Mid-Air Gesture-Based Interface: Developing, Modeling and Use

SW10 Master Thesis

SW105F12

Rasmus Hummersgaard
Lasse Andreasen

Aalborg University

Department of Computer Science

Title:

Mid-Air Gesture-Based Interface: Developing, Modeling and Use

Theme:

HCI Master Thesis

Project Period:

SW10, Spring 2012

Project Group:

SW105F12

Lasse Andreasen

Rasmus Hummersgaard

Supervisor:

Jan Stage

Abstract:

The goal of this project was to study to what extent can mid-air gesture-based interaction be used for interacting with a PC. We developed a mid-air gesture-based interface for basic web browser interaction based on the functionality of a Kinect sensor. The interface was implemented incrementally based on development and evaluation of three prototypes. The interface supports the ability to control the cursor, clicking elements, scrolling up and down and navigating forward and backward in the web history using gestures. Furthermore we modified the original Keystroke-Level Model for prediction of task completion times using the mid-air gesture-based interface and validated the model through an experiment. Lastly we conducted a field study on how users approached, learned and experienced the use of the mid-air gesture-based interface in public.

Total number of pages: 52

Number of reports printed: 4

Finished: June 6, 2012

PREFACE

This paper documents the work by Software Engineering group SW105F12 at the Department of Computer Science, Aalborg University.
This master thesis began at 1th February 2012. However, some of the development and empirical work in article 1 were conducted during the fall of 2011.

We would like to use this opportunity to thank our supervisor Jan Stage for his help and support and all the people that participated in our experiments.

Signatures

Lasse Andreasen

Rasmus Hummersgaard

CONTENTS

CHAPTER 1

INTRODUCTION

The way people are interacting with computerized devices are changing, and gesture-based interfaces have gained increasing interest recently. The adoption of touch screens on mobile devices has contributed to the implementation of gestures-based interaction in our everyday life. Gestures have been implemented in smart phones and tablets with iPhone and iPad as examples, where the user makes finger gestures by tapping elements and swiping their fingers across the screen. Users do not press the left arrow key to navigate to the next image in the gallery or hold down the down arrow to scroll down in the list. Instead they make a gesture by moving the finger across the screen, as if they were flicking through pages in a photo album or scrolling on a real wheel.

The gaming industry has also implemented the use of gestures in their new gaming consoles with Nintendo Wii and Playstation 3 Move as examples, where the user interacts with a controller in mid-air, as if they were using real objects instead of pressing buttons on an original gaming controller. The user can swing the controller, as if it was a real sword, and this is translated into movements of an animated sword in the game.

With the Xbox 360 and Microsoft Kinect the interaction functions without any controllers, and with for example Dance Central 2[1] the user interacts with actual body movements instead of movements of a controller. The user's body movements are compared to predefined poses, and the score depends on timing and how accurately the user can perform the poses.

The same level of implementation of gestures has not been seen for PC's, and the most used standard input devices are still the keyboard and computer mouse. Therefore we have found it interesting to explore the possibility of using gestures for interacting with a PC, and to what extend such a gesture-based interface can be developed using the Kinect sensor, which is expressed in the overall research question:

> *To what extent can mid-air gesture-based interaction be used for interacting with a PC?*

To be able to answer the overall research question it is necessary to develop basic mid-air gesture-based interaction. We conduct this as a proof-of-concept with focus on basic interaction which is expressed in the following research question.

> *To what extent can mid-air gesture-based interaction be developed and replace standard input devices for basic interaction on a PC?*

To narrow down the field of study we focus on one common usage for PC's, which is web browsing[2].
To be able to develop a mid-air gesture-based interface we have chosen to use the Kinect sensor due the fact that the Kinect sensor is becoming more available and we find this technology interesting. The Kinect sensor enables us to track users and their body movements.
The ability to predict the time and performance of different designs of user interfaces can be a vital tool to avoid the implementation of an ineffective design. Additionally a model can be used as a tool for analyzing an already implemented user interface to specify the time consumption of different actions and thereby providing the possibility of improving the user interface. A model also allows designers to compare different interaction forms for a given user interface. The Keystroke-level model (KLM) has been applied for many years for time and performance prediction for interacting with user interfaces and have been modified to fit other forms of interactions than the original keyboard and mouse. To ease the process of designing new user interfaces that uses a mid-air gesture-based interaction, modeling the time and performance is of great importance.
This leads to the following research question:

> *To what extent can the Keystroke-level model be modified to predict time and performance for mid-air gesture-based interaction?*

Because gesture-based interaction is a new area considering PC's, it is important to investigate how people interact with such system. This interaction requires larger movements from the user, since the body is used as a controller, and this makes this form of interaction more visible to other people located near the user than standard input devices such as mouse and keyboard. Therefore we find it important to study the mid-air gesture-based interaction applied in public and locate situations that should be taking into consideration when developing such interface. This leads following research question:

> *How do users experience interacting with a mid-air gesture-based interface in public?*

The following chapter gives a summary of the three articles along with their contribution. The used research methods are described in details and we present our conclusion of this study along with limitations and future work.

CHAPTER 2

CONTRIBUTION

2.1 Article 1

In this article we answer the question:

> *To what extent can mid-air gesture-based interaction be developed and replace standard input devices based on time and performance for basic interaction on a PC?*

To be able to answer this question we developed a mid-air gesture-based interface and conducted a series of experiments. We focused on web browsing on a PC, since this is a common usage. The first part of the study concerned the development of the mid-air gesture-based interface. By studying other articles and basic interaction with a web browser, we developed a set of requirements for the mid-air gesture-based interface in order to replace standard input devices for basic interaction with a web browser on a PC. The requirements consisted of enabling the users to control the cursor and perform gestures for basic web browser functionality, which included clicking elements, scrolling up and down on a web page and moving forward and backward in the web history.

We developed the mid-air gesture-based interface using a Kinect sensor and OpenNI SDK [3]. We chose the Kinect sensor, because it was a new and interesting technology that enabled us to easily track the position of users as well as the joints of the users. With the functionality of a Kinect sensor we implemented an algorithm for recognizing gestures performed with the users' left arms and the functionality of moving the cursor on the screen with the users' right hands. An experiment was conducted with the first prototype of the interface in order to evaluate whether it could replace standard a standard computer mouse. The results showed that users were able to browse a web page, but several improvements would be beneficial regarding the control of the cursor and recognition of gestures.

A second prototype was developed with focus on the problems that were identified in the first experiment, and an additional three techniques for controlling the cursor and recognizing gestures were implemented. We conducted an experiment that focused on the performance of the three new techniques for controlling the cursor and recognizing the gestures compared to each other and the techniques used in the first prototype. We recorded the time used to move the cursor and the precision, when clicking with the "Click"-gesture, using the four different techniques. The four techniques for recognition of gestures were tested and the recognition rate for each of them was recorded. We used the results of the experiment to identify the optimal techniques for controlling the cursor and recognizing gestures.

To enable users to walk-up-and-use this type of interface we studied different techniques to train new users. We found two different approaches, animations and feedback. Both techniques were implemented in the third prototype and tested individually. The results showed that feedback was the most effective and preferred by the participants.

With the final prototype, the optimal technique for recognizing gestures was able to recognize 89% of the performed gestures, and the participants found the gestures easy to learn, easy to remember and the mapping of the gestures logical. When using the mid-air gesture-based interface the time used to move cursor was three times greater than with a standard computer mouse.

2.2 Article 2

The purpose of this article is to answer the following research question:

> *To what extent can the Keystroke-level model be modified to predict time and performance for mid-air gesture-based interaction?*

To answer this question we studied the original Keystroke-Level Model (KLM) to acquire a basic knowledge of the model. We investigated several other studies that used the original KLM to predict time and performance of different user interfaces. Furthermore we studied articles that had modified the original KLM to predict time and performance for other devices than the keyboard and computer mouse. Based on these articles and a detailed analysis of the developed mid-air gesture-based interaction, we defined a new set of operators used to describe the mid-air gesture-based interaction form for two different techniques to control the cursor. The two different techniques for controlling the cursor were used since they both showed good results in the experiments conducted in Article 1. The difference between the two techniques was that the second cursor technique allowed the user to decrease the movement speed of the cursor by switching between two modes.

The new set of operators consisted of operators describing the actions of scrolling up and down, moving backward and forward in the web history and clicking elements. Furthermore operators for describing the action of moving the arm back to the initial position after performing the given gestures were defined. We conducted at first an experiment to define values for each of the operators.

Three tasks were defined for a second experiment, and the task completion time for each task was predicted with the use of our modified KLM. The second experiment was conducted and the empirically determined task completion times were compared to the predicted times. The results showed that the prediction of the first cursor technique and gestures was in the range of -2.6% to 2.0% with the best result of -0.1% prediction error.

In this article we have shown that we are able to give a fairly precise prediction of the time used to complete a task using the first cursor technique. This enables us to predict the task completion time for expert users for a given task without conducting an experiment and comparing task completion time for different interaction forms on the same user interface.

2.3 Article 3

This article concerns the study of the following research question:

> *How do users experience interacting with a mid-air gesture-based interface in public?*

We examined related articles that studied people's interactions with different computerized devices in public spaces. This provided us with basic ideas for which interesting situations that could occur.

We adapted our mid-air gesture-based interface to support buying beverages through a web page that was developed by a student club at Aalborg University. We conducted a field study by using our interface at a social event. The mid-air gesture-based interface was available for 7 hours, and we recorded users' interactions with the interface using a HD-camera. During this time frame 26 different participants used the interface to buy beverages, where several participants used the interface more than once. The participants filled out a survey concerning their opinions about the mid-air gesture-based interface. The video material from the user study was analyzed by both authors in collaboration, and interesting situations were identified. A session consisted of an uninterrupted interaction, resulting in a total of 54 sessions.

Overall the participants were positive towards the mid-air gesture-based interface, and we discovered that especially the first-time users found this type of interaction entertaining. The simple task of buying beverages became more fun and entertaining than when using a standard keyboard and computer mouse. The participants also displayed competitive behavior towards their friends, when using the interface, even though there was no clear measure of their performance, besides they were able to make a purchase.

Based on our observations we introduced five categories which described the users' interaction with the interface and experience when using the interface. The five categories were: Dynamics of approach, Social inhibition, Learning the system, Playing around and Performance and user experience.

CHAPTER 3

RESEARCH METHOD

In this chapter we discuss the research methods used in the empirical studies that form the basis for this master thesis. The theories for the research methods are based on Lazer et. al. [4].

3.1 Laboratory experiment

To determined whether users were able to use the develop mid-air gesture-based interface, laboratory experiments were conducted. This made it possible to monitor the users' performance in a controlled environment thereby allowing us to identify areas that functioned well and areas that could be improved.

3.1.1 General

Laboratory experiments provide the opportunity to control the settings and environment which makes it possible for others to replicate the experiment and either confirm or dismiss the presented results. Due to the controlled environment it is possible to record precise measures on e.g. users' performance. In a laboratory, people are aware of they are being monitored which may cause them to perform better or worse than in a more common environment. The results gathered at a laboratory experiment may not fully reflect real world use.

3.1.2 Method

Laboratory experiments were used to evaluate each of the three prototypes as well as collecting data to develop our modified version of the Keystroke-Level Model. In general, participants were recruited at the Aalborg University for each of the experiments. We used within subject, when we had multiple conditions and used Latin Square [5] to distribute the conditions equally among the participants. To ensure all participants were given the same information,

the test monitor read instructions aloud to the participants each time. Before each of the experiments, multiple pilot tests were conducted to ensure the technical equipment worked as expected as well as the design and procedure of the experiment.

3.2 Interviews

We conducted semi-structured interviews to gather data from the participants after the laboratory test. The interviews focused on the participants' opinions about the designed gestures and techniques for controlling the cursor.

3.2.1 General

Interviews have the ability to go deep and can provide data that otherwise could be difficult to gather. Interviews gives the participants the ability to talk freely and elaborate on their opinions about the given subject in their own words. Semi-structured interviews give the interviewer the ability to follow up on interesting answers and opinions from the interviewee. Interviews are time-consuming since it is necessary to use time to talk to each participant. Interviewees tend to tell what they remember, potentially making the answers the interviewee provides during an interview different from the answers they would give while using the system.

3.2.2 Method

At the end of each laboratory experiment, the participants were interviewed concerning different aspects of the designed gestures and techniques for controlling the cursor. The questions for the interviews were prepared and written down in advance so each participant was asked the same basic questions followed by clarifying questions based on the interviewee's answers. To encourage the participants to provide details and elaborated their answers, we asked the questions, when possible, in an open-ended fashion. The interviews with the participants were held right after the experiments were conducted in order to minimize the amount of details the participants might forget. We recorded the interviews using cameras to be able to review the conversations.

3.3 Field study

To investigate how people experienced a mid-air gesture-based interface, a field study was conducted.

3.3.1 General

Field studies provide the ability to observe users while interacting with a system in an environment where the system is intended to be used. A field study is less controlled, there is no restriction on the participants, or how the system is used. It can also be difficult to analyze the data, because this type of experiment requires qualitative analysis. Since the analysis is done by an analyst observing users, subjective opinions might influence the results.

3.3.2 Method

We placed the mid-air gesture-based interface at a social event at Aalborg University, enabling people to buy beverages using our interface. While users interacted with the interface, a HD-video camera recorded users' behavior as well as their conversations around our interface. We conducted a qualitative analysis of the recorded data, inspired by Peltonen et. al. [6]. The analysis of the video material was conducted by watching the video material several times. First time we noted interesting situations emerging in the video material. Second we noted when the situations occurred, which type of situation we observed, how many people were located in front of the system etc. Both authors watched and noted the findings in the video material in collaboration.

3.4 Survey

To gather opinions from the participants regarding the mid-air gesture-based interface, we conducted a survey. As a tool, we used the Questionnaire for User Interaction Satisfaction (QUIS) [7], which was designed to collect users' satisfaction measures of a specific interface.

3.4.1 General

A survey can be used to collect data from a large number of users at a relative low cost. This makes it possible to get a quick overview over people's opinions on a given subject. Surveys only give limited shallow data since it is not possible to do follow up questions. Because of the impossibility of follow up questions, it is highly important that the questions are clear, error-free and well-written.

3.4.2 Method

After making a purchase with the interface during the field study, the participants were asked to fill out a questionnaire. The questionnaire was to be completed on a laptop located in a more quite place nearby, thereby allowing the user to be anonymous and not be bothered by others. The questionnaire consisted of both open and closed questions. The closed questions were regarding the users' satisfaction on certain aspects of the mid-air gesture-based interface, while the open questions concerned, where such system could be applied and additional comments. The field for additionals comments was added to allow people to elaborate, if needed.

3.5 Research methods applied

Table 3.1 presents an overview of, in which article we have used the different research methods. We highlight strengths and weaknesses of the used research methods. Table 3.1 is based on Lazar et. al. [4] and Wynekoop et. al. [8].

Articles	**Methods**	**Strengths**	**Weaknesses**
Article 1	Laboratory experiment - Evaluation of 3 prototypes	Controlled settings Replicable Precise measures	Unknown relation to real world use Artificial setting
	Interview - Opinions on 3 prototypes	Ability to gather opinions	Time consuming
Article 2	Laboratory experiment - Determination of values	Controlled settings Replicable Precise measures	Unknown relation to real world use Artificial setting
Article 3	Field study - Real world use	Natural setting	Uncontrolled setting Difficult to analyze Subjective
	Survey - Overview of opinions	Easy Low cost	Shallow data

Table 3.1: Use of research methods

CHAPTER 4

CONCLUSION

This chapter presents a conclusion on this master thesis, including limitations and future work.

4.1 Research questions

This master thesis was conducted based on the following overall research question:

> *To what extent can mid-air gesture-based interaction be used for interacting with a PC?*

We answered this question by dividing it into three subquestions.
The first subquestion was:

> *To what extent can mid-air gesture-based interaction be developed and replace standard input devices for basic interaction on a PC?*

We developed a mid-air gesture-based interface using a Kinect sensor with the OpenNI SDK [3], focusing on web browsing on a PC. Using the Kinect sensor and OpenNI SDK we were able to retrieve coordinates of the locations on different joints on a users body. The coordinates of the left arm were used to recognize performed gestures and the coordinates of the right hand to control the cursor. The gestures were mapped to basic web browser functionality, which were clicking elements, going forward and backward in the web history and scrolling up and down. We implemented a recognition algorithm (Dynamic Time Warping-algorithm) [9], and an experiment showed that the optimal technique was able to recognize 89% of the performed gestures, while 13% of the recognized gestures were false hits (unintended gestures). An experiment showed that the time used to move the cursor with the interface was three times greater than with a standard computer mouse. The constrains mentioned above described the extent to which we were able to develop a mid-air gesture-based interaction for web browsing on a PC and to what extent it was able to replace standard input devices.

To predict expert–use of the developed mid-air gesture-based interaction, we defined our second sub-question:

> *To what extent can the Keystroke-Level Model be modified to predict time and performance for mid-air gesture-based interaction?*

We adapted the original Keystroke-Level Model to describe the mid-air gesture-based interface with two techniques for controlling the cursor and validated the model. The results of the validation showed that we were able to predict the task completion time for one of the techniques for controlling the cursor and gestures within the range of -2.6% to 2.0% with the best result only deviated with -0.1%. The other technique for controlling the cursor allowed the user to lower the cursor speed by switching between two modes, and the results of the validation showed that our model was unable to predict the task completion time due to this mode switch.
Our model enables us to predict the task completion time required for expert users to solve a given task on a web page using the mid-air gesture-based interface.

To understand the consequences of using the mid-air gesture-based interface in public, we defined the third sub-question:

> *How do users experience interacting with a mid-air gesture-based interface in public?*

In order to answer the last research question the mid-air gesture-based interface was placed at a social event. The interface was adapted to allow users to buy beverages through a web page during the social event using the mid-air gesture-based interface.
Our observations showed that the users in general found the mid-air gesture-based interface entertaining and exciting. Some participants even displayed competitive behavior against each other: who was the best at performing gestures, making the fastest purchase etc., even though there was no indication of a score. We divided our observations into five categories that described, how the users interacted and experienced our mid-air gesture-based interface. The five categories were: Dynamics of approach, Social inhibition, Learning the system, Playing around and Performance and user experience. The interaction is slower than using a computer mouse, but we observed advantages of using the mid-air gesture-based interface. The trivial task of buying beverages at the university became entertaining and the interface attracted people. Areas that could potential gain from using such interface could be: Medical industry (sterile environment), workshops and kitchens (dirty hands) and public places that wish attention (shopping windows, bars, tourist agency).

To summarize and answer the overall research question we have shown that the users were able to browse a web page using the mid-air gesture-based interface with some limitations for recognizing gestures and the time required to move the cursor. We presented a model for predicting task completion time for expert users using the interface. Furthermore we have observed users' interaction when using such interface in public and identified categories describing the use. Finally we have made suggestions on, where it could be beneficial to use this interface, based on our observations and the users' ideas.

4.2 Limitations

This section describes the limitations that should be taken into considerations, when reviewing our results.

We conducted several laboratory experiments during this study. An effort was made to recruit participants, both females and males of variating ages and IT-experience, but due to limited access to participants the main part of the participants was male computer science students at Aalborg University.

The Keystroke-Level Model assumes that the users are experts and completes the interaction without errors. It is difficult to determine when a user reaches the level of expert, we can therefore not guarantee that all the data used to define values for the model was collected using only expert users. The same issue was present when the validation of the model was conducted.

It must be noted that the results of the field study might have been influenced by the author's subjective opinions.

4.3 Future work

The experience acquired during this study lead to ideas for future work.

Based on suggestions from participants it could be interesting to explore new ways of performing the "Click"-gesture.

The mode switch used in one of the techniques for controlling the cursor showed to be difficult to predict. A possible solution could be to modify the model to incorporate new operator(s) to describe the mode switch.

Furthermore it could be interesting to improve the mode switch to allow users to change the cursor speed more smoothly. Our idea is to determine the speed of the cursor based on the distance between the user's hand and chest. If the user stretches the right hand toward the screen, the speed of the cursor is reduced, while pulling the hand towards the body increases the cursor speed. This would cause the movement speed of the cursor to be more dynamic and the mode switch more smooth.

Due the fact that the training mode was frequently omitted when the mid-air gesture-based interface was placed in public, we have considered another technique for helping users. We suggest an intelligent learning system, where the user only would receive help on how to perform gestures, when the system detected that the user was trying to perform a gesture, but failed. The system would then display how to perform the gesture that was most similar to the movement from the user.

We find it interesting to examine the prediction of our modified version of the KLM for larger and more complex tasks. Furthermore it would be interesting to study whether the developed mid-air gesture-based interaction could be used in other contexts than web browsing.

BIBLIOGRAPHY

[1] H. M. Systems, "Dance Central 2." http://www.dancecentral.com/, October 2011. [Online; accessed 12-May-2012].

[2] T. Beauvisage, "Computer Usage in Daily Life." http://laborange.academia.edu/ThomasBeauvisage/Papers/417510/Computer_usage_in_daily_life, 2009. [Online; accessed 19-December-2011].

[3] OpenNI, "OpenNI." http://www.openni.org/, 2012. [Online; accessed 01-June-2012].

[4] J. H. F. Jonathan Lazar and H. Hochheiser, *Research Methods in Human-Computer Interaction*. Wiley, 1nd ed., 2010.

[5] E. W.Weisstein, "Latin Square From MathWorld–A Wolfram Web Resource." http://mathworld.wolfram.com/LatinSquare.html. [Online; accessed 10-December-2011].

[6] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko, "It's mine, don't touch!: interactions at a large multi-touch display in a city centre," in *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, (New York, NY, USA), pp. 1285–1294, ACM, 2008.

[7] U. of Maryland at College Park, "Questionnaire for User Interation Satisfaction." http://lap.umd.edu/quis/. [Online; accessed 31-May-2012].

[8] J. Wynekoop and S. Conger, "A review of computer aided software engineering research methods," in *Proceedings of the IFIP TC8 WG 8.2 Working Conference on The Information Systems Research Arena of The 90's*, 1990.

[9] M. Müller, "Information Retrieval for Music and Motion." http://www.springer.com/978-3-540-74047-6, October 2007. [Online; accessed 15-November-2011].

Developing a Mid-Air Gesture-Based Interface for Web Browsing

Lasse Andreasen
Aalborg University
Department of Computer Science
DK-9220 Aalborg East, Denmark
landre07@student.aau.dk

Rasmus Hummersgaard
Aalborg University
Department of Computer Science
DK-9220 Aalborg East, Denmark
rhumme06@student.aau.dk

ABSTRACT
Gesture-based interaction is being implemented as standards for gaming consoles, smart phones and tablets. The physical buttons are replaced by gestures on surfaces, with controllers in mid-air or movements in mid-air for interacting with such devices. This paper focuses on using mid-air gesture-based interaction for web browsing on a PC using a Kinect sensor. Three prototypes were developed incrementally. The first prototype was developed as a proof-of-concept of a mid-air gesture-based interaction. The second prototype improved cursor control and gesture recognition, where four different techniques for each were tested, and one identified as the most suited. The third prototype was developed with focus on calibration of users and learning of gestures. To teach users to perform gestures, feedback and animations were used and tested to find the most effective technique. The optimal solution for recognizing gestures resulted in 89% of the performed gestures being recognized correctly. The time to navigate the cursor was found to be three times greater than with a standard computer mouse.

Author Keywords
Mid-air gesture-based interaction, Kinect sensor, Web browsing.

INTRODUCTION
Gesture-based interaction is becoming available on more and more computerized devices. The interaction is changing from pressing buttons to making gestures with fingers on touch screens and movement of controllers and body parts in mid-air. The iPhone and iPad are two well known examples of gesture-based interactions with fingers on touch screens. The user navigates by swiping and tapping on the screen with their finger, for example swiping the finger to the left navigates to the next image or screen. The interaction with touch screens is defined as touch-dependent, as gestures only can be performed, when the user is touching the surface. See Figure 1A.

The gaming consoles, Nintendo Wii [17] and Playstation 3 Move [20], have implemented gesture-based interaction that functions in mid-air with controllers. The user holds the controller, as if it was the object they are using in the game. With Nintendo Wii Sports (tennis) [21] the user holds and moves the controller as a tennis ratchet, which is translated into movements of the virtual tennis ratchet. We have defined the interactions with PS3 Move and Nintendo Wii as controller-dependent mid-air, since the user is required to hold the controllers in order to interact with the game. See Figure 1B.

The Xbox 360 with the Microsoft Kinect [14] uses a mid-air gesture-based interaction that works without controllers, the user must interact using only body movements. Interaction using the Kinect sensor is defined as mid-air, because this only require a user within the view of the Kinect sensor. See Figure 1C.

Figure 1. Gesture-based interactions

The same type of gesture-based interaction has not been seen for PC's, and the most used standard input devices are still the keyboard and computer mouse. Mid-air gesture-based interaction might be applied in different situations where keyboard and a computer mouse is not ideal, such as places where touching an input device is impossible or not favorable. This paper explores the extent to which mid-air gesture-based interaction can be developed and replace standard input devices for basic interaction on a PC. We have based our development of a mid-air gesture-based interaction on the functionality Kinect sensor. The Kinect sensor is a motion sensor, and it was chosen, because it is a respectively new and interesting technology that provides the functionality of tracking a user and identifying different parts of a user's body. By using a motion sensor the user is not required to wear or use any devices. To narrow down the field of our study we focus on a common usage for PC's, web browsing[3], with a standard computer mouse.

The next section describes related work for this study. We then present the incremental development of the three prototypes for a mid-air gesture-based interaction. We describe the design, implementation and evaluation of each of the three prototypes. Finally we discuss the evaluations of the three prototypes and present our conclusion of this study.

RELATED WORK

We have divided related work into four elements: gesture-based interaction, replacing standard input devices, gesture recognition and feedback. Gesture-based interaction concerns gesture-based interaction on different devices. Replacing standard input devices is focused on alternatives to the standard input devices for PC. Gesture recognition is aimed at techniques for recognizing gestures, and where they have been applied with which results. Feedback is focused on techniques for presenting feedback to the users, when they perform gestures.

Gesture-based interaction

Gesture-based interaction has been used to interact with different types of system. Gaming consoles such as PlayStation 3[20] and Xbox 360 with Kinect[14] both offer the possibility of controlling the system using gestures in mid-air.

PC systems have been developed which gives surgeons direct control of computerized systems using gestures, based on data from wireless sensors placed on the surgeon [4] or using cameras [10].

Furthermore Perry et. al. [19] used a web camera to recognize a wave gesture used to browse through a DVD collection.

Different devices have been used to capture gestures, such as cameras [10], touch-screens [8], pens [5] and sensors [4]. Camera-based systems offer the possibility of creating a walk-up-and-use interface since the user is not required to wear any specific clothing or devices.

Web browsing on a PC

Gestures-based interaction is also used in web browsing to perform simple navigation tasks such as moving forward or backward in the web history by performing gestures with the mouse [16]. This is also supported by different add-ons for Mozilla Firefox, such as FireGestures [9].

Replacing standard input devices

When controlling a cursor using standard input devices such as a mouse, the movement of the mouse is mapped to the movement of the cursor. Older laptops from IBM used a TrackPoint [13] to control the cursor. When the Track-Point is pushed in a direction, the cursor moves in the given direction. The more pressure that is applied to the TrackPoint the faster the cursor moves.

Dynamic cursor control means when the cursor approaches the area of interest, the cursor speed is decreased. Young Hong et al.[12] described whether it was beneficial to use dynamic cursor control or not. The results showed that dynamic cursor control decreases the time it took navigating from point A to point B.

Gesture recognition

Bao et al. [1] described a system capable of recognizing 26 alphabetical hand gestures using the Dynamic Time Warping-algorithm. Wobbrock et al. [22] developed an algorithm called $1 recognizer, capable of recognizing gestures. The $1 recognizer was compared to both the DTW and Rubine algorithm showing that $1 recognizer and DTW performed very well (success rate of 99.02% and 99.15% respectively) whereas Rubine was significantly more inaccurate.

Feedback

In order for users to be able to use a gesture-based interface, the users must be able to perform the specific gestures correctly. Feedback has showed to be a way of teaching users how to perform gestures. Freeman et al. [8] conducted an experiment with 22 participants (8 female) showing that feedforward and feedback were more effective compared to using a video demonstration to teach users how to perform a predefined set of gestures. This is also supported by Bau et al. [2] which got very similar results. Kratz et al. [15] described a study, where they filtered and smoothed the input from a user's finger on the touch screen, before visualizing it on the screen. The experiment with 12 participants showed that the participants were able to perform more correct gestures with the smoothed data as feedback than with the original data.

PROTOTYPE 1

The purpose of Prototype 1 was to create a proof-of-concept of a mid-air gesture-based interaction for basic web browsing on a PC and to compare the use of a standard computer mouse against the developed interaction when browsing a web page. Prototype 1 enables the user to control the cursor and perform gestures mapped to different web browser functionality, when the user is within the view of the Kinect sensor.

Prototype 1 was developed in C# with the use of OpenNI Software Development Kit[18] and the Kinect sensor connected with a PC. OpenNI SDK is capable of retrieving coordinates from 15 different joints of a user's body, which are the hands, elbows, shoulders, torso, head, hips, knees and feet using the depth image provided by the Kinect sensor. OpenNI SDK is also able to detect, when a user enters and leaves the view of the Kinect sensor.

Design

For interacting with a web browser a technique for choosing elements and browser specific functionality is required, and this is described in Gestures. Furthermore the ability of navigating to elements is required, and we have chosen to use a basic computer mouse as inspiration, and this is described in Control of cursor movement.

Gestures

We have defined the gestures that must be supported, by studying the functionality of a web browser. When using a web browser the user must be able to click different elements on a web page, such as links and images, and scrolling up and down. This has been transformed into three gestures: "Click", "Scroll Up" and "Scroll Down". When interacting with a web browser, one of the most used functions in Mozilla Firefox is going back in the web history[7]. Moyle et al.[16] also states that using gestures to move forward and backward in the web history is less time consuming than using the standard backward and forward buttons. This has lead to two additional gestures, "Go Backward"-gesture and "Go Forward"-gesture.

All gestures start in an initial position shown in the first part of e.g. Figure 2. Moving the left hand towards the Kinect sensor is defined as ”Click”-gesture (Figure 2). Moving the left hand upward or downward is defined respectively as ”Scroll Up”-gesture and ”Scroll Down”-gesture (Figure 3 and 4). Moving the left hand to the left or the right is defined respectively as ”Go Backward”-gesture and ”Go Forward”-gesture (Figure 5 and 6).

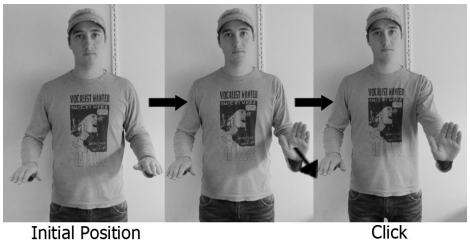

Figure 2. Illustration of ”Click”-gesture

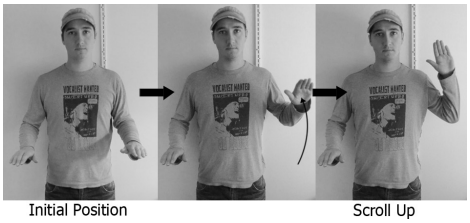

Figure 3. Illustration of ”Scroll Up”-gesture

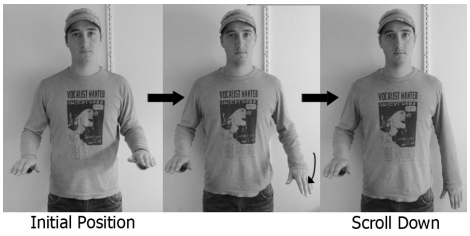

Figure 4. Illustration of ”Scroll Down”-gesture

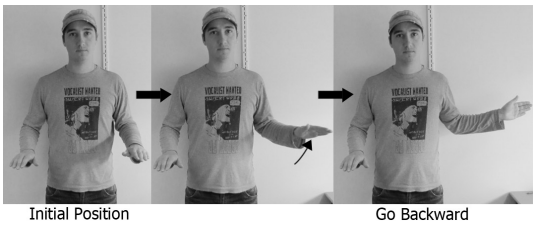

Figure 5. Illustration of ”Go Backward”-gesture

Recognition of gestures

To recognize gestures a suitable algorithm is required, and for this purpose the Dynamic Time Warping-algorithm (DTW) has been found sufficient. The DTW makes it possible to recognize gestures performed at different speeds. Given two sequences, a candidate and a predefined, the DTW is able to determine the distance between the candidate and the predefined sequence. The DTW calculates distances between points in the two sequences and uses these distances to create a matrix, and the length of an optimal path through this matrix determines the similarity of the two sequences.

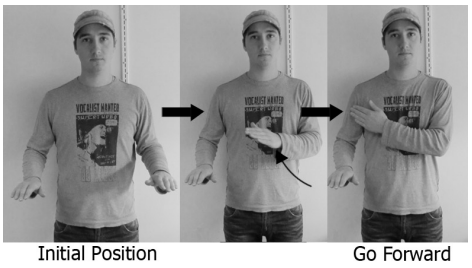

Figure 6. Illustration of ”Go Forward”’-gesture

Control of cursor movement

For interacting with the browser the position of the user’s right hand according to the Kinect sensor was directly mapped to the position of the cursor on the screen. The cursor sensitivity was adjusted to allow the user to position the cursor in each corner of the screen without having to move around.

Implementation

The implementation is divided into two main parts, which are the recognition of gestures and control of cursor movement.

Recognition of gestures

For recognizing gestures Prototype 1 uses four different joints of the user’s body, both shoulders, left elbow and left hand, but when a user moves around within the view of the Kinect sensor, the coordinates of these joints change. The shoulder joints are used to create a point located between the shoulders, from which a distance to the left elbow and hand is calculated. These distances are divided with the user’s shoulder width to ensure that a user also can make gestures close to and far away from the Kinect sensor. The gestures are then defined as changes in the distances to the point between the shoulders over a sequence of 30 continuous observations. This ensures the user is able to perform gestures anywhere within the view of the Kinect.

In order for Prototype 1 to consider a candidate sequence as a match, the DTW calculates the distance between the two sequences which must be below a predefined value. If more than one candidate sequence is considered a match, the candidate sequence with the lowest distance to a predefined sequence is chosen, and the associated action is performed.

Control of cursor movement

To position the cursor the X- and Y-coordinates of the user’s right hand joint is used. In order to reduce the jitter caused by incorrect data from the Kinect sensor and shaking of the human hand a low-pass filter is applied to stabilize the data received from the Kinect sensor.

Evaluation

Participant:	Sex:	Age:	Education:	Test order:
A	M	27	Technical	T-G-M
B	M	24	Technical	T-G-M
C	F	20	Non-technical	T-G-M
D	F	23	Non-technical	T-G-M
E	M	26	Technical	M-T-G
F	M	23	Technical	M-T-G
G	F	20	Non-technical	M-T-G
H	F	23	Non-technical	M-T-G

Table 1. Demographical data of participants

The goal of this evaluation was to identify advantages and disadvantages of the developed mid-air gesture-based interaction and compare it against mouse-based interaction with a standard computer mouse.

System

The system consisted of a 42" screen, a PC, a mouse, a keyboard and a Kinect sensor. The screen was placed on a table with the Kinect sensor mounted on top of the screen. The developed mid-air gesture-based interaction was evaluated using a web page called: "dk-kogebogen.dk", a website containing recipes, and this web page can be navigated with the use of cursor and clicking elements. The different recipes are divided into larger and smaller categories, such as low-fat and low-fat soups. "dk-kogebogen.dk" uses three different font-sizes for links, which we have named small, medium and large.

Participants

A total of eight participants, 4 female and 4 male with variating age and experience with IT, participated, and all of the participants were students. Table 1 shows the demographical data on the participants. In this table the participants' educations are described as technical or non-technical, an example one participant was a software-student and was described as technical, and another participant was pedagogue-student and was described as non-technical.

Setting

The evaluation was conducted in a usability laboratory of the university. Two cameras were set up to record the participants, one camera facing the participants and the other camera recording the participants from behind. A screen capture software was used to record the movement of the cursor throughout the evaluation using mid-air gesture-based interaction as well as mouse-based interaction.

Procedure

The evaluation was divided into two parts, one part using mid-air gesture-based interaction and another part using mouse-based interaction. The evaluation of mid-air gesture-based interaction was performed with the participants standing in front of the screen and using the mid-air gesture-based interaction as the only input device. During the mouse-based interaction evaluation, the participant was seated in front of the screen and given a mouse as input device. Half of the participants started using mid-air gesture-based interaction, and the other half started using mouse-based interaction. The procedures for mid-air gesture-based and mouse-based interaction are shown below:

1. Training using mid-air gesture-based interaction.
2. Test using mid-air gesture-based interaction.
3. Test using mouse-based interaction.

This is referred to as T-G-M in Table 1.

1. Test using mouse-based interaction.
2. Training using mid-air gesture-based interaction.
3. Test using mid-air gesture-based interaction.

This is referred to as M-T-G in Table 1.

The training using the mid-air gesture-based interaction did always take place right before the evaluation using mid-air gesture-based interaction to ensure the participants were presented with same conditions. The tasks presented to the participants consisted of different selection and navigation tasks. A test monitor was responsible for dictating the tasks to the participants during the test session as well as helping the participants during the training session. After completing the test the participants were interviewed and asked about their opinions concerning the gestures and movement of the cursor with the mid-air gesture-based interaction.

Data collection and data analysis

The number of recognized gestures and the number attempts, meaning when a participant tried to perform a gesture, but the gesture was not recognized, were noted. The recognized gestures were further divided into gestures and false hits, where false hits refers to the situation, where the system recognized a movement from the participant that was not intended as a gesture. An example of a false hit was, when a participant scratched his nose, and this was recognized as a "Scroll Up"-gesture. When a participant successfully performed a "Click"-gesture, we noted the font size of the link, whether the participant missed the link or not. The time, the participants used to complete the tasks, was recorded for both the mouse-based and mid-air gesture-based interaction.

A success and false hits rate was calculated based on the data recognized gestures, number of attempts and false hits.

Results

The results are divided into three part: Effectiveness, efficiency and Satisfaction.

Effectiveness

Effectiveness refers to how effective the system recognized the gestures. The overall success rate of performing gestures was calculated to be 58%, meaning that 223 of the 383 attempts to perform gestures were correctly recognized by the system. The overall false hit rate was 7%. Furthermore the relation between link sizes and number of times, where the participants performed the "Click" gesture, but missed the link, were analyzed, but no significant differences between the link sizes were found. The participants were able to click their targets with an accuracy of 72% on average.

Interaction	Move Cursor	Click	Total
Keystroke-level	01.100	0.200	01.300
Mouse-based	01.825	00.325	02.150
Gesture-based	05.620	00.850	06.470
Gesture-based w/ errors	-	-	08.770

Table 2. Time for Keystroke-level model and the average time for mouse- and mid-air gesture-based interaction

Interaction	Going Back	Going Forward
Keystroke-level	01.300	01.300
Mouse-based	01.665	01.135
Gesture-based	00.800	00.867
Gesture-based w/ errors	11.150	00.867

Table 3. Time for Keystroke-level model and the average time for mouse- and mid-air gesture-based interaction (continued)

Efficiency

Efficiency refers to how efficient the participants were able to interact with the web browser with the mid-air gesture-based interaction. The recorded times for the tasks showed a significant increase with mid-air gesture-based interaction according to mouse-based interaction. To get an understanding of this increase, the mouse-based and mid-air gesture-based interaction were divided into four elements, based on the Keystroke-Level Model[6], in order to locate the time-consuming parts of the interaction. The results are summarized in Table 2 and 3.

Satisfaction

Satisfaction refers to the overall impression of the mid-air gesture-based interaction from the participants. The quotes presented were translated from Danish. The participants responses to the mid-air gesture-based interaction were very positive. The participants found the gestures easy to remember and the mapping from the movement to the action logical. One participant stated: "I think that they were easy to remember, because it made good sense that the backward gesture reminded of the back arrow". Overall the participants found the control of the cursor to be easy. However some participants had difficulty holding the cursor still over a link while performing a "Click"-gesture with the other hand, and one participant stated: "It was easy, but it was difficult to be precise, when you should click at the same time".

Summary

Based on the results of the evaluation of Prototype 1 it was seen that the participants were able to use the mid-air gesture-based interaction to control a web browser. However based on observations and interviews two main problems were identified, recognition of gestures and precision with the cursor.

PROTOTYPE 2

Prototype 2 was developed with focus on the problems found in the evaluation with Prototype 1.

Recognition of gestures

With Prototype 1 58% of the gestures were recognized in the experiment, meaning that on average 3 out of 5 gestures were recognized. To increase the success rate the DTW can be changed, so it accepts more imprecise gestures, referred to as a loose setting of the DTW. Another idea is to have several variations of predefined sequences of each gesture, and this is inspired by the article of Hinrichs et. al. [11]. By combining these ideas three additional techniques for recognizing gestures are developed:

- **G1:** one variation per gesture and a strict setting of the DTW (Prototype 1).
- **G2:** one variation per gesture and a loose setting of the DTW.
- **G3:** multiple variations per gesture and a strict setting of the DTW.
- **G4:** multiple variations per gesture and a loose setting of the DTW.

The loose setting of the DTW is close to the setting, where the DTW starts recognizing unexpected gestures, and the strict setting of the DTW is the setting used in Prototype 1. The variations of each gestures are chosen based on the most common variations seen in the experiment with Prototype 1, and the number of variations per gesture is between 3 and 5.

Control of cursor movement

To solve the issue with cursor precision three new techniques for controlling the cursor were developed based on related work [12][13]. The first parameter introduced was dynamic cursor speed, meaning instead of having the cursor speed fixed, the cursor speed can be reduced, when the user wants to be more precise. The second parameter was directional control, so the cursor moves in the direction the hand is moved from a predefined initial point instead of using direct mapping. Three new techniques were developed by combining dynamic cursor speed and directional mapping of the cursor movement:

- **C1:** fixed cursor speed and direct mapping of the hand (Prototype 1).
- **C2:** dynamic cursor speed and direct mapping of the hand.
- **C3:** dynamic cursor speed and directional control.
- **C4:** fixed cursor speed and directional control.

In the following of the implementation of C2, C3 and C4 is described.

C2 follows the hand in the same manner as C1, but it has the ability of dynamic cursor speed, meaning that when the user holds the hand still for one second, the cursor speed is reduced, meaning that a larger movement is required to move the cursor. The cursor speed is returned to normal, when the user makes a movement larger than a specified threshold.

C3 was implemented as directional control, meaning the cursor moves in the direction of which the user moves the hand from the initial point. For this technique the initial point is defined as a safe area, where the user can hold the hand inside and not move the cursor. When the user moves the hand

Participant:	Sex:	Age:	Education:	Test order:
A	Male	24	Technical	1-2-4-3
B	Male	22	Non-technical	2-3-1-4
C	Female	21	Technical	4-1-3-2
D	Female	23	Non-technical	3-4-1-2
E	Male	24	Technical	1-2-4-3
F	Male	24	Non-technical	2-3-1-4
G	Female	23	Technical	4-1-3-2
H	Female	25	Non-technical	3-4-1-2

Table 4. Table of participants

outside of the safe area, the cursor moves in the same direction, as which the hand is moved, Figure 7. The safe area is defined to be in front of the elbow, meaning when the user holds the arm in a 90 degree angle, the hand is inside the safe area. Furthermore C3 has dynamic cursor speed, so the user is able to increasing and decreasing the cursor speed depending on the distance from the hand to the safe area.

C4 is similar to C3, but do not have the ability of dynamic cursor speed, so the cursor moves at a fixed speed, no matter how far the user moves her hand from the safe area.

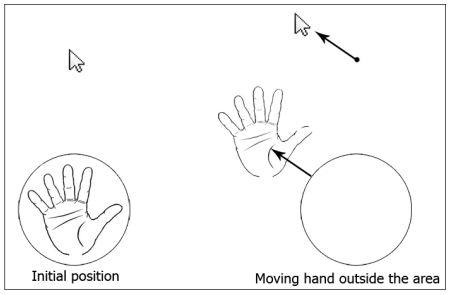

Figure 7. Illustration of C3 and C4

Evaluation

The purpose of this evaluation was to find the most suited technique for controlling the cursor movement and recognition of gestures.

System

The system contained the same elements from the previous evaluation, a 42" screen, a PC, a mouse, a keyboard and a Kinect. The mid-air gesture-based interaction used in this evaluation was a further developed version of the one used in the previous evaluation.

Participants

For this evaluation 8 people participated, 4 females and 4 males with variating experience with IT. Two male and two females with a high IT experience and two male and two female with low IT experience. Table 4 shows the demographical data on the participants. In Table 4 Test order refers to the order, of which the participants tried the four techniques for both recognition of gestures and control of cursor movement. None of the participants have used the mid-air gesture-based interaction before.

Setting

The setting for this evaluation was identical to the setting of the previous evaluation, and the same system and setup as for the evaluation of mid-air gesture-based interaction was used.

Procedure

The evaluation was divided into two parts, recognition of gestures-test and control of cursor movement-test. The order of which the participants tried the different techniques, were defined using Latin Square[23].

Recognition of gestures-test

The participants started with the recognizing gestures-test after receiving a training session, where the five different gestures were demonstrated by the test monitor. The gestures that the test monitor demonstrated, could be recognized by all four techniques for recognizing gestures, and the training session ended, when the participant was able to perform each gesture correct one time.

During the test the test monitor would ask the participant to perform a gesture, and the participant should try to perform the gesture, until it was recognized by the system. When the gesture was recognized the test monitor would ask the participant to perform a new gesture, and this continued, until all gestures had been recognized five times.

Control of cursor movement-test

For the control of cursor movement-test the participants were given a training session of each technique before the actual test of that specific technique. The training consisted of an explanation of the technique and allowing the participant to play around with the technique for approximately 30 seconds.

A task for a technique consisted of four steps that the participant should complete ten times with each technique:

1. Move the cursor to Circle 1.
2. Move the cursor to Circle 2 that appears after completing Step 1.
3. Hold the cursor within Circle 2 for one second.
4. Click the link that appears after completing Step 3.

The distances between Circle 1 and Circle 2 and Circle 2 and the link were always kept the same. The reason for requiring the participant to hold the cursor within Circle 2 for one second was to ensure that the participant did not accidentally hit Circle 2, but needed to be precise with the cursor.

After completing the control of the cursor movement-test the participants were interviewed and asked about their opinions concerning the techniques for controlling the cursor.

Data collection and data analysis

To compare the different cursor techniques the time used with each technique was recorded along with the number of missed click, situations where the participant clicked out side of the target. This was used to calculate an average time for each technique. For the different gesture techniques the number of attempts that the participants use, and false hits rates were recorded. The number of attempts was used to calculate a success rate and the false hits to a false hits rate for each technique.

Results

The results are divided according to the recognition of gestures-test and control of cursor movement-test.

Recognition of gestures-test

For each technique of recognizing gestures the success rate (SR) and false hits rate (FHR) have been calculated and inserted into Figure 8. It must be noted that the results for G1 can not be compared with the results from the evaluation of Prototype 1, even though it is same technique. The reason for this is that the evaluation of Prototype 1 was a more realistic use of the system, whereas the evaluation of Prototype 2 was a performance test.

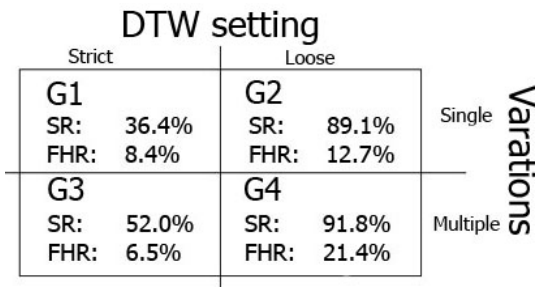

Figure 8. Results of the four techniques for recognition of gestures

Figure 8 shows that either G2 or G4 is the optimal technique when considering a high success rate. To find the optimal solution we have conducted an Tukey's pair-wise comparison test on the success rates of the four techniques. The test revealed a significant difference between strict (G1 and G3) and loose (G2 and G4) setting ($0.00004<p<0.009$), but no significant difference between single and multiple gestures ($0.18<p<0.77$). This means that the loose setting outperformed the strict setting in success rate, but we can not conclude that single outperformed multiple in success rate.

Since loose setting outperformed the strict setting in success rate, we investigated the false hits rate between single (G2) and multiple (G4). A two-sample t-test between G2 and G4 revealed that there was a significant difference in false hits rate ($p<0.048$), meaning that G2 outperformed G4 in false hits rate. Therefore, G2 is the optimal technique for recognizing gestures.

Control of cursor movement-test

For the techniques for controlling the cursor the time to move the cursor and percentage of missed clicks according to all recorded clicks have been calculated and inserted into Figure 9.

Figure 9 indicates that the optimal technique is either C1 or C2 based on movement time. To find the optimal solution we conducted a Tukey pair-wise comparison test that revealed that there was a significant difference between direct mapping and directional mapping ($0.001<p<0.04$). This means that the direct mapping outperformed directional mapping in movement time. For direct mapping, a two-sample t-test revealed there was a significant difference in movement time ($p<0.022$) and in miss clicks ($p<0.028$) between C1 and C2. Therefore, the optimal technique regarding movement time of

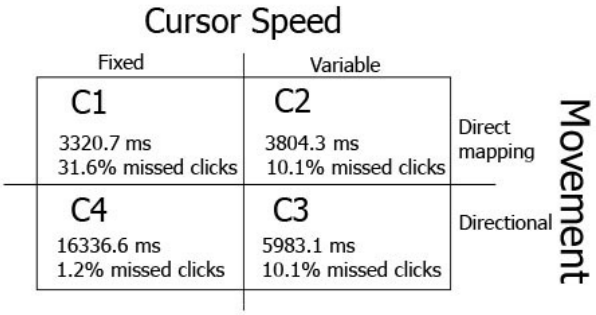

Figure 9. Results of the four techniques for control of the cursor movement

the cursor is C1, and the optimal technique regarding precision of the cursor is C2.

Satisfaction

According to the participants C2 was preferred: "I think it was the easiest. It is because you had more control over the adjustment." "That was certainly the easiest. The thing about it was that it became slower when the hand was kept still." Out of the remaining three C4 received the most negative comment: "Way too slow." "It was hard, because you did not have control over when it (the cursor) should stop". The participants also expressed positive opinions about C1.

Summary

By analyzing the recorded material from the test of Prototype 2, two time consuming ares were identified. The need of requiring a user to be calibrated by the system before beginning the interaction, meaning that the user should stand in a predefined pose and wait for the system to determine position of the users joints. The time spent while being taught how to perform gestures by a test monitor.

PROTOTYPE 3

The purpose of this prototype was to support walk-up-and-use and eliminate the need of having a person teaching the mid-air gesture-based interaction to the users.

Walk-up-and-use

To eliminate the required calibration at the beginning of the interaction, different approaches were tested and the OpenNI SDK studied. Since the second prototype was developed, a new version of the OpenNI SDK was released that offers the possibility of calibrating without the need of standing in a predefined pose. Using this feature the system calibrates a user automatically, when the user enters the view of the Kinect sensor. When calibrating automatically a method is required to determine which user should be in control, meaning that if two users enters the view of the Kinect sensor, which user should be able to perform gestures and control the cursor. The user in control is the person closest to the Kinect sensor. The idea behind is that a person would not stand in front of a user, if they did not wanted to be in control. If people wants to watch the interaction they place them selves behind the controlling user. The torso of the users is chosen to determine the smallest distance to the Kinect sensor, and the torso is used to

prevent situation, where an observer points at an element, but has no interest in controlling the system.

Teaching gestures

Based on the article by Freeman et. al. [8], animations and feedback were considered as two possible techniques of teaching users how to perform the gestures.

Animations

The animations were designed as viewing a user from the front performing the different gestures. The idea was that users watched the animations and tried to imitate the movements shown in the animations.

Visual and audio feedback

The idea with feedback was to guide the users while they tried to perform gestures and show, if they were performing the gestures correctly or not. A 2D representation of the predefined gestures was presented to the user on top of a live video feed of themselves from the Kinect sensor. The gestures were presented as white lines, meaning that the user had to move the hand along those lines. When a user began to perform a gesture, the appertaining white line was painted green, according to how much of the gesture was performed. When the user started to deviate from the predefined gesture, the green color disappeared, and the line turned white again. An example is shown in Figure 10, where the two gestures "Swipe Up" and "Swipe Down" are represented as two white lines initiating from the user's hand and going respectively up and down. The first part illustrates a user performing the "Swipe Up"-gesture, and as seen the white line is painted green according to how much the user has performed the gesture. The second part is an image from the implemented version of feedback.

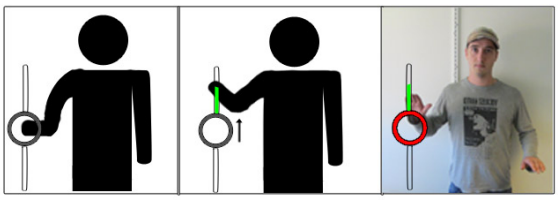

Figure 10. Example of visual feedback

An audio feedback was given, meaning that when a user started to perform a gesture, the system played a continuous tone that changed in frequency according to the user's progress in performing the gesture. When the gesture was recognized, a notable sound was played indicating the recognition of a gesture.

Evaluation

The purpose of this evaluation was to study the advantages and disadvantages of animations and feedback as methods for teaching users to perform gestures.

System

The system for this evaluation was similar to the previous evaluation with a further development of the mid-air gesture-based interaction.

Subject:	Sex:	Age:	Education:	Test order:
A	Male	24	Technical	F1 - A2
B	Male	22	Technical	F2 - A1
C	Male	28	Technical	A1 - F2
D	Male	23	Technical	A2 - F1
E	Male	24	Technical	F1 - A2
F	Male	24	Technical	F2 - A1
G	Male	27	Technical	A1 - F2
H	Male	25	Technical	A2 - F1
I	Male	23	Technical	F1 - A2
J	Male	25	Technical	A2 - F1

Table 5. Table of participants

Participants

For this evaluation 10 people participated, where all of them were male with high experience with IT. Table 5 shows the demographical data for the participants. In Table 5 Test order refers to the order of which the participants were presented with the animations and the feedback. Furthermore it refers to which gesture set, either 1 or 2, the participant tried with feedback and animations. F1 means feedback with gesture set 1, and A2 means animations with gesture set 2. None of the participants have used the mid-air gesture-based interaction before.

Setting

The setting for this evaluation was identical to the setting of the evaluation of Prototype 1, and the same setup as for the evaluation of mid-air gesture-based interaction was used.

Procedure

The participants were given an introduction to the system by the test monitor. Half of the participants started using feedback, and the other half started using animations. Two different sets of gestures were used for testing, and the gesture sets were distributed equally between the two techniques for learning gestures, meaning that one participant would try animation and gesture set 1, and another participant would try the same gesture set and feedback. The reason for this was to reduce the influence of one gesture set possibly being easier to perform for the participants than another gesture set.

The participants were asked to perform a gesture, until it was recognized by the system. When the specific gesture was recognized, the participants were asked to wait for the next task. The test of one technique ended, when each gesture had been recognized 20 times, giving a total of 40 successfully recognized gestures.

Data collection and data analysis

To compare animations and feedback the number of attempts that the participants used was recorded during the evaluation. The total number of attempts was used to calculate how many additional attempts than required the participants used.

Results

For animations the participants used 559 attempts to perform the required 400 gestures, which means they used 39.8% more attempts than required. For feedback the participants

used 513 attempts, which gives 28.3% more attempts than required. Therefore, feedback performed on average 10% better than animation.

Figure 11 shows the average amount of additional attempts the participants used.

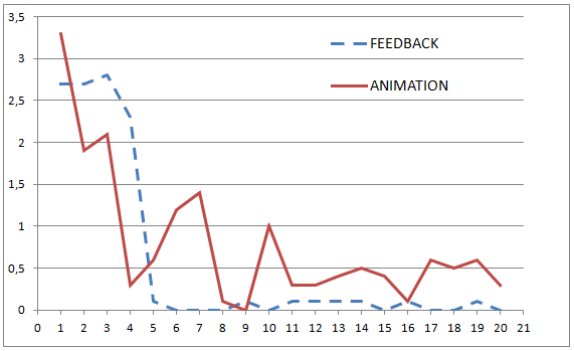

Figure 11. Average amount of additional attempts

Satisfaction

Overall the participants preferred feedback when taught how to perform the gestures. A participant stated: ”I think feedback was the best, because it provided me with feedback on the location of my hand (in the movement)”. The participants said that it was helpful that they were given immediately feedback of their current progress while performing gestures. A participant stated: ”It was shown, how you should move your hand, and I was able to see, where I was in the process.”

The participants stated that it was easy to identify the required movement from the animations, ”I think that it was easiest to see the animations, because then I had an idea of what I should do with my body.” However the participants stated, it was more difficult to replicate it precise: ”You could not tell, whether you did the right thing or were about to do it right”.

DISCUSSION

This discussion is divided into four parts, which are discussions of the results from the three evaluations and a section concerning other observations that were not related to the results.

Evaluation of Prototype 1

From Table 2 and 3 it can be seen that the most time consuming part of Prototype 1 is controlling the cursor movement, whereas making a gesture to go backward and forward in the web history is faster than using mouse-based interaction.

The results showed that 58% of the attempts were recognized, and the gestures were evaluated to find out, whether they were to difficult to perform and remember. The participants’ quotes indicate that this was not a problem, as the gestures were referred to as intuitive and their mapping logical.

An interesting aspect with the interviews was that the all of participants complained about the situation, where they missed a link, and said that this was a major source of irritation, but when comparing to the recognition of gestures, they were able to hit the link with an accuracy of 72%, while only 58% of their gestures were recognized.

Evaluation of Prototype 2

The results regarding recognition of gestures showed that a loose setting of the DTW increased the success rate compared to a strict setting. Furthermore, when using a loose setting, one variation per gesture outperformed multiple variations per gesture. Therefore, the optimal solution was found to be G2 with success rate of 89.1% and a false hit rate of 13.1%.

The results from the evaluation of techniques for controlling cursor movement showed that C1 required the least time to move the cursor, whereas C2 reduced the amount of missed clicks by approximately 20% while being approximately 484 ms slower. Therefore, the optimal technique for controlling the cursor is either C1 or C2 depending on the requirements.

Hong et. al.[12] showed that dynamic cursor control decreased the time required for moving the cursor from point A to point B with a standard computer mouse. This complies with our results for cursor technique C3 and C4 that used directional control and fixed and dynamic cursor speed, where C3 (dynamic) was on average three times faster than C4 (fixed). However with cursor technique C1 and C2, direct mapping and fixed and dynamic cursor speed, the results displayed a disagreement, because C1 (fixed) was on average 484 ms faster than C2 (dynamic). To find a reason for this disagreement we have analyzed cursor technique C2 and assigned the difference to the implementation of C2, because C2 requires one second for activating the mode with reduced cursor speed. If the switch between the two modes could be done automatically, for example when nearing a link or button, the time for C2 could potentially be reduced. Because the difference in navigation time between C1 and C2 is 484 ms, and the time for activating mode switch is one second, this implies that the required navigation time for C2 could be reduced to below the navigation time for C1 and would thereby comply with the results from Hong et. al.[12].

Evaluation of with Prototype 3

The results from the evaluation of Prototype 3 showed that the use of feedback was more efficient than animations, when teaching users to perform gestures. Our results are also supported by Freeman et. al. [8].

As Figure 11 shows, the amount of attempts for feedback stabilizes after the fifth successfully performed gesture and lies within 0 - 1 attempt on average, whereas animations continue to encounter spikes. We analyzed the video material to find an explanation of the spikes observed in the data for animations. We experienced that when a participant made a wrong gesture with animations, they often tried several times, before making adjustments to perform a correct gesture. With animations the participants did not receive any information on, where they were wrong in the movement and therefore, based on the quotes, found it difficult to make adjustments to make the gestures correctly.

The quotes from the participants indicate that a combination of the two would be beneficial. The animations gave a clear impression of how to perform the gestures, because the actual body movement was shown, whereas the feedback helped the participants to be more precise and to understand what they did right and wrong.

Observations

Other observations were made during the evaluations and the most interesting are consistency of gestures and ignoring the possibility of moving.

Consistency of gestures

During the evaluations with Prototype 1 and 2 some participants were able to guess gestures based on previously performed gestures. When some participants were showed the ”Scroll Up”-gesture, they performed the gesture and immediately performed the reflected gesture ”Scroll Down” without any introduction to this gesture. The rest of the participants remained still and waited for new information from the test monitor. This is an indication of the gestures being intuitive and that the use of consistency can increase the learnability of gestures.

Ignoring the possibility of moving

The participants tended to stay in the same place, while using the mid-air gesture-based interaction and would rather stay in an uncomfortable position to hold the cursor at a link than taking a few steps to the side. One participant commented on this: ”When you are focused on the screen, it seems strange that you should move around”. This indicates that people should be notified more strongly of the opportunity of moving around while using the system, if this is required, or the interface should be designed, so this is not necessary.

CONCLUSION

This paper explored the extent to which mid-air gesture-based interaction can be developed and replace standard input devices for basic interaction on a PC. This study was conducted as a proof-of-concept, where three prototypes were developed incrementally. Prototype 1 focused on real use of such a system, allowing the user to control the cursor movement and using five gestures for interaction with a web browser.

Prototype 2 concerned the time-consuming parts of the interaction, namely controlling the cursor movement and recognition of gestures.

Prototype 3 had focus on reducing the time required for calibrating and how to learn the gestures.

With the final prototype evaluations showed that the prototype was capable of recognizing 89.1% of all gestures. The prototype also allowed switching between users without required recalibration. The participants stated that they found the gestures easy to remember and their mappings logically. They also stated that the cursor was easy to control.

However with the final prototype 13.1% gestures were recognized that were not intended by the user. The participants also experienced issues with missed clicks, where on average 10.1% (C2) and 31.6%(C1) of all clicks were clicks that missed the targets. When comparing our mid-air gesture-based interaction with a standard computer mouse the time required to move the cursor from point A to B with our mid-air gesture-based interaction is approximately three times greater.

Our conclusion is that it is possible to develop and replace standard input devices with mid-air gesture-based interaction for basic interaction with a PC with the above limitations.

A limitation with this study is that the evaluations were conducted in a laboratory with 8-10 participants that all were students and a main part computer science students. We can therefore not guarantee that the findings in this study can be applied to the general public. To make a general statement evaluations with a larger number of variated participants is required.

ACKNOWLEDGMENTS

A special thanks to the participants that helped throughout the evaluations.

REFERENCES

1. Bao, J., Song, A., Guo, Y., and Tang, H. Dynamic hand gesture recognition based on surf tracking. In *Electric Information and Control Engineering (ICEICE), 2011 International Conference on* (april 2011), 338 –341.
2. Bau, O., and Mackay, W. E. Octopocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST ’08, ACM (New York, NY, USA, 2008), 37–46.
3. Beauvisage, T. Computer Usage in Daily Life. **`http://laborange.academia.edu/ThomasBeauvisage/Papers/417510/Computer_usage_in_daily_life`**, 2009. [Online; accessed 19-December-2011].
4. Bigdelou, A., Schwarz, L., and Navab, N. An adaptive solution for intra-operative gesture-based human-machine interaction. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, IUI ’12, ACM (New York, NY, USA, 2012), 75–84.
5. Bragdon, A., and Ko, H.-S. Gesture select:: acquiring remote targets on large displays without pointing. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI ’11, ACM (New York, NY, USA, 2011), 187–196.
6. Card, S. K., Moran, T. P., and Newell, A. The keystroke-level model for user performance time with interactive systems. *Commun. ACM 23*, 7 (July 1980), 396–410.
7. Firefox, M. What are the most popular Firefox menu items? **`http://mozillalabs.com/testpilot/2010/03/17/popular-menu-buttons/`**. [Online; accessed 04-January-2012].
8. Freeman, D., Benko, H., Morris, M. R., and Wigdor, D. Shadowguides: visualizations for in-situ learning of multi-touch and whole-hand gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS ’09, ACM (New York, NY, USA, 2009), 165–172.
9. Gomita. FireGestures 1.6.16 . **`https://addons.mozilla.org/da/firefox/addon/firegestures/`**, 2010. [Online; accessed 16-May-2012].
10. Grätzel, C., Fong, T., Grange, S., and Baur, C. A non-contact mouse for surgeon-computer interaction. *Technol. Health Care 12*, 3 (Aug. 2004), 245–257.
11. Hinrichs, U., and Carpendale, S. Gestures in the wild: studying multi-touch gesture sequences on interactive tabletop exhibits. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI ’11, ACM (New York, NY, USA, 2011), 3023–3032.

12. Hong, J.-Y., Chae, H.-S., Yoo, S., Kim, M.-J., and Han, K.-H. Does dynamic cursor control gain improve the performance of selection task in wearable computing? In *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on* (oct. 2005), 224 – 225.

13. IBM. TrackPoint. **http://www.pc.ibm.com/ww/healthycomputing/trkpnt.html**. [Online; accessed 13-December-2011].

14. Kinect, M. Introduction to Kinect. **http://www.xbox.com/da-DK/Kinect/GetStarted**. [Online; accessed 08-May-2012].

15. Kratz, S., and Ballagas, R. Unravelling seams: improving mobile gesture recognition with visual feedback techniques. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, ACM (New York, NY, USA, 2009), 937–940.

16. Moyle, M., and Cockburn, A. The design and evaluation of a flick gesture for 'back' and 'forward' in web browsers. In *Proceedings of the Fourth Australasian user interface conference on User interfaces 2003 - Volume 18*, AUIC '03, Australian Computer Society, Inc. (Darlinghurst, Australia, Australia, 2003), 39–46.

17. Nintendo. Controls for Wii. **http://www.nintendo.com/wii/what-is-wii/#/controls**. [Online; accessed 28-December-2011].

18. OpenNI. OpenNI. **http://www.openni.org/**, 2012. [Online; accessed 08-May-2012].

19. Perry, M., Beckett, S., O'Hara, K., and Subramanian, S. Wavewindow: public, performative gestural interaction. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, ACM (New York, NY, USA, 2010), 109–112.

20. Sony. This Is How I Move. **http://us.playstation.com/ps3/playstation-move**. [Online; accessed 30-December-2011].

21. Wii, N. Wii Sports. **http://wiisports.nintendo.com/games_section/**, 2006. [Online; accessed 05-June-2012].

22. Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, ACM (New York, NY, USA, 2007), 159–168.

23. W.Weisstein, E. Latin Square From MathWorld–A Wolfram Web Resource. **http://mathworld.wolfram.com/LatinSquare.html**. [Online; accessed 10-December-2011].

A Keystroke-Level Model for a Mid-Air Gesture-Based Interface for Web Browsing

Lasse Andreasen
Aalborg University
Department of Computer Science
DK-9220 Aalborg East, Denmark
landre07@student.aau.dk

Rasmus Hummersgaard
Aalborg University
Department of Computer Science
DK-9220 Aalborg East, Denmark
rhumme06@student.aau.dk

ABSTRACT

Predicting time and performance is an essential element when designing new user interfaces. Models, such as Keystroke-Level Model (KLM) and Fitts Law, have been used for predicting time and performance with keyboards and computer mouse. KLM has also been studied and modified for predicting interaction on other devices such as mobile phones and touch screens. We present a modified version of the KLM which can be used to give estimates of time and performance for mid-air gesture-based interaction using a Kinect sensor. We analyze the mid-air gesture-based interaction and propose new operators and values. We also present guidelines for placing operators, when predicting tasks using the model. The operator values are based on empirical data collected through an experiment with users of the mid-air gesture-based interaction. The validation of the model showed a prediction error between -2.6% to 2.0% on three different tasks for basic web browsing.

Author Keywords

Keystroke-Level Model (KLM), mid-air gesture-based interaction, user performance, Kinect

INTRODUCTION

The KLM [19] is used to predict time and performance for interacting with a PC. KLM defines times for different actions, such as moving the cursor and pressing a key. The predicted time used to complete a task is the sum of the values for each of the operators in the given task.

Applying a model for predicting time and performance during the design of a system can potential save a company millions of dollars. In 1982 researchers from Bell Laboratories used Keystroke-Level Model (KLM) to study the work routines at NYNEX. They applied the KLM in order to examine the design of a new system for NYNEX which lead to the conclusion that the old system outperformed the newly designed system, thereby saving the company for a bad investment [8].

The KLM has been applied in several applications domains and for various purposes, and the following are examples of such studies. Bälter [2] used the KLM for analysis of email organization and created an extended model along with rules for email organization and optimization of this. Haunold et. al. [9] used the KLM for studying the task of transforming analog graphical data into digital spatial data. They verified the model and introduced two new operators for pointing in maps and clicking a 16 button cursor. The KLM is also often used for evaluation and comparison of designs, e.g. [6, 10, 4, 3].

Since the development of KLM, KLM has been used and modified to predict interaction with new ways of interacting, such as mobile phones and touch screens, e.g. [14], [20] and [1].

Gesture-based interaction is becoming more widely applied. The iPhone and iPad are well known examples of gesture-based interaction on mobile devices, where the user performs gestures by swiping the finger across the screen and tapping to interact. Gesture-based interaction has also gained interest within the industry of gaming consoles. The interaction with gaming consoles are changing from using classic controllers into mid-air movements. The mid-air movements are performed either with or without a controller. The Nintendo Wii [17] and PS3 Move [18] track a controller in mid-air, and the movement of the controller is used as input. The Xbox 360 with Kinect sensor [12] tracks the player instead of a controller thereby making the player's body the controller. A similar level of implementation mid-air gesture-based interaction for PC's has not been seen.

In this paper we focus on the extent to which the Keystroke-Level Model can be modified to predict time and performance for mid-air gesture-based interaction. To narrow down the field of this study we focus on predicting time and performance for basic web browsing tasks on a PC using a mid-air gesture-based interaction that is based on the functionality of a Kinect sensor. The mid-air gesture-based interaction is shown in Figure 1.

Figure 1. The mid-air gesture-based interaction

In the following section we present related work for this

study. Based on the original model, we analyze the developed mid-air gesture-based interaction in order to identify the operators needed to describe this interaction form. Values for each of the operators are estimated based on an experiment conducted with 10 participants. Finally we evaluate our model, discuss the findings and conclude on this study.

RELATED WORK

For the related work we are focusing on the appliance of KLM in different application domains, and we have divided this into three parts. The first part concerns the original KLM, which is used for interaction with keyboard and computer mouse on a PC. The second part is focused on studies that use or extend the KLM for mobile devices. In the third part we focus on studies that make use of the KLM for touch screens.

To present an overview of the selected articles we have placed them in Table 1. For this table Mid-air Gesture-based refers to interaction with gestures without using any handheld devices and instead only movements of the body.

PC:	**Mobile:**	**Touch:**	**Mid-air Gesture-based:**
[2],[9], [8], [6], [10]	[14], [20], [11], [13], [15]	[1], [4], [3]	

Table 1. Table of selected articles

KLM for PC interaction

With the possibility of modeling user tasks designers can analyze user complexity for interactive systems and identify areas, where the required time can be reduced. Being able to model user tasks has been given much attention for several years, and Card et. al. [19] published the GOMS (Goal, Operators, Methods, Selection of rule) model for such a purpose in their book in 1983. From the GOMS model an instance called the Keystroke-level model (KLM), was developed. The KLM is designed to estimate the time it takes to complete simple input tasks with keyboard and computer mouse.

Bälter [2] used the KLM for analysis of email organization and created an extended model along with rules for email organization and optimization of this.

Haunold et. al. [9] used the KLM for studying the task of transforming analog graphical data into digital spatial data. They conducted an experiment with 7 users that should solve 38 tasks using a digitizing program based on AUTOCAD. Based on the analysis of the tasks they introduced two new operators, one for pointing in maps and one for clicking a 16-button cursor. Finally they validated their model with an average prediction error of 5%.

KLM for Mobile interaction

The Keystroke-level model has also been applied for mobile devices. Hollies et. al. [11] extended the original KLM to a model for mobile devices. They extended the model with new operators and revisited values for existing operators based on 7 studies. They finally validated their model with a new experiment, where the results showed a prediction error of 3% and 5%.

Luo et. al. [14] studied the KLM and its applicability to predict task execution times on stylus-based interfaces on handheld devices. They modeled four tasks using CogTool [5] and performed an experiment with 10 expert users of a Palm Vx PDA in order to verify their KLM's. Their results showed an average of 3.7% error for the predicted and the measured data, and they concluded that the original KLM should be updated with a Grafitti-stroke operator for making a stroke with stylus-based interfaces.

Teo et. al. [20] used the data from Luo and John to make a further investigation of the KLM on handheld devices. Through an analysis of overlapping operators and Mental Prepare they found a error of 0.4% for predicted and measured data for the KLM.

Furthermore Li et. al. [13] conducted an experiment with three participants that all had at least four years of experience with mobile phones and GPS-applications on mobile phones. The participants should solve tasks using a GPS-application on two different mobile phones, and by analyzing the interactions Li et. al. identified 14 new operators for mobile phone interaction.

Siewiorek et. al. [15] extended the KLM for prediction of user time and energy consumption, referred to as Keystroke-level Energy Model (KLEM). They conducted an experiment with 10 participants that should solve tasks on an iPaq and Tungsten that were connected to a power supply for measuring the energy consumption. The resulting predictions with the KLEM were within 13% of measured user time and energy for the iPaq and Tungsten.

KLM for Touch interaction

Evgeniy [1] showed that the KLM can be applied for middle-sized touch screens with acceptable accuracy level. Evgeniy developed a touch interface for controlling a HDD/DVD recorder through Internet Connection Sharing and performed an experiment with 16 participants. The results showed that KLM prediction error was less than 2-5%, and Evgeniy concluded that the KLM can be used for middle-sized touch screens.

KLM for Mid-air Gesture-based interaction

As it can be seen in Table 1 we have found no articles concerning KLM with mid-air gesture-based interaction. We find it therefore interesting to study this area, and the contribution of this article is our research of this area, resulting in a model for gesture-based interaction that is based on the original KLM.

MID-AIR GESTURE-BASED INTERACTION

We have developed a mid-air gesture-based interaction for controlling the basic functions of a web browser on a PC. The gesture-based interaction is developed with the functionality of Kinect sensor [12] that is used track a user and the user's

movements. The user can control the cursor with the right hand and make gestures with the left hand.

Control of cursor movement

To select items on a web page, it is necessary to be able to control the cursor. We have developed two cursor techniques for controlling the cursor, C1 and C2.

C1 directly maps the position of the user's hand to the position of the cursor on the screen based on the location of the hand within the view of the Kinect sensor. The sensitivity of the cursor has been adjusted to allow the user to touch every corner of the screen with the cursor without requiring the user to move around.

C2 is developed with the purpose of enabling the user to be more precise, when desired. **C2** functions very similar to **C1**, but utilize dynamic cursor speed by allowing the user to switch between two modes, Mode 1 and Mode 2. Mode 1 uses direct mapping and the same fixed cursor speed as **C1**. Mode 2 uses direct mapping and a reduced cursor speed, meaning that a larger physical movement is required to move the cursor the same distance than with Mode 1. Mode 2 is activated, when the user holds her hand still for one second, and deactivated when the user performs a movement larger than a specified threshold.

- **C1**: Direct mapping with fixed cursor speed.
- **C2**: Direct mapping with dynamic cursor speed.

Gestures

In order to support basic web browser interaction, a set of gestures is required. In order to select elements on a web page, the user must be able to perform the functionality of a mouse click. To view content on web pages that require vertical scrolling, the interface must support this functionality. According to Mozilla Labs [7] going backward in the web history is one of the most used functions in Mozilla Firefox, when browsing. Based on Moyle et. al. [16], using gestures to move backward and forward is less time consuming than using the standard backward and forward buttons in web browsers. As a result, we have developed a set of five gesture for basic web browsing:

- "Click"-gesture, which functions as clicking on a computer mouse (Figure 2).
- "Scroll Up"-gesture, which functions as pressing the page up-button on a keyboard (Figure 3).
- "Scroll Down"-gesture, which functions as pressing the page down-button on a keyboard (Figure 4).
- "Go Forward"-gesture, which navigates forward in the web history of the web browser (Figure 5).
- "Go Backward"-gesture, which navigates backwards in the web history of the web browser (Figure 6).

All gestures start in the same initial position, holding the left arm in a 90 degree angle, followed by moving the arm in a certain direction. As an example, in order to perform the "Click"-gesture, the user is required to place the left arm in the initial position and then pushing the left hand towards the screen.

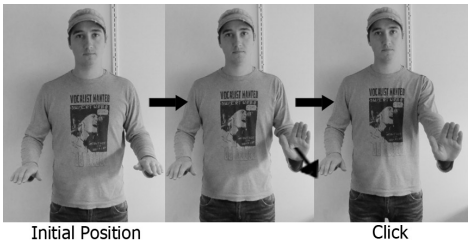

Figure 2. Illustration of "Click" gesture

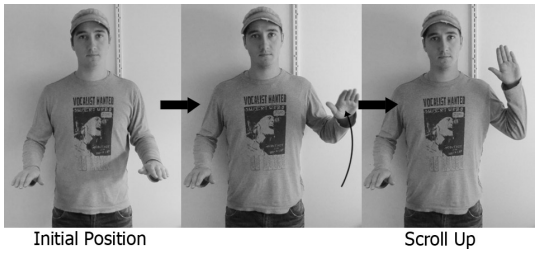

Figure 3. Illustration of "Scroll Up"-gesture

Figure 4. Illustration of "Scroll Down"-gesture

MODEL OPERATORS

The original KLM consists of the six following operators[19]:

- ***K*** - Keystroke, pressing a key on a keyboard, ranging from 0.12 - 1.2 seconds based on the user, and 0.28 recommended for most users.
- ***P*** - Pointing with the mouse to a target on the display, 1.1 seconds.
- ***H*** - Homing hand(s), the time required to move the hand between keyboard and mouse, 0.40 second.
- ***D***(n_D, l_D) - Drawing n_D straight-line segments with a total length l_D cm, $0.9n_D + 0.16l_D$ seconds.
- ***M*** - Mentally prepare, mental preparation of the task for the user, 1.35 seconds.
- ***R***(t) - Response by system, waiting for the system in time t.

By analyzing the mid-air gesture-based interaction the operators needed to describe interaction have been found. The following subsections describe the omitted, modified and new operators.

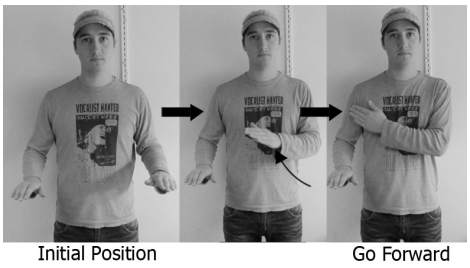

Figure 5. Illustration of "Go Forward"-gesture

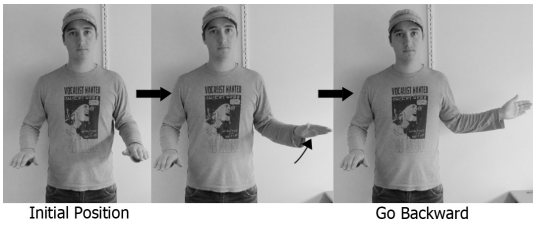

Figure 6. Illustration of "Go Backward"-gesture

Omitted operators

A subset of the operators is not relevant to describe the interaction with a mid-air gesture-based interface.

Since the mid-air gesture-based interface does not support the use of a physical keyboard, the ***K***-operator is removed.

The use of a physical keyboard on PC also leads to the next operator, the ***H***-operator. The ***H***-operator is omitted since there is no switching between input devices, as there is only one input devices, namely the user's hands.

The ***D***(n_D, l_D)-operator is removed since drawing is not relevant.

Unchanged operators

The ***M***-operator is kept as in the original model.

Several other studies have shown that the value of this operator is applicable for other application domains and devices. As an example, Hollies et. al. [11] adopted the original value of 1.35 seconds and was able to predict the time used to complete tasks on a mobile phone with average deviations of 5% and 3%. Haunold et. al. [9] adopted the original value for the ***M***-operator for manual map digitizing with an average prediction error of 5%.

Modified operators

The ***P***-operator is originally used to describe the time it takes to move a cursor from object A to object B using a standard computer mouse. Since the cursor is controlled using the right hand in mid-air, it is likely that the original timing value is not applicable. With the mid-air gesture-based interface the user is required to move the right hand more to move the cursor than with a standard computer mouse. The value for the ***P***-operator must be examined to ensure it describes the time required to move the cursor from object A to object B using the mid-air gesture-based interface.

New operators

The mid-air gesture-based interface differs from the standard keyboard and mouse interaction which means that a set of new operators must be introduced. The mid-air gesture-based interface is designed, so no human touch is required opposite to when interacting with keyboard and computer mouse. Furthermore the mid-air gesture-based interface requires larger movements in order to interact than with the standard keyboard and computer mouse.

New operators are required for our gestures, because the original model does not support gestures. The ***C***- ("Click"-gesture), ***SU***- ("Scroll Up"-gesture), ***SD***- ("Scroll Down"-gesture), ***GF***- ("Go Forward"-gesture) and ***GB***- ("Go Backward"-gesture) operators are therefore introduced as gesture-based operators.

When a user has performed a gesture, the user must return the left arm to initial position to be able to perform a new gesture. Because each gesture requires a different movement with the left arm, five new operators are developed that describe the time needed to return the hand to initial position from each gesture. The names for the five operators consist of the name of the gesture that has been performed and IP (Initial Position), e.g. for "Scroll Up"-gesture the operator is called SU_{IP}. The movement of returning the left arm to the initial position might seem similar to the ***H***-operator from the original KLM, where the user moves the hand between the keyboard and mouse. However because we have five different gestures, we have developed an operator for each gesture.

The modified KLM

The template of the final model is shown in Table 2.

Operators	
P (Point)	
Gestures	C (Click) SU (Scroll Up) SD (Scroll Down) GF (Go Forward) GB (Go Backward)
IP	C_{IP} (IP from Click) SU_{IP} (IP from Scroll Up) SD_{IP} (IP from Scroll Down) GF_{IP} (IP from Go Forward) GB_{IP} (IP from Go Backward)
M (Mental Prepare)	
$R(t)$ (System response)	

Table 2. Overview of the proposed operators

The execution time is calculated by adding the times for the operators used to describe a given task together. As an example, if one would move the cursor from object A to object B, click object B and move backward in the web history the equation would be:

$$T_{execute} = T_M + T_P + T_C + T_{C_{IP}} + T_{GB} \qquad (1)$$

The equation above assumes that there is no wait for the system to respond, and $T_{execute}$ will be the total time for the interaction.

Participant:	Sex:	Age:	Started with:
A	M	29	C1
B	M	24	C2
C	M	26	C1
D	M	25	C2
E	M	24	C1
F	M	24	C2
G	M	25	C1
H	M	24	C2
I	M	27	C1
J	M	24	C2

Table 3. Demographical data of participants

USER STUDIES FOR TIME MEASUREMENTS

To be able to predict an interaction with the mid-air gesture-based interface the times for each of the operators must be defined. To gather the necessary data to calculate the times for the operators, user studies were carried out. The goal of the user studies was to record the times, the participants used to perform each of the actions, and thereby calculated an average time for the operators.

System

The system used for testing consisted of a PC running the mid-air gesture-based interface connected to a 42" screen, where a Kinect sensor was mounted on top. A keyboard and computer mouse were connected to the PC as well.

Participants

A total of ten participants, all males with variating ages ranging from 24 - 29, participated, and all of the participants were male students at technical educations. Table 3 shows the demographical data on the participants.

Due to a technical error, the data from participant "J" was corrupted and therefore omitted.

Setting

The experiment was conducted in a usability laboratory of the university. Two cameras were to record the participants while testing, one camera facing the participants and the other camera recording the participants from behind. A screen capture software was used to record the cursor movement and the image feed from the Kinect sensor.

Procedure

The experiment with the mid-air gesture-based interaction was performed with the participants standing in front of the screen and using the mid-air gesture-based interaction as the only input device. The experiment was divided into two parts, one part for gestures and another part for pointing with the cursor. Before each of the two parts, a training session was carried out with each of the participants. For both parts the participants were told to return their left arm to initial position after each performed gesture.

The first part of the experiment was conducted to determine the time used to perform the four gestures: "Scroll Up", "Scroll Down", "Go Backward" and "Go Forward" and the time to return the left arm to initial position for each of the four gestures. The participant was asked perform one of the four gestures until the gesture was recognized. After the system had recognized the gesture the test monitor asked the participant to perform another gesture. The test was finished after a total of 15 successful attempts to perform each gesture.

The second part of the experiment was conducted in order to determine the time used to move the cursor as well as the time used to perform the "Click"-gesture and to return the left arm to initial position from the "Click"-gesture. Each participant was asked to perform 15 tasks with both techniques to control the cursor, C1 and C2. Each task followed the same structure:

- Place cursor in Circle 1.
- Circle 2 appears.
- Move cursor to Circle 2.
- Keep the cursor within Circle 2 for one second.
- A link appears with the label "Click"
- Move cursor to the link.
- Click the link.

To ensure the time logged between Circle 1 and Circle 2 was correct, and not reflecting the participants accidentally hitting Circle 2, the participants were required to keep the cursor within Circle 2 for one second. The distance to and size of Circle 2 changed throughout the experiment to ensure that the difficulty of the tasks varied.

Data collection and data analysis

To be able to determine the time required to perform each of the gestures, the video material from each of the participants was analyzed. The start of a gesture was defined to be when a participant started to move the left arm to perform a gesture. The end of a gesture was defined to be the exact time the system recognized the performed gesture. The times for the operators for returning to initial position for the five gestures were defined as the duration, from when a given gesture was recognized by the system, until the participant had placed the left arm in the initial position. Data from gestures that were not recognized by the system, was omitted.

The time used to move the cursor was logged automatically by the system during the experiment. The timer was started when the cursor left Circle 1 and ended the last time the cursor entered Circle 2. If the participant overshot Circle 2 the timer was not stopped until the participant stayed in Circle 2 for one second. Furthermore a timer was started when the participant left Circle 2 and ended when the user hit the link, labeled "Click", the last time. The reason for required the participant to stay in Circle 2 for one second was to remove the possibility of the participant hitting Circle 2 by accident.

OPERATOR VALUES

Based on the user studies the values for each of the model operators have been found and are shown in Table 4. The value for the ***P***-operator has been divided according to the two techniques (C1 and C2) for controlling the cursor.

Operators		Values (sec)
P		C1:1.504 C2:1.706
Gestures	C	0.426
	SU	0.345
	SD	0.556
	GF	0.325
	GB	0.292
IP	C_{IP}	0.906
	SU_{IP}	1.073
	SD_{IP}	1.094
	GF_{IP}	1.008
	GB_{IP}	1.092
M		1.35
$R(t)$		Variable

Table 4. Overview of the proposed operator values in seconds

P-operator

As shown in Table 4 the ***P***-operator for C2 is 202 ms greater than C1. The data from the user study has been thoroughly analyzed and shows that the increase in time for the ***P***-operator is caused by two things. The fact that when controlling the cursor in Mode 2 the cursor moves a lot slower than with C1 which causes an increase in time to move the cursor from object A to object B. If the user wants to increase the speed of the cursor they are required to switch mode by performing a large movement which often causes the cursor to "jump" to another place on the screen forcing the user to use additional time to recover from the switch in modes.

Gesture- and Initial Position-operators

The values are based on the average times used by each participant from all conditions (C1 and C2).

Summary

In general Table 4 shows that it is faster to use C1 than C2 to control the cursor. The reason for designing and implementing C2 was the fact that it could be difficult to hit targets with C1 due to the relatively high cursor speed and the fact that it is hard to keep the hand still in mid-air. The difficulty of hitting targets with respectively C1 and C2 is defined by counting the times the participants tried to click an object and missed (referred to as missed clicks). Using C1 the participant had a miss clicked percentage of 5.6% and using C2 of 2.2% of the total amount of clicks for each technique.

GUIDELINES FOR USING KLM

We have developed guidelines for using our KLM based on the original rules for placing the ***M***-operator in the KLM[19]. Furthermore we have introduced guidelines for placing the new ***IP***-operators.

- Rule 0: Place ***M***'s in front of all ***P***'s and all gesture-based operators (***C***, ***SU***, ***SD***, ***GF***, ***GB***).
- Rule 1: If an operator following an ***M*** is fully anticipated in an operator just previous to ***M*** , then delete the ***M*** (e.g., ***PMC*** becomes ***PC***).
- Rule 2: If a string of ***MPC***'s belongs to a cognitive unit (e.g., writing a known word), then delete all ***M***'s but the first.
- Rule 3: If gestures are performed sequentially with the same hand, an ***IP***-operator for the given gesture should be placed between each of the gesture operators (***C***, ***SD***, ***SU***, ***GB***, ***GF***)

EVALUATION

In order to validate our values for the operators we created a scenario, where users should solve a task using our developed mid-air gesture-based interaction. This section describes the tasks used to validate the model and KLM prediction and the empirical validation of the model.

For validation we recruited 9 different participants, all males and students at computer science with an average age of 24 (23-27). None of the participants had taken part in the earlier user studies for estimating the values of the operators. The participants received training before the validation with an introduction to the mid-air gesture-based interaction and a training task.

Tasks for validation

The task presented to the participants was divided into three subtasks with a total of 19 operators. Task 1 was to type a four letter word on the virtual keyboard by moving the cursor to each of the four letters and perform the "Click"-gesture. Task 2 was defined to be scrolling down on the web page, by performing the "Scroll Down"-gesture one time, followed by moving the cursor to a button and performing a click. Task 3 was defined to be going back in the web history, scrolling up one time and going forward in the web history.

The tasks were developed with the purpose of being tasks for basic web browsing on a PC.

The participants were required to complete the tasks three times without errors with both cursor techniques. If the participants made an error, e.g. performed a wrong gesture, the task was repeated. The times that the participants used for completing each task, were logged.

Placing the M- and IP-operators

One of the more difficult aspects of using the KLM is to determine where to place the ***M***- and ***IP***-operators in the interaction. In order to place the operators when using the model, the user interface as well as the tasks must be defined and analyzed.

Task 1 was defined as shown in Equation 2.

$$T_{execute} = T_M + 4 * T_P + 4 * T_C \quad (2)$$

One ***M***-operator which gives the user time to determine how to spell the four letter word as well as time to recall how to move the cursor and perform the "Click"-gesture. Four ***P***-operators, since the user is required to move the cursor to four different letters Four ***C***-operators, one for each of the four letters that must be clicked. The reason for not placing

	Cursor technique 1 (C1)	Cursor technique 2 (C2)
Task 1:	8.92	9.728
Task 2:	3.66	3.888
Task 3:	4.327	4.327

Table 5. Predicted execution times in seconds

	Cursor technique 1 (C1)	Cursor technique 2 (C2)
Task 1:	8.692 (-2.6%)	10.585 (9.0%)
Task 2:	3.654 (-0.1%)	5.551 (35.4%)
Task 3:	4.415 (2.0%)	4.327 (-3.1%)

Table 6. Empirical execution times in seconds

an ***M***-operator before each of the four ***P***- and ***C***-operator is the fact that typing in a word is a string and belongs to one cognitive unit, as suggested by Rule 2.

The equation of Task 2 was defined as shown in Equation 3.

$$T_{execute} = T_M + T_{SD} + T_P + T_C \quad (3)$$

One ***M***-operator gives the user time to recall how to perform the "Scroll Down"-gesture as well as how to move the cursor and perform the "Click"-gesture. Only one ***M***-operator is placed in Task 2 before the task of scrolling down followed by pointing the cursor and click.

Task 3 was defined as shown in Equation 4.

$$T_{execute} = T_M + T_{GB} + T_{GB_{IP}} + T_{SU} + T_{SU_{IP}} + T_{GF} \quad (4)$$

One ***M***-operator at the beginning of the task which gives the user time to recall how to perform the gestures. One ***GB***-, ***SU***- and ***GF***-operator. The reason for placing ***IP***-operators after the ***GB***- and ***SU***-operator is the fact that when the user has to perform several gestures sequentially with the same arm, the user must move the left arm back to the initial position before performing the next gesture. The ***GF***-operator does not require an ***IP***-operator, as it is the last gesture.

KLM prediction

Based on the guidelines for using the KLM and the values for each of the operators, the predicted times for each of the three tasks have been calculated.

The predicted times for each of the tasks are shown in Table 5.

Empirical validation

From the empirical validation the average times for each task and the deviation of the predicted results have been calculated and are shown in Table 6.

DISCUSSION

This section describes the discussion of this study concerning the mode switch and comparison with related work.

Mode Switch

When viewing the results for Task 1 in Table 6, it is noticeable that the deviation for C2 is approximately 3.5 times larger than with C1 with 9.0%. For Task 2 the deviation is more obvious with a 35.4% deviation from the predicted time.

To find a reason for these deviations we have studied the tasks and C2 further, and find that switching between modes is the main cause.

Task 1

For Task 1 we have seen several examples of, people staying in Mode 2 with reduced cursor speed and then realizing after moving the cursor a distance that the distance to travel is too great, whereupon they switch to Mode 1.

The fact that some participants change their mind on which mode to use during a task (moving the cursor) increases the task completion time. The model assumes that the user decide which mode to use before beginning to solve a task, not that the user change their mind during a task.

Task 2

The reason for Task 2 is deviating by 35.4%, is, based on our observations, an unintentional mode switch. While the participants were performing the "Scroll Down"-gesture, they kept the cursor still which resulted in an unintentional mode switch to Mode 2 (reduced cursor speed). After completing the "Scroll Down"-gesture, the participants started to move the cursor and realized that they had activated Mode 2, resulting in a mode switch to Mode 1 before moving the cursor closer to the "Enter"-button. This unintentional mode switch increases the task completion time more which is not included in the model.

To summarize, the model is not able to handle that the users switch mode during a task or unintentional mode switch. This suggests that further study of mode switching is required to obtain a more precise prediction with C2.

Comparison with related work

To compare our results with other studies we have chosen only to consider the tasks for C1, because of the above mentioned problems with mode switching in C2.

On a PC, Haunold et. al. [9] modified the original KLM to suit the task of transforming analog graphical data into digital spatial data. They introduced new operators and validated their model with an average prediction error of 5%.

For mobile devices, Hollies et. al. [11] extended the original KLM. They added, modified and removed operators and validated their model with a prediction error of 5% and 3% on two tasks.

With a touch screen, Evgeniy [1] showed that the KLM can be applied with acceptable accuracy level. The results showed that KLM prediction error was less than 2%, and Evgeniy concluded that the KLM can be used for middle-sized touch screens.

The results of -2.6% to 2% from the validation of our model concerning C1 comply with the above mentioned results from other studies on different devices.

CONCLUSION

This paper explored the extent of which the KLM could be modified for allowing time and performance prediction for mid-air gesture-based interaction. We studied the original KLM and introduced a modified version of KLM for mid-air gesture-based interaction along with guidelines for placing operators. The operator values used for the modified version of KLM were empirical determined through an experiment conducted with 9 participants. To evaluate the model we conducted a validation using 9 new participants, where we presented them with three basic web browsing tasks, which they should complete using gestures and two different techniques for controlling the cursor, C1 and C2. The results of our validation with C1 and the gestures showed an acceptable level of prediction with an error rate between -2.6% and 2.0%, and this complies with results from other studies that modified the original KLM. Furthermore the validation of our model showed that the modeling of C2 was more difficult than expected because of mode switching.

It must be noted that the empirical data was collected in a laboratory with only male participants, studying computer science. The validation was also performed with male students of computer science. We can therefore not guarantee that our model can be applied to the general public. The original and our modified KLM requires users to be experts, and it can be difficult to evaluate, when a user is considered an expert user.

The results showed that the mode switch in C2 requires further study, which can involve a new implementation of the mode switch or other operators to describe the mode switch. Furthermore it could be interesting to validate the model with more complex web browsing tasks.

ACKNOWLEDGMENTS

A special thanks to the participants, who helped throughout our experiment and validation.

REFERENCES

1. Abdulin, E. Using the keystroke-level model for designing user interface on middle-sized touch screens. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, CHI EA '11, ACM (New York, NY, USA, 2011), 673–686.
2. Bälter, O. Keystroke level analysis of email message organization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '00, ACM (New York, NY, USA, 2000), 105–112.
3. Bragdon, A., and Ko, H.-S. Gesture select: acquiring remote targets on large displays without pointing. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, ACM (New York, NY, USA, 2011), 187–196.
4. Bragdon, A., Nelson, E., Li, Y., and Hinckley, K. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, ACM (New York, NY, USA, 2011), 403–412.
5. CogTool. Welcome to CogTool. `http://cogtool.hcii.cs.cmu.edu/`. [Online; accessed 04-June-2012].
6. e Laila, U., Shah, S., and Ishaque, N. Using keystroke-level model to analyze ios optimization techniques. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for* (dec. 2011), 373 –377.
7. Firefox, M. What are the most popular Firefox menu items? `http://mozillalabs.com/testpilot/2010/03/17/popular-menu-buttons/`. [Online; accessed 04-January-2012].
8. Gray, W. D., John, B. E., and Atwood, M. E. The precis of project ernestine or an overview of a validation of goms. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '92, ACM (New York, NY, USA, 1992), 307–312.
9. Haunold, P., and Kuhn, W. A keystroke level analysis of a graphics application: manual map digitizing. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, ACM (New York, NY, USA, 1994), 337–343.
10. Hinckley, K., Guimbretiere, F., Baudisch, P., Sarin, R., Agrawala, M., and Cutrell, E. The springboard: multiple modes in one spring-loaded control. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, ACM (New York, NY, USA, 2006), 181–190.
11. Holleis, P., Otto, F., Hussmann, H., and Schmidt, A. Keystroke-level model for advanced mobile phone interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, ACM (New York, NY, USA, 2007), 1505–1514.
12. Kinect, M. Introduction to Kinect. `http://www.xbox.com/da-DK/Kinect/GetStarted`. [Online; accessed 28-December-2011].
13. Li, H., Liu, Y., Liu, J., Wang, X., Li, Y., and Rau, P.-L. P. Extended klm for mobile phone interaction: a user study result. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, CHI EA '10, ACM (New York, NY, USA, 2010), 3517–3522.
14. Luo, L., and John, B. E. Predicting task execution time on handheld devices using the keystroke-level model. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI EA '05, ACM (New York, NY, USA, 2005), 1605–1608.
15. Luo, L., and Siewiorek, D. P. Klem: A method for predicting user interaction time and system energy consumption during application design. In *Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers*, ISWC '07, IEEE Computer Society (Washington, DC, USA, 2007), 1–8.

16. Moyle, M., and Cockburn, A. The design and evaluation of a flick gesture for 'back' and 'forward' in web browsers. In *Proceedings of the Fourth Australasian user interface conference on User interfaces 2003 - Volume 18*, AUIC '03, Australian Computer Society, Inc. (Darlinghurst, Australia, Australia, 2003), 39–46.

17. Nintendo. Controls for Wii. `http://www.nintendo.com/wii/what-is-wii/#/controls`. [Online; accessed 28-December-2011].

18. Sony. This Is How I Move. `http://us.playstation.com/ps3/playstation-move`. [Online; accessed 30-December-2011].

19. Stuart K. Card, T. P. M., and Newell, A. *The Psychology of Human-Computer Interaction*. LEA, 1983.

20. Teo, L., and John, B. E. Comparisons of keystroke-level model predictions to observed data. In *CHI '06 extended abstracts on Human factors in computing systems*, CHI EA '06, ACM (New York, NY, USA, 2006), 1421–1426.

Exploring Interaction with a Mid-Air Gesture-Based Interface in the Wild

Lasse Andreasen
Aalborg University
Department of Computer Science
DK-9220 Aalborg East, Denmark
landre07@student.aau.dk

Rasmus Hummersgaard
Aalborg University
Department of Computer Science
DK-9220 Aalborg East, Denmark
rhumme06@student.aau.dk

ABSTRACT

We have developed a mid-air gesture-based interface for interacting with a web browser on PC. The user interacts with the web browser by making gestures in mid-air, and this is developed based on the functionality of the Kinect sensor.

We placed the mid-air gesture-based interface at a social event at Aalborg university where people could buy beverages through a web page. We conducted a field study for a period of 7 hours with a total of 26 different users. Video material from the study was analyzed in order to better understand how users interact with a mid-air gesture-based interface in public. We divided our observations into five different categories that described how users approached, experienced social inhibition, learned how to use the interface, played around and in general experienced the interface. Overall people had a positive impression of the interface.

Author Keywords

Mid-air gesture-based interaction, public spaces, gestures, field study

INTRODUCTION

Gesture-based interaction is being adopted as a common interaction on a variety of computerized devices. Smartphones are using fingers as input for gesture-based interaction, where the user swipes the finger across the screen. A more visible version of gesture-based interaction is adapted by the gaming consoles, where the user makes larger movements in mid-air with controllers (PS3 Move [12], Nintendo Wii [6]) or only the body (Xbox 360 with Kinect [4]) as input.

When interacting with a mid-air gesture-based interface, the user and the user's actions become visible to others. The visibility when using such interface in public can have both positive and negative effects.

The visibility of this interaction can have the positive effect of attracting more people to approach the system, described as the Honey-Pot Effect by Brignull et. al. [10] that used a laptop and a large display to enable people to write comments and opinions. A positive result of gathering people around the system is that people collaborate. Peltonen et. al. [8] experienced with their large multi-touch screen that people that were observing a user, commented and gave advices on how to interact.

A negative effect is that being visible and attracting attention can lead to a restricted interaction. Perry et. al. [9] developed a system supporting wave gestures to browse DVD's and observed during a field study in public that some users were too embarrassed to perform a wave gesture, even though they had watched the initial training video.

This paper explores how users experience interacting with a mid-air gesture-based interface in public.

Inspired by the interaction with a Xbox 360 and the Kinect sensor, where the user's body acts as the controller, we have developed a mid-air gesture-based interface enabling people to browse web pages on a PC. The users were able to control the cursor, click elements and go forward and backward in the web history. The system consisted of a 42" wide-screen TV with a Kinect sensor mounted on the top. To investigate how users interact with such system, the system was installed at a social event, where people could buy beverages using the mid-air gesture-based interface as shown in Figure 1.

Figure 1. Users buying beverages using the mid-air gesture-based interface

We analyzed video material collected over a period of seven hours to identify interesting aspects of the interaction.

In the following sections we present related work followed by a description of our developed mid-air gesture-based interface for web browsing on a PC. We describe how the field study was conducted along with the procedure of analyzing the collected video material. We report our findings, discuss the results and relates them to other studies from related work. Finally we conclude on this study.

RELATED WORK

For related work we study other articles that investigated, how people interact with different systems in public. We have divided the related work depending on the input devices: laptop, touch screen, tabletop and mid-air gesture-based.

Large display using laptop in public

Brignull et. al. [10] developed a system enabling people to write their opinions onto a large shared display. People were able to write opinions and add comments to existing opinions that could be read by the audience. They conducted two studies in a public setting. During the first study their observed that people hesitated to interact with the system, which resulted in the authors adding their own opinions which created a momentum effect. Furthermore they observed that people were able to learn how to use the system by observing others. During the second study, people were interviewed. The results of the second study complied with the results from the first study and showed that people in general were positive towards the system, but social embarrassment played a big role.

Large touch screen using gestures in public

Peltonen et. al. [8] observed the interaction with a system called CityWall placed in public. CityWall consisted of a large touch screen providing the ability to interact with Flickr content through gestures. CityWall was available for eight days and they recorded a total of 1199 users interacting with the system. The analysis of the obtained data resulted in several findings concerning: Dynamics in approach, interacting at the display with others, transition between activities and participants and roles and social configurations.

Tabletop gestures in public

Hinrichs et. al. [2] presented their findings from a field study of a tabletop system enabling people to browse through a media collection. The study was conducted over a period of eight days with a total of approximately 20 hours of video data collected by two cameras. Findings from the study indicated that a versatile many-to-one mapping between gestures and actions were important. Furthermore they presented data showing the difference between children and adults, single- and bimanual interaction and symmetric and asymmetric actions.

Jacucci et. al. [3] developed a system called Worlds of information allowing multiple users to interact with a touch screen at the same time. They described some of the challenges of developing a multi-touch application for walk-up-and-use displays as well as how people learned and interacted with such system.

Mid-air Gestures in public

Perry et. al. [9] developed a system called WaveWindow which enabled people to interact with a screen placed behind a window in public. Interacting with the screen was done using a wave gesture as well as knocking on the window. Based on their observations they proposed design recommendation made for gestural interaction in public.

Rubegni et. al. [11] developed a system called USIAlumni Faces that projected a virtual yearbook onto a large public screen. People could interact with the system using a Wii remote and an infrared pen hidden in a toy case. They set up their system for a university alumni event, where over 200 persons used their system. They described, how people interacting with the system attracted more people, and the use of gestures made the system more visible to others. Furthermore they also described the social aspect of the system working as a conservation starter with acquaintances and strangers, and how people applied the observe-and-learn model.

Hardy et. al. [1] developed a system using a webcam enabling people to play an asteroids game and show weather information by moving the arm to the left or right. They invited individuals and groups to come and interact with the system and observed 46 participants. Their study showed, among other things, that people were more interested in the display when others were using it and the fact that people tend to make larger and quicker movements when the system did not respond as expected.

MID-AIR GESTURE-BASED INTERACTION

We have developed a gesture-based interaction for controlling a web browser on a PC. The gesture-based interaction is based on the functionality of Kinect sensor[4], which is used to track a user and the user's movements. The user can control the cursor with the right hand and make gestures with the left hand. The cursor on the screen is positioned based on the position of the user's hand within the view of the Kinect sensor. When the user holds the hand in upper left corner of the view of the Kinect, the cursor is placed in the upper left corner. We have adjusted the cursor sensitivity to allow the user to touch every corner of the screen with the cursor without requiring the user to move around.

For this system we have also developed three gestures for interacting with the web browser, which are:

- "Click"-gesture, which functions as clicking using a computer mouse (Figure 2).
- "Go Forward"-gesture, which navigates forward in the web history of the web browser (Figure 3).
- "Go Backward"-gesture, which navigates backward in the web history of the web browser (Figure 4).

All gestures start in the same initial position, which meant holding the left arm in a 90 degree angle and then moving the arm in a certain direction. As an example the "Go Forward"-gesture, the user should place the left arm in the initial position and then move the left hand towards the right shoulder.

The system functions in such a way that user, which is closest to the screen, is in charge.

The system has two modes, purchase mode and training mode. Training mode is displayed in Figure 5, and in training mode the user is able to view the camera image from the Kinect sensor, instructions on how to use the system and animations of a person performing the three supported gestures. On the camera image from the Kinect sensor illustrations on

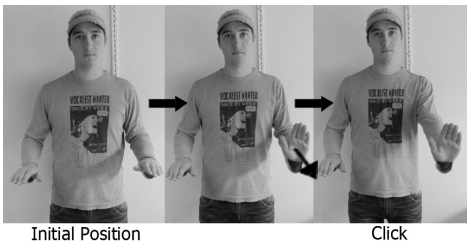

Figure 2. Illustration of "Click" gesture

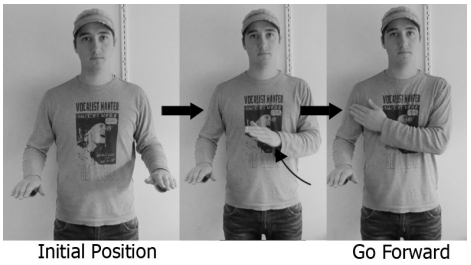

Figure 3. Illustration of "Go Forward"-gesture

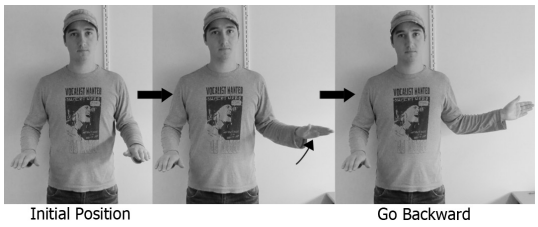

Figure 4. Illustration of "Go Backward"-gesture

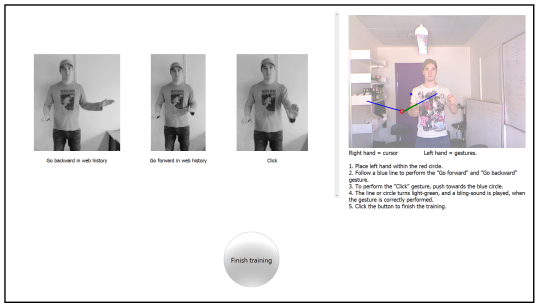

Figure 5. Screenshot of training mode

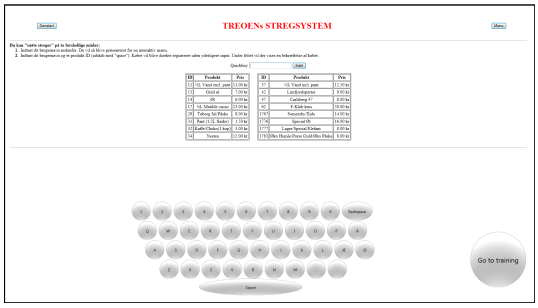

Figure 6. Screenshot of purchase mode

how the user should move her left arm to perform gestures, are drawn.

The purchase mode, shown in Figure 6, displays a web page called "Stregsystemet" in the top half of the screen while displaying a virtual keyboard on the bottom half. The web page, "Stregsystemet", is used at the university to buy different beverages by the students. The web page enables the students to buy beverages by typing in their user name followed by clicking the link of the desired item. To support the possibility to type in a user name, the virtual keyboard was added.

Our system also supports the functionality of scrolling up and down on web pages, however the web page "Stregsystemet" does not require this functionality, and the gestures have therefore been omitted.

FIELD STUDY

The purpose of the field study was to study the interaction with this type of system in a public space.

System

The system used for testing consisted of a 42" screen, a PC running the mid-air gesture-based interface, a mouse, a keyboard and a Kinect sensor. The screen was placed on a table with the Kinect mounted on top of the screen.

Participants

In order to motivate people to use our system we held a competition. Each time a person used our system, their chance of winning was increased. At the end of the evening the winner was announced and given the price.

Setting

The field study was conducted in the canteen at Cassiopeia - House of Computer Science at Aalborg University. During the afternoon and evening a social event was held in the canteen. A camera was placed next to the screen to record the users from the front, enabling us to capture the behaviors and voices of the users.

Procedure

The system was set up and available from 01 pm to 08 pm, and during this time people were free to interact with the system. Both authors were present during the field study in order to ensure the system functioned as expected, but did not interfere with any interaction with the system from the users. When the system was not in use the authors made sure the system was switched to training mode to be ready for the next user.

Data collection and data analysis

To study the interaction with our system we used a high-definition digital camera for recording the users' movements and voices. During the field study we asked the users to fill out a questionnaire concerning the mid-air gesture-based interaction and where they believed such interaction could be used.

Before analyzing the video material we removed the parts, where no users were using the system. The rest of the material was divided into sessions, each session consisting of an

uninterrupted use of the system. Both authors watched the sessions in collaboration and listed interesting situations that emerged in the videos. We then analyzed the sessions again in collaboration and noted, when the situations occurred, the type of situation and the number of people involved, including people that observed the situation. It must be noted that observers only contained people that were within the view of the camera. However since the system was set up on a plateau in the canteen, we are confident that we have recorded the main part of the observers.

FINDINGS

This section describes the findings of our analysis of the video material from the field study. A total of 26 different users used the system in addition to the people observing the interaction. Some users used the system more than once, so we recorded a total of 54 sessions with the system. The section focuses on how people approached, interacted and collaborated with others at the screen.

Dynamics of approach

Approach refers to how people noticed the interface, how people approached the interface and how people were taking turns.

Noticing the system

An observation regarding how people approached the system was that when a crowd was gathered around the system, it attracted more attention, and more people were likely to attend the crowd and thereby the system, also referred to as the Honey-pot Effect. The authors of Brignull et. al. [10] initially used their own system and observed that this created a momentum effect, where more people approached the system. Peltonen et. al. [8] also made a similar observation with CityWall, where people were standing with their backs to the large touch screen and waited for the rain to stop. A boy walked by the screen and touched the screen, after which he uttered: ”Oooh”, thereby getting the attention of his friends and the people around the screen, which then noticed the screen. Users of CityWall also claimed in the interviews that the system was hard to notice, when nobody was using it, but when they started interaction with the screen, they could see that it attracted a lot of attention from passers-by.

Approaching the system

Generally people approached the system by first observing the system and users from a distance followed by walking up to the interface to interact. People tended to come in pairs or larger groups when approaching the system, only 15% of the first-time users approached the system alone, when the system was not in use. This result is similar to the observations made by Peltonen et. al. [8], where only 18% of the users approached the system alone. Like Peltonen et. al. [8] we have not found any explanation why people seldom approached the system individually.

Transitions between users

Transitions between users refers to how users were able to determine who was next in line to use the system. When people approached the system, they placed themselves around the screen instead of a normal queue so they could watch the interaction. Even though there was no explicit queue, people were in general able to determine when it was their turn without any conflicts. At one occasion, we observed a conflict between a male and a female that were standing equally close to the system when user completed a purchase, Figure 7A. After the user left, the male took a step towards the screen, Figure 7B, after which the female stated: ”It was not your turn” and pushed the male away to claim the control of the system, Figure 7C.

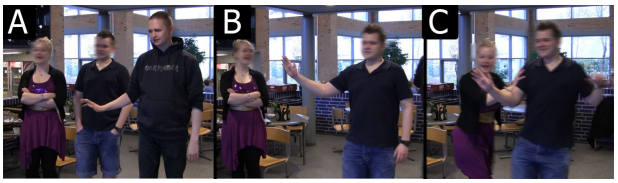

Figure 7. Screenshot of conflict in transition between users

Social inhibition

Social inhibition refers to the embarrassment of using such a system in public. The system requires larger movements and gestures that can be spotted by others, and this can have a negative effect on people’s willingness to interact.

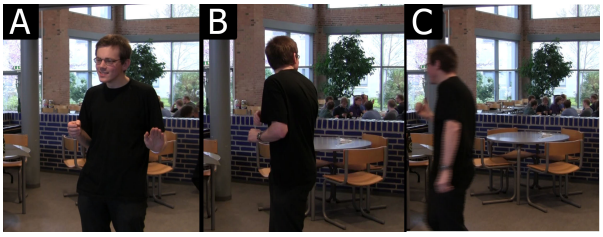

Figure 8. Screenshot of social inhibition

Figure 8A shows an example of a user using the system a couple of minutes until three other people arrived. The user noticed the arrival of the others in Figure 8B. The user stopped the interaction right away and left as shown in Figure 8C. This suggests that the user felt uncomfortable interacting with the system while others were present. Perry et. al. [9] observed another example of social inhibition where a mother and teenage daughter did not feel comfortable using the mid-air gesture-based interface. They approached the mid-air gesture-based interface and watched the training video. After the completion of the video, they both tried to encourage each other to interact with the interface: ”Go on mum, you try it” Both of them rejected and exited the location.

Learning the system

Learning the system refers to how people learned how to perform the gestures and control the cursor.

Skipping training

When nobody was using the system we switched the system to training mode, where the interaction was explained and demonstrated. However we noticed during analysis of the video that almost none of the users (85% of all first-time

users) made use of the training mode presented when they approached the system. The first thing people did when beginning the interaction was to move the cursor down towards the ”Finish Training”-button and make movements to perform the ”Click”-gesture. Instead of following the instructions on the screen, the main part of the users used the trial-and-error approach. This indicates that when developing a walk-up-and-use system, people should be able to interact with the system without first having to complete a training session. A possible solution could be to introduce a more intelligent training system which gives the users hints on how to perform a given gesture when the users try to perform it. If a user tries to perform a gesture but fails, the system could e.g. demonstrate how to perform the gesture that was similar to the user's movement. By doing so, the user would only receive training when needed and only on the gestures the user found difficult to perform.

We observed an three ways for learning how to interact: Observe-then-act, Collaboration and Competition and Teacher/Apprentice.

Observe-then-act

Many of the users started with watching others interact with the system before trying it themselves by imitating, how the previous person used the system. Figure 9A shows a person in a purple shirt standing behind and observing another user interacting with the system. The observing person then waited for the other user to leave, before he approached the system and started to interact shown in Figure 9B.

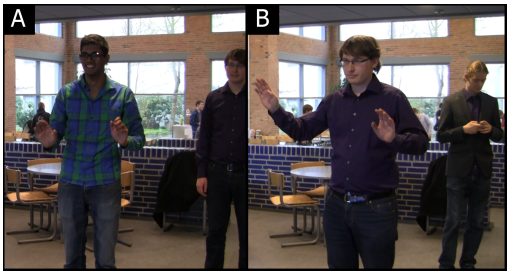

Figure 9. Screenshot of observe-then-act

Likewise, Hinrichs et. al. [2] made similar observations of imitation from users of their multi-touch tabletop. They observed that an adult visitor using both hands to herd as many items as possible to his corner of the screen. A little girl observed this and started to imitate his gesture immediately.

Collaboration and competition

We observed several examples of group members encouraging each other to interact. People that did not directly interact with the system, were involved by making suggestion on how to interact, even though they did not have any experience with system themselves. As an example, two males had approached the system, one using the system, while the other was observing. The male using the system was having problems with performing a ”Click”-gesture”, and the observer suggested: ”I think you should push your hand out in a more straight line”, where after the user was successful in making the gesture. Without the additional input from the observer, the user might have given up interacting with the system. This was observed with people that left the system, when having problems with the interaction instead of reading the instructions, but with the above example the user was given input that helped and encouraged him to keep interacting. Other observers were more direct in their help, such as: ”I think, you should try the training”.

We also observed the interaction between group members in more competitive way. In one occasion an observer was telling the current user how to perform the ”Click”-gesture while the user had trouble, which might seem helpful but the conservation between the two was in a competitive language. When the current user was finished interacting the observer placed him self in front of the screen confident in order to show how to perform the gesture. Another example of this competitive behavior was that after watching one group member having trouble buying an item, another group member takes over and makes a purchase. After successfully buying beverages he yells at the other group member: ”How hard can it be?”.

Teacher/Apprentice

Other studies have showed that people tend to teach each other how to interact with a new system when placed in public [11, 3]. Similar situations were observed during the analysis of the video material. Often two or more users helped each other to learn how to perform the gestures and control the cursor, one as the teacher and another as the apprentice, where the teacher had tried the system before.

Figure 10. Screenshot of teacher/apprentice

Figure 10 shows an occasion, where a new user (apprentice) approached the system while another user (teacher) was present. The apprentice walked up to the screen and held both hands in front of her. The teacher told her that the cursor should be controlled with the right hand and that the gesture should be performed with the left hand. Afterwards the teacher demonstrated how to hold her hand in a comfortable position to control the cursor.

Playing around

As the users were able to see a live video feed of themselves while being in training mode, some of them found it entertaining to perform a dance or in other ways play around in front of the system. Figure 11 shows two different users playing with the system.

Figure 11. Screenshot of users playing around

Other found it amusing just to move the cursor around and perform gestures without actually purchasing anything. Generally this type of interaction seemed to entertain first time users.

Teasing

The users often approached the system in small groups of 2-3 people. When one of the group members began to interact with the system, the others tended to tease the current user. During the training, the users were able to see a live feed of themselves which e.g. lead to others making ”bunny ears” above the head of the current user. During the actual use of the system the teasing of the current user was a bit different. In many occasions the group members, which were not in control, were verbally teasing the current user, as ”Do you not know how to make a forward punch?”, ”It is easy, come on” and cheered, when the user made a correct gesture. In one occasion the current user had difficulties performing the ”Click”-gesture correctly, which entertained the others from the group. Others went in front of the current user for no other reason than just to tease by gaining control of the system. Since the user located closest to the sensor was the user in control it was an easy way to disturb the current user. It must be noted that this only occurred with people, who approached the system together, or when it was clear that the persons knew each other.

Victorious

When analyzing the video material from the field study, it was clear that a part of the users was celebrating when they successfully bought an item using the system, e.g. one user high-fived an observer when finished buying an item, Figure 12.

Figure 12. Screenshot of users celebrating

The fact that some of the users were celebrating suggests that when using the mid-air gesture-based interaction, the task of buying an item becomes more like a game to the users than when using a standard input device, even though there was no score of their interaction presented other than they were able to make a purchase.

Performance and user experience

The following subsection is primarily based on the questionnaires from the users in the field study. The general impression of the system was positive. 70% of the users gave the system a rating above 6 on a scale from 1-10.

Gestures

In general, the users found the gestures intuitive, easy to learn and easy to remember. 66% rated the intuitiveness of the gestures from 7-10. 74% of the users rated the learnability of the gestures from 7-10 and 87% of the users gave the gestures a rating from 7-10, concerning how easy the gesture were to remember. Even though the gestures were intuitive, easy to learn and easy to remember, some users found it difficult to perform the gestures. 60% of the users rated the difficulty of performing the gestures from 5-10. This indicates that the choice of gestures were acceptable, but the system should be able recognize less precise gestures. It must be noted that almost all of the users were first time users. The analysis of the video material showed that some users tended to increase the speed of their movements when performing a gesture that was not recognized by the system. One user failed at performing the ”Click”-gesture a couple of times, and this made him perform the gesture numerous times at very high speed which the system was unable to recognize. A possible solution to this problem could be to present the user with a hint telling him/her to slow down the movement of the arm when needed. Hardy et. al. [1] also experienced using a mid-air gesture-based interface that when users encountered problems, they performed gestures faster and more erratic instead of slowing down.

The ”Click”-gesture

Some of the users found it difficult to hold the cursor still while performing the ”Click”-gesture. The ”Click”-gesture does not require the user to fully stretch the left arm towards the screen, however several of the users did so. When fully stretching the left arm, it increases the difficulty of holding the right hand still resulting in the users clicking next to the desired element. The results from the questionnaires showed that several users would prefer the ability to perform a functionality of a mouse click by snapping the fingers or closing the hand.

Cursor techniques

The questionnaires showed that the users found the cursor relatively easy to control, 92% gave the control of the cursor a rating above 5. However one user stated that the links and buttons on the web page were to hard to hit and the whole web page should be scaled up. This could potentially decrease the difficulty of hitting the links and buttons.

Application domain

The users were asked to give their opinion about where such an interaction form could be used. All suggestions can be divided into three categories:

- Places, where people get dirty hands.
- Places, where people must not get dirty hands.
- Places, where people are unable or not allowed to touch anything.

The following examples are based on conversations and the questionnaires from the users.

When working at places like kitchens and workshops, people often get dirty hands. The mid-air gesture-based interaction could be used while having dirty hands enabling a cook to browse through recipes online while cooking or a mechanic to browse technical documentation while having oil on his/her hands. At operation rooms the surgeon must stay sterile while performing operations. With mid-air gesture-based interaction the surgeon could browse through X-rays while operating without requiring an assistant to do so. It could be possible to use this interaction form in shops allowing people to interact with systems that are placed behind glass. People could then e.g. browse through real estates, pricing list and find tourist attractions.

DISCUSSION AND CONCLUSION

We have placed a mid-air gesture-based interface at a social event at a university and studied how people interacted with such interface. By analyzing the collected data we observed several interesting situations. A positive effect of the interaction with the system being visible to others often attracted more users. A negative effect was that some people were anxious about interacting with the system while other were observing. Instead of finishing what they were doing, they aborted their interaction and left, when other approached the system.

Even though the system had a training mode, where new users could learn the interaction, we discovered that this training mode was seldom used. New users often observed other users before trying themselves, thereby acquiring a basic idea of how to interact. On other occasions, another user taught the new user how to move the cursor as well as how to perform the different gestures.

We also experienced that some users found the type of interaction entertaining. Some users were dancing in front of the screen, while others were just interacting with the system without actually buying anything. A part of the users that was using the system to buy beverages, even celebrated a successful purchase. We also observed that people in groups were competitive towards each other. This suggests that the trivial task of buying beverages became more interesting and similar to a game compared to buying beverages with a keyboard and computer mouse, even though there was no indication of performance.

In general the users responded positively to the mid-air gesture-based interaction. Overall the users were satisfied with, how the cursor was controlled and how the gestures were designed, even though some users found it difficult to perform the ”Click”-gesture and suggested alternative ways of performing a click.

The mid-air gesture-based interface seemed to fascinate people which caused them to come forward to observe and participate, while some users seemed anxious for interacting with such system in public.

The users made different suggestions, where such mid-air gesture-based interaction could be used, and these suggestions comply with very recent work in different areas. For the categoy of getting dirty hands Panger [7] is working on an article (Work-In-Progress) that focuses on using a mid-air gesture-based interface in kitchens to follow recipes while cooking. For the other category for sterile environment and not getting dirty hands, King’s College London [5] has recently announced that they are working on applying the Kinect technology in operation rooms.

The analysis of the collected data might have been influenced by the subjective opinions from the authors. The fact that the study was conducted during a social event where alcoholic drinks were sold, can have had an influence on the results.

ACKNOWLEDGMENTS

A special thanks to the people that participated in the study.

REFERENCES

1. Hardy, J., Rukzio, E., and Davies, N. Real world responses to interactive gesture based public displays. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, MUM ’11, ACM (New York, NY, USA, 2011), 33–39.
2. Hinrichs, U., and Carpendale, S. Gestures in the wild: studying multi-touch gesture sequences on interactive tabletop exhibits. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI ’11, ACM (New York, NY, USA, 2011), 3023–3032.
3. Jacucci, G., Morrison, A., Richard, G. T., Kleimola, J., Peltonen, P., Parisi, L., and Laitinen, T. Worlds of information: designing for engagement at a public multi-touch display. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI ’10, ACM (New York, NY, USA, 2010), 2267–2276.
4. Kinect, M. Introduction to Kinect. `http://www.xbox.com/da-DK/Kinect/GetStarted`. [Online; accessed 28-May-2012].
5. London, K. C. Pioneering touchless technology. `http://www.kcl.ac.uk/newsevents/news/newsrecords/2012/05May/Pioneering-touchless-technology-.aspx`, 2012. [Online; accessed 04-June-2012].
6. Nintendo. Controls for Wii. `http://www.nintendo.com/wii/what-is-wii/#/controls`. [Online; accessed 28-May-2012].

7. Panger, G. Kinect in the kitchen: testing depth camera interactions in practical home environments. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, CHI EA '12, ACM (New York, NY, USA, 2012), 1985–1990.

8. Peltonen, P., Kurvinen, E., Salovaara, A., Jacucci, G., Ilmonen, T., Evans, J., Oulasvirta, A., and Saarikko, P. It's mine, don't touch!: interactions at a large multi-touch display in a city centre. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, ACM (New York, NY, USA, 2008), 1285–1294.

9. Perry, M., Beckett, S., O'Hara, K., and Subramanian, S. Wavewindow: public, performative gestural interaction. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, ACM (New York, NY, USA, 2010), 109–112.

10. Rogers, H. B. . Y. Enticing People to Interact with Large Public Displays in Public Spaces. **http://www.shengdongzhao.com/courses/wp-content/uploads/2012/03/Enticing.People.to_.Interact.with_.Large_.Public.Displays.in_.Public.Spaces.pdf**, 2003. [Online; accessed 25-May-2012].

11. Rubegni, E., Memarovic, N., and Langheinrich, M. Talking to strangers: Using large public displays to facilitate social interaction. In *14th International Conference on Human-Computer Interaction (HCII 2011)*, Springer, Springer (7 2011).

12. Sony. This Is How I Move. **http://us.playstation.com/ps3/playstation-move**. [Online; accessed 28-May-2012].