Creating a Touch Interface on an Ordinary Flat Surface Utilizing the Microsoft Kinect and a Projector

Supervisor: Kalin Stefanov
Members: Dan Carino, Erik Dahlström, Niclas Ericsson, Bálint Zsiga

Abstract

This project is focused on examining how the Kinect can be integrated to create a touch surface on a flat surface with a projector. To examine this, we developed a drawing application where the user could draw shapes that the program could recognize and into a perfect shape. We improved the application and the user experience in several steps by implementing various techniques like the Kalman filter. After the the development was done, we evaluated the application by performing user tests. These test sessions were split up into three parts; an initial testing part with simple tasks, a template task and a final task where the test participant were supposed to form an opinion about some statements. When we had done these tests, we evaluated the results. Finally we looked at some future work and implementations that can be done within the area.

Table of contents

1. Introduction

Touch surfaces have become increasingly common as user interfaces, especially among mobile phones and tablets. They are used on a daily basis since the interface is intuitive and easy to use for all ages. Therefore it would be interesting to find ways to use touch interfaces in other areas, for example when doing presentations on whiteboards or tables.

1.1 Background and previous work

With the spread of touch interfaces, touch gestures are becoming more and more intuitive. We see potential in extending these gestures to objects other than screens. We believe the possibility of using touch gestures on everyday objects can offer numerous applications in the future. Studies have shown that touch interfaces can be created by only using a projector and a camera (Dai et al, 2012), something that shows the possibilities of creating touch interfaces with ordinary everyday technology.

For our project we will be creating an interactive table by using a 2nd generation Kinect, that will sense the movement of the user, and a projector to show the interface. We are interested in doing this project because we want to explore the possibilities of Kinect's gesture recognition and tracking capabilities, and we also want to see how users create and interact with virtual shapes in this context.

Among previous work, there has been some successful studies centered around multi-touch tables (Cyborra et al, 2013), and also some that utilize the Kinect to create augmented reality applications (Hsieh et al, 2014). These studies show that touch interfaces are used in more contexts today and are of general interest. Also, other studies have shown that touch user interfaces *"can provide more immersive, natural and user-friendly interfaces"* (Seo et al, 2013).

1.2 Question formulation

Our main question is as follows:

> *"How can the Kinect be used to create a touch surface on a flat surface with a projector?"*

1.3 Aims and delimitations

Our aim is to make a regular surface interactive, with a Kinect and a projector. This will be a proof of concept prototype that creates and manipulates simple two dimensional geometric shapes (rectangles and circles) from hand gestures and movements.

2. Method

2.1 System setup and functionality

The hardware that was used was a 2nd generation Kinect and a projector. The projector was situated on a shelf, projecting down onto a table. The Kinect was placed on a lower shelf, pointing towards the projected area of the table.

The software and libraries we used were the following:

- Kinect v2.0 SDK
- Microsoft Visual Studio
- Gestures.js, a library for recognizing drawn shapes
- Code for creating shape objects
- An implemented Kalman Filter

The final prototype had the following functions:

- Plane calibration
- Shape creation
- Shape manipulation
    - Moving
    - Resizing

On startup, the application request the user to calibrate so that the system knows which area is intended to be touchable, This requires the user to indicate where the corners of the touchable area are. Once calibrated, the user is able to create shapes. Shapes are created by simply drawing it on the area. The system would then try to recognize the shape and create a similar shape based on the points on the canvas. The shapes can then be manipulated in two ways; they can be moved by holding your hand over the shape on the projected area within the given threshold, then moving your hand to the place on the table you want the shape to be. The second way of manipulating is to resize the shape. This is done by touching the projected image with both hands and move the hands away from each other; this will make the shape larger. Moving the hands closer to each other will make the shape smaller.

2.2 System implementation

We began by experimenting with the different demos that were included in the Kinect SDK in order to became more familiar with how the Kinect and its coordinate system works. We decided that we wanted to continue working with the HTML/Javascript library since we had most experience with it. At the same time, we started to look after Javascript code that could recognize which shape the user was drawing on a HTML canvas. In our search we discovered a library called Gestures.js which could do what we required. With these resources we started writing code for a desktop application in order to quickly create the functionality as we were still unfamiliar with the Kinect SDK. We first made it possible to create rectangles from the points drawn on the HTML canvas, and discovered code for saving these shapes as objects on the canvas. To begin with, we only did this for rectangular shapes, and after making everything work we extended it to include circles. We also made it possible to change the size of the objects and remove them from the canvas by dropping it over an image of a trashbin. However, the latter did not get implemented in the final version which was used during the test sessions because of time constraints.

When the desktop application contained all the functionality, that we wanted to have in the final application, we started to integrate the drawing and shape recognition code with the code for the Kinect. First, we implemented some simple drawing with the Kinect, where the user could draw on a HTML canvas with the hand. We had to map the drawing to the HTML canvas since the metric coordinates from the Kinect did not match the pixel values that the canvas required. When this was done we integrated the shape recognition. This was followed by merging the different code parts since we had worked on them separately.

A video demonstration of our system:
https://vimeo.com/117327109

Fig 1 - The entire setup

Fig 2 - The projector used

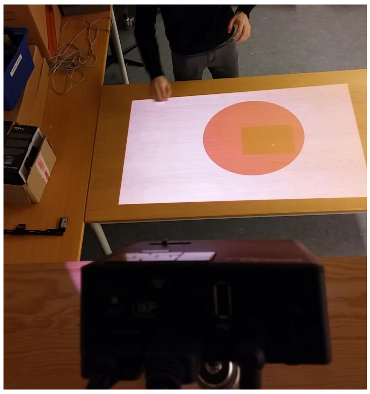

Fig 3 - Example of projection onto the table

Fig 4 - The 2nd Generation Kinect

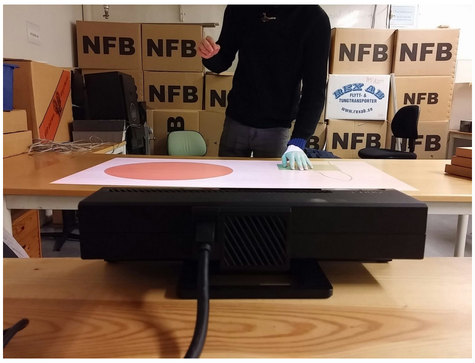

Fig 5 - The position of the user relative to the Kinect

Fig 6 - Drawing with the system

2.2 Description of tests

Our user tests consisted of three parts. In the first part, the test participant got to become familiar with the system. The second part consisted of a number of tasks where the participant would try to create shapes that matched predefined templates which we had drawn on a paper beforehand. In the third and final test assignment, the participant got to answer questions about the experience.

2.2.1 Initial test assignment

The initial test assignment were made to make the test participant becoming familiar with the setup and the interface and to test the degree of intuitivity of the gestures. The assignment consisted of eight tasks:

1. Draw a rectangle of any size
2. Draw a circle of any size
3. Move the circle to the upper right corner
4. Move the rectangle to the center
5. Make the rectangle larger and then smaller
6. Move the rectangle to the left side
7. Move the circle to the center
8. Make the circle smaller

During this test, the users were supposed to figure out how to use the system on their own. If they got stuck, we explained how to use it correctly. By doing this we could see behaviors that were intuitive for the users and which things in the system that did not correlate to these behaviors.

2.2.2 Template assignment

After the initial tasks, we placed a piece of paper on the table on which we had drawn various shapes with a marker pen. The user was supposed to create shapes that match the various template shapes by drawing and resizing, and finally move them to their corresponding position. With this assignment we wanted to test the accuracy of the system when the users already know how to use it.

2.2.3 Questions

During the final test evaluation, we asked the test participant a number of statements that the participant would form an opinion about. The first 6 questions are parts of the System Usability Scale [1], while the last 3 questions aim to measure the participants experience with touch devices. The answers derived on a scale from 1 to 5 were 1 represented *"strongly disagree"* and 5 represented for *"strongly agree"*. The statements we presented for the user were as follows:

1. I thought the system was easy to use
2. I think that I would need the support of a technical person to be able to use this system
3. I would imagine that most people would learn to use this system very quickly
4. I found the system very cumbersome to use
5. I felt very confident using the system
6. I needed to learn a lot of things before I could get going with this system
7. I'm an experienced tablet user
8. I'm an experienced smartphone user
9. I'm an experienced multi-touch table user

3. Results

3.1 Results from tests

3.1.1 Initial test assignment

During the first assignment, the function that most participants had trouble with was the resize function. All five participants had trouble with understanding how the resizing worked, leading us to explain to them how to do it.

Two out of five participants wanted at times to resize the shape using the one handed gesture that is used on smartphones, which did not work with the system. All participants tried to resize the shapes by putting their hands outside the boundaries of the shape, which did not work either since the system required the user to start the resizing by putting both hands inside the shape.

Other things that the participants thought were confusing included the manner in which the system recognized a touch. Most users thought that the natural way would be to move their hand onto the interface from the side, but this did not work because of the way the system was implemented; the Kinect would recognize a touch if the hand was close enough to the calibrated plane of the table, meaning that it is necessary to come from the top, instead of the side, in order to activate the touch at the correct point.

The participants also had some problems when it came to creating circles. All participants started to draw the circle from the top, something that our system could not recognize since it required the user to start drawing from the bottom. Some of the attempts where the users tried to create rectangles also failed. This was caused by a bug in the system that mistook some rectangular shapes for triangles, but since the triangle recognition was not implemented nothing was drawn on the canvas. This happened during almost all test sessions, and all participants were a bit distracted by it.

Despite these problems, all five users adapted to the system and, in the end, were able to use it without bigger issues.

3.1.2 Template assignment

One of the biggest issues that happened during all test sessions in the template assignment was, as mentioned above, when the user needed to create a circle. We also noticed, if not for the first time, that precision drawing was hard to do for the users. Some users wanted to change the width or height of the shape without scaling the whole object, something that couldn't be done with our system.

As a result of the threshold problem that has been already introduced, the assignment was not easy, but as it can be seen on the figure below, there were participants who managed to finish the task rather accurately.

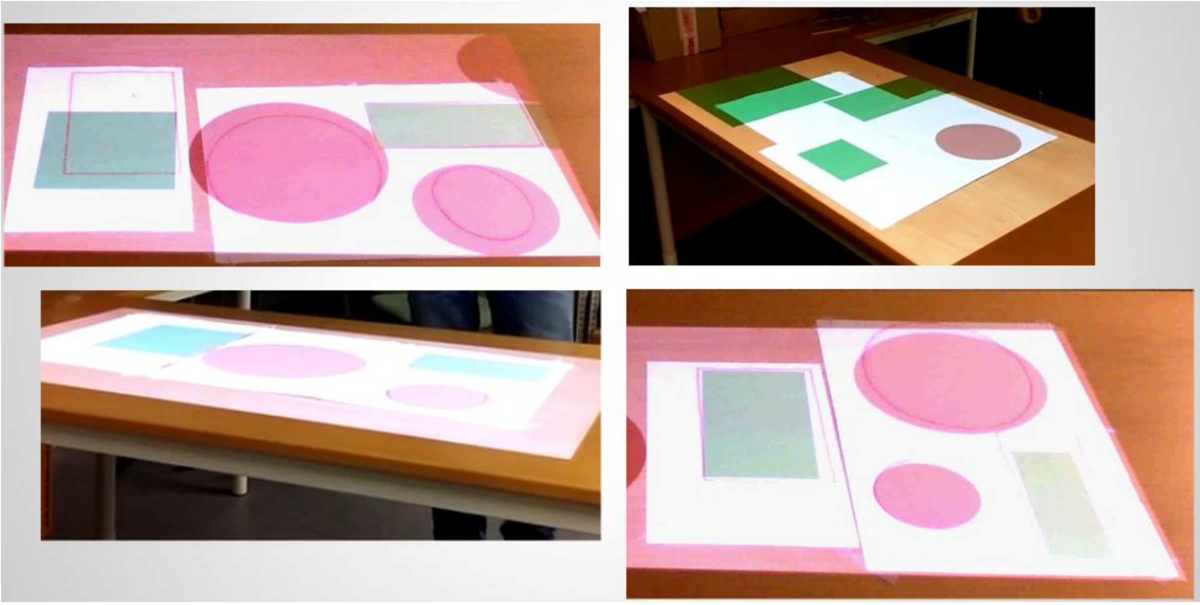

Fig 7 - Results of the template test

As it can be seen in the top left image of Fig 7, our system lacked the function of rescaling shapes, that caused some difficulties for the users as well.

3.1.3 Questions

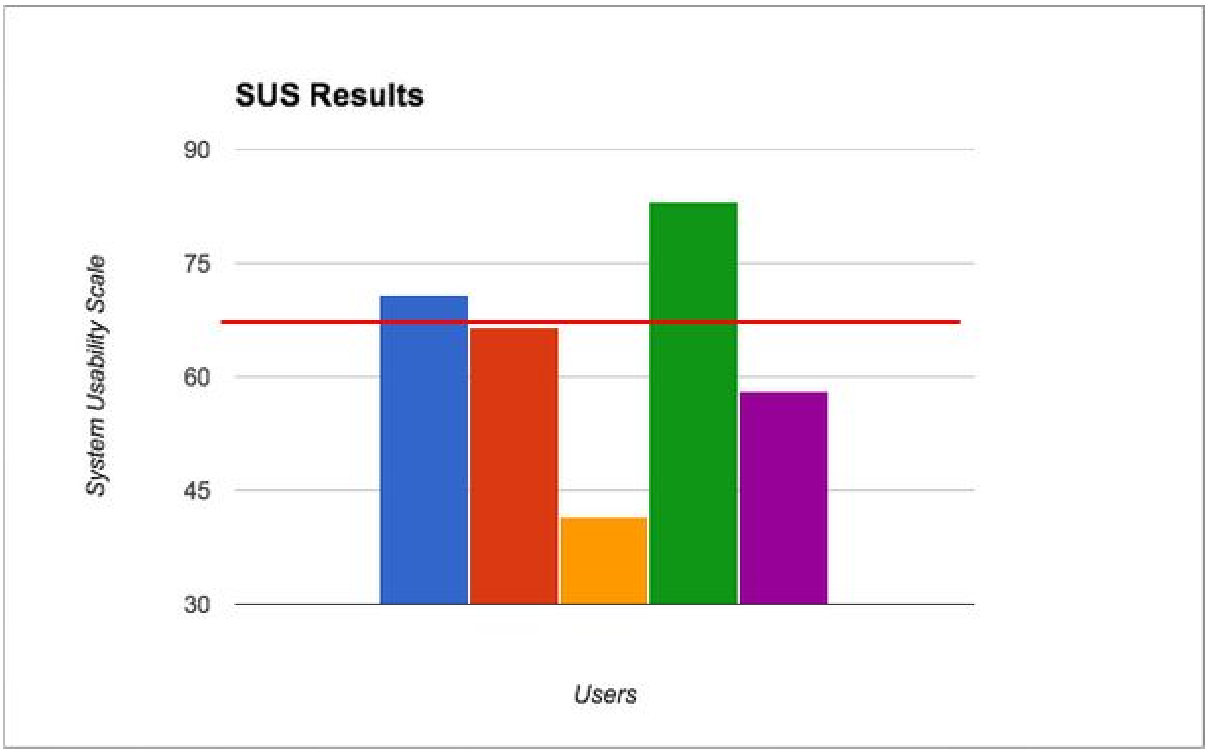

Fig 8 - Results of System Usability Scale. The red lines shows the score of a system with average usability.

In total, the average system usability score was 64, which is 4 points below the score of a system with average usability (Brooke, 1990).

Almost all participants were experienced smartphone users; four of five participants said that they *"strongly agree"* to this statement. However, the same participants also *"strongly disagreed"* that they were experienced multi-touch table users.

Three out of five participants said that they strongly agreed to the statement *"I needed to learn a lot of things before I could get going with this system"*.

Four out of five participants agreed or strongly agreed when it came to the statement *"I think that I would need the support of a technical person to be able to use this system"*. Despite this, three out of five participants strongly agreed to the statement *"I would imagine that most people would learn to use this system very quickly"*.

The statement *"I thought the system was easy to use"* got an average score of 3 out of 5, meaning that the participants neither agreed or disagreed.

4. Discussion

4.1 Discussion of the test results

The system usability score that our system got can be compared to the average score of 68. This may be due to the flaws that we discovered during the user tests.

One interesting fact is that all the user were familiar with smartphones but they were much more unfamiliar when it came to multi-touch tables. Despite this the users managed quite well to use our system. This indicates again, as we saw in previous studies, the high usability factor of the touch interface.

The result also indicates that the system wasn't very easy to accurately control. Even though the gestures were intuitive to the participants, the accuracy problem of the system created difficulties for the users to complete the tasks they were asked to do. This resulted in the users resorting to less common types of interaction, which did not work because it was not implemented within our system. Since the users initially used the correct gestures, it is likely that it is a better solution to firstly resolve the accuracy problems rather than implement other types of gestures as well.

4.1 Strengths and weaknesses

Since we have not been able to acquire the material needed for projection from below, we resorted to have the projector standing, point downwards in an angle projecting on a table. This has caused some problem with skewering of the projected image which makes the projected image and the image, that the Kinect picks up, to misalign.

Another problem that has arisen is the problem of calibrating the program to the plane we want to draw on, i.e. the table. It has been difficult to make sure that these too are aligned. For example, the script often did not recreate the plane to be parallel to the ground. Instead it was tipping away or towards the Kinect. Meaning that if the plane tilted toward the Kinect, the *"cursor"* would accelerate toward the Kinect when the user's hand moved closer to it.

The noise that came from the Kinect sensor, which caused the drawing to be jumpy and inaccurate, also became a problem. Initially it was quite hard to draw straight lines. To solve this problem, we implemented a so-called Kalman filter to do estimation calculations and reduce the jumpiness of the drawing. This made the drawing a lot better than before.

4.2 The work process

We decided to, for the most part, work together during this project, mostly because we were fairly new to Kinect development. It therefore felt best to do the things together. When we had become more familiar with the software and javascript code we started to divide the work between us.

We also decided that we would not use any revision control since it seemed a little too much work for our rather small project. It is hard to say if this was a good or bad choice, since the manual merging took a bit of time to do and it was also kind of hard to know what different persons had changed in the code since last time. On the other hand it is hard to know how much time the revision control would take if we would have decided to use it.

5 Conclusion

5.1 General conclusions

In conclusion, as all the users initially tried the right questures, we can say that we have a positive answer for our first research question (Are the chosen gestures intuitive?). Although the users tried different gestures as well (e.g. one hand sizing gesture), they only did that after having unsuccessful tries with the right ones because of other usability problems.

The root of these problems were obviously the too big touch threshold above the table. As it has been already mentioned in a previous point, the system only registers the proper touch points if the touches arrive from above. Although as it turned out, that was not natural for any of the test participants. They rather approached the table from the sides in an acute angle, that caused unwanted touches and activated unwanted functionalities. Even if the users were able to use the system quite accurately after learning it’s limitations, they did it in a not necessary natural way. To achieve better usability, the threshold above the table should be decreased.

5.2 Future work

For future work includes having a set up where the projector is directly below the drawing area, as this will remove the problems with skewing and calibration. It will also avoid projection onto the hands and the shadows under the hands that are created by projection over the table.

The system can also be improved by utilizing the depth data instead of tracking the body joints, which we did for this project. This would make the tracking more precise, it would also be possible to have several users at the same time.

Other future implementations would be to further develop and test this system in collaborative work scenarios, since it’ve been shown in other studies that multi-touch tables *“promote collaborative learning in very effective ways”* (Candela et al, 2012). This could also include the extension of multi-touch functionalities so that several users can manipulate different objects at the same time.

References

Cyborra, D., Albert, M., Steinicke, F. and Bruder, G. (2013). Touch & move: A portable stereoscopic multi-touch table. 2013 IEEE Virtual Reality (VR).
Hsieh, C., Kuo, T., Wang, H., Wu, Y. and Chang, L. (2014). Touch interface for markless AR based on Kinect. Fifth International Conference on Graphic and Image Processing (ICGIP 2013).

Candela, E., Pérez, M., Romero, C., López, D., Herranz, G., Contero, M. and Raya, M. (2012). HumanTop: a multi-object tracking tabletop. Multimedia Tools and Applications, 70(3), pp.1837-1868.

Dai, J. and Chung, R. (2012). Making any planar surface into a touch-sensitive display by a mere projector and camera. 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops.

Seo, D. and Lee, J. (2013). Direct hand touchable interactions in augmented reality environments for natural and intuitive user experiences. Expert Systems with Applications, 40(9), pp.3784-3793.

John Brooke. (1990). SUS - A quick and dirty usability scale, Redhatch Consulting Ltd., 12 Beaconsfield Way, Earley, READING RG6 2UX United Kingdom