

A Cost Efficient Solution for Surface Technology with Mobile Connect

Nitin Sharma

Department of Computer Science, Jaypee Institute of Information Technology University, Noida, India
Email: nitinsh99@gmail.com

Manshul V Belani

Department of Computer Science, Jaypee Institute of Information Technology University, Noida, India
Email: manshulvbelani@gmail.com

Ms. Indu Chawla

Superviser, Sr. Lecturer, Department of Computer Science, Jaypee Institute of Information Technology University,
Noida, India
Email: indu.chawla@jiit.ac.in

Abstract—For the past few decades we have seen the evolution of user interfaces from CLI (Command Line Interface) to GUI (Graphical User Interface) to the most recent NUI (Natural User Interfaces). One such example is Surface Computing. Surface computing is a new way of working with computers that moves beyond the traditional mouse-and-keyboard experience. It is a natural user interface that allows people to interact with digital content with their hands, with gestures and by putting real-world objects on the surface. Though very convenient and simple to use, the cost of this technology is one of the major hindrances in making it accessible to common users. Microsoft PixelSense is an excellent example of this technology. For making this technology cost efficient and more accessible, we have designed an interactive table. It allows two mobile phones to interact with each other, i.e. one can share mobile content just by placing two phones on the table and dragging the content from one phone to another using their fingers. Instead of the usual resistive and capacitive touchscreens used these days, it uses an imaging touch screen, i.e. image processing is used to determine the fingertip coordinates and handle data according to these coordinates thereby decreasing the cost of this technology sharply.

Index Terms—Android, Image Processing, Sensors, Surface Technology, Surface Technology with Mobile Connect (STMC).

I. INTRODUCTION

The latest revolution in the field of human computer interaction is the shifting from graphical user interfaces to Natural User Interfaces. NUI's refer to a user interface based on nature or natural elements like speech, gestures etc. that is effectively invisible to its users, or becomes invisible with successive learned interactions [1]. One such example of natural user interface is Surface Technology.

Surface Technology is the term for the use of a specialized computer GUI in which traditional GUI elements are replaced by intuitive, everyday objects [2]. Instead of a keyboard and mouse, the user interacts directly with a touch-sensitive screen. It more closely replicates the familiar hands-on experience of everyday object manipulation.

For instance you are in a coffee shop hanging out with friends and you want to view and share some holiday pictures. Wouldn't it be a great experience to view your pictures on the coffee table right in front of you? Moreover you can also drag them to other devices placed on the table. This is what surface technology can provide you with and much more.

Surface work has included customized solutions from vendors such as GestureTek, Applied Minds for Northrop Grumman. Major computer vendor platforms are in various stages of release: the iTable by PQLabs, Linux MPX, the Ideum MT-50, interactive bar by spinTOUCH, and Microsoft PixelSense (formerly known as Microsoft Surface) [3].

Microsoft PixelSense is an interactive surface computing platform that has the ability to optically recognize objects placed on top of it. It allows one or more people to use touch and real world objects, and share digital content at the same time [4]. However one of its major drawbacks is its cost. Due to its high cost Microsoft PixelSense is currently being marketed and sold directly to only large scale leisure, entertainment and retail companies [5].

To overcome this drawback and to make this technology more accessible to common man, we've designed Surface Technology with Mobile Connect (STMC). STMC provides its users with a natural interface and that too at an affordable cost. It is an interactive table which detects the devices placed on it and helps in sharing data between them.

Microsoft PixelSense has been described in Section II. Section III discusses the overview and architecture of STMC. The approach we have followed has been discussed in Section IV. Section V discusses the

elaborate implementation details of our interactive table. A comparison in terms of cost of the new approach with the existing approach has been presented in section VI.

II. RELATED WORK

The new concept that's driving the web sphere crazy is the new and totally innovative Microsoft Surface Computing. Microsoft Surface (now known as Microsoft PixelSense) is basically a surface computer that erases the borders between physical and virtual worlds. It is the new approach of interacting with Digital Content in a physical manner using abstract technology. Microsoft is the first company ever to bring such an idea to the market as a commercial product [2]. Microsoft Surface 1.0, the first version of PixelSense, was announced on May 29, 2007 at the D5 Conference [4].

Essentially, Microsoft Surface is a computer embedded in a medium-sized table, with a large, flat display on top that is touch-sensitive. The software reacts to the touch of any object, including human fingers, and can track the presence and movement of many different objects at the same time [2].

It is an interactive table top that can do everything a network computer can do. It has four key features: direct interaction, multi-touch ability, multi-user ability, and object recognition [5].

Direct interaction allows user to use their hands and natural gestures to interact with content without the use of mouse or keyboard [2].

Multi-touch feature enables the Surface to recognize many points of contact simultaneously [2].

Multi User Experience allows for multi-users at once, i.e. more than one user can interact with the table at one. [2].

Object recognition, enables users to place physical objects on the surface to trigger different types of digital responses, including the transfer of digital content. [2].



Figure 1- Microsoft Surface Table

Working of Microsoft Surface:

1. A contact (finger/blob/tag/object) is placed on the display[6]
2. IR back light unit provides light (through the optical sheets, LCD and protection glass) that hits the contact.[6]

3. Light reflected back from the contact is seen by the integrated sensors[6]
4. Sensors convert the light signal into an electrical signal/value[6]
5. Values reported from all of the sensors are used to create a picture of what is on the display[6]
6. The picture is analyzed using image processing techniques[6]
7. The output is sent to the PC. It includes the corrected sensor image and various contact types (fingers/blobs/tags)[6]

TABLE 1 – Microsoft PixelSense Overview

Developer(s)	Microsoft
Initial Release	Microsoft Surface 1.0 (April 17 2008)
Development Status	Commercial Applications
Operating system	Microsoft Surface 1.0: Windows Vista (32-bit) Microsoft Surface 2.0: Windows 7 Professional for Embedded Systems (64-bit)

III. SURFACE TECHNOLOGY WITH MOBILE CONNECT OVERVIEW

Surface Technology with Mobile Connect is a cheaper parallel version of the Microsoft's Surface Technology. It consists of a glass table with sensors on all four sides. The sensors help in detection of devices placed on the table. In this case two phones (say A and B) are placed on the table. With the help of the sensors the coordinates of the device (phone) are determined and a field is created around the phones in which the files stored in the phone are displayed on the table itself. This is done with the help of an android application (This can also be done for other non-touch devices like Wi-Fi digital cameras, Wi-Fi pen drives or hard disks with the help of a simple client server application).

In order to transfer the files one can simply drag and drop those from the source phone (A) to the destination phone (B). Any phone can work as the source or destination phone as per the user's need. This is basically done by determining the coordinates of the user's fingertip with the help of a contour detection algorithm. The file starts moving along with the fingertip if the fingertip is pointing at the file. Once the file being dragged reaches the field created around the destination phone it is transferred to the phone with the help of an android application.

All this time the server screen is being projected on the glass tabletop which gives the feeling as if one is working on a touch table.

Since an imaging touchscreen (using image processing to determine touch) is used in place of more frequently used resistive or capacitive ones, the cost of the setup is reduced manifold.

Overall the application is an integration of surface technology, image processing, android application development, sensors and algorithms to create a user-friendly interactive multipurpose table.

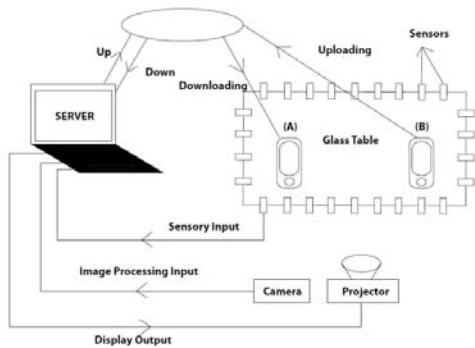


Figure 2 – STMC Architecture

The basic components of this interactive table being:

1. **Glass Table:** It is the top layer which acts as the surface. Also it acts as a virtual touch screen using which one can drag and drop files.
2. **Sensors:** Infrared Sensors (reflectance sensors) placed around the table edges help in detecting phones as soon as they are placed on the table.
3. **Web Camera:** The webcam helps in finger detection using image processing.
4. **Projector:** The projector displays the screen of the server onto the glass table to give user the impression of working on a touch screen.

IV. PROPOSED APPROACH

The project has been divided into four modules namely device detection, Transfer of data to server, Fingertip detection and Transfer of data to other device.

In the first module the device is detected with the help of infrared reflectance sensors. Sensors are placed at the edges of the table. As soon as the devices are placed on the table and come in the range of sensors, the application triggers.

In the second module, data is transferred from the device to the server. As soon as the device is detected by the sensors, the data from the device is transferred to the server via an android application. The server receives this data via a client server application. Thereafter this data is displayed on our user interface screen. Since the whole setup is connected to a projector which projects the server screen on the glass tabletop, the user screen can be seen on the tabletop the whole time.

The third module deals with the detection of the user's fingertip and hence determining its coordinates (x, y). Using these coordinates we can move the data displayed on the table with the movement of one's

finger. This use of image processing to determine position of fingertip reduces the requirement of a large number of touchscreen sensors thereby reducing the cost. The data is moving along the fingertip on the user interface screen and is being projected on the tabletop simultaneously.

Finally the data has to be transferred to the other device kept on the table. This has also been achieved by an android application. As soon as the coordinates of the user's fingertip reach the vicinity of the device, the data is transferred to the other device. The server uses another client server application for this transfer and the device receives it via an android application.

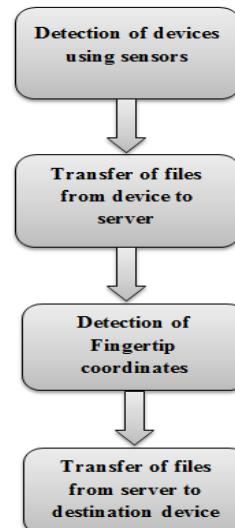


Figure 3- Proposed STMC

V. IMPLEMENTATION

A. Device Detection

One of the important aspects of the STMC is its device detection. Our main goal is to make such devices affordable and accessible to general user. The approach to reduce the overall cost of such setup involves replacing the touch sensitive capacitive displays by cheaper sensors. The sensors will act as device detectors. We have taken a 18" X 12" table and placed 10 sensors on its edges (6 across its length and 4 across its width). Figure-2 illustrates the distribution of sensors and sensor types on our table.

1. Once the sensors are activated they continuously look for any near field detection of any object. These sensors are typically low range object detecting sensors and each sensor has its own area of detection. The range of the sensors used by us was 15cm, i.e. any object at a distance of more than 15cm from the sensors could not be detected by the sensor.
2. As soon as a device is put on the table, it falls within the detection range of one or more sensors.

3. The readings of sensors are continuously being read using ‘Processing-2.0b7’ tool [7]. When an object comes in the sensor’s range, the value of the sensors goes below a certain value. In our case we got the best results with value 80. As soon as the value for one or more sensors reaches below 80 for fifteen times continuously, we consider the object to be in sensor’s range. We’ve taken 15 readings before confirming presence because at times values go below this value even in absence of device.

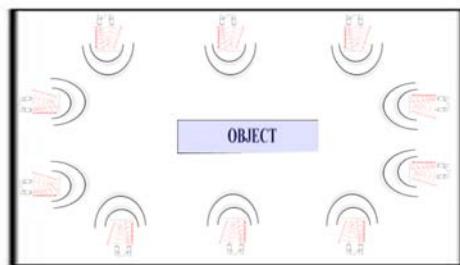


Figure 4 -Illustrating the sensor distribution on the table and the detection field of the sensor.

This is a crucial step; since the user can put any device or even his/her hand on the table. The sensors must be intelligent enough to report only the detection of required device such as a mobile or any other media source. We overcame this problem by using what we call “Device Registration”. So, each device, which has to be put for data sharing on the table, has to be first pre-registered with the backend server running on the cloud or main processing unit. Hence only the registered device can trigger the data transfer process on the main processing unit.

B. Transfer Data to Server

As soon as we put a registered device in the detection range of a (or more) sensor, the server process continuously running on the main processing unit is informed by the sensor. The server then transmits a message to each of the registered device asking which particular device wants to transmit the data. The android app running on the registered device receives this message and responds with the appropriate response. This request-response finally results in the data transfer from android device to the main server [8].

The android application first of all determines all the files present in a particular folder of the phone and appends it to an array list. It then sends the number of files present in the list to the output stream using socket programming. Then it writes the names of all files in the data output stream. Then a loop is run which determines the size of each file and sends the file into outstream in the form of bytes. The loop iterates as many times as the number of files present in the array list. At the server end a program is running which continuously receives in its input stream the bytes being transferred from this output stream.

We have experimented with different types of data and were able to transfer various file formats like jpeg, png, mp3, pdf, doc, zip, mp4 successfully.

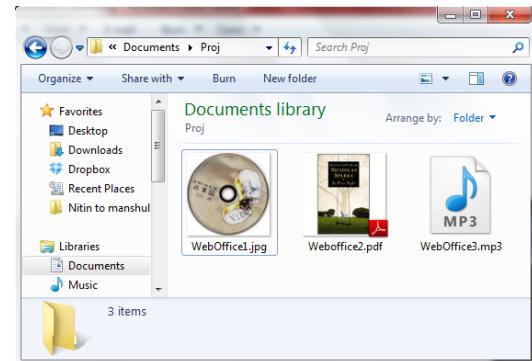


Figure 5- Files Received at Server End

C. 5.3 Fingertip Detection

The next step is to determine the position of the fingertip on the tabletop. Contour Detection algorithm [9] is used for determining fingertip position. This can be done using the openCV [10] library which uses C++. Following steps were required for fingertip detection:

1. Capture live video using webcam and extract every frame.
2. Convert the RGB image of every frame to gray scale image.
3. Convert the image to binary image.
4. Use this binary image to extract out all the contours corresponding to probable hand regions.
5. Compute the contour having the largest area among all these and consider it to be the hand.
6. Use curvature determination to find out the peaks and valleys in the hand. This is done by finding the dot product between every two vectors on the contour of the hand. The peaks and valleys will have a dot product close to zero.

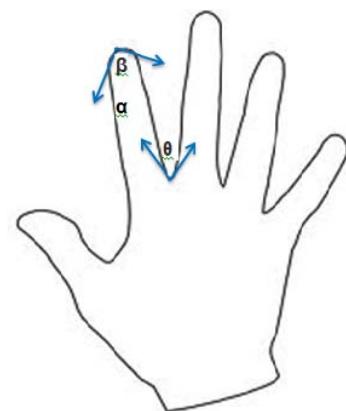


Figure 6 - Peak and Valley detection

7. Differentiate between peaks and valleys by taking cross product of vectors. The values for peaks and valleys will have opposite signs.

Thus we can determine the coordinates of the fingertip.

D. Transfer Data to Other Device

If the fingertip lies on a particular file, the file starts moving along with the fingertip. As soon as the fingertip (along with the file) reaches near the other device, the file gets transferred to the other device. This has been achieved using simple image processing algorithm.

1. The coordinates of the fingertip are continuously being recorded in a file.
2. This file is continuously being read by the code displaying the file on canvas. The position of the file on canvas changes along with changing coordinates of fingertip.
3. Once the coordinate of the file reaches close to the destination device, the file is transferred to it. In our case this is done using the android application running on the destination phone. The server first of all sends the name of this file into the output stream. At the same time the android app running on the destination phone reads this name into its input stream. After sending the name of the file, the server then sends the content of the file in the form of bytes to its output stream. The android app running on the destination phone receives these bytes in its input stream and stores these with the specified name in a particular folder.

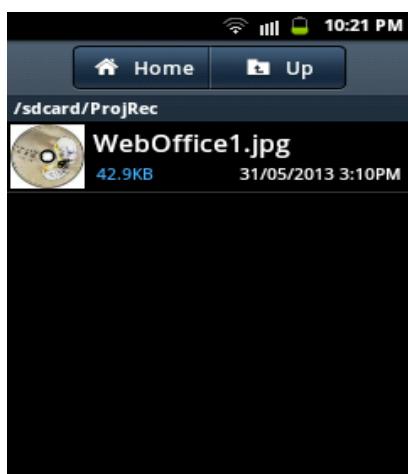


Figure 7 – File transferred to destination phone

VI. RESULTS AND ANALYSIS

We have been able to successfully create an interactive table which provides a natural user interface for sharing of data. This was achieved with the help of infra-red sensors to detect devices on the table, an image processing technique to provide a touch interface (imaging touchscreen) and android

applications for transfer of data from one device to another.

Also we have been able to create this table at a cost much lower than that for comparable existing technologies. The cost of our approach is projected to be nine to ten times lower as compared to the existing technologies which cost around \$10000 [11].

On the other hand the total cost of the solution provided here can be approximated to as shown in Table1.

TABLE 2 – Approximated Cost of STMC

COMPONENT	COST (Approx)
Glass Table	Rs.3000
Sensors	Rs. 3000
Arduino Kit	Rs. 1200
Processor	Rs. 25000
Web Camera	Rs. 500
Projector	Rs. 25000
Wi-FiRouter	Rs. 2000
Total	Rs.59,700 (\$1000 approx)

VII. CONCLUSIONS

We have created an interactive table using which we can transfer the desired content between two mobile phones by just dragging and dropping the icon of the file. When we place a mobile on the table; files from a particular folder of the phone are displayed on the table surface. As we move our fingertip, the file also starts moving with it and as soon as it reaches the vicinity of the other phone, it gets transferred to the other phone. We have experimented so far with various types of files like jpeg, png, mp3, pdf, doc, zip, mp4.

The major contribution of the work is its very nominal cost as compared to existing similar technologies. This would help in bringing surface technology into our daily lives.

VIII. FUTURE WORK

Future directions include use of such interactive tables in many other spheres of life. In the field of education, class desks can be replaced by such interactive tables to make learning a fun experience. In the field of hospitality such tables can be placed in hotels, restaurants etc. where users can choose from available options and also make payments. Such tables can also be used for entertainment purposes in the form of interactive Music Players, Photo albums and Board Games.

Also we can make such tables multitouch as well as compatible for more than two devices. The scope of the table can further be increased to compatibility with non-android devices. Moreover security is still an issue with such devices and hence there's a need to introduce some

security measures to protect a user's personal data when using such tables.

ACKNOWLEDGMENT

We would like to thank the Department of Computer Science and Information Technology of JIIT, Noida for giving us the permission to commence this project in the first instance and for providing us the necessary resources specially the hardware components.

REFERENCES

- [1] http://en.wikipedia.org/wiki/Natural_user_interface
- [2] Rajesh C, Muneshwara M.S, Anand R, Bharathi R, Anil G.N., "Surface Computing: A Multi Touch Technology (MTT)- Hands on an Interactive 3D Table", Coimbatore Institute of Information Technology Presented at the International Conference on Computing and Control Engineering (ICCCE), 2012.
- [3] http://en.wikipedia.org/wiki/Surface_computing
- [4] http://en.wikipedia.org/wiki/Microsoft_PixelSense
- [5] Allison Jeffers Gainey, "Microsoft Surface: Multi-Touch Technology", Winthrop University, Rock Hill, SC, USA, 2008.
- [6] <http://www.microsoft.com/enus/pixelsense/pixelseNSE.aspx>
- [7] <http://teslau.wordpress.com/2013/02/24/processing-g-arduino/>
- [8] RonyFerzli, Ibrahim Khalife, "Mobile Cloud Computing Educational Tool For Image/Video Processing Algorithms", Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE, 2011, pp. 529-533.
- [9] Shahzad Malik, "Real-time Hand Tracking and Finger Tracking for Interaction", University of Toronto, 2003.
- [10] <http://opencv.org/>
- [11] B.Sindhu&N.Sneha, "Surface Computing", Prakasam Engineering College, Kandukur, 2010.
- [12] Ankit Gupta, Kumar AshisPati, "Finger Tips Detection and Gesture Recognition", Indian Institute of Technology, Kanpur, 2009.
- [13] Jagdish Lal Raheja, Karen Das, Ankit Chaudhary, "An Efficient Real Time Method of Fingertip Detection", 7th International Conference on Trends in Industrial Measurements and Automation (TIMA 2011), CSIR Complex, Chennai, India, 2011.



Nitin Sharma is Master's candidate at the University of Florida Graduate School. His research interests include Mobile networking, cloud computing, mobile ad-hoc networking. He is also interested in developing applications for mobile platforms. He received his B-tech degree from Jaypee Institute of Information Technology at Noida, India.



Manshul V Belani has been recruited in Dell International Services as Svc IT Dev Program Sr Assoc II. Her interest areas include Web based application, Image Processing, Algorithms, Object Oriented Programming. She received his B-tech degree from Jaypee Institute of Information Technology at Noida, India.



Indu Chawla is a Senior Lecturer at Jaypee Institute of Information Technology at Noida, India. Her primary research interests are in Database and Algorithms. She received her B-tech degree from Punjab Technical University, Jalandhar and her M-tech degree in Information Technology from Indraprastha University, Delhi.