mentioned functionalities are implemented by taking following variables into account:

a. contourPresentFlag: default is false, if true then signals that contour is present in the screen.

b. time: default to 0, records the time since leftClickFlag is true.

c. initialPosition: records x-y coordinates of the point where contour was initially found when contourPresentFlag is set to true.

d. finalPosition: records x-y coordinates of the point where contour was removed.

e. changeInPostion: difference in finalPosition and initialPosition

For all the mouse functionalities contourPresentFlag has to be true. Based on the variable states, left/right click pressed/release signals are sent to OS for implementation. Table 1 shows state of the variables for normal mouse functionalities.

**Table 1. Variable state for normal mouse functionality**

| Mouse Functionality | time | changeInPosition | |
|---|---|---|---|
| | | Δx(px) | Δy(px) |
| Left Click | < 1s | ~ 0 | ~ 0 |
| Right Click | > 1s | ~ 0 | ~ 0 |
| Scroll Up | - | ~ 0 | < 20 |
| Scroll down | - | ~ 0 | > 20 |
| Drag & drop | - | ≠ 0 | ≠ 0 |

Double click functionality can be achieved using right click and selecting first option in the menu. Figure 4 and 5 shows the left and right click functionality respectively.

**Figure 4 Left click functionality**

**Figure 5 Right click functionality**

Drag and drop functionality is performed when stylus (or finger) moves from its initial position of entering the screen area then left click pressed command is sent to OS, while left click released command is sent when finger/stylus is removed from the screen area, i.e. when there are no more contours in the image. Figure 6 shows implementation of drag and drop functionality.

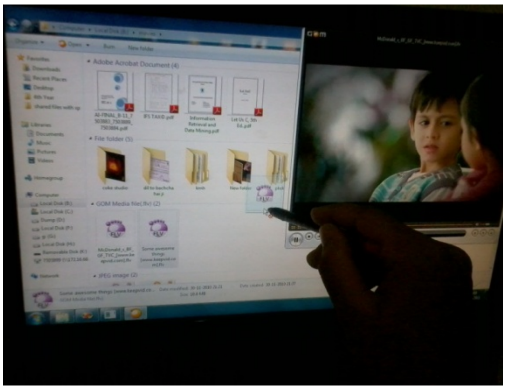

**Figure 6 Drag and drop functionality**

If stylus (or finger) moves vertically upwards (or downwards) from its initial position of entering the screen area then scroll up (or down) functionality is performed by sending arrow key up/down command to OS.

**Processing of multiple contours in the image:** In case of multiple contours, multi-gesture processing is done. Each contour is tracked and is matched with arithmetic templates rather than traditional pattern templates. Here arithmetic templates are used in the sense that multi-gesture is recognized by calculating change in distance between multiple contours and slope formed by them.

Table 2 shows for each of the three gestures, value of changes in the angle (or slope) formed by multiple contours and the respective changes in their distance. For example from Table 1, if change in distance between contours is found to be less than 25 pixels and change in angle(or slope) is less than 10 degrees then Zoom-in gesture is identified. Similarly, for rotate clockwise (or anticlockwise), if change in distance between the contours is less than 15 pixels and change in angle is more than 35 degrees clockwise (or anticlockwise) then rotate clockwise (or anticlockwise) gesture is identified and respective commands are sent to OS.

**Table 2 Arithematic templates for multiple contours**

| Gesture | Change in angle | Change in distance |
|---|---|---|
| Zoom-in | < 10° | < 25 pixels |
| Zoom-out | < 10° | > 25 pixels |
| Rotate anti/clockwise | > 35° | < 15 pixels |

**Testing SmartTouch approach:** SmartTouch gesture-based interactions have been tested on various application-software like Adobe Reader, Microsoft Paint etc. The objective was to test both single gesture and multiple gestures wherein users can naturally use their fingers or stylus to issue commands and type text. Details of features tested are presented in Table 2.

**Table 2 SmartTouch testing on application software**

| Application Software | Tested Features | |
|---|---|---|
| | Simple Touch Features | Multi Touch/Gesture Based Features |
| Adobe Reader | Selection, Right Click, Left Click | Scroll Up, Scroll Down, Zoom in, Zoom Out, Scroll Left/Right |
| MS-Paint | Draw, Right Click, Left Click, Selection of tools, Use of tools | Scroll Up, Scroll Down, Zoom in, Zoom Out |
| Google Chrome/ Internet Explorer/Mozilla Firefox | Selection, Right Click, Left Click, Click on Link, Drag & Drop | Scroll Up, Scroll Down, Zoom in, Zoom Out |
| Windows Photo Viewer | Right Click, Left Click | Scroll Up, Scroll Down, Zoom in, Zoom Out, Rotate Clockwise/Anti Clockwise, Swipe (for changing pics forward and backward) |
| MS-Office (MS-Excel, MS-Word) | Right Click, Left Click, Selection | Scroll Up, Scroll Down, Zoom in, Zoom Out |
| VLC Media Player | Right Click, Left Click, Drag & Drop | |
| KM Player | Right Click, Left Click, Drag & Drop | Movie Forward/Backward, Volume Up/Down |
| Desktop | Selection, Right Click, Left Click, Drag & Drop | |
| Windows Explorer | Left Click, Right Click, Drag & Drop, Selection | |

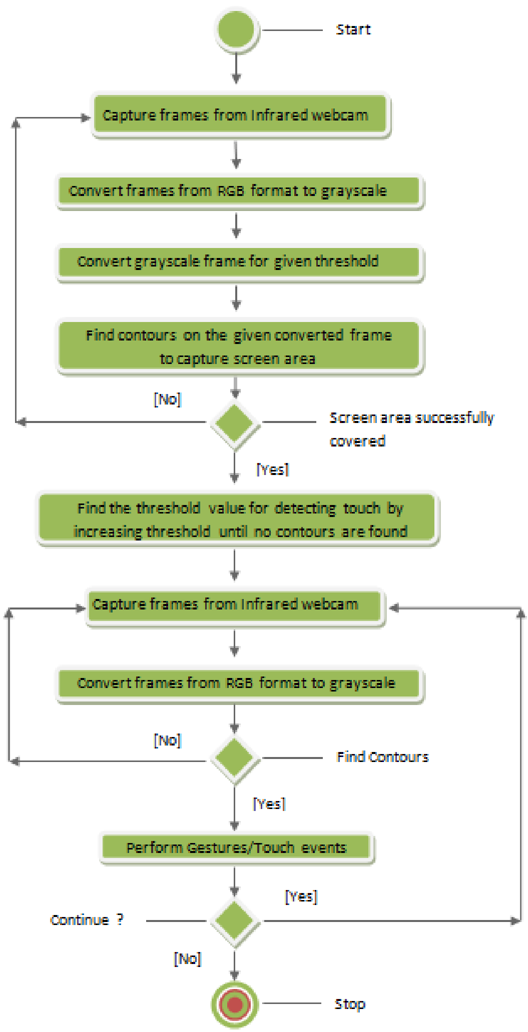


**Figure 7 Activity diagram of SmartTouch**

## IV. Findings and Conclusion

### *A. Comparative advantage of SmartTouch*

SmartTouch is cost effective and efficient multi-touch and multi-gesture enabled design. It uses single IR webcam to capture images of the screen area. IR LEDs emits IR rays parallel to LCD screen. These IR rays when strikes stylus (or finger) causes illumination of stylus (or finger) that is detected as touch by SmartTouch system. In uTouch[5], skin color detection is used to identify finger and a point of touch is determined by point of intersection by images from two webcams. Skin color detection is dependent on background of the image, so when background of an image is similar to skin color then it may cause problem. Secondly, it requires two cameras which increases cost of the system and computation complexity. Images from two cameras need to be merged. In this case, point of intersection of the finger on merged image gives the point of touch. Memory requirements are more which may result in memory leakages. Diamond Touch[1] requires the special capacitive electrode alignment on sheet to be placed on table to implement touch capabilities. Although approach being novel and effective for multi-user environment, yet it is costly. Holo Wall[2] uses a special acrylic screen for identifying objects close to screen. Holo Wall approach being simple and effective for large screens requires high screen cost and needs camera setup & projector to be placed behind the screen area. This constraint restricts its use with PCs and LCD screens.

SmartTouch uses IR rays and IR webcam. It is independent of skin color so there is no effect of background color. It uses single camera, few IR LEDs and easily available LCD screen to implement a touch screen. Working LCD screens can also be easily converted to touch screen by using the proposed hardware setup. SmartTouch enables both multi-gesture and multi-touch. The approach presented in this paper helps to not only convert normal screen to touch screen in a cost-effective manner but can also reduce dumping of normal screens which is done to replace them with touch screens. This can reduce electronic waste (e-waste).

### *B. Applications and future scope*

The most unique characteristic of SmartTouch is the fact that it requires a cost effective and easy hardware setup to give a touch screen experience that is further empowered with multi touch abilities at absolutely no additional cost. To us this hardware setup cost Rs.500 as compared to conventional touch screens that costs up to Rs.15000/- . Thus this technology being cost effective and efficient is designed so that it can reach masses that can integrate it with their Personal Computers and use their monitors as a touch screen for gesture interactive applications. In future, it can be improved to make it work in direct sunlight. In current scenario it won't work in direct sunlight. Distance between screen and IR camera can be further reduced to improve efficiency in case of multi-touch. It can also be made compatible for tablets (reducing overall size of setup). In our approach, a multi-gesture is recognized by calculating change in distance between multiple contours and slope formed by them. Other way of mapping multi-gesture can be to map the points on the predefined pattern. This technique can be useful for gestures involving curves, like formation of "8". Since current specifications of SmartTouch do not require implementation of complex gestures, so arithmetic templates are sufficient enough. In future, one can also enhance the approach to support complex gestures by shifting to pattern matching.

## ACKNOWLEDGMENT

We would like to extend our heartiest thanks to Mr. Usama Gufran for his valuable comments and feedback in the paper.

## REFERENCES

[1] Paul Dietz and Darren Leigh, "DiamondTouch: A Multi-User Touch Technology", UIST 2001, the 14th Annual ACM Symposium on User Interface Software and Technology, Orlando, Florida USA, pp. 219-226, November 11-14, 2001

[2] Nobuyuki Matsushita, Jun Rekimoto," HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall", UIST 97 Albert, Canada, pp 209-210

[3] Klemmer, S.R., M. W. Newman, R. Farrell, M. Bilezikjian, J.A. Landay, "The Designer's Outpost: A Tangible Interface for Collaborative Web Site Design", ACM Syposium on User Interface Software and Technology, pp. 1-10, 2001

[4] Ullmer, B., H. Ishii, "The metaDESK: Models and Prototypes for Tangible User Interfaces", ACM Symposium on User Interface Software and Technology, pp. 223-232, 1997

[5] Usama Ghufran, "uTouch: A cost effective touch screen solution for gesture based interaction", ICCMS, 2011