IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. XI (Mar-Apr. 2014), PP 48-56
www.iosrjournals.org

Design and Implementation of Low-Cost Multi-Touch Surface Computing Using Image Processing

Prinkan Pal[1], Subramanya Vikas[1] and Pavan Raju[1]
[1] *Electronics and Communication Engineering, Visvevaraya Technological University, Belgaum, Karnataka 590018, India*

Abstract : *Touch sensing devices have reached the pinnacle of success in the present era thereby revolutionizing human machine interaction. Capacitive touch-screens are popular because of its portability whereas image processing based touch-screens are massive but for low-cost surface computers with the capability of both touch and object-recognition, the latter is preferred because it uses optical solutions to sense touch.Multi-touch devices are in vogue and therefore the concept of surface computing is gaining popularity since it provides an enhanced multi-touch and multi-user experience.Multi-touch denotes a set of interaction techniques that allow computer users to control multimedia applications with several fingers. The concept of gesture recognition thus comes into the picture. Our objective is to implement a cost-effective multi-touch surface computing system. The image processing algorithms used are implemented in MATLAB and the simulation of gesture recognition is carried out using Open Source Computer Vision Library (OpenCV).*

Keywords: *Multi-touch, Gesture Recognition, Surface Computing, TUIO, Fingertip Blob, Image Processing based Touch-Screens.*

I. Introduction

Nowadays, we are overthrown with huge amounts of information that needs to be accessible in public spaces. In these spaces, there are users with all kinds of backgrounds accessing the information. [1]For example, in most museums there is a lot of digital information available on the exhibit but it is difficult to make them accessible to the audience using traditional devices such as information kiosks with a touch-screen. Interactive computerized systems in public places should take in consideration the special characteristics of these places: everyone can enter them, and there might be many different users at once. The user makes different gestures with his/her fingers on the surface of the table. The video of those gestures are captured and processed instantly to obtain the gesture patterns and respective Cartesian coordinates using an image processing software. The principal of optics used in obtaining the patterns from the surface is known as Front Diffused Illumination (DI). Each gesture has been dedicated to carry out certain tasks. The gestures are hence recognized and the calculated coordinates are then sent to the respective applications using TUIO protocols, powered by OpenCV libraries.

II. Existing Methodologies

Research on multi-touch sensing surfaces using image processing has started back in 1982 and ever since then numerous optical solutions have been proposed leading to different table-top hardware constructions elaborated in the following sub-sections [2].

2.1 Frustrated Total Internal Reflection (FTIR)

This method makes use of an acrylic sheet over which a compliant surface of higher refractive index compared to the acrylic sheet is placed. Infrared (IR) LEDs are placed throughout the thickness of the acrylic. When the IR rays are emitted by these LEDs the rays undergo Total Internal Reflection (TIR) inside the acrylic. When a user touches the surface these rays are said to be "frustrated" at the point of touch or in other words the rays are hindered from TIR. These IR rays scatters downward where they are captured by an IR camera placed underneath and are identified as blobs.

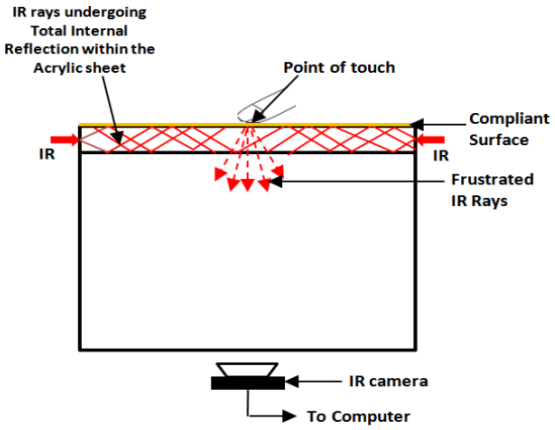

Fig. 1 Setup for FTIR

1.2 Diffused Surface Illumination (DSI)

In this method a special type of acrylic sheet having small particles within the material is used. These particles acts like thousands of small mirrors which help in distributing the IR rays evenly across the touch surface. The setup is almost same as FTIR; however a compliant surface is not used since the IR rays need not undergo total internal reflection. When a finger is brought in contact with the surface a shadow is formed which is registered as a blob by the IR camera.

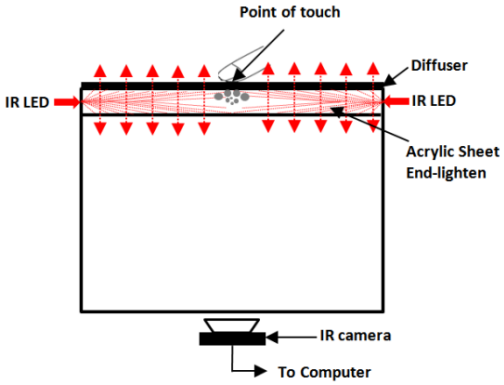

Fig. 2 Setup for DSI

2.3 Front Diffused Illumination (FDI)

A diffuser is kept on top of the acrylic surface. Infrared light from the surrounding is shined on top of the touch surface (acrylic sheet). A shadow is created when a finger or an object touches the surface. The shadow is then sensed by the IR camera and processed further.

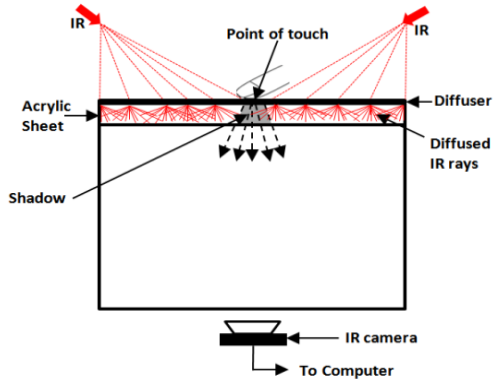

Fig. 3 Setup for Front DI

1.3 Rear Diffused Illumination (RDI)

In this case, infrared light is shined at the screen from below the touch surface. A diffuser is placed on top or below the surface. When an object touches the surface it reflects more light compared to the diffuser or other background objects. The reflected light is then sensed by the IR camera as blobs.

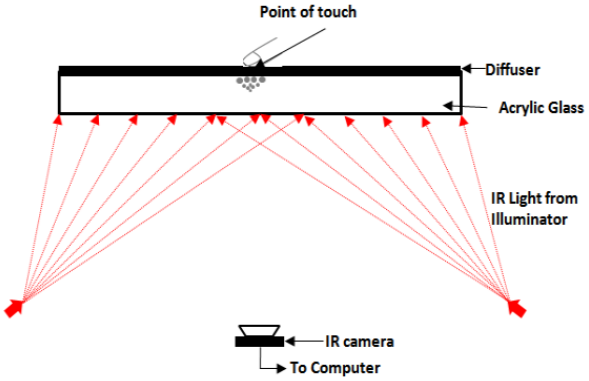

Fig. 4 Setup for Rear DI

1.4 Laser Light Plane (LLP)

Infrared light is shined above the surface using a laser. When a finger is placed on the surface the laser rays come in contact with the finger and scatters downwards. These scattered rays are picked by an IR camera placed underneath which identifies these rays as blobs.

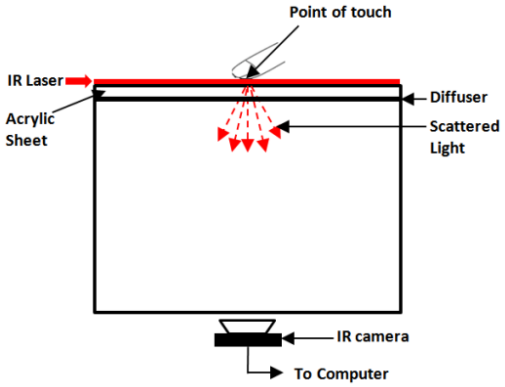

Fig. 5 Setup for LLP

All the above mentioned optical solutions possess few constructional similarities like the use of IR LEDs, IR cameras and acrylic touch surface. These increases constructional complexity and cost. Use of infrared rays also prevents the system from being energy efficient and environment-friendly. Keeping the adversities in mind, we have designed a setup with reduced construction complexity at a low-cost and the same is presented in the next section of this paper.

III. Cost-Effective Hardware Setup

In the construction of our hardware we avoided the use of IR LEDs, rather we used natural light for normal room lighting conditions and cost-effective low power white LEDs in case of inadequate or dim lighting conditions. Thus the costly acrylic surface is replaced by normal glass pane and use of diffuser and compliant surface is absolutely unnecessary. Hence the IR camera is replaced by a basic VGA webcam with a capture rate of 30 frames per second and resolution of 640 x 480. A tracing paper is placed below the glass surface to prevent the reflection from the glass surface.The tracing paper also serves the purpose of projection of the display when a projector is interfaced.

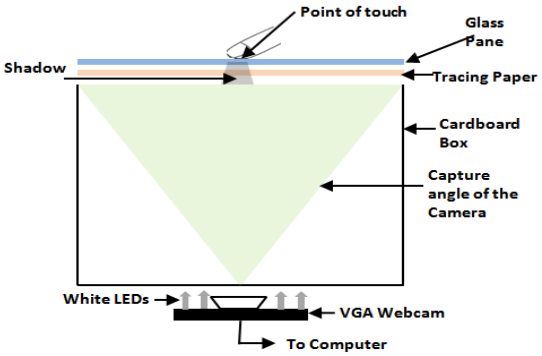

Fig. 6 Our Cost-Effective Hardware Setup of a Multi-Touch table.

IV. General Software Architecture Of Multi-Touch Technology

Every action on the user interface is expected to turn into some useful command that the currently running application (or the operating system) performs. We can identify two layers of software at this point: the device interface software and the application software. The job of the device interface software is to turn the physical interface action into a logical software representation, a command. [4] Then it becomes the job of the application software to respond to that command appropriately. The interface software was mainly implemented using OpenCV.

The software architecture consists of four distinct layers explained in the following subsections.

4.1 Hardware Abstraction Layer

This layer takes raw input data from the underlying hardware. The data is then searched for finger, hand and/or object positions, which are then transmitted to the next layer.

4.2 Transformation Layer

This layer transforms the position data from device to screen coordinates. This is achieved, with a perspective transformation which is obtained in a calibration procedure. Especially with camera based systems, a transformation has to be performed on the low-level data, which is still in the form of an image.

4.3 Interpretation Layer

The interpretation layer translates the movements of hands and fingersinto gestures, thereby assigning a meaning to pure motion. To doso, this layer needs knowledge about regions on the screen. Foreach region, a list of gestures to match is maintained. When thecorrect events occur within a region, the corresponding gesture istriggered and passed to the next layer. As the mapping from motionto meaning can be expected to change for different input devices, acapability description has to be supplied which provides this mapping.

4.4 Widget Layer

The widget layer is the application layer that performs the task of generating visible output for the user. It receives events from and registers regions with the interpretation layer.

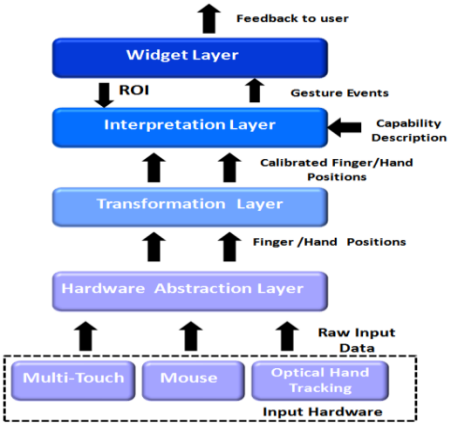

Fig. 7 Multi-Touch Sensing Software Architecture

Programming for multi-touch input is much like any other form of coding; however there are certain protocols, methods, and standards in the multi-touch world of programming. [5] Through the work of NUI Group and other organizations, frameworks have been developed for several languages, such as Action Script 3, Python, C, C++, C#, and Java. Multi-touch programming is two-fold: reading and translating the "blob" input from the camera or other input device, and relaying this information through pre-defined protocols to frameworks which allow this raw blob data to be assembled into gestures that high-level language can then use to interact with an application. TUIO (Tangible User Interface Protocol) has become the industry standard for tracking blob data. The implementation of the software architecture has been elaborated in section 5 and 6 of the thesis.

V. Image Processing Algorithm For Blob Detection And Tracking With Results

After the hardware is setup, the webcam is interfaced to the computer through USB 2.0 port. Still image snapshots are taken by the camera and are processed using image processing algorithms implemented in MATLAB to obtain the blobs from which the point of touch is accurately located. These image processing steps

forms the Hardware Abstraction and the Transformation Layer explained in section 4.1 and 4.2 respectively. [3] [7] The algorithms used for obtaining the blobs from the RGB images are shown in figure 6 and summarized in the steps below.

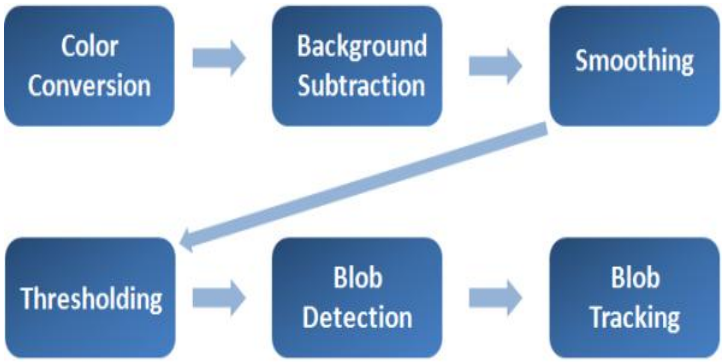

Fig.8 Image Processing Steps for extraction, detection and tracking of the blobs.

5.1 Color Conversion

The image captured by the camera is in the format of red, green and blue (RGB). Since we are only bothered about the finger touch points on the surface color is not required. Therefore, we convert the color image to gray scale image where the entire pixel values ranges between 0 (black) to 255 (white), denoting the luminous intensity (Y) of each pixel as shown in figure 9a and 9b. This simplifies further processing of the image and processing time is reduced approximately by a factor of 3.

5.2 Background Subtraction

It is a technique used to identify the background of an image from the desired objects in the foreground. Before the system starts, an image of the surface without any object or fingers placed on the surface is captured and saved as the background image. Later on, this background image is subtracted from composite frames to eliminate the background and retain only the foreground elements as shown in figure 9c.

5.3 Smoothing

Camera images typically contain few stray pixels that are brighter than their surroundings. If not removed from the image, these may falsely look like fingertips. Smoothing the image works to filter them out and bring their brightness closer to their surroundings.

5.4 Thresholding

The smoothened gray scale image is converted into a monochromatic or black and white image by a conditional threshold value denoted by δ. From experimentation we have found the value of δ to be 20 for normal lighting conditions and 30 in case of inadequate lighting. This improves the identification of the fingertip location on the screen and removes the false bright spots identified during smoothing.

5.5 Blob Detection

At this point, the image has been turned into a strictly black and white image. White spots (blobs) represent the areas where the dark spots are formed on the screen. These spots are the locations of the user's fingertips on the surface of the screen. The black and white image is scanned line by line to detect white dots. After scanning a line, we have a group of line blobs. These are the sequence of white pixels on a particular line. After scanning the second line, another sequence of line blobs is identified. These are then compared to the line blobs on the previous line. If they overlap, they are merged to form blobs. This is repeated until all lines in the image are scanned. The result is a collection of white blobs in the 2-dimensional array.

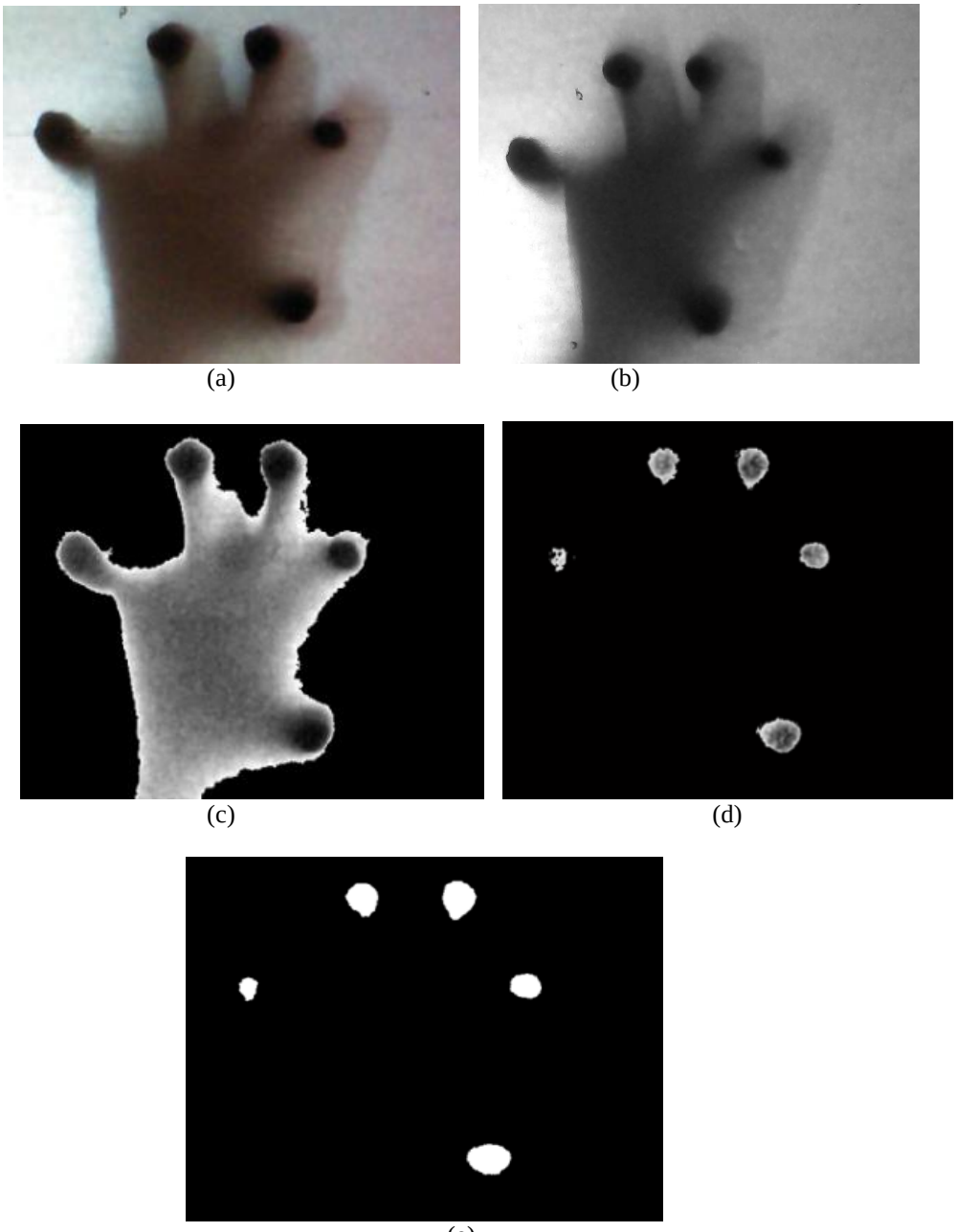

Fig. 9 Images obtained at each processing step (a) Captured RGB image by the camera, (b) Grayscale Image, (c) Background Subtracted Image, (d) Smoothened image, (e) Thresholded image.

5.6 Blob Tracking

The algorithm assigns a unique ID for each blob and keeps updating that ID by tracking the blob in all frames. The algorithm computes the distance between the center of the blob in the new frame and the center of the blob in the previous frame. Under any scenario there can be two cases as explained below.

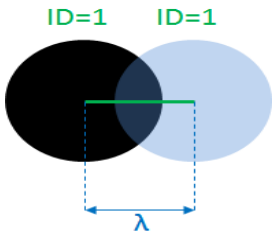

Fig. 10 Case 1: A user moves the same finger on the touch surface.

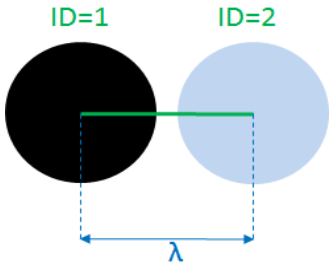

Fig. 11 Case 2: A user places multiple fingers on the touch surface at different instances of time.

In case 1, when the user drags the same finger touching the surface inorder to make some gesture, the distance λ between the center of the blob in a particular frame and that in the consecutive frame is below the certain threshold Δ so the blob in the next frame is also assigned the same ID, hence both the blobs have the same ID. This also helps the gesture recognition module to interpret the kind of gesture made by the user.

In case 2, when the user places multiple fingers in a sequence at different instances of time, the distance λ between the center of the blobs in consecutive screens is above the threshold Δ, so the blob in the next frame is assigned a new ID.

To detemine the value of Δ, two fingers are kept on the surface at closest proximity possible and a snapshot is taken using the camera of our setup. On analyzing the image it was observed that the distance between the two centers is 60 pixels. Hence under any condition a particular finger cannot traverse a distance of 60 pixels between consecutive frames. Therefore, from experimentation we obtained the value of Δ to be 60 pixels.

A conclusion section must be included and should indicate clearly the advantages, limitations, and possible applications of the paper. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extentions.

VI. Front-End To Back-End Communication

The primary goal of gesture recognition research is to create a system which can identify specific human gestures and use them to convey information or for device control. In order to ensure accurate gesture recognition and an intuitive interface a number of constraints are applied to the model.[8][9] The multi-touch setup should provide the capability for more than one user to interact with it working on independent (but maybe collaborative) applications and multiple such setups working in tandem. The camera obtains the frames of gestures at a rate greater than 30 frames per second which is then processed in the image processing software to obtain the blobs or touch points as explained in section 5. Recognizing the gesture is performed at the Interpretation Layer elaborated in section 4.3.

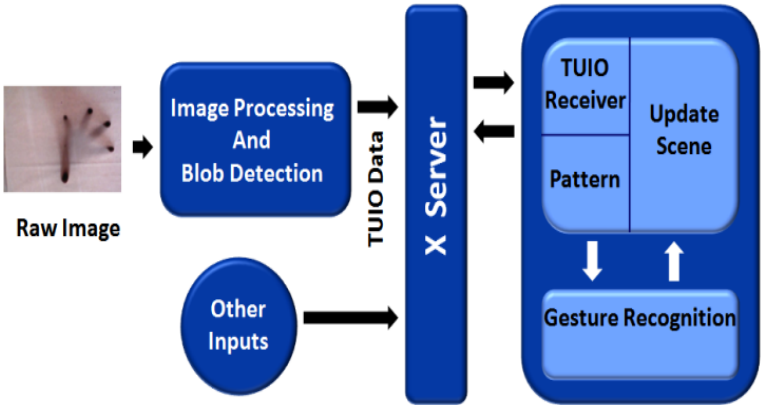

Fig.12 Blob detection to Gesture Recognition framework outline.

The image processing algorithm detects and tracks the blobs in each frame. Now our system is capable of sensing touches but still it is incapable of interpreting the touches and responding for the same. [6] This is where gesture recognition comes into the picture which is generally carried out using three basic steps as explained in the following subsections.

6.1 Detection of Intention

Gestures should only be interpreted when they are made within the application window. With the implementation of support for multi-touch interface in X Input Extension Protocol, it is the job of the X server to relay the touch event sequence to the correct application.

6.2 Gesture Segmentation

The same set of gestures in the same application can map to several meanings depending on the context of the touch events. Thus the touch events should be again patterned into parts depending on the object of intention. These patterned data will be sent to the gesture recognition module.

6.3 Gesture Classification

The gesture recognition module will work upon the patterned data to map it to the correct command. There are various techniques for gesture recognition which can be used alone or in combination like Hidden Markov Models, Artificial Neural Networks and Finite State Machine etc.

OpenCV provides a set of libraries which has pre-defined gesture recognition modules. Our algorithm obtains the location of touches on the surface in form of Cartesian co-ordinates (x, y) considering the surface to be a 2-dimensional plane. [10] This location is then sent to the OpenCV gesture recognition framework using Tangible User Input Output (TUIO) protocol. The TUIO server shown in figure 12 converts the results of blob tracking to TUIO messages and warps these messages in a UDP packet. [11]Any application software can interface to the multi-touch implementation using the TUIO protocol. The protocol is made up of a TUIO server which acts as an interface between the blob locations and the gesture recognition module, and a TUIO client that is implemented at the application end. The application software implements the client side of the protocol and monitors a specific UDP port to receive the UDP packets. The TUIO approach makes it easier for developers to develop applications for multi-touch without the need to implement the detection part. The protocol has been implemented using Open Source Control and is therefore usable on any GUI platforms.

VII. Multi-Touch Applications

The Widget layer of the architecture is defined by the multi-touch application that runs at the front-end or in other words that the user operates. Amongst the several multi-touch applications available we have used the "Virtual Piano" application for our experimentation which is elucidated below.

7.1 Virtual Piano. The piano application for multi-touch table is one of the most popular and primitive applications which gives us the feel of a playing a piano in the real world. The application was developed using ActionScript3 and implements the TUIO client side for recognizing events. The notes and the tones produced do not differ by any means from that of the one in the real world.

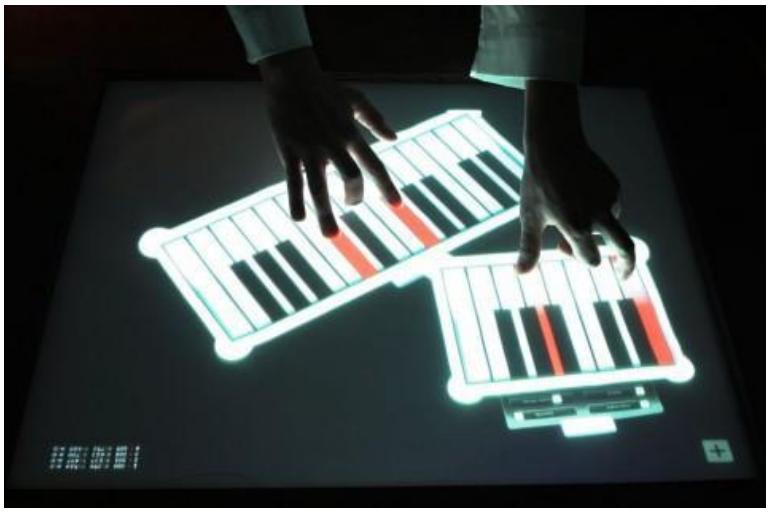

Fig. 13 Auser playing a virtual piano on multi-touch surface producing musical tunes.

VIII. Conclusion

The hardware and software implementation has been thoroughly tested for a wide variety of user fingertip sizes and lighting conditions with appropriate results. The system has succeeded in recognizing up to ten fingertip blobs in a single frame at any instant.

The touchscreen can be used widely as gaming consoles, information kiosks, auxiliary to a personal computer, etc.The future of surface computing also promises to gift engineers with development tools. At

present surface computers are being used in hospitals, educational organizations, railway stations, museums, restaurants, coffee-shops and at all those places where digitalization and automation has made sharing of information via human machine interaction a necessity. Hence a cost-effective implementation for the same will make the technology affordable and intensify its popularity. The use of image processing touchscreens is not feasible for portable electronic gadgets or handheld devices due its massiveness but is the only option for surface computers.

Surface technology and surface computing is an interesting field of research where one can come up with innovative ideas and applications in order to make human computer interaction much simpler. Our future work lies in the conversion of recognized gestures into text and speech which will help the visually impaired people to convey and obtain information. [12] Multi-touch technology can be useful for the implementation of gesture to text conversion, object recognition, pattern recognition and many more resulting in the revolution which will merge the physical world to the digital world thereby making life simpler and faster.

References

Proceedings Papers:

[1]. Han, Jerfferson Y. "Low Cost Multi-Touch Sensing through Frustrated Total Internal Reflection." Symposium on User Interface Software and Technology: Proceedings of the 18th annual ACM symposium on User interface software and technology. Seattle,WA, USA, 2005. 115-118.

[2]. Ang, Zhi-Yang; Yuen, Chai Tong; Ng, Tze-Yu; Ng, Yee-Long; Ho, Jee-Hou; "Development of a multi-touch table for natural user interface", Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2011 IEEE Conference, pp - 201 – 206

[3]. Saravanan, C., "Color Image to Grayscale Image Conversion*",* Computer Engineering and Applications (ICCEA), 2010 Second International Conference, pp - 196 – 199

[4]. S. S. A. Sheikh, S. M. Hanana, Y. Al-Hosany, and B. Soudan, "Design and implementation of an FTIR camera-based multitouch display," *GCC Conference & Exhibition*, 2009 5th IEEE, pp – 1-6.

[5]. M. Pollefeys, R. Koch, and L.J.V. Gool, "Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters," *Proc. Sixth IEEE Int'l Conf. Computer Vision (ICCV '98)*, pp. 90-95, 1998.

[6]. Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon; 2011 10th IEEE Int'l Symp. Mixed and Augmented Reality, pp. 127–136

[7]. Brink, A.D.: Grey-level thresholding of images using a correlation criterion: Pattern Recognition Letters, 9, 335-341 (1989)

[8]. Y. Sato, Y. Kobayashi and H. Koike, "Fast Tracking of Hands and Fingertips in Infrared Images for Augmented Desk Interface," *Proc. 4th IEEE Int',l Conf. Automatic Face and Gesture Recognition*(FG 2000), IEEE Press, Piscataway, N.J., 2000, pp. 462-467.

[9]. F. Echtler, M. Huber, and G. Klinker, "Hand tracking for enhanced gesture recognition on interactive multi-touch surfaces", Technical Report TUM-I0721, Technische Universitat Munchen, Department of Computer Science, 2007.

[10]. H. Benko, A. D. Wilson, and P. Baudisch, "Precise selection techniques for multi-touch screens", In Proc. CHI, pp. 1263-1272, 2006. (Pubitemid 44032225)

[11]. Gomes, A., Amador, G. ; TouchAll: A Multi-Touch, Gestures, and Fiducials API for Flash/Action Script 3.0; Multimedia and Ubiquitous Engineering (MUE), 2011 5th FTRA International Conference on ;Pages 53 – 58

[12]. D. WANG, M. ZHANG, Z. XIONG, "The review of multitouch technology research", Application Research Of Computers, Vol. 26, No.7, 2009, pp. 2404-2406, doi:10.3969/j.issn.1001-3695.2009.07.002.