Using NFC Technology for Fast-Tracking Large-Size Multi-Touch Screens

Mario Munoz Organero, *Member IEEE*, Samuel King Opoku

Abstract—**Multi-touch technology interfaces are becoming ubiquitous since their integration in smart-phones, tablets and many consumer electronics devices. Depending on the size of the tangible interface, the computing resource restrictions and the resolution required, there are different technological alternatives for multi-touch screens such as camera based systems, capacity, resistive or pressure sensing systems or LED based systems. This paper presents a different alternative to deploy medium to large size multi-touch screens with low computational requirements for tracking the user movements based on the Near Field Communication (NFC) technology. The paper makes a review of the main related technologies to create medium to large size multi-touch screens identifying their pros and cons and proposes a different design based on NFC tags and mobile readers. A multi-touch algorithm is designed and implemented.**

***Index Terms*— NFC, Multi touch, Bluetooth, J2ME, Algorithm, SDK**

I. INTRODUCTION

Multi-touch refers to a touch-sensitive device that can independently detect and optionally resolve the position of two or more touches on a screen at the same time. Multi-touch screens capture and react to simultaneous interactions of several fingers or hands of the users allowing user-friendly intuitive human-computer interfaces. Depending on the size of the tangible interface, the computing resource restrictions and the resolution required, there are different technological alternatives for multi-touch screens such as camera based systems, capacity, resistive or pressure sensing systems or LED based systems [3]. Camera based systems [6]-[9] use a camera either above or behind a screen to recognize movements on the screen. Recently, the Frustrated Total Internal Reflection (FTIR) is used for obtaining higher resolution [10]-[12]. Capacitive sensing systems [13]-[15] detect the capacity of the screen by means of antenna sets. Infrared LED based systems consist of infrared transmitters and receivers [16]-[18]. Camera based systems present low resolution and high computational requirements for image processing but are able to cover large-size screens. Capacitive sensing and infrared systems are better adapted for small to medium size screens but do not scale for large size screens. This paper presents an NFC based multi-touch system that localizes and tracks the movements of the hands of the user in a medium to large projection screen by using NFC phones that read NFC tags behind the screen. The NFC enabled mobile phones will run a J2ME application that reads the IDs of the NFC tags and uses Bluetooth to send them to the computer controlling the projection system. A Java based desktop application will receive the NFC tag IDs and perform gesture and shape identification of each of the hands of the users and consequently control the projection system.

NFC is an intuitive way of mobile communication that has achieved high level of consumer appreciation. NFC operates in three modes, namely, Tag emulation, Read/Write and Peer-to-peer. However only one mode can be manually selected and work with at a time [1]. J2ME is the Java platform used in order to develop NFC application for mobile phones in most of the pre-commercial devices. J2ME includes, among other configurations, CLDC which is supported by MIDP profile. Pre-commercial SDK's widely used for NFC applications are Nokia 6131 NFC SDK 1.1 and Series 40 Nokia 6212 NFC SDK. Both SDKs support Eclipse and NetBeans IDEs but the former contains additional Java packages which are implemented as stubs whereas the latter contains NFC Manager [2].

Among the APIs needed for NFC application is the Contactless Communication API (JSR 257). This supports tag emulation and read/write modes operation. Coupled with the Security and Trust Services API (JSR177), the tag emulation can be done through APDU communication. An extension of Contactless Communication supports peer-to-peer operational mode of NFC [5]. If Bluetooth is required, the Bluetooth API (JSR 82) has to be integrated with the previously mentioned APIs. The limited capabilities of mobile devices such as memory and processing power sometimes make it necessary to support mobile devices with backend servers. J2ME supports J2SE and J2EE as backend servers. Mobile phones thus communicate with the server through Bluetooth for short range communication technologies. Bluetooth does not require line-of-sight communication unlike infra-red.

II. NFC MULTI-TOUCH SYSTEM

The basic infrastructure for the NFC multi-touch proposed system is showed in figure 1. It consists of at least 2 NFC mobile phones, a PC with a Bluetooth dongle and a projector. See figure 1. The rear view of the projecting screen is captured in figure 2. The projecting screen is filled with NFC tags that can be read by the user NFC mobile devices. The user touches the projecting screens with the mobile devices and each tag read from the rear of the screen is sent to the controlling PC using BT. Each mobile device sends data independently. The movements of each hand of the user can be autonomously tracked. Multiusers interactions are supported since the system can associate mobile device IDs to each user.

All authors are with the Telematics engineering department at Universidad Carlos III de Madrid (munozm@it.uc3m.es and samuel.k.opoku@gmail.com)

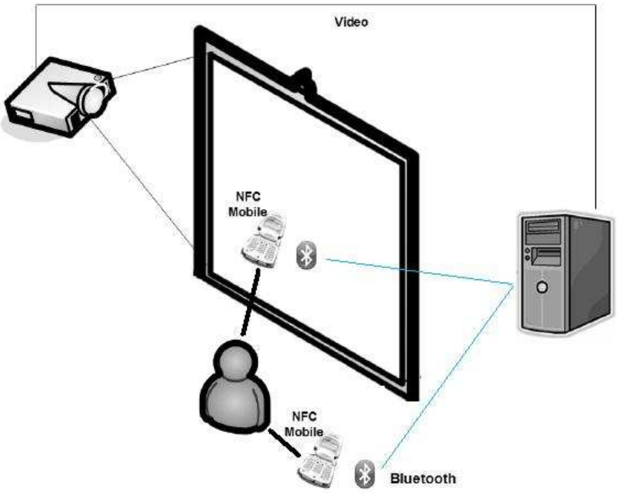


Fig. 1: NFC multi-touch system

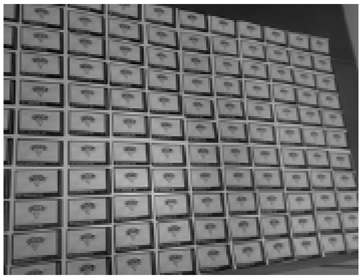


Fig. 2: Rear view of the projecting screen with NFC tags

The mobile phone application was implemented in J2ME (CLDC 1.1 / MIDP 2.0). The Contactless Communication API (JSR 257) was used for accessing NFC tags by the phones. The backend server application was implemented using J2SE and the Bluecove API was used for the communication with the Nokia NFC phones.

Some of the main features of this multi-touch system are:

- Large screens are supported
- Low computing resources required in the controlling part of the system since no image processing is needed to track the users' movements
- Multi-user environment

The multi-touch modeling and algorithm used to implement the system are described in the following sections.

III. NFC MULTI-TOUCH MODELLING

Given S as the number of tags covering the surface of the projecting screen a subset T, where $\mathbf{T \geq 16}$, are used for computing multi-touch interactions: two tags are used for implementing "Resize" and "Rotate" options and $\mathbf{M \geq 14}$ denote the number of tags used in multi-touching implementation. It follows that $\mathbf{M + 2 = T \geq 16}$ tags are used in calculations in order to implement the system. The close proximity nature [4] of NFC tags requires that a tag is closely placed adjacent to other to have a continuous plane. Figure 3 shows two examples of how the $\mathbf{M}$ tags are arranged (depending on the location of the NFC reading devices).

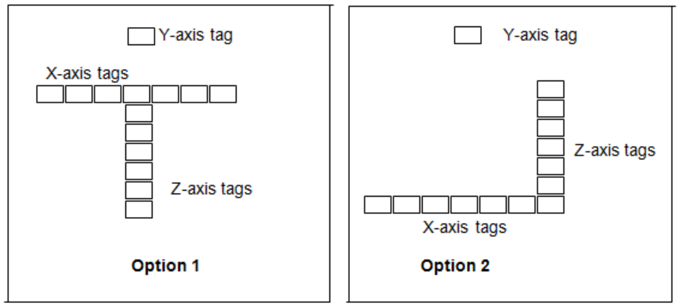


Fig. 3: Three-Dimensional Arrangements of NFC Tags

The UID's of tags are used to track the position and movement of the mobile phones. The tag UID read by the mobile phones at each position determines the coordinates (or location) of the device. The horizontal resizing is achieved by using x-axis tags whereas z-axis tags are used to achieve vertical resizing. Turning of images is achieved by both y-axis and x-axis tags.

Figure 4 shows different orientations for the x-axis and z-axis traversed tags as a generalization of figure 3.

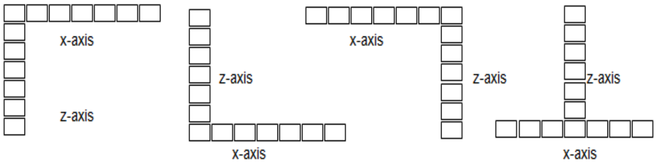


Fig. 4: Different Tag paths

IV. MULTI-TOUCH ALGORITHM

In a concise form, the multi-touch implementation is achieved through these three steps:

1. Authentication of Mobile Phone pair.

2. Touch the desired "Manipulate" tag – 'Resize' or 'Rotate'

   a. 'Resize': widen the space between the paired phones to enlarge the image. To reduce the size of the image, narrow the space between the paired phones. The phones always move over either x-axis or z-axis in opposite directions.

   Let the previous UID touched by mobile phones $\mathbf{m_1}$ and $\mathbf{m_2}$ respectively be $\mathbf{U_{m1}{}^{p}}$ and $\mathbf{U_{m2}{}^{p}}$ and the corresponding current UIDs respectively be $\mathbf{U_{m1}{}^{c}}$ and $\mathbf{U_{m2}{}^{c}}$. Using the position, P, of the UIDs such that

$$\llbracket\, \mathbf{P(U\rrbracket_{m1}{}^{p}) < \llbracket\, P(U\rrbracket_{m2}{}^{p})}$$

If

$$\llbracket\llbracket\, [\, \mathbf{P(U\rrbracket_{m1}\rrbracket^{c}) < \llbracket\, P(U\rrbracket_{m1}{}^{p})and\, \llbracket\, P(U\rrbracket_{m2}{}^{c}) < \llbracket\, P(U\rrbracket_{m2}{}^{p})}\,\big]$$

or

$$\llbracket\llbracket\, [\, \mathbf{P(U\rrbracket_{m1}\rrbracket^{c}) > \llbracket\, P(U\rrbracket_{m1}{}^{p})and\, \llbracket\, P(U\rrbracket_{m2}{}^{c}) > \llbracket\, P(U\rrbracket_{m2}{}^{p})}\,\big]$$

, the phones move in the same direction and resizing cannot be done.

If

$[[P(U]_{m1}]^{c}) < [P(U]_{m1}{}^{p}) \text{and} [P(U]_{m2}{}^{c}) > [P(U]_{m2}{}^{p})]$

meaning the phones move in opposition direction and the distance is widen. Thus the image size has to be enlarged.

If

$[[P(U]_{m1}]^{c}) > [P(U]_{m1}{}^{p}) \text{and} [P(U]_{m2}{}^{c}) < [P(U]_{m2}{}^{p})]$

meaning the phones move in opposition direction and the distance is narrowed. Thus the image size has to be reduced.

b. 'Rotate': Let $N_y \geq 1$, the total number of tags and $P_y{}^{m}$ be the tag in the middle position of the tags in the y-axis plane so that $P_y{}^{m} = P_{y\frac{N_y+1}{2}}$ and $N_y \text{ Modulo } 2 = 1$ with m_1 and m_2 representing the two mobile phones. Giving that

$P_x{}^{m}$ be the position of the middle tag in x-axis plane

$P_x{}^{i}$ be the position of any tag in x-axis plane such that $P_x{}^{i} < P_x{}^{m}$

$P_x{}^{j}$ be the position of any tag in x-axis plane such that $P_x{}^{j} > P_x{}^{m}$

$[P(U]_{m1}{}^{p})$ be the position of the previous tag read by m_1

$[P(U]_{m2}{}^{p})$ be the position of the previous tag read by m_2

$[P(U]_{m1}{}^{c})$ be the position of the current tag read by m_1

$[P(U]_{m2}{}^{c})$ be the position of the current tag read by m_2

The image is rotated through clockwise 90^{0}:

If

$[P(U]_{m2}{}^{p}) = P_y{}^{m} \text{ and } [P(U]_{m2}{}^{c}) = P_x{}^{j} \text{ and } [P(U]_{m1}{}^{p}) = P_x{}^{m} \text{ and } [P(U]_{m1}{}^{c}) = P_x{}^{i}$

OR

If

$[P(U]_{m2}{}^{p}) = P_x{}^{j} \text{ and } [P(U]_{m2}{}^{c}) = P_x{}^{m} \text{ and } [P(U]_{m1}{}^{p}) = P_x{}^{i} \text{ and } [P(U]_{m1}{}^{c}) = P_y{}^{m}$

OR

If

$[P(U]_{m2}{}^{p}) = P_x{}^{m} \text{ and } [P(U]_{m2}{}^{c}) = P_x{}^{i} \text{ and } [P(U]_{m1}{}^{p}) = P_y{}^{m} \text{and } [P(U]_{m1}{}^{c}) = P_x{}^{j}$

The image is rotated through anticlockwise 90^{0}:

If

$[P(U]_{m2}{}^{p}) = P_x{}^{m} \text{ and } [P(U]_{m2}{}^{c}) = P_x{}^{j} \text{ and } [P(U]_{m1}{}^{p}) = P_y{}^{m} \text{ and } [P(U]_{m1}{}^{c}) = P_x{}^{i}$

OR

If

$[P(U]_{m2}{}^{p}) = P_x{}^{j} \text{ and } [P(U]_{m2}{}^{c}) = P_y{}^{m} \text{ and } [P(U]_{m1}{}^{p}) = P_x{}^{i} \text{ and } [P(U]_{m1}{}^{c}) = P_x{}^{m}$

OR

If

$[P(U]_{m2}{}^{p}) = P_y{}^{m} \text{and } [P(U]_{m2}{}^{c}) = P_x{}^{i} \text{ and } [P(U]_{m1}{}^{p}) = P_x{}^{m} \text{ and } [P(U]_{m1}{}^{c}) = P_x{}^{j}$

The image is rotated through 180^{0}:

If

$[P(U]_{m2}{}^{p}) = P_x{}^{j} \text{ and } [P(U]_{m2}{}^{c}) = P_x{}^{i}\text{and } [P(U]_{m1}{}^{p}) = P_x{}^{i} \text{ and } [P(U]_{m1}{}^{c}) = P_x{}^{j}$

OR

If

$[P(U]_{m2}{}^{p}) - P_x{}^{i}\text{and } [P(U]_{m2}{}^{c}) - P_x{}^{j} \text{ and } [P(U]_{m1}{}^{p}) - P_x{}^{j}\text{and } [P(U]_{m1}{}^{c}) - P_x{}^{i}$

3. Exit to close the program or continue by going to step 2

V. SYSTEM INTERACTION

The backend server interacts with the tags through the phone. The phones touch the tags and send the UID of the touched tag to the server. There is no direct communication between the pair phone. Their interaction is controlled by the backend server.

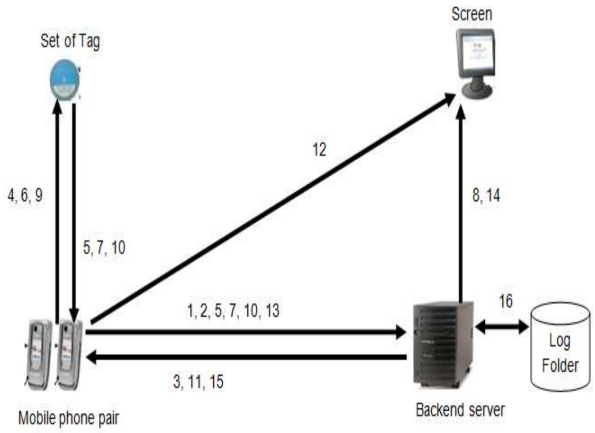


Fig. 5: Interaction between System Components

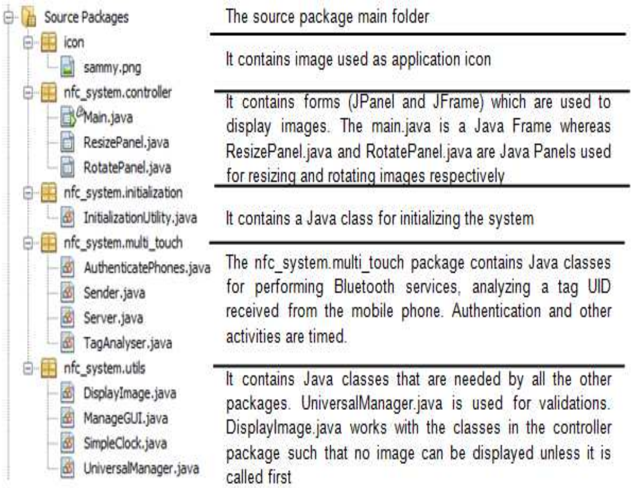
Description of the above notations

1. Phones connect to the server through Bluetooth communication technology
2. Communicating phones authenticate themselves with the backend server
3. Authentication acknowledged and phones are ready to be used
4. Initialize the system the first use of it
5. Tag UIDs are sent to the server through one of the authenticated phones to initialize the system
6. **Optional:** Select the desired contents of the screen by touching hot areas of a welcome screen
7. Tag UID in the hot area is sent through the phone(s) to the server
8. Selected image is displayed at the centre of the screen

9. **Optional:** Touch either 'Resize' or 'Rotate' tag. The default manipulation is 'Resize'
10. Tag UID is sent through phone(s) to the server
11. Server is ready to read and detect phones movement
12. Phones move across the screen over the axes tags place in front of the screen
13. Movement "co-ordinates" are sent to the server
14. Image is manipulated depending on the movements of the hands
15. Phone mode is set preparing it for the next action either to go to step 6, 9 or 12
16. Activity details are stored for traceability and auditing purposes

VI. SYSTEM IMPLEMENTATION

The server packages are described in the figure below



Source Packages	The source package main folder
icon sammy.png	It contains image used as application icon
nfc_system.controller Main.java ResizePanel.java RotatePanel.java	It contains forms (JPanel and JFrame) which are used to display images. The main.java is a Java Frame whereas ResizePanel.java and RotatePanel.java are Java Panels used for resizing and rotating images respectively
nfc_system.initialization InitializationUtility.java	It contains a Java class for initializing the system
nfc_system.multi_touch AuthenticatePhones.java Sender.java Server.java TagAnalyser.java	The nfc_system.multi_touch package contains Java classes for performing Bluetooth services, analyzing a tag UID received from the mobile phone. Authentication and other activities are timed.
nfc_system.utils DisplayImage.java ManageGUI.java SimpleClock.java UniversalManager.java	It contains Java classes that are needed by all the other packages. UniversalManager.java is used for validations. DisplayImage.java works with the classes in the controller package such that no image can be displayed unless it is called first

Fig. 6: Source Packages of Backend Server

The client side, on the other hand, has only one package. The aim of reducing computations and hence the number of packages and classes is the limitations of memory and processing capacity of mobile devices especially mobile phones.

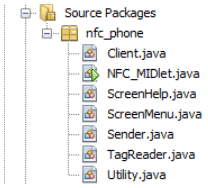


Fig. 7: Source Package of the Mobile Phone System

A. Connectivity between client and server

Connectivity is the first stage of the whole process. This ensures that the mobile phones and backend server can communicate before any process can be initiated. The connectivity process first finds the device and then uses the device to find the service. These steps help in determining error if there exists any. The errors that can be ascertained are battery power level, no device in the range or the backend server is not running.

B. *Authentication of mobile phone pair*

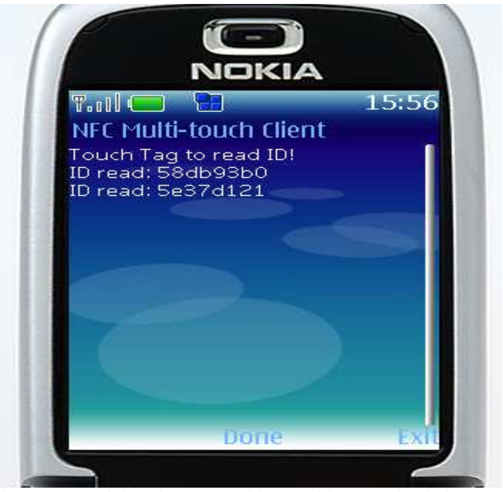
Authentication ensures that the communicating phone pair comes from the same user and thus exactly two phones are used. This process is needed to avoid system confusion. In authentication, the user needs to enter the same three-digit number for the two phones. Digits are used because they are easy to implement. Three digits ensures that there are one thousand (10x10x10=1000 where 10 is the numerical values from 0 to 9) ways to select the three digit ensuring that there is low tendency for different users to enter the same numbers.

C. *Initialization of the system*

This process ensures that all the tags needed to be used are captured into the system before using. The system thus knows what to do specifically and therefore guaranteeing consistency. Validation and vulnerability checks are thus easily performed without wasting much time. The system could have been done without initialization and the user will tell the backend server what to do by sending information. This will make the system inconsistent since the information sent to the backend server will defer from user to user.

D. *Manipulation of screen images*

This is the climax of the whole process ensuring that screen images are resized or rotated. There are default settings that help the system to function as expected. If no image is selected, the system has default image which will be used. If manipulation option (either resizing or rotation) is not selected, the default option is resizing. The two authenticated phones are used to touch different tags and depending on the initialization of the system, the image is resized (by reducing or increasing its height or width) or rotated (clockwise or anticlockwise 90^0 based on the current angle of rotation)



Touch Tag to read ID!
ID read: 58db93b0
ID read: 5e37d121

Fig. 8: One of the phones used in touching manipulation tags

VII. SYSTEM TESTING

A. Resizing an Image

The default setting of any image when displayed for the first time is shown below. This ensures that there is room for expansion and reduction depending on the focal point of the lens of the user.

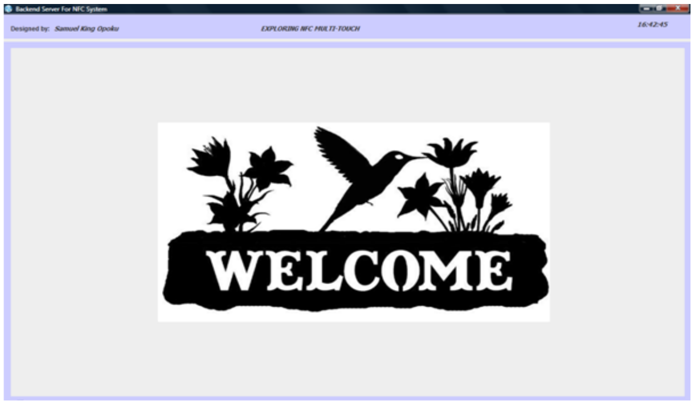

Backend Server For NFC System
Designed by: Samuel King Opoku
EXPLORING NFC MULTI-TOUCH
16:42:45
WELCOME

Fig. 9: Default size of the image

The minimum size allowed in relation to the main screen. The system is implemented such that after the minimum size is obtained, there is no way the size of the image can be decreased. This mechanism ensures that collision detection is controlled

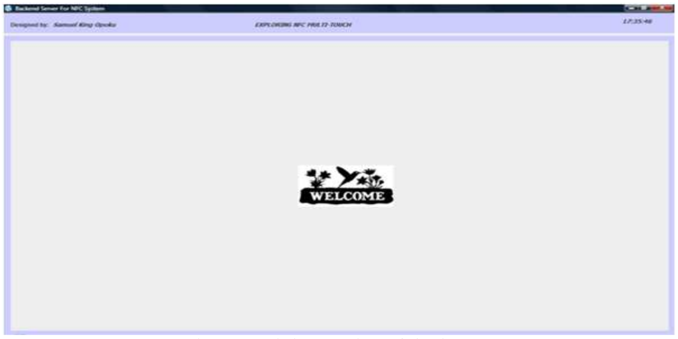

Backend Server For NFC System
Designed by: Samuel King Opoku
EXPLORING NFC MULTI-TOUCH
17:35:46
WELCOME

Fig. 10: Minimum size of the image

B. Rotating an Image

Images can be rotated through 90^0 in either clockwise or anticlockwise direction. Anticlockwise 90^0 results the same output as clockwise 270^0. Similarly, the outcome of clockwise 90^0 rotation is the same result as anticlockwise 270^0. In effect, there are only four sets of display associated with image rotation, 0^0, 90^0, 180^0 and 270^0. It is worth mention that 360^0 is equivalent to 0^0. The four sets of image rotation as described above are illustrated below with the various manipulation options.

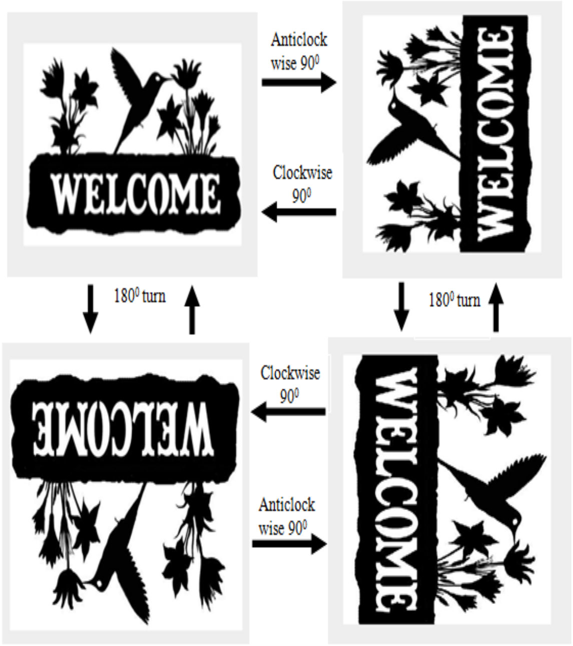


Fig. 11: Rotating Image

VIII. CONCLUSIONS

NFC is a feasible technology for deploying medium to large size multi-touch screens with some particular features that make it adapted to multiuser, low computing resource systems. Some of the main features of the NFC multi-touch system described in the paper are:

- Large screens are supported
- Low computing resources required in the controlling part of the system since no image processing is needed to track the users' movements
- Multi-user environment

The system requires minimal initial calibration to initialize the location of the tags on the screen. The tracking of the movements of the users are not based on image processing, minimizing the impact of light conditions for the system.

ACKNOWLEDGEMENTS

The research leading to these results has received funding by the ARTEMISA project TIN2009-14378-C02-02 within the Spanish "Plan Nacional de I+D+I", and the Madrid regional community projects S2009/TIC-1650 and CCG10-UC3M/TIC-4992.

REFERENCES

[1] S. Erkki and T. Vili, "*The Impact of NFC on Multimodal Social Media Application*", 2nd International Workshop on Near Field Communication, IEEE, p.51-56, 2010

[2] Nokia Forum, http://discussion.forum.nokia.com/tools/

[3] Touch Base, "*Windows 7 Touch Implementation*", 6th November 2009, http://touch-base.com/documentation/Windows%207%20Touch%20Implementation.htm

[4] M. Sallinen, E. Strommer and Ylisaukko-oja A., "Application Scenario for NFC: Mobile Tool for Industrial Worker", Second International Conference on Sensor Technologies and Applications, SENSORCOMM '08, IEEE, p.586 – 591, 2008

[5] O. C. Enrique, "An Introduction to Near-Field Communication and the Contactless Communication API", NFC Article, Sun Developer

Network (SDN), 2008, http://java.sun.com/developer/technicalArticles/javame/nfc/

[6] C. Shen, K. Ryall, C. Forlines, A. Esenther, F. D. Vernier, K. Everitt, M. Wu, D. Wigdor, M. R. Morris, M. Hancock, and E. Tse, "Informing the Design of Direct-Touch Tabletops," Computer Graphics and Applications, IEEE, vol. 26, pp. 36-46, 2006

[7] G. D. Morrison, "A camera-based input device for large interactive displays," Computer Graphics and Applications, IEEE, vol. 25, pp. 52-57, 2005.

[8] J. K. Parker, R. L. Mandryk, and K. M. Inkpen, "Integrating Point and Touch for Interaction with Digital Tabletop Displays," Computer Graphics and Applications, IEEE, vol. 26, pp. 28-35, 2006.

[9] Agarwal, A ,S. Izadi, Chandraker. M, and Blake. A, "High Precision Multi-touch Sensing on Surfaces using Overhead Cameras." Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop, pp. 197-200, 2007

[10] Jefferson Y. Han, " Low-cost multi-touch sensing through frustrated total internal reflection," Proceedings of the 18th annual ACM symposium on User interface software and technology, ACM, pp. 115-118, 2005

[11] De Maria. E. A. A, E. Gho, Maidana. C.E, Rodriguez. C.A, Szklanny. F.I, and Tantignone. H.R, "Real Time FPGA based Thresholding Segmentation in a Multi Touch System." Programmable Logic, 2008 4th Southern Conference on, pp. 237-240, 2008

[12] Hofer. R, D. Naeff, and A. Kunz, "FLATIR: FTIR multi-touch detection on a discrete distributed sensor array," Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, ACM, pp. 317-322, 2009

[13] J. W. Roach, P. K. Paripati, and M. Wade, "Model-based object recognition using a large-field passive tactile sensor," Systems, Man and Cybernetics, IEEE Transactions on, vol. 19, pp. 846-853, 1989.

[14] P. T. Krein and R. D. Meadows, "The electroquasistatics of the capacitive touch panel," Industry Applications, IEEE Transactions on, vol. 26, pp. 529-534, 1990.

[15] D. Maxime, C. Jacques, and Wu. Ke, "An Analytical Solution to Circular Touch

[16] D. Pasquariello, M. C. J. M. Vissenberg, and G. J. Destura, "RemoteTouch: A Laser Input User- Display Interaction Technology," Display Technology, Journal of, vol. 4, pp. 39-46, 2008.

[17] Izadi. S, A. Butler, S. Hodges, West. D, Hall. M, Buxton. B, and Molloy. M "Experiences with building a thin form-factor touch and tangible tabletop." Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on, pp. 181-184., 2008

[18] Byoungjoo Lee; Insung Hong; Yoonsik Uhm; Sehyun Park; , "The multi-touch system with high applicability using tri-axial coordinate infrared LEDs," Consumer Electronics, IEEE Transactions on , vol.55, no.4, pp.2416-2424, November 2009