IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 4 (April. 2013), ||V1 || PP 06-16

Design of Low Cost Interactive Multi-Touch Table With Rfid Technology for Hotels

Rajat Verma
Department of Computer Science and Engineering, SRM University, Kattankulathur,Chennai, India

Abstract: With the introduction of multi-touch, a new form of human computer interaction is introduced. Due to recent innovations multi-touch technology has become affordable. Unlike interaction on a desktop computer multi-touch allows multiple users to interact with the same device at the same time. To demonstrate the vast possibilities of multi-touch technologies an '*interactive RFID-based Multi-touch Device'* can be constructed which can be utilised in a hotel. The objective of the project is to enhance the customer's dining experience and to provide an enjoyable and user friendly interface thereby reducing human effort and time. The multi-touch table constructed is a camera based multi-touch device which is designed using front-side illumination technique. Card associated with a RFID tag will be used, priced to a specific amount which is issued to the customer at the reception desk. Each order table at the restaurant will have a multi-touch device along with the RFID reader. Customer can now interact with the multi-touch device by showing his RFID card and place an order by selecting from the menu displayed on the order table. This project proposes the methodology of designing an interactive system along with applications to verify the effectiveness of the same.

Keywords: *multi-touch, RFID, interactive, front-side illumination*

I. INTRODUCTION

Multi-touch refers to the ability of a touch-sensing surface (usually a touch screen or a track pad) to detect or sense input from two or more points of contact simultaneously. Multi-touch consists of a touch screen (screen, table, wall, etc.) or touchpad, as well as software that recognizes multiple simultaneous touch points, as opposed to the standard touch screen (e.g. computer touchpad, ATM), which recognizes only one touch point. This effect is achieved through a variety of means, including but not limited to: heat, finger pressure, high capture rate cameras, infrared light, optic capture, tuned electromagnetic induction, ultrasonic receivers, transducer microphones, laser rangefinders, and shadow capture. Multi-touch displays are interactive graphics devices that combine camera and tactile technologies for direct on-screen manipulation. Multi-touch technology is not entirely new, since the 1970s it has been available in different forms. Due to the improvement of processing power of modern desktop computers, it no longer requires expensive equipment. Modern computer interaction consists of a monitor, keyboard and a mouse. Limited by the operating system, it allows us only to control a single pointer on the screen. With multi-touch, multiple input pointers are introduced to the system which all can be controlled independently. Depending on the size of the display multi-touch allows multiple persons to interact with the same display at the same time.

Radio-frequency identification (RFID) is the use of a wireless non-contact system that uses radio-frequency electromagnetic fields to transfer data from a tag attached to an object, for the purposes of automatic identification and tracking. It has been extensively applied to various fields such as product surveillance, supply chain, automobile parts tracking and telemedicine. A RFID system is composed of three major components, including reader, tag and middleware. Unlike traditional bar-code system, RFID systems can carry dynamic as well as static data. Some tags require no battery and are powered and read at short ranges via magnetic fields (electromagnetic induction). Others use a local power source and emit radio waves (electromagnetic radiation at radio frequencies).Tag contains electronically stored information which may be read from up to several meters away. Unlike a bar code, the tag does not need to be within line of sight of the reader and may be embedded in the tracked object.

The project consists of a multi-touch table integrated with a RFID reader. A card associated with a RFID tag will be used. Two computer systems will be required, one to be used as a client and other one as server. Network connectivity needs to be established between the reception desk, multi-touch table or order table and chef desk. This whole setup can be utilised in a hotel/restaurant. A customer can buy a card (with the RFID tag) priced to a specific amount which he thinks is required or necessary for future transactions. Each table in a restaurant will have a multi-touch device along with the RFID reader. Customer will show the card on this device and his details will be displayed along with the menu card on the table. He can then place the order by selecting from the menu according to the balance left in his card. This order will be sent to the chef's system. The chef's system will be setup in such a way that he will be able to identify the table from which the request has been made and confirm the order. Also the estimated time required for processing the order will be sent to the multi-touch table from which the customer gave the order. Meanwhile, the customer can interact with the various touch applications like playing piano, using finger gestures to create ripple effect, solving a picture puzzle etc.

II. SYSTEM OVERVIEW

In this section, we will overview the proposed rfid-based multi-touch interactive system from two different angles. Figure 1 shows the structure of the proposed rfid based interactive display system. One is the hardware configuration that contains two main components, RFID reader and multi-touch table. The rfid reader is user to identify the user or customer which uses the rfid tag and the multi-touch table is an interface for user to operate the proposed interactive display system. The other part is from software viewpoint where the client and server are connected in an *ad-hoc network* and applications are run on both sides along with the blob tracker for multi-touch.

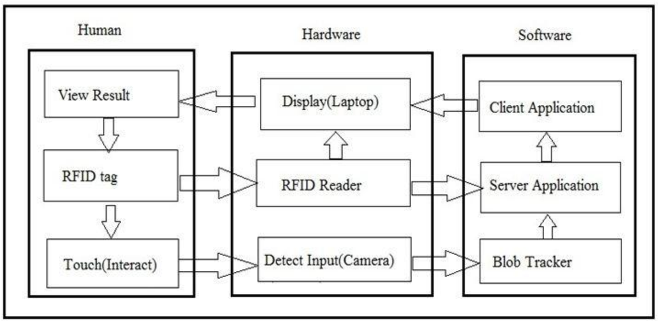


Fig. 1 A graphical overview of RFID-based multi-touch interaction pipeline

Figure 2 shows the flowchart of the rfid-based interactive multi-touch system. The receptionist at the reception desk will login into the system by entering the appropriate username and password. The customer arriving at the Reception Desk has two options: New Registration, for a new customer or Recharge for an existing customer. Accordingly details like first name, last name, email id, mobile no, amount and the rfid tag unique identification number for the customer will be added or updated in the existing database of the hotel. Now the customer is ready to place an order at the restaurant. The rfid reader is connected to the system available at the customer side which is also in conjunction with the multi-touch table. At the same time connection needs to be established between the customer table and the chef system using adhoc(computer-to–computer) network by manually configuring the IP addresses of both the systems. The customer will show his rfid card issued to him before at the reception to the rfid reader at the restaurant table. As soon as she shows the card, user details will be displayed including the amount present in his account along with the menu card. Now the customer will place the order by selecting different options available on the menu using the multi-touch table according to the amount available in his account. As soon as he places the order, the ordered amount will be deducted from customer’s account updating the database and timer will start showing the estimated time required for the service to be processed. The customer’s request will be sent to chef system. After receiving the order, estimated time required for processing the request (preparing the order) will be sent to the customer table by the chef.

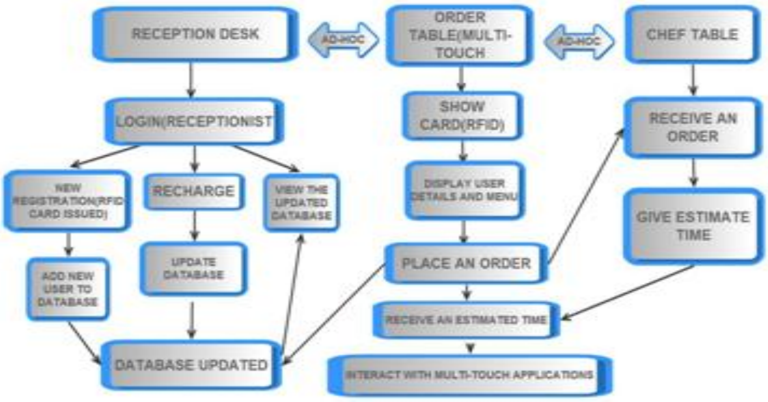


Fig. 2 A flowchart of the proposed interactive RFID-based multi-touch device.

III. HARDWARE AND SOFTWARE CONFIGURATION

This project aims at developing the rfid based multi-touch interactive system which combines display technology with sensors which are capable of tracking multiple points of input. The idea is that this would allow users to interact with the computer in a natural way .Using this display more than one user can interact with the system simultaneously.

Hardware and software specifications for this project are as follows:-
Hardware Requirements:-

- Web Camera of minimum resolution 640 X 480
- Acrylic sheet or glass(thickness ~6-10 mm)
- Two computer systems with minimum of 1 GB RAM.
- Light source for increasing light intensity when ambient light is not sufficient and to overcome shadow problem.
- RFID-Reader module with RS232 output.
- RFID tag/cards.
- USB to serial adapter.
- A display projector (if needed).

Software Specifications:-

- Touchlib, free open source multi-touch framework libraries like CONFIG.BAT, GATEWAY.BAT AND SERVER.BAT are required for configuring this multi-touch device.
- VISUAL BASIC-6.0 and Video Ocx Tools have been used in application development for the multi-touch device.
- Hyper terminal to test connection with rfid reader.
- Client and server side applications using VB6.0 for reception, table and chef desk.
- Windows socket programming used for both client and server side applications.

3.1 Design considerations of the RFID-based multi-touch device.

There are various techniques available for the construction of the multi-touch device. But from the starting of this project, my prime area of consideration was the camera based multi-touch techniques. The camera based multi-touch techniques have been widely utilized in various multi-touch devices. Microsoft Surface which released previously and created a lot of fuss was based on a camera based multi-touch technique (precisely saying rear-side illumination technique).Front-side illumination technique is camera based multi-touch technique but because of its impreciseness and low reliability, its potential has not been fully utilized. In my project I have used front-side illumination technique only but with certain modifications so as to increase the reliability and preciseness of the technique. Before designing a multi-touch device one has to consider various factors such as the environment in which the device would be used, the target audience, the cost which would be involved in the construction of the device, etc. The following considerations were kept in mind while making the device: There may be a case when the surrounding light may not seem sufficient. To tackle that case an array small bulbs are put on the boundary of the touch sensitive surface. The cost of construction of the device needed to be kept low; hence I decided not to use the projector. Instead the camera inputs are directly supplied to the computer.

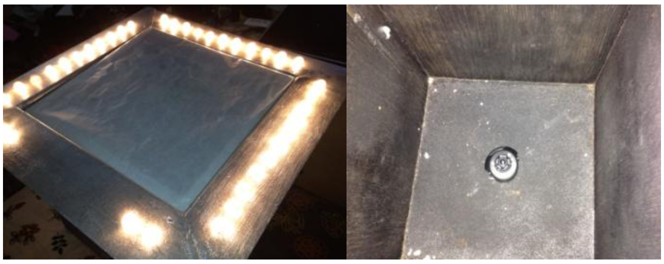


Fig. 3 Multi-touch table outside and inside view.

3.1.1 Camera-based multi-touch techniques

Camera based multi-touch devices share the same concept of processing and filtering captured images on patterns. The pipeline begins when the user views the scene on the panel and decides to interact. To interact with the device, the user touches the device panel. On the hardware level the camera registers touch. Because the captured frame might not only include the contact points but also a (static) background, it is required to perform image processing on each captured frame. The captured frame is converted to a gray scale image and the static background is removed by subtracting the current frame with a reference frame. As a result the frame only shows white contours (blobs) which are the points of contact. By tracking these blobs the system becomes capable of sensing touch. In order to track the blobs, the positions are transmitted to the blob tracker which matches blobs from earlier frames with the current frame.After processing, events will be triggered which can be used in a multi-touch capable application. The application modifies objects in the current scene according to the new blob positions. The result is returned to the user through the display. The performance of a camera based multi-touch device depends on the used hardware and software. When a user touches a multi-touch device, it expects the device to respond directly. The responsiveness of the device depends on the time it needs to process the user's input and present the users a result through the display. In the interaction pipeline two levels are important, the hardware and the software level. Using a camera which is capable of 30 frames per second allows smooth interaction with the system. However, this requires a system that can handle image processing and blob tracking in 1/30 of a second. A combination of smart algorithms implemented in software and fast hardware helps to minimize the latency and increase the responsiveness of the device.

3.1.1.1 Front-side illumination technique

A technique for designing a touch interface, based on diffused illumination is known as Front-side Illumination (FI). This technique is based on light being diffused. However instead of using infrared light sources, it depends on the ambient light from the environment. With FI the diffuser is attached to the front side of the display. The ambient light illuminates the diffuser, which from the camera's point of view, results in an evenly colored rectangle. By touching the surface, shadows will appear underneath the fingertips because the ambient light cannot reach it. These points are detected by the camera and a touch is registered. Figure 4. shows a basic setup for FI. Because the used touch library requires touched spots to be colored white, a simple invert filter is being applied. FI can be seen as the cheapest way of creating a camera based multi-touch capable device.

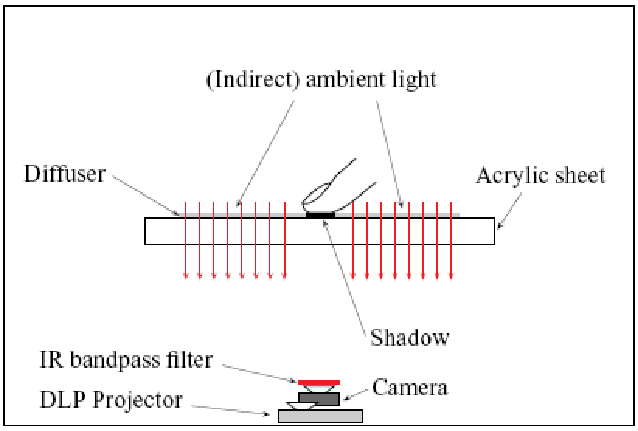


Fig. 4 Schematic view of the multi-touch panel using front side illumination

3.1.2 Choosing an appropriate camera device

When choosing a camera it is recommended to find out which sensor is used and whether the data sheets are available for this sensor. Whilst high-end consumer USB cameras are capable of transmitting images of VGA resolution (640X480 pixels) at reasonable frame rates, they often introduce latency. Because this reduces the responsiveness of the multi-touch devices, it is recommended to use a FireWire based camera instead. Depending on the size of the display and the projected image it is recommended to use at least a camera running a VGA resolution because of precision. The frame rate should be at least 30 frames per second to allow smooth interaction.

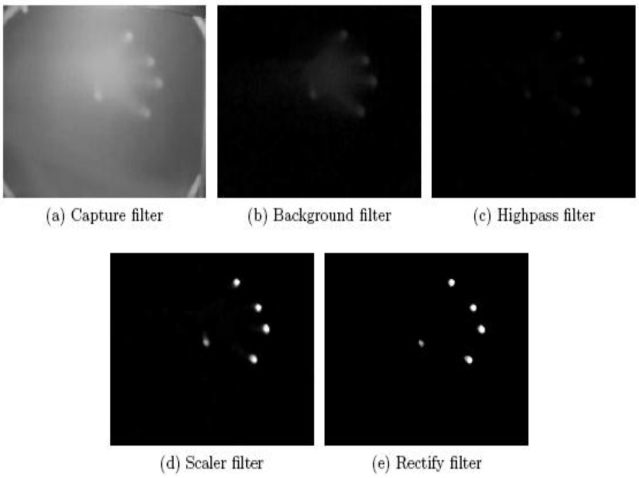
3.2 Multi-touch detection and processing.

To perform camera based multi-touch detection and processing several frameworks are available. In this paper we describe the used multi-touch framework, how a multi-touch framework connects to a multi-touch application and the different types of gestures.

3.2.1 TouchLib

Our multi-touch system uses Touchlib which is a free open source cross platform multi-touch framework which provides video processing and blob tracking for multi-touch devices based on FTIR and DI. Video processing in Touchlib is done through the Intel's OpenCV graphics library. Touchlib currently runs on

MS Windows, Linux and Mac OS X. When Touchlib is used for the first time (in an application) it stores a copy of the current frame into the memory. This frame is used as a reference frame and used to perform background subtraction on the next frames. Front-side illumination requires more video processing filters before blob tracking can be applied. First a capture filter is selected depending on the used interface (USB or Firewire). If the system uses FI it is required to add an invert filter because Touchlib expects white colored blobs. The next filter is the background subtraction filter. After removing the (static) background a high pass filter is applied. This filter compares the contrast in the image and only allows `high' values to pass through (depending on a pre-set value). As a result only the touched spots will be visible. Depending on the used diffuser and infrared LEDs the result might show weak blobs, in this case we can add a scalar filter to amplify the brightness of the blobs. Finally, the rectify filter is applied resulting in clear blobs ready for tracking. It is used to filter out noise and reduce the gray scale image to a black and white image only displaying the actually touched areas.
An example of the Touchlib image processing results can be found in Figure 5.


(a) Capture filter (b) Background filter (c) Highpass filter

(d) Scaler filter (e) Rectify filter

Fig. 5. Example of Touchlib filter chain using front side illumination

3.2.2 Blob detection and tracking

In Touchlib the blob tracker handles the blob detection and tracking. Blob detection is done to detect touch in a camera image. In order to follow the movement of touch, the blob tracker compares the detected touch locations in a frame with the positions of the previous frame. In each time step, Touchlib requests a frame from the video camera. After a frame has been processed, the resulting image is used to perform blob tracking. By using the OpenCV function cvFindContours() we obtain a list of contours found in the current frame. All found contours (and their properties) are stored in a contours list. Each contour is checked on whether it is a fiducial marker or a touch. On each contour Touchlib tries to fit a polygon which matches the outlines of the contour. If the fitted polygon is build out of four points (we are talking here of square fiducial marker) it might be a possible fiducial marker. Next it will check whether the angle of all the corners matches approximately 90 degrees. If the result is true, the contour is considered a fiducial and the position (center) will be calculated. It will also assign a unique tag identifier based on the pattern found within the square. If the polygon is more complex than four points, it is assumed to be a touch. Touchlib will fit an ellipse on the contour. The properties of the ellipse are used to determine the position, orientation and the size of a blob. If the size of the blob fits the minimum and maximum requirements on height and width, the blob will be added to the blob list.

In order to track blobs it is required to have at least two data sets that contain blobs in different states. We first define the two data sets.
The first data set contains the blob list from a previous frame and is defined as follows:
$p_1; p_2; p_3;\ldots\ldots\ldots\ldots\ldots; p_n$
where n is the number of active blobs in the previous frame. The second set contains the blobs list of the current frame and is defined as follows:
$c_1; c_2; c_3;\ldots\ldots\ldots\ldots\ldots; c_m$
where m is the number of active blobs in the current frame.
After each frame, the data from set p is replaced by set c. Set c will be filled with the new blob list.

3.2.3 Generating events

In the last part of the blob tracking, Touchlib prepares the events to be triggered. First Touchlib walks through the list of currently active blobs. If a blob is new in the current frame (it did not match with a blob in the previous frame) it will contain an ID with the value -1. In that case a new ID will be generated and assigned.

After the assignment a “touch event” of the blob will be dispatched. If a blob in the current frame matches a blob of the previous frame, it will use the same identifier of the previous frame. Depending on the amount of movement an update event will be dispatched. If the delta of movement is higher than the set value of distanceThreshold it means the blob ‘traveled’ over a larger distance than one would assume to be correct. In this case instead of an update event a “touch event” is dispatched instead. If the delta of movement from the current and the previous frame is below the value of minimumDisplacementThreshold, no update event will occur. After Touchlib processed the list of currently active blobs, it compares the list with the blob list of the previous frame. If a blob was active in the previous frame but has not been found in the current frame, the blob is marked for deletion and a “touch up event” will be dispatched.

3.2.4 Multi-touch applications.

There can be various touch applications which can be created to work with the multi-touch table which is capable of tracking multiple points of input. The customer can interact with these applications while his order is being processed. I have used many such applications to demonstrate the usage of the interactive surface. Figure 6(a) shows how the touch surface can be used to solve a picture puzzle where the user has to arrange the tiles in puzzle using touch gestures to get the desired picture.

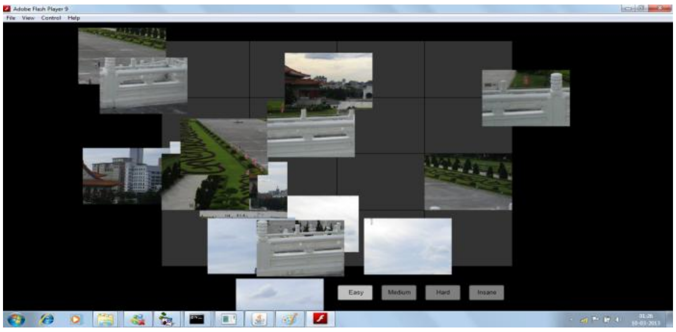


Fig. 6(a) Picture Puzzle

The next figure shows the piano application where user can interact and play with the piano using his fingers.

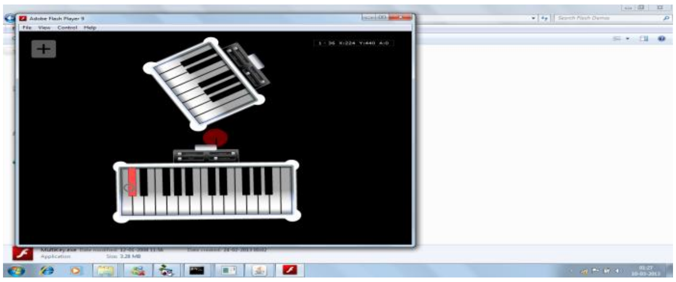


Fig. 6 (b) Playing piano.

Another application which demonstrates the usage of finger gestures with the multi-touch surface can be seen in Figure 6(c) where the user interacts with the pictures and can zoom in and out, pan, pinch, and rotate the photos clockwise or anti-clockwise and perform various other gestures.

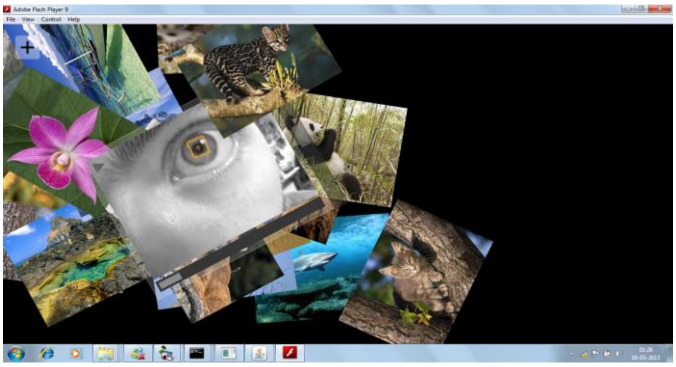
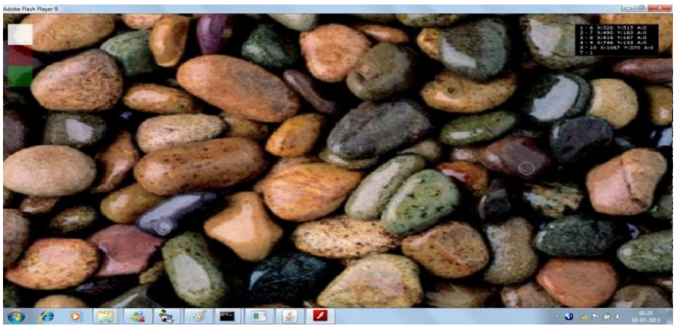


Figure 6(c) Usage of multi-touch gestures with photos.

Furthermore, an application demonstrating the ripple effect is shown in Fig 6(d) which we seen in most of the android-based smart phones where ripples are created as the user interacts and touches the screen or surface. Also note the coordinates of the various finger touch points are tracked by the touch table and shown at the upper right corner of the ripple application.



```
1 - 6 X:326 Y:515 A:0
2 - 7 X:490 Y:182 A:0
3 - 8 X:918 Y:187 A:0
4 - 9 X:746 Y:133 A:0
5 - 10 X:1067 Y:370 A:0
T - 1
```

Figure 6(d) Ripple Effect application.

3.2.5 Lens Correction

Our current system is using a wide-angle lens which suffers from a radial distortion. Because Touchlib tries to correct a non-linear radial distortion with a linear method based on grid points, it will fail near the corners of the screen. This is the result of the increasing distortion when a point is further away from the image center. Fortunately multiple algorithms are available to correct a radial distortion caused by the lens. For our system a new Touchlib filter was developed which applies image correction on the video stream. The filter uses standard functions from the OpenCV library. Before it is possible to correct an image frame it is required to obtain more information about the camera parameters. The camera parameters describe the camera configuration and consist out of two parts, namely the intrinsic and extrinsic camera parameters. The intrinsic parameters describe the properties of the camera which include the focal length, the image center (principle point, the effective pixel size and the radial distortion coefficient of the lens. The extrinsic parameters describe the relationship between the camera and the world. These include the rotation matrix and translation vector. In order to find the camera parameters it is required to calibrate the camera by presenting it a known pattern from different angles and positions. In our case we used a chessboard pattern which is obtained from the OpenCV examples. The pattern is printed on an A4-sized paper and glued onto a steady surface. By presenting the chessboard pattern to the camera to the OpenCV calibration tool, it will recognize the pattern and assign coordinates to the intersections of the chessboard pattern. By showing multiple views of the pattern from different angles, the tool tries to map all the coordinates in a three dimensional space. From the mapping of these points, the tool is capable of calculating the intrinsic parameters and the distortions coefficients.

3.2.6 Gesture based Interaction

In comparisons with traditional desktop computing with mouse and keyboard, a multi-touch device provides additional input methods. Instead of working as a point and click device it can be improved with gesture based interaction. In the definition of direct gestures we describe a gesture pattern that allows a user to manipulate objects directly. The manipulation can be performed with multiple pointers. For example, when using a multi-touch system it is very common to use a pinching like motion in order to change the size of an object. Depending on the distance between two fingers, the object increases or decreases in size. Examples of common used direct manipulation gestures are shown in Figure 7.

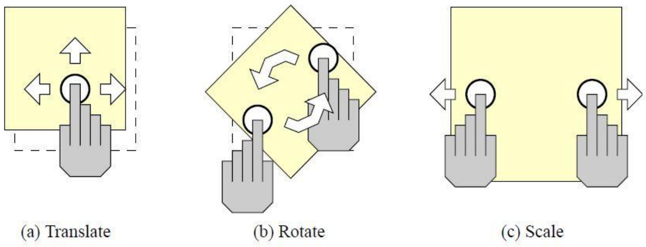


Fig.7 Direct manipulation gesture set used for object manipulation.

Symbolic gestures are patterns based on the gesture location and trajectory. Patterns can be made in any shape such as triangles, circles or even text characters. Symbolic gestures are commonly used in applications to simplify tasks. An example can be found in modern web browsers that feature mouse gesture recognition. When pressing a predefined button, the mouse gesture recognition mode is enabled. The user can now draw a gesture on the browser window. After releasing the button, the gesture recognition engine will try to match the gesture pattern and complete the task which the gesture is bind to (such as switching between pages and tabs). Neural networks are commonly used for pattern recognition. A basic feed forward neural network (FFNW) consists out of three layers the input layer, the hidden layer and the output layer. Before it is possible to use a FFNW for gesture recognition, it is required to define a data format which will be fed to the FFNW. For symbolic gestures path information is stored as a list of x and y values. Depending on the precision of the system, the path list can increase in size rapidly. To reduce the complexity, the number of entries is reduced. The entries in the path list will now represent a rough estimation of the original path. For each point in the path the angle is calculated. The input layer of a FFNW contains 16 neurons, the same number as points we reduced our path list to. Each angle value is passed to one neuron of the input layer. Internally each neuron from the input layer is connected to all of the neurons in the hidden layer. The neurons of the hidden layer contain a value (weight). Next each of the neurons in the hidden layer is connected to one of the neurons of the output layer. The number neurons in the output layer depend on the number of gestures that are required to be recognized. When a gesture is performed a list of angles is passed to the FFNW. The value of the input layer is now multiplied with the value of the hidden layer. The result is passed to the output layer. Since each gesture results in a unique value in the output layer, it is possible to recognize patterns. When training the FFNW with gesture pattern the weight values in the hidden layer are adjusted. The training is completed when each pattern only triggers one neuron in the output layer. A popular gesture recognition program for the X Window System is Xstroke. Xstroke is a full-screen gesture recognition program which allows users to bind commands to a gesture. The technique which Xstroke uses to recognize gestures is region based. Internally Xstroke holds a 3x3 grid which is numbered from 1 to 9. The grid is centered on the gesture and will scale to the extends of the obtained gesture path. The last gesture recognition technique that will be discussed is based on direction. When drawing a gesture, the path can be described as a sequence of directions. In our example we limit the number of possible direction to eight. Each direction is numbered from 0 up to 7 (counted in clockwise order, the first direction is pointing to the right and equal to zero). When drawing a gesture shaped as an N, the ideal direction sequence would be 616 which is equal to up, downright, up. Depending on the accuracy of the input device and user, the recorded sequence might match or not. In order to compare both sequences the Levenshtein cost method is used. The Levenshtein cost method compares the recorded sequence with the list of known sequences from the gesture library. For each entry in the gesture library a Levenshtein distance value will be calculated which is based on how similar both strings are. The match with the lowest Levenshtein cost will be selected as the preferred match. By specifying a maximum cost value it is possible to discard wrong matches.

3.3 RFID Reader/RFID tag

An RFID system consists of a tag reader (also called the interrogator) and a tag. All communication between the tag and reader occurs completely through a wireless link that is sometimes called an air interface. Through a sequence of commands sent and received between both devices (called the inventory round), an RFID reader can identify the electronic product code (EPC) of an RFID tag. For passive tags, the basic idea is that the interrogator initiates an interrogation round with a query command. The query command essentially “wakes up” the tag, which responds with the appropriate information. Figure 8 shows a basic block diagram of the tag/reader system. Figure 9 shows the RFID reader with RS 232 port used in the project.

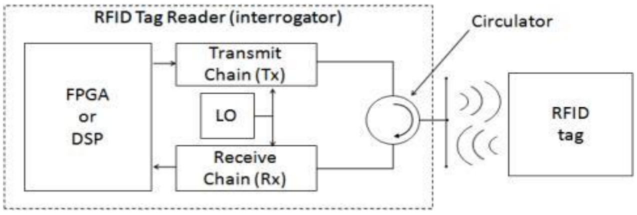


Fig. 8 Block diagram of typical RFID tag/reader system

Fig. 9 RFID reader with RS 232 kit.

Note from Figure 8 that many RFID readers and measurement systems actually use a three-port RF component called a circulator that gives both transmit and receive front ends the ability to use the same antenna. Note that with many RFID, standards, timing information between transmit and receive commands is defined by strict guidelines. In fact, a sort of “handshaking” is required between the tag and reader to complete an interrogation round. This actually creates a unique test challenge because the instrumentation must be capable of the same behaviour. On an interrogator, an embedded processor is required to decode and generate commands within a tight timing interval. As discussed in a later section, this design is quite similar to field-programmable gate array (FPGA)-enabled RFID measurement systems, which use similar embedded processing to fully emulate either a tag or a reader. *RFID tags* come in many shapes and sizes each suited to a specific application, but all RFID tags can be generally grouped into two main categories, regardless if they are encased, a sticky label or just a solid button like tag .RFID tags are either “*passive*” (no battery) or “*active*” (self-powered by a battery). RFID tags also can be read-only (stored data can be read but not changed), read/write (stored data can be altered or rewritten), or a combination, in which some data is permanently stored while other memory is left accessible for later encoding and updates. *Passive RFID Tags* have no internal power supply. An electrical current induced in the antenna by the incoming radio frequency signal provides just enough power for the integrated circuit (IC) in the RFID tag to power up and transmit a response. RFID tags communicate in various ways with the RFID reader. The aerial (antenna) has to be designed to both collect power from the incoming signal and also to transmit the outbound signal. Lack of an onboard power supply means that the passive RFID tag can be quite small: commercially available products exist that can be embedded under the skin.

In this project I will be using a *passive RFID tag*. These do not require batteries and have an unlimited life span. As we have already seen there are two important components of a RFID tag – A microchip and a coil (antenna). The antenna receives power and RF signals from the RFID reader and sends those signals to the chip. The chip receives those signals, computes them and sends back the data to RFID reader. We can figure out the precise working of a RFID tag through this diagram.

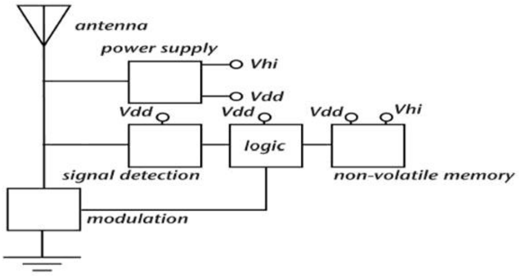


Fig. 10 Working of RFID tag.

To recognize the identity of an *RFID tag*, RFID reader sends radio signals which is captured by the coil (working as antenna) for the tag. The coil receives these signals as alternating current and passes to the chip. The chip extracts both the power and the information from this alternating current. By communicating with the non volatile memory of the chip that stores unique id as well as other information, it sends back the required signal to the antenna which is then transmitted to the RFID reader. There are three main roles a RFID reader plays other than signalling RFID tag to transmit desired information back to the RFID reader. Firstly, a RFID reader has the responsibility of keeping RFID tags powered up. Secondly, a RFID reader demodulates incoming signals from the RFID tag down. This process slows the incoming signals down enough so that the RFID reader is able to process the signals. Finally, after the incoming signals are slowed down, RFID then has the responsibility of decoding the incoming signals into the words people can interpret.

3.4 Usage of AD-HOC Network.

An ad-hoc network is a wireless local area network that is built spontaneously as devices connect. Instead of relying on a base station to coordinate the flow of messages to each node in the network, the individual network nodes forward packets to and from each other. It typically refers to any set of networks where all devices have equal status on a network and are free to associate with any other ad hoc network device in link range. In the Windows operating system, ad-hoc is a communication mode (setting) that allows computers to directly communicate with each other without a router. Here I am using the option ‘Set up an ad-hoc (computer to computer) network’ available in Windows 7 Operating System. To connect the systems using ad-hoc, I am manually configuring the IP addresses of the systems by changing the adapter settings of wireless network connection(configuring the IPv4 settings). By doing such, the systems corresponding to chef table and order table(multi-touch table) get connected in an ad-hoc network. The dynamic change of data in the corresponding database takes place with respect to transactions incurred by the customer using his RFID card along with the notifications being sent to both the customer and the reception desk.

3.5 Programming Language Interfaces/Working Modules

The client side or reception desk module will consist of programs and algorithms which are written in accordance with communication performed over a ad-hoc network. The serial communication ports will be able to transfer and receive the data via the RFID reader and the same informed can be sent via the network to the server side or the chef system using windows socket programming. The coding parts are written in Visual Basic 6.0 which creates several forms or applications to be used by the interactive system and are implemented in conjunction with the Touchlib software designed for the multi-touch table. The three modules for the reception desk, multi-touch table, chef table are integrated using the algorithms specially designed for this purpose and various forms which demonstrate the use of such an application is shown in the project. Figure 11 shows the various modules developed for *RFID integrated multi-touch device*. The *first module (reception desk)* registers the new customer and even has the option of recharging the card for an existing customer. The *second module (order table or multi-touch table module)* depicts the system where the menu is displayed at the touch interface and the customer can place the order using the touch table by performing one click gesture for the dish to be selected along with specifying the quantity of the same. The corresponding transaction for the order placed will be deducted from his RFID card and the total amount will be displayed. The placed order will be sent to the *chef table* where he can view the details for the order like from which table the order has been placed along with the quantity specified for each of the dishes. The customer will also be notified of the estimated processing time for the order during which he can interact with the multi-touch table and play with the various applications shown in the previous section.

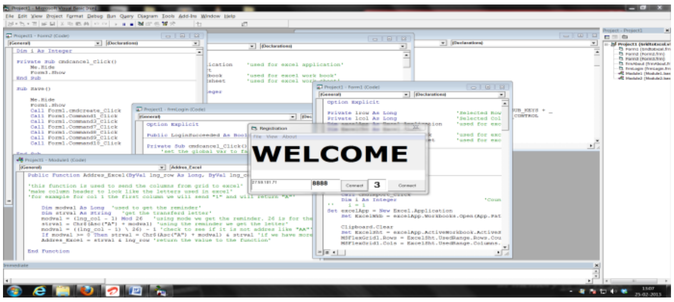
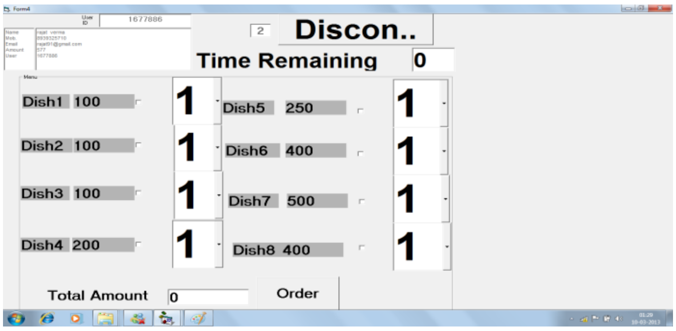


Figure 11(a) Reception Desk module.



User ID 1677886

Name rajat verma
Mob. 8939325710
Email rajat91@gmail.com
Amount 577
User 1677886

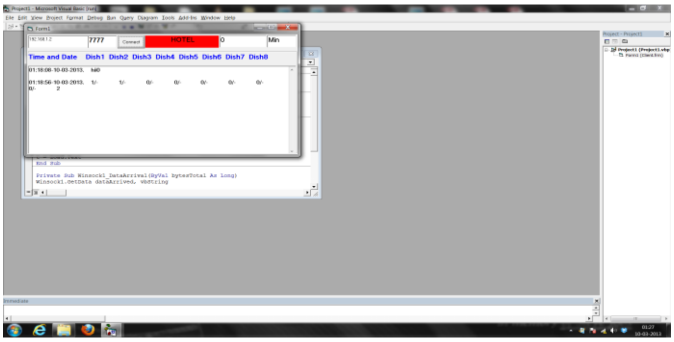
2

Time Remaining 0

Dish1	100	1	Dish5	250	1
Dish2	100	1	Dish6	400	1
Dish3	100	1	Dish7	500	1
Dish4	200	1	Dish8	400	1

Total Amount 0

Figure 11(b) Order Table (multi-touch) module


Project1 - Microsoft Visual Basic [run]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Form1
Connect
Immediate

192.168.1.2 7777 HOTEL 0 Min

```
Time and Date   Dish1 Dish2 Dish3 Dish4 Dish5 Dish6 Dish7 Dish8
01:18:08-10-03-2013,   hi0
01:18:56-10-03-2013,   1/-   1/-   0/-   0/-   0/-   0/-   0/-
0/-      2
```

```
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Winsock1.GetData dataArrived, vbString
```

Figure 11(c) Chef Table module

IV. CONCLUSIONS AND FUTURE WORK

A novel interactive system that integrates RFID and multi-touch technologies is proposed in this paper. Customers can now have a whole new experience by using the touch interface for ordering their food at restaurants and paying via the RFID card issued to them. With the construction of our own camera based multi-touch table we have demonstrated that multi-touch technology has become affordable. The precision, reliability and responsiveness of a multi-touch table depends on the used hardware. Because of the use of a camera based multi-touch technology, ambient light has a large influence in the tracking performance. Compared to desktop applications, users were able to manipulate objects in a natural way of touch. Instead of correcting the entire camera image, it is also possible to apply position correction on the output of the blob tracker. Currently image processing is done on the CPU. In order to reduce the load on the CPU it is possible to use graphics processing units (GPU) for image processing. The current blob tracker used in Touchlib is not scalable. Depending on the used hardware, Touchlib will fail to track fifty or more blobs in real-time. A smarter algorithm would not only take the distance into account but also the region and direction of a blob. It would also be interesting to use the GPU for blob tracking. Based on our own experiences the touch surface should be improved. The friction with the acrylic makes it hard to use the system for a longer period of time

.

V. ACKNOWLEDGMENTS

This research project would not have been possible without the support of many people. The author wishes to express his gratitude to his supervisor, Prof.C.Malathy of Computer Science and Engineering Department who was abundantly helpful and offered invaluable assistance, support and guidance. The author wishes to express his love and gratitude to his beloved families for their understanding & endless love, through the duration of his studies.

REFERENCES

[1]. Han, J.: Low-cost multi-touch sensing through frustrated total internal reflection. In: UIST ’05: Proceedings of the 18th annual ACM symposium on User interface software and technology, New York, ACM Press (2005) 115–118.
[2]. RFID :A Technical Overview and its application to the enterprise by Ron Weinstein.
[3]. Jefferson Y.Han Multi-touch Interaction wall In Siggraph ’06:ACM Siggraph 2006 Emerging technologies.
[4]. Jazz Mutant .Lemur Input Device.2006
[5]. Didier Brun .Mouse Gesture Recognition 2007 URL http://www.bytearray.org/?p=91
[6]. Touchlib:www.multigesture.net
[7]. Experiences with interactive multi-touch tables : Wim Fikkert, Michiel Hakvoort, Paul Van Der Vet
[8]. http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=4 for RFID working.
[9]. http://www.engineersgarage.com/insight/how-rfid-tag-works?page=5 Rfid tags.
[10]. Hyperterminal http://help.expedient.net/dialup/hyperterminal.shtml
[11]. Configuration http://www.avitresearch.co.uk/USBTTL/hyperterminal.html
[12]. http://www.tradeindia.com/fp641264/Rfid-Reader-Kit-Rs232-Interface.html RFID reader kit with RS 232 interface

ABOUT THE AUTHOR

Rajat Verma, a final year Computer Science and Engineering student from SRM University is an Oracle Certified Java Associate and Microsoft Technology Associate. He has developed several learning based applications for Texas Instruments, the World leader in Digital and Analog Technology. Recently his Research Project has been awarded with ‘Best Engineering Project’ at International Project Competition and Exhibition, Chennai.