New Interaction Techniques
Spring 2001

Index

New Interaction Techniques: Overview
Roope Raisamo 1

Interaction Technique papers

Driving Experience in the 3D Environments:
The Use of Car-like Controls in Virtual Reality
Antti Kokkonen 2

Special Character Text Input in Mobile Devices
Niku Yliluoma 7

Hand Gestures Used In Manipulation System
Mahmoud Haj-Rashid and Abridzak H. Mohamed 13

Future User Interface for Wearable Computer
Veli Ijäs 19

Simple Operation Email Composing System without Keyboard and Mouse:
With Multi-modality Input Methods
June Miyazaki 24

Conducting a Wizard of Oz test on Ubiquitous Computing system Ovimies
Kaj Mäkelä and Esa-Pekka Salonen 32

Space Alignment Tool: A New Manipulation Tool for Graph Edititing
Jukka Raisamo and Tommi Järvinen 38

Combined Voluntary Gaze Direction and Facial Muscle Activity as
a New Hands-free Technique for Human-Computer Interaction
Marko Illi, Poika Isokoski, and Veikko Surakka 47

Using a Mouse With Tactile Feedback
Elina Haapanen 53

Technotes

Foot as a Supportive Input Technique for Navigation Tasks
Jiri Hlusi 57

Autocomplete in Text Input – Comparison between Finnish and English
Juha Jääskeläinen and Sami Pekkola 59

New Interaction Technique for Flexible Text Copying
Teemu Tanila 61

Wearable E-Mail Client
Mikael Rinnetmäki 63

New Interaction Techniques: Overview

Roope Raisamo
University of Tampere
Department of Computer and Information Sciences
Tampere Unit for Computer-Human Interaction (TAUCHI)
Pinninkatu 53B
FIN-33014 TAMPERE, FINLAND
E-mail: rr@cs.uta.fi

The course “New Interaction Techniques” [1] was organized for the second time in the spring of 2001 at the Department of Computer and Information Sciences in the University of Tampere. The course is one of the advanced courses on human-computer interaction by Tampere Unit for Computer-Human Interaction (TAUCHI). The present report collects the research and papers that were completed in this course.

The course was designed so that in the beginning of the spring semester we had a short period of lectures. The lectures introduced the concept of interaction technique, multimodal human-computer interaction, and some novel input technologies, namely speech technologies, two-handed input, haptic interaction, VR Techniques, eye tracking, text input methods, and wearable computing. The purpose of the lectures was to clarify what was to be done in the practical phase of the course, and to show some potential uses for input technologies.

This time it was possible to pass the course by writing a research paper, or by developing a new interaction technique and reporting on it in 2-page ACM UIST [2] technote style. The papers were also formatted following the UIST guidelines [3]. The aim was that we could submit the best papers in the UIST conference that had the submission deadline conveniently at the same time as our spring semester ended. The reason for writing all the other papers in the same format was to get a professional course report that would be distributed to all the students and other interested people.

Paper categories in the course were the following:

- *interaction technique papers* that either introduced a novel interaction technique or reported on an empirical study that was related to interaction techniques; and
- *technote papers* that explain briefly an interaction technique that was developed in a programming project.

The papers have been grouped in these categories in this publication.

In total, about 20 students participated in the lectures. Of them, 16 finished the course. Several papers were submitted to UIST or other relevant conferences.

ACKNOWLEDGMENTS
I’d like to thank the researchers in the TAUCHI unit and other groups who volunteered to give a visiting lecture on their research topic in this course. Aulikki Hyrskykari, Markku Turunen, Poika Isokoski, Veikko Surakka, Timo Partala, Ismo Rakkolainen, Marja Salmimaa, and Toni Pakkanen did a great job in helping to achieve the goals of the course. Hopefully all the acknowledged people also benefited from the student papers that were written on the paper topics that they proposed and were willing to guide.

REFERENCES

1. New Interaction Techniques course home page, http://www.cs.uta.fi/interact/.
2. ACM UIST Conference pages, http://www.acm.org/uist/.
3. Doe, J. and Dupond, M., Standard UIST’96 Conference Format: Preparing Submissions for Papers. http://www.acm.org/uist/sample.rtf

Driving Experience in the 3D Environments: The Use of Car-like Controls in Virtual Reality

Antti Kokkonen
University of Tampere
Department of Computer Science
Tampere, Finland
+358 40 71 383
antti.kokkonen@uta.fi

ABSTRACT

In this paper, we present new ways to interact with a computer similar to driving a car and compare these techniques to the standard ways to control the computer like mouse and keyboard. We think that controlling the computer with both hands and feet in the same way that one controls a car offers new possibilities especially for three dimensional (3D) World Wide Web (WWW) navigation. The car metaphor methods studied in this paper include using devices such as a steering wheel, foot pedals and transmission stick (joystick). We suggest that the use of driving-like interaction as a control method to move through 3D worlds is a promising alternative compared with the currently used ones and offer excellent prospects for the future. To prove this we studied if moving around Virtual Reality (VR) could be made more efficient and more intuitive to the user using the driving-like interaction technique.

Keywords

Multimodal Human-Computer interaction, Car metaphor, Steering wheel, Pedals, Joystick, Three dimensional (3D) navigation, World Wide Web (WWW), Virtual Reality (VR), Virtual Reality Modeling Language (VRML)

INTRODUCTION

The World Wide Web (WWW) has become the entertainment and information center of the World today. The increasing use of media elements like video and audio over traditional text and images in WebPages has been the trend during the last few years. One of the most recent additions to the Web has been the coming of the three dimensional (3D) WWW-pages using Virtual Reality Modeling Language (VRML).[1]

If the visions of the future development come true, three-dimensional WWW could take the user to places that would otherwise be impossible for one to visit. These places could be anywhere in the World or even historical location, for example ancient Rome, created only in the Virtual Environment (VE). To make the dreams come true the ongoing development of the visual technology demands the necessary up-to-date hardware and also new ways to interact with the system.

The common and currently the most used control methods mentioned earlier are efficient and familiar to the user. It has been shown that new techniques require learning and the user to get used to different habits. However the use of a mouse and keyboard forces the designers and users to concentrate heavily on the graphical user interface (GUI). Because of this, we suggest that the use of the car metaphor as a control method for three-dimensional WWW-navigation described in this paper is a promising alternative to the old ones and offer excellent prospects for future. Moving through the VE is mostly turning to left or right and moving forwards and backwards (and possibly up and down). As this is can be seen as "driving" in a 3D world in a way very much like real world (RL) car driving experience, the car metaphor seems to offer a more intuitive way to control the movement.

To make the most out of the car-like use of the computer one would need "a dashboard" which would include a steering wheel, foot pedals and a transmission stick (Joystick in this case). The use of the devices in the dashboard lets the user concentrate on the information instead of trying to figure how the GUI works. More advanced dashboard could also include buttons and switches to give more controls to the user to enhance the virtual experience.

The interfaces of popular 3D games, such as Quake or Half-Life, give some aspect to the designing of the interaction methods. Many computer users move through 3D worlds on a daily basis and have attained a certain savvy for navigating with simple keypad commands and mouse gestures. The challenge is to make the basic functionality needed to navigate and view in 3D accessible and usable to web users also with little 3D user interface experience.

PREVIOUS WORK

The virtual environments demand a lot from both the system and the user. Rendering of the Virtual World requires a lot from the machinery. When the idea is to make the computer disappear and give an impression of a "real" VR the information flow will burden all the senses of the user. The Full VR Experience including Head-Mounted Displays (HMD), touch sensations, etc. referred to here has been critiqued for being unsuitable for day-to-day use for the possibility of shutting the user off from social life. The VRML pages can provide a more gentle approach to the average user and also to the hardware.[2,8]

The paradigm for a usual 3D-navigation tool is that of a user "walking" around a virtual world. The walk paradigm constrains the up direction of the scene and the height of the camera above a plane perpendicular to that direction. The primary functions needed for the walk paradigm are:

- the ability to move forward and backward
- the ability to turn left and right
- the ability to tilt the camera up and down.
-

A VRML browser called WebSpace presented a graphical dashboard across the bottom of the viewing window, to provide access to the most important navigation tools. [3]
The comparisons of hand and foot have showen that the movement times of the foot are approximately twice to those of the hand. The previous human-computer studies of the use of foot as an input method are somewhat irrelevant to the car metaphor as the studied foot devices were meant for a pointing tasks usually done with a mouse. [5,6,7] As the pedals in our Dashboard would be used as one would drive a car, studies involving real life driving proved to be more interesting. [9,10,11]

THE INTERACTION TECHNIQUE

The purpose of this paper is to introduce new interaction technique to control computer displaying the 3D environment. The basic idea is to make the movement in a VR easy and natural to the user. This would be done by giving the user the feel of driving by using a car metaphor as the basis for the interaction. To make the system work one would need software to receive the user input from various devices and then to transport the commands to the program, for example a VRML browser which would show the actual virtual world. Implementation of the software (and therefore the whole interaction system) will not be done for this paper. However we pay attention to the possible uses of the existing devices to drive the computer as easy as a car.

The Car Metaphor

As walking through the 3D world could also be thought as driving, the use of car metaphor seems logical choice to control the movement. The user would move around in the virtual world like one would drive a car in the real world. As the basic functions would be moving forward and backward and turning left and right, car interface with steering wheel and pedals would be a natural choice to control the computer for this purpose. All these devices meant to control the virtual car make up a control panel which we call *Dashboard.*

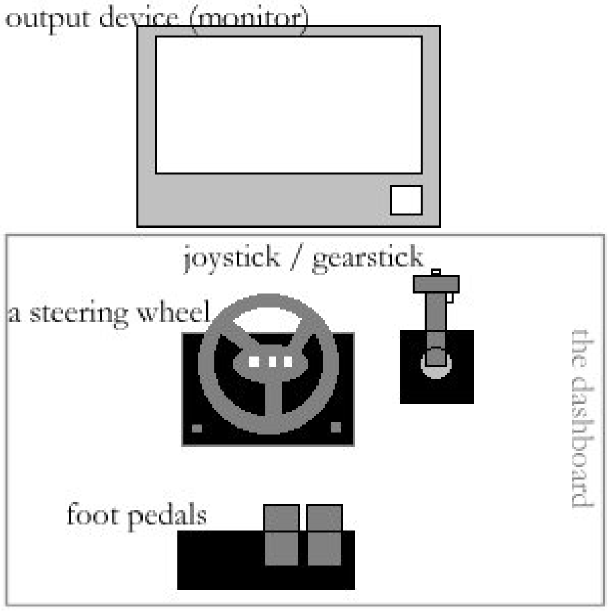


Figure 1. Dashboard.

The Dashboard

Dashboard is a name which we use when talking about all the devices used to control the computer in our interaction technique. Dashboard has the devices needed to control the "car". Steering wheel for turning the car to left and right and pedals to move the car. The system could also have a gearstick to shift to reverse and back to the forward motion. While a joystick would be used as a "gearstick", other functions could also be attached to it, like changing the camera view up and down. Buttons of the joystick (and also the buttons on steering wheel if such exist) could be used when clicking is needed to make choices.

The dashboard could also have different kinds of switches, which would be used to control the blinkers or the window wipers of the car in real environment. These extra devices could have various different functions when controlling the virtual car. These extra controllers and buttons would enable more ways to interact with the environment than just moving around.

If one would like to take the most out of the car-like control method, the dashboard system could also include some or all the parts found inside a real car, like light and ventilation switches or even a cup holder. Of course these extra options could have other functions than the real ones.

Various instruments could also be used to show information about the system to the user, perhaps a gas level would not be as essential as in the real world, but some warning lights and possibly a speedometer could be useful.

Use of the Devices

The devices of the system are similar to the ones found in a normal car, only built for computer use. Steering wheel, pedals and joystick are common hardware mostly used to control the system when playing computer games. The idea is to use the hardware already on production and available to reduce the cost and increase the availability of the necessary equipment.

Steering Wheel

The steering wheel would be used to control the right and left turning in the system. As some steering wheel products have buttons attached these could also be used to other functions like selecting or otherwise interact with the system. Using the wheel is easy to learn and the hand movement needed for this turning motion is natural to us, as the use of vehicles in the world shows. We believe that turning to left and right in the virtual world by turning the wheel to the same direction would be intuitive to any user no matter the age or the experience.

If the wheel has behind the wheel switches, sometimes called formula-like gears, these could also be taken to use. One possibility would be to change the direction of the view without changing the direction of moving. As one would have hands on the wheel most of the time while using the system, taking the most out of the possibilities of the wheel seems like a good choice. To back-up our idea of emphasizing the use of wheel it has been shown that wheel-attached audio device controls in real cars increase the safety of driving [13]. Like car manufacturers want the drivers to keep the attention on the road, we think that the user should have the opportunity to focus mostly on the information and not to heavily concentrate on the input and the devices.

Foot Pedals

The pedals in the system would be used to control the movement like one uses the gas and break pedals in a real car. Other functions could also be used, but the physical limitations of foot put some limitations on the possibilities [6]. As using hands for more precise actions is more effective, keeping the foot movement to the minimum is a better choice for us [5]. Using feet to press and release the pedals to move forward (and backward) or possibly making simple click choices when needed are the ways to use the pedals in our system. For a bit different approach than the car-model one pedal could be used for forward and the other for backward motion if we would like to put the gear changer to other use. This kind of use of pedals could even be better choice than the car-like approach, because probably one would not want to shift gears while exploring the virtual world (if the idea would not be to actually simulate driving).

Joystick

To simulate the gear change of the real gar we would use joystick in this system. One perhaps even better possibility would be that to use the gearstick attached to the wheel device if such exists. If the pedals would be used for both forward and backward movement, the gearstick could have other functions like tilting the view up and down. Joystick would be placed near the wheel to limit the necessary hand movement to the minimum.

The buttons of the joystick offer a lot of opportunities for the user to interact with the system. However as not all joysticks have lot of buttons, the functions attached to the buttons and switches could be left freely configurable for the user. As most joysticks have at least two buttons, some options should be set as default, for example button number 1 would be used for select/ choice in the program.

Other devices

Other devices like switches and buttons could have various different functions in the system. Still we leave these out of the initial system, because the use of these devices would possibly need some unnecessary hardware, which could have to be built for this purpose. The keyboard and mouse are probably present in the system if no other alternative control methods would be used (like speech recognition for example). And the buttons of the steering wheel and joystick already offer enough controls for the system.

Giving feedback to the user

Using force feedback and sounds with the devices it is possible to enhance the driving-like experience of the user even further. Although the typical computer user gets the most of the essential information through the visual system, usually from a monitor. If we assume that all the users don’t have the opportunity to use get other feedback than the basic computer graphics displayed on screen, we must concentrate on the sense of sight and keep the use of senses in background at this point. But as combining information from different sensory streams has been shown to facilitate the processing of that information, we must study the use of other senses in the system as well.

Sounds in the system

Sounds can be used to inform the user of activities perhaps invisible to the eye. If a object in virtual world can be manipulated, system could tell that with simple sound in addition to the visual output. As we want to give the user experience of driving, sounds could be used to generate one kind of engine sounds to express the speed to the user.

Force feedback interaction

Force feedback has been successfully used in game industry for a while now. Force feedback provides another sense that can be used for multimodal interaction in addition to hearing and sight. Haptic interaction is one of the most fundamental ways in which people perceive and effect changes in the real world around them. Therefore the haptic system might also be of great use when interacting with computers, especially in a virtual world. It is possible that the integration of touch input and output to human computer 3D interfaces is one of the solutions to the problem of developing the best multimodal interfaces.

In the car metaphor force feedback could be used to inform the user about moving, collisions, obstacles, etc.. Also the speed of moving, object manipulation and other changes in the virtual world could affect the feel gotten from the devices. In general force feedback would be used to make the driving-like experience more real to the user.

EVALUATION

As the software needed to test the system wasn't implemented for the paper no evaluation has been done to test the interaction technique. If the implementation would be done in the future, The new car-like interaction technique should be tested against at least the traditional mouse interface of the WWW/ VRML browser as well as the control method using mouse and keyboard familiar from popular 3D games such as Quake and Half-Life.

DISCUSSION

As real-like controls are more and more used for driving games and other simulations we believe that the technique could prove to be more efficient and more intuitive way to control the movement in VR. One could criticize the need of all the hardware, but we want to emphasize the ease of use of the devices and the user's change to control the system intuitively. As this novel interaction technique has very little drawbacks we came up with, the implementation of the system in the future seems like a good idea.

CONCLUSION

We think that controlling the computer with both hands and feet in a same way that one controls a car offers new possibilities especially for three dimensional World Wide Web navigation. The use of car-like controls to provide the basic functionality needed to navigate and view the virtual worlds make the possibilities accessible and usable to web users also with little 3D user interface experience. We believe that the use the car metaphor we presented in this paper as a control method to move through 3D worlds is promising alternative for the currently used ones and offer an alternative choice to the future.

ACKNOWLEDGMENTS

We would like to thank all the people who provided helpful comments on previous versions of this paper concerning the technical aspects and also the language.

REFERENCES AND FURTHER READINGS

1. Virtual Reality Modeling Language (VRML) Specifications. <http://www.vrml.org/Specifications/> (22nd March 2001)
2. Future Multimedia User Interfaces. MacIntyre, B., Feiner, S.. Multimedia Systems, 1996. <http://www.cs.columbia.edu/graphics/publications/mmSystems.pdf > (22nd March 2001)
3. Mohageg M., Myers, R., Marrin, C., Kent, J., Mott, D., and Isaacs P.. A User Interface for Accessing 3D Content on the World Wide Web. Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems, DESIGN BRIEFINGS: Design for Communication and Communication for Design, Vol. 1, pp. 466-472, 1996. <http://www.acm.org/sigchi/chi96/proceedings/desbrief/Mohageg/mfm_txt.htm> (22nd March 2001) [VR / WWW]
4. Darken, R. P., Sibert, J. L.. Wayfinding Strategies and Behaviors in Large Virtual Worlds. Proceedings of ACM CHI'96 Conf. on Human Factors in Computing Systems. <http://www.seas.gwu.edu/faculty/sibert/darken/publications/Strategies_CHI96/strategies.html> [VR]
5. Hoffman, E. R.. A comparison of hand and foot movement times. Ergonomics, 1991. Vol. 34, no. 4. 397-406.
6. Pearson, G., Weiser, M.. Exploratory Evaluation of a Planar Foot-Operated Cursor-Positioning Device Input Devices. Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems 1988 p.13-18 [POINTING WITH FEET]
7. Pearson, G., Weiser, M.. Exploratory Evaluations of Two Versions of a Foot-Operated Cursor-Positioning Device in a Target-Selection Task. CHI+GI'87 Poster, Session Papers and Abstracts. ACM SIGCHI Bulletin 1988 v.19 n.3 p.70-75 [POINTING WITH FEET]
8. Weiser, M. The World is Not a Desktop. Perspectives article for ACM Interactions, Nov. 7th 1993 <http://www.ubiq.com/hypertext/weiser/ACMInteractions2.html> (22nd March 2001) [VR]
9. Brackett Jr., R. Q., Koppa, R. J.. Preliminary Study of Brake Pedal Location Accuracy Safety: Driver Behavior. Proceedings of the Human Factors Society 32nd Annual Meeting 1988 v.1 p.976-980 [PEDALS]
10. Koppa, R. J.. Human Factors (Chater 3, Revised Monograph on Traffic Flow Theory). <http://www.tfhrc.gov/its/tft/chap3.pdf> (22nd March 2001) [REAL LIFE DRIVING]
11. Brackett, R. Q, Pezoldt, V. J., Roush, L.. Initial Driver Foot Placement as an Antecedent to Pedal Error SAFETY: Transportation Systems Safety. Proceedings of the Human Factors Society 36th Annual Meeting 1992 v.2 p.970-974 [PEDALS]
12. Kay, J. S.. STRIPE: Remote Driving Using Limited Image Data. Doctoral Consortium. Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems 1995 v.2 p.59-60. <http://www.acm.org/sigchi/chi95/Electronic/documnts/doctoral/jsk_bdy.htm> <http://www.cs.cmu.edu/afs/cs/project/alv/member/www/projects/STRIPE.html> (26th April 2001) [STEERING WHEEL]
13. Farber, E., Salata, S.. Comparing Ease-of-Use of Steering Wheel-Mounted and Conventional Radio

Controls General Sessions: Human Factors Inside the Automobile. Proceedings of the Human Factors Society 32nd Annual Meeting 1988 v.1 p.578-582 [STEERING WHEEL CONTROLS]

14. Zwahlen, H. T.. Evaluation of Pushbutton Arrangements in Automobiles SAFETY: Contemporary Safety Issues in Driving. Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting 1993 v.2 p.969-973 [STEERING WHEEL CONTROLS]
15. Risden, K., Czerwinski, M., Worley, S., Hamilton, L., Kubiniec, J., Hoffman, H., Mickel, N., Loftus, E.. Interactive Advertising: Patterns of Use and Effectiveness Persuasion. Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems 1998 v.1 p.219-224 [USE OF CAR METAPHOR]
16. Ramstein, C., Arcand, J.-F., Deveault, M.. Adaptive User Interfaces with Force Feedback VIDEOS: New Directions. Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems 1996 v.2 p.406-407 [FORCE FEEDBACK]
17. Ramstein, C.. An Architecture Model for Multimodal Interfaces with Force Feedback I.14 Virtual Reality 2. Proceedings of the Sixth. International Conference on Human-Computer Interaction 1995 v.I. Human and Future Computing p.455-460 [FORCE FEEDBACK]
18. Rosenberg, L. Ph.D., Brave, S.. Using Force Feedback to Enhance Human Performance in Graphical User Interfaces. Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems, SHORT PAPERS: Thought Pieces for Interaction Technology, Vol. 2, pp. 291-292, 1996. <http://www.acm.org/sigchi/chi96/proceedings/shortpap/Rosenberg/rl_txt.htm> (22nd March 2001) [FORCE FEEDBACK]
19. Kerk, C. J., Chaffin D. B.. Evaluation of Limiting Strength Constraints in a Comprehensive Biomechanical Model INDUSTRIAL ERGONOMICS: Strength. Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting 1993 v.2 p.739-743 [ERGONOMICS]
20. Shehab, R. L., Schlegel, R. E.. Test of Fitts' Law in a Dual-Task Paradigm GENERAL SESSIONS: Surface Transportation / . Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting 1993 v.1 p.559-563 [STEERING WHEEL]
21. Sellen, A. J., Kurtenbach, G. P., Buxton, W. A. S.. The Role of Visual and Kinesthetic Feedback in the Prevention of Mode Errors Interactive Technologies and Techniques: Input. Proceedings of IFIP INTERACT'90: Human-Computer Interaction 1990 p.667-673 [PEDALS]
22. McGee, M. R.. The Effective Combination of Haptic and Auditory Textural Information. Haptic Workshop at the University of Glasgow, 2000. <http://www.dcs.gla.ac.uk/~mcgeemr/workshop.doc> (9th May 2001) [HAPTIC INTERACTION]

Special Character Text Input in Mobile Devices

Niku Yliluoma
TietoEnator Resource Management Support
Kalkkipellontie 6
02601, Espoo, Finland
+358 400 904 528
niku.yliluoma@tietoenator.com

ABSTRACT
The mobile devices should provide alternatives for the users that need international characters in their writing. In this paper the most common techniques and ways of entering mobile text are presented, and some ways of easing the issue of writing Scandinavian letters in mobile devices are suggested.

The two most common text input techniques, handwriting recognition and soft keyboard are studied and presented to make the suggestions more easily understood. One of the most commonly used PDAs, 3Com's Palm™, is used as an example of how the text input is done today.

Keywords
Soft keyboard, handwriting, text input, stylus input, Personal Digital Assistant, PDA, handheld computers, portable devices, mobile systems

INTRODUCTION
A PalmV™ user often faces troubles when typing Scandinavian letters. Neither the handwriting recognition nor the soft keyboard provides an easy way of writing Finnish, which contains surprisingly many Scandinavian letters. Writing gets quite slow and irritating, when error rate becomes high.

The market of small computers is rapidly growing and some visionaries see Personal Digital Assistants (PDAs) as the most important thing in which to invest development resources in the near future. Text input is crucial to the success of all portable devices.

There are two common options for text input: handwriting recognition and soft keyboard. A technique that could serve all users in their mobile situations as well as the all-familiar QWERTY keyboard attached to a computer, will most likely be used in most of the mobile devices in the future.

If text entry is made as easy as possible also when it comes to international characters, such as Scandinavian and German special letters, the usability of the portable devices will leap onto a higher level – and the products will be more succesful in these countries.

If the issue of text input in mobile devices were researched more widely, there would be a need to study the situation on special characters from all Nordic languages and German. The research could even go as far as integrating text input between western and the more awkward languages, such as Japanese and Chinese. In this paper, the issue is presented only from the narrow perspective of a Finnish user.

RELATED WORK
In addition to the two leading ways of text input in PDAs, some very innovative text input technologies, such as scrolling, non-keyboard input [1] and text input with joysticks [4] are arising. These are left for further studies, and the paper concentrates on the most common text input techniques. The same applies to the so called 'multi tap' method, which is commonly used in mobile phones, and has been proven to be less efficient than the predictive text input/word completion techniques [11].

BACKGROUND
Mobile devices bring the almost ancient needs in human-computer interaction back to life. In the early 1950's, handwriting recognition was the major research area – before the text keyboard took its place as the common way of computer text input. [10]

The development in the area of text input boosted in the mid 1980's. After that time, the need for easier text input methods has been increasing, because the rise of smaller and smaller computing devices.

Even though the need for more advanced technology for text input has been around for so long, the development has concentrated on other things, such as making hardware smaller. Now that the science has reached a point where devices can be made so small that a traditional keyboard is no longer an option, the text input methods rise again as a very important development issue.

Another thing that makes (especially the international letter) input issues more important is usability and personalization. People require more from their computers – both the traditional and the new technology.

SPECIAL CHARACTERS

Most text input methods and devices are developed *in* and especially *for* the English speaking world. This can be seen in the science around the subject, also – almost all studies and empirical tests are made with English language.

Still, a considerable number of users need special characters, such as Scandinavian letters. This issue has not been taken into account well enough in the design of the mobile devices and software.

Scandinavian Letters in Mobile Devices

The irritation of the people who do not speak and write English as their native language is common, when they need to draw strange things on the device to get their own letters 'ä', 'ö', 'ü' etc. to display correctly. Or, they need to change to another keyboard on the run to tap a single letter. It should be possible to select an own keyboard before starting to use the device - or teach the device to understand the way the person is used to type letters…

The traditional computer users do not have to change to an infrequent letters keyboard, when they want to type in letters that are special to their own language. Why should it be different in mobile computers? Very often the keyboard that is used for typing in the mobile devices, is a soft one – a piece of software that displays the keys on the device screen. Thus, selecting between different keyboards seems even easier in the world of mobile computers than in the desktop environment.

Another way of entering text is handwriting recognition. The device recognizes the unistrokes that the user draws on a special area on the screen. The surface recognizes what is drawn. The functionality is based on algorithms – these should be developed to learn the user's behavior, and let him/her personalize the device for easier and more efficient use.

In Finnish, the problem letters are 'ä' and 'ö'. On a traditional (Finnish) desktop QWERTY keyboard these letters are easily typed by one hit on a key. On the mobile devices, the problem is that they do not provide a local keyboard or an easy way of handwriting the letters.

SOFT KEYBOARD LAYOUTS

The QWERTY keyboard is today the one that people are most accustomed to in typing on a traditional computer. Several studies have shown that this keyboard layout is not the most efficient one for mobile computing, where typing is usually done with one hand only. [2,7,8]

There are several competitive layouts for QWERTY. These layouts have been designed in order to make one hand typing as fast and as flawless as possible. Again, Scandinavian letters have not been taken into account in the design of most of them. For example, the frequency of letter-to-letter transitions has mainly been studied in the English language, the most frequent letters are the ones in English etc.

OPTI Keyboard

OPTI keyboard is developed to optimize text input with one hand. In some tests (MacKenzie et al. [7]) it has proven to be the fastest in word per minute (wpm) rates for text entry.

Figure 1: OPTI keyboard

Fitaly Keyboard

Textware Solutions sells a commercial keyboard named Fitaly, which is claimed to be the optimal solution for entry with a single finger or with a pen, as is the case on a pen computer or a computer with a touchscreen.

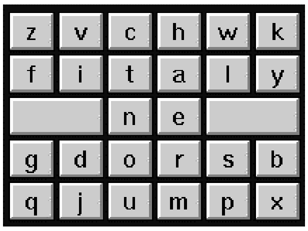


Figure 2: Fitaly keyboard

ABC Keyboard

In the ABC keyboard, the letters are in alphabetical order, in two columns. This makes learning the places of the keys easier (good for the novice user), but makes letter-to-letter transition slow (bad for the expert user). This keyboard layout is an exception in the group of layouts that support a single or only few languages.

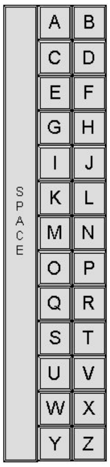


Figure 3: ABC keyboard

Virtual Oval Keyboard

The virtual oval keyboard [2] aims at making the typing faster and more flawless by shortening the distance between the letters, and still providing the familiar QWERTY layout.

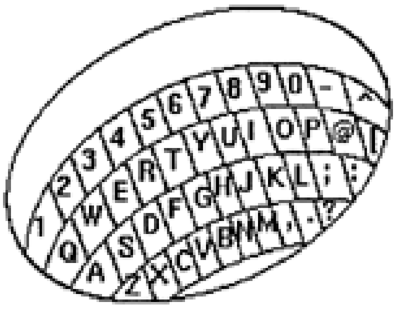


Figure 4: Virtual oval keyboard

Soft Keyboard in Palm™

In Palm™, an on-screen keyboard can be made visible by tapping the “abc” or the ”123” icon at the bottom corner of the data entry field. There are three kinds of soft keyboads in Palm™: the traditional QWERTY for basic text characters, and separate keyboards for numbers/special characters and international characters.

In Palm™, the soft keyboard appears in the English mode, and international letters can be typed after selecting the “int’l” keyboard. This is very inconvenient for the user, and simply makes typing Finnish with the soft keyboard too slow.

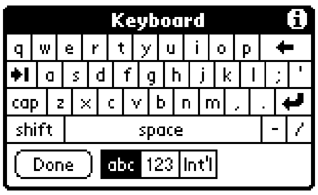


Figure 5: Palm™'s on-screen text keyboard

TEXT ENTRY WITH DIFFERENT KEYBOARD LAYOUTS

The different layouts have been tested for text entry speeds and error rates. MacKenzie et al. [7,8] have confirmed the predictions that QWERTY is the fastest and most accurate for novice users, but the others can be a little faster, when the user is more experienced.

Most of the devices, in which one hand typing is required, are for novice consumer users. The ease of use is the most crucial feature from the moment the person takes the deivice in his/her hand. The biggest problem for any new keyboard layout is the (often) lengthy learning phase that a person has to go through when starting to use a layout different from QWERTY.

Due to these facts, it is easy to predict that QWERTY layout will remain as the most commonly used keyboard layout also in the mobile devices. For example, in the two most succesful handheld computers, Compaq’s iPAQ and 3Com’s Palm™, this familiar layout for soft keyboard text entry is used. Still, the studies have clearly shown that the word per minute averages in one-hand typing are better with optimized layouts.

It seems reasonable to design more optimal layouts not only for one-hand typing but also for languages that have special characters. This issue is shortly covered in the following subsections. New keyboard layouts are not presented, but a few important facts are pointed out that could increase text input efficiency, not only for different languages but also for mobile text input in general.

Placement of Characters

Learned Pattern of Characters

None of the new keyboard designs seem to take into account the fact that people who are accustomed to using the QWERTY keyboard have most likely formed some sort of a pattern in their mind about on which side of the keyboard a certain letter is placed. For example, when a person writes the word “assist”, he/she will intuitively look for the first three letters on the right side of the keyboard, because they are there on the familiar QWERTY keyboard.

The QWERTY will most certainly stay as almost the only keyboard for the traditional computer, and the traditional computers will remain, even if the PDAs got a lot better. Thus, combining the traditional and the new layout pattern in some way would in the future, when people are using both kinds of devices, support both the old and the new text input methods.

This kind of a design approach would probably make the new input system’s learning phase shorter and also increase the experts’ word per minute rates. This would be a good issue to study further.

Most Frequent Letters in Different Languages

The most frequent letter in the English language is ‘e’ [6]. The language frequencies differ from language to language. For example, one of the most infrequent letters, ‘j’ appears quite often in Scandinavian words. This fact has to be taken into account in the design of optimized keyboard layouts.

Both Opti and Fitaly designers have placed ‘j’ key far from the center, where the most frequent letters are. This cannot be the optimal solution for Scandinavian languages. It would be a good subject for research to find out the optimal character placements for different languages.

The Shape of the Keyboard

Since PDAs very often are rectangular, it seems reasonable to make the soft keyboard in them the same shape. If the different shapes, for example the oval keyboard or the round-shaped Cirrin keyboard [9], prove to be more efficient, and the space around them on the screen can be designed for some purpose, they will provide a very interesting alternative for the traditional shape.

The oval keyboard proves that also the QWERTY layout can be designed in a way that all characters are quite near each other. In the future, all the issues pointed out in this section should be taken into account when evaluating and designing better keyboard layouts for mobile devices.

HANDWRITING RECOGNITION

Handwritten letters are written in the mobile device with a stylus[1]. A small part of the PDA's screen surface is usually devoted for handwriting. The ways of writing the letters are different in different brands, but the letters resemble the normal ones very much (see figure 6 below) – this makes learning easier.

Writing letters with different character sets has been proven to be quite similar in speed and accuracy [3].

Handwriting Recognition in Palm™

In Palm™, there are only two areas for handwriting, for text and for number typing. Characters are written in a text input system called Graffiti®.

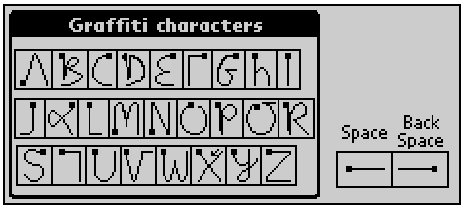


Figure 6: Basic Graffiti® characters in Palm™

The data entry fields are part of the physical design, and they do not steal too much of the screen area (figure 7). Capital letters are written on the text area in exactly the same way as small letters – after dragging the stylus upwards (once for making the case change only for the following letter or twice for a permanent case change).

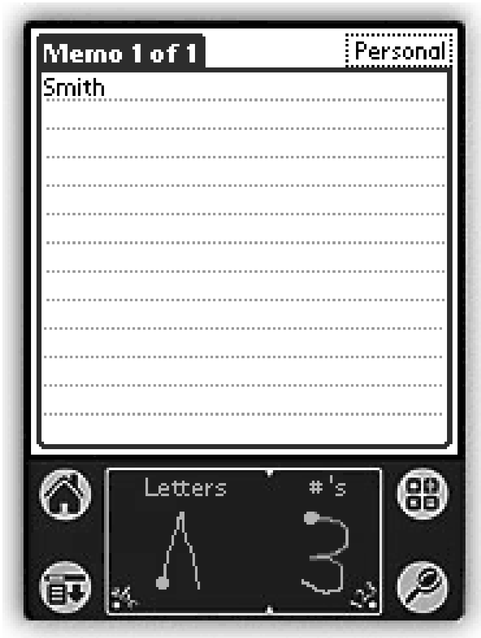


Figure 7: Palm™'s handwriting fields are at the bottom of the screen

Palm™'s way of entering text into a relatively big input area makes it easy to write without looking at the device. This is good for work use – writing meeting memos etc.

IMPROVED HANDWRITING LETTERS

There are many ways of writing Scandinavian letters more easily than how it is done in mobile devices today. In this section some ideas are presented for improving speed and accuracy in special character input, when Graffiti® strokes are used (for example, in Palm™).

Present Situation

In Graffiti®, the characters are written on top of each other on the recognizer area. After looking at the Graffiti® strokes more carefully, it is obvious that the normal way of writing the dots on top of 'a' or 'o' (which would be the most convenient) cannot be used in Graffiti®. For example, two taps after writing the letter 'a':

would result in: "**a.**", because the first tap would turn on the special character (dots, commas etc.) recognizer and the second one would then be recognized as a dot. A line drawn after writing the letter 'a':

would result in "**a** " ('a' and a blank), or if the line was drawn in the other direction:

[1] A pen-like stick for drawing on the recognition surface of a PDA or some other computing device.

nothing would be written, because the ‘a’ would be erased by the following stroke.

The letter ‘ä’ is thus handwritten in Palm™ with the following Graffiti® strokes (strokes are written on top of each other):

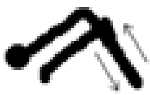
And the letter ‘ö’:

Another Example

In Compaq’s iPAQ, the dots on Scandinavian letters are tapped just as in normal handwriting (two taps on top of the letter). This is a familiar way, and reduces the number of errors made in typing. The taps have to be made quite accurately on top of the basic letter to avoid errors.

The Double Stroke

A big improvement to the strange dot stroke, which very easily gets mixed with the ‘w’ or the ‘y’ character would be, if the letters were written in the following way:

‘ä’:  and ‘ö’: 

The letter is written another time backwards on top of the basic one, without lifting the stylus. None of the Graffiti® letters (see figure 7) resemble these double strokes, so the letters would not be mixed very easily.

Tests in writing letters in a Palm™ device showed that the Double Stroke ‘ä’ resulted in various letters: ‘b’, ‘d’, ‘o’, and the shortcut symbol. Very often no letter was recognized. The Double Stroke ‘ö’ resulted in letters such as ‘d’, ‘g’, ‘p’, ‘w’ or no letter recognition.

Continuous Line

Another option would be to draw the line above the basic letter without lifting the stylus in the middle:

‘ä’:  and ‘ö’: 

Testing the Continuous Line ‘ä’ stroke in a Palm™ resulted in ‘n’ in all cases. ‘ö’ resulted in ‘q’ in almost all cases.

The very small tests imply that with some more research and small recognition software changes the Double Stroke or the Continuous Line model could be possible to implement and made part of an enhanced version of Graffiti®.

A Learning Recognizer

The way in which people write letters differs on even the simplest strokes. The recognizer/panel should be able to learn from the user’s stylus movements. This could be done automatically as the user types text in, or in a technically simpler way, by allowing the user to teach the device his/her way of writing certain characters.

In today’s devices, there are some functions that are something in this direction. For example Palm™ allows the user to create shortcuts to pieces of text. But run-time learning devices would ease the use a whole lot more.

OTHER IMPROVEMENTS

Locale Selection

When a new desktop computer or a mobile device is taken into use, preferences have to be set. The operating system in the traditional computer asks for a default locale for the system, and sets up e.g. time information and keyboard settings based on that. The time and some other things have to be set in the handheld computers as well – why is there no locale selection available?

After having selected the locale for the handheld, the user would be able to use many features for easier use. These include language-based optimization for text input, word prediction and word completion.

Language-based Optimization

If the device knows the language the user is writing, it can help optimize the input by correcting text automatically and suggesting grammar and spelling changes.

Every experienced MS Word user is familiar with the function that highlights spelling and grammar errors in the text. This function remains to be seen on the mobile devices.

Word Completion

The same kind of thing as correcting text or highlighting errors as one does the typing, is completing the words by suggestions from a dictionary. This feature also requires knowledge of the language used, and also a dictionary (a default or user-created/modified one). It has been proven useful in research – James and Reischel showed that the word per minute rates on expert users are more than double the ones with traditional methods with word prediction [5].

In Finnish language, this feature is not easy to implement, because the words are long and there are often complex endings added to the basic word.

Word Prediction

Some mobile phones are today capable of predicting words that the user is writing. These devices do know the language that is being used. Even though using this feature – as all new ways of doing things – requires training and getting used to, it would make the usability of mobile text input better.

Keyboard Selection

Research shows that people are efficient in producing text with the devices they have become accustomed to. The QWERTY keyboard is thus the fastest and least error prone soft keyboard for the novice users. [8] This implies that also a local keyboard should be available in the mobile devices – the Scandinavians are used to finding the Scandinavian letters on the right hand side of their QWERTY keyboard. If a more optimized keyboard (see above) could be provided, the use would probably become an even more comfortable experience.

DISCUSSION

A great deal of text in today's mobile devices is transfered into the handheld by one button. That button is the one that starts the synchronization between the mobile device and the desktop computer. This is a fact that may reduce the efforts to make mobile text input easier. When text can be typed in with the traditional keyboard and then transfered into the handheld, it seems easy enough to get data in.

This fact has probably slowed down the development of mobile text input to some extent. In the future, when the handheld devices become more independent and become somewhat rivals or replacements to the desktop computers, text input by synchronizing the handheld and the desktop computer is no longer an excuse for leaving the issue of making mobile text entry easy for less attention.

CONCLUSIONS

This paper has pointed out the needs and suggested some improvements to the mobile text input technologies. Research and design efforts should be made on optimizing and possibly implementing the methods and improvements. Different soft keyboard layouts and easier written special characters should be created.

The PDA user should be able to choose between different keyboards and handwriting recognizers – depending on his/her native language (or the language the user prefers to write in).

An enthusiastic goal would be to design a device independent, user definable text entry system, which would support all languages. Even small steps taken into this direction are valuable – and make the user experience more pleasant.

ACKNOWLEDGMENTS

Thank you to TietoEnator for supporting this work and to my colleague Aleksi Vesaikko-Rodriguez for presenting innovative ideas for text input methods.

REFERENCES

1. Goldstein, M., Book, R., Alsiö, G., and Tessa, S. Non-Keyboard QWERTY Touch Typing: A Portable Input Interface For The Mobile User, in *CHI '99*, Pitsspurgh, PA, USA, 1999, ACM, 32-39.
2. Hashimoto, M., and Togasi, M. A Virtual Oval Keyboard And a Vector Input Method for Pen-Based Character Input, in *CHI '95*, Denver, CO, USA, 1995, ACM, 254-255.
3. Isokoski, P., Model for Unistroke Writing Time, in *Proceedings of the CHI '01*, ACM, 2001.
4. Isokoski, P., and Raisamo, R. Device Independent Text Input: A Rationale and an Example, in *Proceedings of AVI 2000 Conference on Advanced Visual Interfaces*, New York, 2000, ACM, 76 – 83.
5. James, C.L., and Reischel, K.M. Text Input for Mobile Devices: Comparing Model Prediction to Actual Performance, in *SIGCHI '01*, Seattle, WA, USA, March-April, 2001, 365-371.
6. MacKenzie I.S., Chang L., A Performance Comparison of Two Handwriting Recognizers, *Interacting with Computers*, Guelph, Ontario, Canada, 1999, 283-297.
7. MacKenzie, I.S., Zhang S.X., The Design and Evaluation of a High-Performance Soft Keyboard, in *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*, New York, USA, 1999, ACM, 25-31.
8. MacKenzie, I. S., Zhang, S. X., & Soukoreff, R. W. Text Entry Using Soft Keyboards. *Behaviour & Information Technology*, Guelph, Ontario, Canada, 1999, 235-244.
9. Mankoff, J., and Abowd G.D. Cirrin: A Word-Level Unistroke Keyboard for Pen Input, in *UIST '98*, San Francisco, CA, USA, 1998, ACM, 213-214.
10. Sibert, J., Buffa, M.G., Crane, H.D., Doster, W., Rhyne, J., Ward, and J.R. Panel in *CHI '87*, 1987, ACM.
11. Silfverberg, M., MacKenzie, I.S., and Korhonen, P. Predicting Text Entry Speed on Mobile Phones, in *Proceedings of CHI '00*, The Hague, Amsterdam, April 2000, ACM, 9-16.

Hand Gestures Used In Manipulation System

Mahmoud Haj-Rashid
Department of Computer Science
University of Tampere
Väinämöisenkatu 11 C 331
FIN-33540 Tampere, Finland
mr68653@uta.fi

Abdrizak H. Mohamed
Department of Economics
University of Tampere
Laakavuorentie 1 D 34
FIN-00970 Helsinki, Finland
Am61739@uta.fi

ABSRACT

Gestures use for communicating among people without having difficulties. So the use of hand gestures provides an attractive to cumbersome interface devices for human-computer interaction (HCI). Using hand gestures, which are important human gestures, is a key technique to build a friendly user interface.

Thereby, the ease and naturalness desired for HCI can be got by the visual interpretation of hand gestures. We survey the literature on visual interpretation of hand gestures in the context of its role in HCI. This discussion is organized on the basis of the method used for modeling, analyzing, and recognizing gestures. We examine in our paper the important differences between the gesture interpretation methods, and we figured out these differences increase depending on whether a 3D model of the human hand or an image appearance model of the human hand is used. Our paper focuses on gestural systems in addition to the other applications of vision-based gesture recognition. Through our research we discuss ways of future research in gesture recognition, with its integration with other natural modes of human-computer interaction.

Keywords

Human-computer interaction, appearance-based gesture recognition, gesture analysis.

1. INTRODUCTION

Recognition of hand gestures may provide a more natural human-computer interface, allowing people to point, or rotate a CAD model by rotating their hands. Interactive computer games would be enhanced if the computer could understand players' hand gestures. Gesture recognition may even be useful to control household appliances.

Until recently, most of the work on vision based gestural HCI has been focused on the recognition of the static hand gestures or postures.

For the broadest possible application, a gesture recognition algorithm should be fast to compute.

We organize the survey by breaking the discussion into the following main components based on the general view of a gesture recognition system as following:

- Gesture modeling
- Gesture analysis
- Gesture recognition
- Gesture based systems and applications

2. GESTURE MODELING

In order to systematically discuss the literature on gesture interpretation, it is important to consider what model the authors have used for hand gesture. In fact, the scope of gestural interface for HCI is directly related to the proper modeling of hand gestures. How to model hand gestures depends primarily on the intended application within the HCI context? For a given application, a very coarse and simple model may be sufficient. However, if the purpose is natural-like interaction, a model has to be established that allows many if not all-natural gestures to be interpreted by computer. The following discussion addresses the question of modeling of hand gestures for HCI.

2.1 Definition of Gestures

Outside the HCI framework, hand gestures cannot be easily defined. The definitions, if they exist, are particularly related to the communicational aspect of human hand and body movements. Webster's Dictionary, for example, defines gestures as "…the use of motions of the limbs or body as a means of expression; a movement usually of the body or limbs the expresses or emphasizes an idea, sentiment, or attitude." Psychological and social studies tend to narrow this broad definition and relate it even more man's expression and social interaction [15]. However, in the domain of HCI the notion of gestures is somewhat different. In a computer controlled environment one wants to use the human hand to perform task that mimic both the natural use of the hand as a manipulator, and its use in human-machine communication (control of computer/machine functions through gestures). Classical definitions of gestures, on the other hand, are rarely, if ever, concerned with the former mentioned use of the human hand (so called practical gestures [15]

Hand gestures are a means of communication, similar to spoken language. The production and perception of gestures can thus be described using a model commonly found in the field of spoken language recognition. An interpretation of this model, applied to gestures, is depicted in **Fig.1**

Gesture – mental concept ⇒ Hand Movement ⇒ Visual Images.

Fig 1 shows production and perception of gestures. Hand gestures originated (as a mental concept) are expressed through hand, motion and perceived as visual images.

According to the model, gestures originate as a gesture's mental concept, possibly in conjunction with speech. They are expressed through the motion of arms and hands, the same way speech is produced by air stream modulation through the human vocal tract. Also, observers perceive gesture as stream of visual images which they interpret using the knowledge they possess about those gestures.

In the context of visual interpretation of gestures, it may then be useful to consider the following definition of gestures: Each realization of one gesture can then been seen as a trajectory in the model parameter space. The stochastic property in the definition of gestures affirms their natural characters: no two realizations of gesture will result in the same hand and arm motion or the same set of visual images.

The gesture analysis and gesture recognition problems can than be posed in terms of the parameters involved in above definition.

2.2. Modeling of Gesture Temporally

We have to put in account that the temporal characteristics of gestures are very important as long as the human gestures are dynamic process. This may help in the temporal segmentation of gestures from other unintentional hand movements. If we look at the definition of gestural hands carefully we will figure out that the temporal modeling of gesture is equivalent to determining the gesture interval. It seems so amazing that the psychological studies are fairly consistent about the temporal nature of hand gestures. Kendon [15] calls this interval a "gesture phrase." It has been established that three phases make a gesture:

- preparation
- nucleus (peak or stroke), and
- retraction.

The preparation phase consist of a preparatory movement that sets the hand in motion from some resting position. The nucleus of a gesture has some " definite from and enhanced dynamic qualities" [15]. Finally, the hand either returns to the resting position or repositions for the new gesture phase. We should remember that what has been said not applicable to all, there's an exceptional rule which called "bets".

The above discussion can guide us in the process of temporal discrimination of gestures. The three temporal phases are distinguishable through the general hand-motion: "preparation" and "retraction" are characterized by the rapid change in position of the hand, while the "stroke," in general, exhibits relatively slower hand motion. However, the complexity of gestural interpretation usually imposed more stringent constraints on the allowed temporal variability of hand gestures. Hence, a work in vision-based gesture HCI sometimes reduces gestures to their static equivalents, ignoring their dynamic nature.

2.3. 3D Hand Model

The 3D-hand model has often been a choice for hand gesture and arms. They can usually include two large classes.

2.3.1. Volumetric models:

Volumetric models are meant to describe the 3D visual appearance of the human hand and arms. They are commonly found in the field of computer animation, but have recently also been used in computer vision applications. In the field of computer vision volumetric models of the human body are used for analysis-by-synthesis tracking and recognition of the body's posture by synthesizing the 3d model of the human body in question and then varying its parameters until the modal and the real human body appear as the same visual images. Most of the volumetric models used in computer animation are complex 3D surfaces (NURBS or non-uniform rational B-splines) which enclose the parts of the human body they model

2. 3.2. Skeletal models.

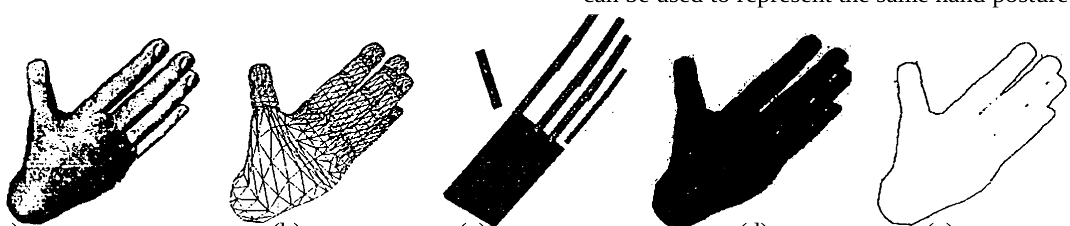
Skeletal models are extensively studied in human hand morphology and biomechanics. The human hands skeletal consists of 27 bones divided in three classes:

Carpal (8 bones)

Metacarpals (5 bones)

Phalanges (14 bones)

The **Fig2** shows the hand models. Different hand models can be used to represent the same hand posture.



(a) (b) (c) (d) (e)

Fig 2 Hand models. Different hand models can be used to represent the same hand posture.

(a) 3D textured volumetric model.

(b) 3D wireframe volumetric model.

(c) 3D skeletal model.

(d) Binary silhouette

(e) Contour.

The joints connecting the bones naturally appear different degrees of freedom. Most of the joints connecting carpal has very limited freedom of movements.

2.4. Appearance-Based Model

The second group of models is based on appearance of hands in visual images. This means that the model parameters are not immediately derive from the 3D spatial description of the hand. The gestures are modeled by relating the appearance of any gesture to the appearance of the set of predefined, template gestures.

A large variety of models belong to this group. Some of them are based on deforming 2D templates of the human hands, or even body [7], [13], [16], [17]. Deforming 2D templates are the sets of points on the outline of an object, used as interpolation nodes for the object outline approximation.

3. GESTURE ANALYSIS

We consider, in this section, on the analysis phase where the goal is to estimate the parameters of the gesture model using measurements from the video images of a human operator engaged in HCI. Two generally sequential tasks are involved in the analysis **Fig. 3**. The first task involves “detection” or extracting relevant images features from the raw image or image sequence. The second task uses these images for computing the model parameters.

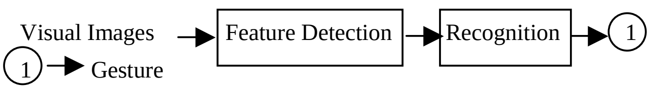


Fig 3 shows analysis and recognition of gestures.

3.1. Feature Detection

Feature detection stage is concerned with detection of features, which are used for estimation of parameters of the chosen gestural model. In the detection process, it is necessary to localize the gesture. Once the gesture is localized, the desired set of features can be detected.

3.1.1. LOCALIZATION

Gesture localization is a process in which the person performing the gestures extracted from the rest of the visual image. Two types of cues are often used in the localization process:

- Color cues and
- Motion cues.

Color cues are applicable because of the characteristic color footprint of the human skin. The color footprint usually is more distinctive and less sensitive to illumination changes in the hue-saturation space than in the standard (camera capture) RGB (Red, Green and Blue) color space. Most of the color segmentation techniques rely on histogram matching or employ a simple look-up table approach based on the training data for the skin and possibly its surrounding areas. The major drawback of color-based on the localization techniques is the variability of the skin color footprint in different lighting condition [2].

Motion cue is also commonly applied for gesturer localization and is used in conjunction with certain assumptions about the gesture. For example, in the HCI context, it is the case that only one-person gestures at any given time. Moreover, the gesture is usually stationary with respect to the background. Hence the main component of motion in the visual image is usually the motion of the hand and can thus be used to localize her/him.

3.1.2. FEATURES AND DITECTIONS

Although different gesture models are based on different types of parameters, the image features employed to compute those parameters are often very similar. For example, models that use finger trajectories and 3D hand models require fingertips to be extracted first. Mainly color or gray scale images that encompass hands or gestures themselves are used often as the features. One approach to using whole images as features is related to building the motion energy images. So, let’s know what motion energy images mean?

In fact, the motion energy images are 2D images that unify the motion information of sequences of 2D images by accumulating the motion of characteristic image points over the sequence [4].

Contour is another group used as a feature. Several different edge schemes can be used to produce contours. Contours are often used in 3D model-based analyses. While the fingertip used as a feature in gesture analyses. Locations of fingertip can be used to both the 3D hand models and 2D appearance based gestural models. However, it is not trivial to detect the fingertip location by using 3D or 2D space. In fact, the simple solution to fingertip detection problem is to use marked gloves or color markers.

4. GESTURE RECOGNITION

The definition of gesture recognition is the phase that the data analyzed from the visual images of gestures is recognized as specific gesture. The two tasks that associated to the recognition process are:

1. The task of optimal partition of the parameters space and
2. The task of implementation of the recognition procedure.

The first task is usually addressed through deferent *learning from-examples* training procedures. The task of optimal partition of the model parameter space is related to the choice of the gestural models and their parameters. However, most of the gestural models are not implicitly designed with the recognition process in mind. This is particularly true for the models of the static gestures or hand postures [11].

While the second task is the key concern in the implementation of the recognition procedure, which is computational efficiency. To perform recognition of those gestures, some type of parameter clustering technique creating from vector quantization is usually used. But let's light on vector quantization!

In vector quantization is briefly an n-dimensional space divided into convex sets using n-dimensional hyperplans, based on training examples and some metric for determining the nearest neighbor.

5. APPLICATIONS AND SYSTEMS

Many potential applications have driven current interest in gesture interface for HCI **Fig. 4.** Hand gestures can simply improve the interaction in desktop computer applications as a mode of HCI by taking the computer mouse (hand held devices) away and using instead of that gestural interface. Of course, they can also replace joysticks and buttons in the control's devices. Moreover, the main reason in the development of gestural interfaces has come from the growth of applications has been in virtual environment (VEs) [2] [19].

The communicative tasks of gestures are an accurate, and tend to be a supportive element of speech. At this point, we should have exceptions of deitic gestures, which play a major role in human communication. However, some applications have emerged recently which take advantage of the communicative role of gestures. The aspect of manipulative gestures also prevails in their current use for HCI. Both manipulative actions and communication have used hand-gestures in natural environment as we have seen so far.

Manipulators of virtual object (VOs) have been described by most applications of hand-gestures as shown in (figure 9). Computer graphics such as 2D and 3D objects can be virtual object [6], [12], [9] in addition, virtual objects can be windows [17] and abstractions of physical objects, like device control panels [3], [8], [9] or robotic arms [7], [11], [14], as well. A combination of coarse tracking and communicative gestures is currently used to perform manipulations of such objects through HCI. For instance, to rotate an object we should direct the computer by giving two comments:

Select object

Rotate object

The first action uses a hand tracking to move a pointer in the virtual environment to the area that close to the object. The second action is to rotate the object, here the user rotates his hand back forth producing a metaphor rotational manipulation [6].

5.1. THE COMMUNICATIVE GESTURES USES FOR MANIPULATIVE ACTION:

First of all, communicative gestures imply a certain vocabulary of gestures that has to be learnt, while the manipulative ones are neutral hand movements. One has to consider the complexity of analysis and recognition of each type of gestural models. Modeling of both manipulative and communicative gestures is well suited to 3D hand model-based gestural models, while the appearance-based models of gestures are mostly applicable to the communicative ones.

Because the, 3D hand model-based gesture models are in some how more expensive than the appearance-based models, therefore, to achieve a real-time performance; one has to resort to the less preferable appearance-based models of gestures. Nowadays, the computing power is growing up, and simplified hand models [4], have been taken in account for applications that use communicative gesture recognition [5]. Recognition of communicative gestures uses models are simple to be analyzed in real-time.

Characteristics of the systems aimed at the applications of hand gestures for HCI are shown in **table (*).**

After looking at the **table (*)** we can say that it summarizes the basic modeling technique used for the gestures, the class of gesture commands that are interpreted, and the reported performance in terms of the speed of processing.

*** Table shows systems that employ hand gestures for HCI:**

Application	Gestural Modeling Technique	Gestural commands	Complexity (Speed)
Gesture Computer [7]	Image moments & fingertip position	Tracking and one metaphor combined with speech	Real-time
Hand sign recognition[24]	Most expressive feature cameras of images	40 signs	n.a.
Finger pen	Finger tip template	Tracking only	n.a.

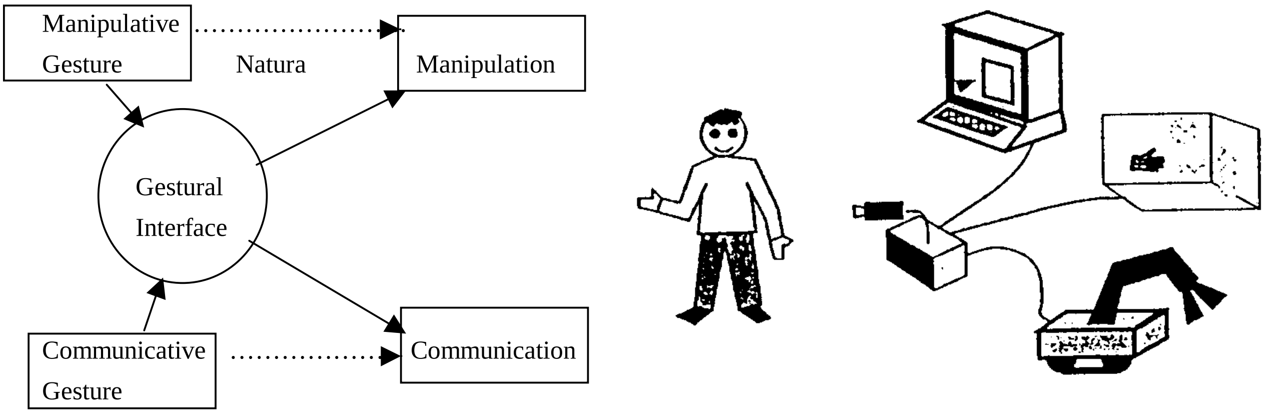


Fig.4 *shows Applications of gestural interface for HCI. The manipulative and communicative gestures in HCI can be used in direct manipulations of objects.*

We should emphasize that not all of the applications of hand gestures for HCI are meant to imply manipulative actions. Moreover, gestures for HCI can be used to convey messages for the purpose of their analysis, storage or transmission. The reduction of bandwidth is one of the major issues in applications of the video teleconferencing (VTC). One coding techniques is model-based coding where image sequences are described by the states (for instance, position, scale and orientation) of all physical objects in the scene [1], [10]. Only the updates of the descriptors are sent, then the computer will generate model of physical object using the received data. Therefore, model-based coding for VTC requires the human bodies be modeled in the right way. Depending on the amount of desired detail, this can be achieved by only coarse models of the upper body and arms, or finely tuned models of the human faces or hands. In that time, modeling of hand gestures can be of substantial value for thus applications.

Of course, there is another great application using human gestures, known Sign Language, which naturally employs human gestures as means of communications. Those kind of application have a real role in communication with people who can’t speak or hear.

Automatically some devices that could translate sign language hand gestures into speech signals would for sure have a positive effect on individuals. However, the reason, which could be more practical for using the sign language, is well-defined structure, compared to other natural gestures human use. This fact yields us that the appearance-based modeling techniques are particularly suited for such sign language interpretation, as was proven in several current applications.

There are many prospective applications of vision based hand gesture analysis. The applications used so far in task of hand gesture are first steps in HCI. In previous applications we light on importance of research in issues that need incorporating natural hand gestures into the HCI.

6. DISCUSSION AND CONCLUSION

Human Computer Interaction is still in its first stage if we compare it with other fields in computer science such as programming, software engineering and algorithmatics. The visual interpretation of hand gestures has a role in development of natural interfaces to computer controlled environments. Therefore, the number of different approaches to video-based hand gesture recognition has grown hugely in recent years. For that reason there is a need to analyze and use of many gestural interfaces. We tried in this paper to emphasis to the modeling, analyze and recognition of hand gestures for visual interpretation. The discussion has produced two classes of models in our paper, which employed in visual interpretation of hand gestures. The first class is related to 3D models of the human hand, while the second one uses the appearance of the human hand in the image. 3D hand models enrich description and discrimination capability that let a big class of gestures to be recognized as leading to natural human computer interaction. However, the computation of 3D model parameters from visual image under real time constrains is still belong to an unclear goal.

The vision based gesture interfaces have proposed many simple HCI systems. However, according to the practical viewpoint, the development of such systems is still at in initial life. Although most current systems use hand gestures for the manipulation of objects, the complexity of interpretation of gestures dictates the performed solution.

For instances, the hand gestures for HCI in most cases are restricted to single hand and produced by as single user in the system. Of course, this will reduce the effectiveness of the interaction. By doing this paper we figured out that integration of hand gestures with speech would make gestural HCI more usable.

Finally, we believe that development and effectiveness of natural hand gesture interface will need real research which connects advances in computer vision with the basic study of human computer interaction.

7. ACKNOWLEDGEMENTS

We like to thank our professor Roope Raisamo for his support.

REFERENCES

[1] J.F. Abramatic, P. lettelier, and M. Nadler, “A Narrow-Band Video Communication System for the transmission of Sign Lanuage Over Ordinary Telephone Lines, “ Image Sequences processing and Dynamic Scene analysis, T.S. Huang, ed, pp. 314-336. Berline and Heidelberg: Spring-Verlag, 1983.

[2] J.A. Adam “Virtual Reality,” IEEE Spectrum, vol. 30, no. 10, pp. 22-29, 1993.

[3] S. Ahmad, “A usable Real-Time 3D Hand Tracker,“ IEEE Asilomar Conf., 1994.

[4] A. Azarabayeini C. Wren, and A. Pentland, “Real-Time 3D Tracking of the Human Body,” Proc. IMAGE’COM 98, Bordeaux, France, 1996.

[5] D.A. Becker and A. Pentland, “Using a Virtual Enviroment to Teach Cancer Patients T’ai Chi, Relaxation, killington, Vt., Oct. 1992.

[6] U. Bröckl-Fox, “Real-Time 3D Interaction With Up to 16 Degrees of Freedom From Monocular Image Flows,” Proc. Int’l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 172-178, June 1995.

[7] R. Cipolla and N:J. Hollinghurst, “Human-Robot Interface Pointing With Uncalbrated Stereo Vision, “ Image and Vision Computting, vol. 14, pp. 171-178, Mar. 1996.

[8] W.T. Freeman and C.D. Weisman, ”Television Cotrol by Hand Gestures,” Proc. Int’l Workshop on Automatic Face and Gesture Recognition, Zurich,Switzerland. pp. 179-183, june 1995.

[9] M. Fukomoto, Y. Suenaga, amd K.Mase, ” Finger Pointer”: Pointing Interface by Image Proccessing,” Computer and Graphics, vol.18, no.5, pp.633-642, 1994.

[10] H. Harashima and F. Kishino, ” Intelligent Image Coding and Communications With Realistic Sensations-Recent Trends,” IEICE Trans., vol. E74, pp. 1.582-1,592, June 1991.

[11] E. Hunter, J. Schlenzing, and R. Jain, ”Posture Estimation in Reduced-Model Gesture Input Systems,” Proc. Int’l Workshop on Automatic Face and Gesture Recognition, June 1995.

[12] K. Ishibuchi, H. Takemura, and F. Kishino, ”Real Time Hand Gesture Recognition Using 3D Prediction Model,” Proc. 1993 Int’l conf. Seystems, Man, and Cybernetics, Le Touquet, France, pp.324-328, Oct. 17-20, 1993.

[13] S.X. Ju, M.J. Black, and Y.Y. oob, ”Cardboard People: A Parametrized Model of Articulated Image Motion,” Proc. Int’l Conf. Automatic Face and Gesture Recognition, Killington, Vt., pp. 38-43, oct. 1996.

[14] S.B. Kang and K. Ikeuchi, ”Toward Automatic Robot Instruction for Perception-Recignizing a Grasp From Observation,” IEEE Trans. Robotics and Automation, vol.9, pp. 432-443, Aug. 1993.

[15] A. Kendon, ”Current Issues in the Study of Gesture.” The Biological Foundations of Gestures: Motor and Semiotic Aspects, J. –L. Nespoulous, P. Peron, and A. R. Lecours, eds., pp. 23-47. Lawrence Erlbaum Assoc., 1986.

[16] C. Kervrann and F. Heitz, ”Learning Structure and Deformation Modes of Nonrigid Objects in Long Image Sequences,2 Proc. Int’l Workshop on Automatic Face and Gesture Recognition, June 1995.

[17] R. Kieldsen and J. Kender, ”Visual Hand Gesture Recognition For Window System Control,” Proc. Int’l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp.184-188, June 1995.

[18] R. Kieldsen and J. Kender, ”Finding Skin in Clor Images.” Proc. Int’l Workshop on Automatic Face and Gesture Recognition, Killington, Vt., pp. 312-317, Oct. 1996.

[19] M.W. Krueger, Artifial Reality II. Addition-Wesley, 1991.

N. Magnenat-Thalmann and D. Thalman, Computer Animation: Theory and Practice. Newyork: Springer-Verlag, 2nd rev. ed., 1990.

D. McNeil and Levy, ”Conceptual Representations in Language Activity and Gesture,” Speech, Place and Action: Studies in Deixis and Related Topics, J. Jarvella and W. Klein, eds. Wiley, 1982.

Future User Interface for Wearable Computer

Veli Ijäs
Department of Computer Science
University of Tampere
New Interaction Techniques-course
veli.ijas@uta.fi

ABSTRACT

Mobile technology might be the most interesting area in computer science nowadays. Especially in Finland the development of wireless technology gets a lot of attention. That is at least partly because of the fact that our Crown Jewel Nokia with a notable share of mobile phone markets has a very great impression on our economy. In this paper I deal with one part of mobile technology: wearable computers and focus especially on their user interfaces. I'm going to present different solutions of handling context-aware data in wearable computers. Based on previous studies I form a model of multimodal user interface.

KEYWORDS

Wearable computer, user interface, context-aware computing

INTRODUCTION

There have lately been a lot of articles about the progress of mobile technology. Companies all over the world are presenting all the time different kinds of products based on wireless techniques. A good example of that is the largest information technology event in he world, Hanover CeBit Fair in 20. – 28. March 2001. In connection with fair there are commonly plannings on what kind of schedule will gprs, umts or many other kinds of devices come to consumer. At the same time expectations of consumers are increasing. However, it doesn't matter how powerful the technologies multimedia computing, it is the user interface that ultimately determines how these systems will be used [1].

In the effect of progress also the development of wearable computers has progressed. As late as few years ago almost all the research had centred in special applications like how wearable computers can help make aircraft manufacturing workers more productive. This kind of use sets strict limits for user interfaces, too. Nowadays it is made more and more plans for devices to daily life for common people. In this context has also tried out different kinds of user interfaces.

WEARABLE DISPLAY OVERVIEW

There has been examined various kinds of displays on wearable computers. All of these have some good features, but no one of them is above others. It seems that versatile user interface needs qualities from many different systems and in the future when technology still goes on it will be natural to employ multimodal interfaces in mobile computing devices.[2]

Voice-only interaction in a wearable computer

The researchers from Georgia Institute of Technology, Atlanta has studied how an interface could be constructed, which effectively make use of context-awareness [3]. They opinion is that a common focus is the attempt to break away from the traditional desktop computing paradigm. Their new aim has been to provide an interface that can take the responsibility of location and serves the user.

They have had good ideas about using context awareness in a wearable computer. They have developed for instance CyberGuide system, an interactive map, which transmit information from immediate area and Savoir, an information access system with a voice-only interface.

Anind K. Dey et al. mention in their research [4] that voice-only interface has a lot of good qualities. The application with voice-only interface might require a hands-free mode of interaction. That is a very important issue when we are talking about wearable computing. Second, telephone service is one of the few truly robust and ubiquitous network technologies, so it makes sense to extend information services away from the desktop by providing a telephone interface.

The main technologies, which are required to carry out voice-only interaction, are speech synthesis and speech recognition. Speech synthesis has fairly good developed technologies and they are widely available. Speech recognition whereas sets limitations for applications. A. K. Dey et al.[4] therefore decided to use a Wizard of Oz approach, with a human operator performing the speech recognition.

After careful preparations they built a prototype of voice-only interface and used it in a series of usability studies. On the basis of tests they made following observations. The old general principles still apply; feedback is important as well as direct control over an application's actions. They noticed that users remember information worse by working with voice-only interface than working with GUI. This causes problems expressing data in voice-only interface. Issues should be expressed packages short enough and in the same time information should be sufficiently.

As we noticed from the research of Anind K. Dey et al. there are useful properties in voice-only interface but there still are some difficulties, too. One way to solve these problems could be join voice-only and GUI together. That kind of system could still be hands-free if commands were given by talking. Feedback whereas would be much more informative and easier to remember in GUI. Speech synthesis still wouldn't take away but it could act together with GUI.

A wrist watch as a user interface

A wrist watch is practical place of device because we always carry it with us and it is easy to view. The value of smart watches has also been noticed commercially and there are several products with different kinds of qualities as MP3 players, digital cameras, or Personal Information Management applications available. In any case before a wrist watch will be useful enough as a computer display the resolution must become much better.

C. Narayanaswami and M. T. Raghunath have surveyed input devices suitable for wrist watch. In their opinion a touch screen would be a good input mechanism and more elegant compared to buttons After all the relative sizes of a watch face and a human finger limit the number of distinguishable touch zones and a selection mechanism causes problems. In their research [5] they constructed a wrist watch with a high-resolution (600dpi) touch screen and roller wheel and tested it with many kinds of applications.

Before designing the applications they spent a lot of time trying to simplify the user interface by taking advantage of the high resolution display and the chosen input devices. They fixed a lot of attention on usability and according to results they have succeeded quite well. They draw the conclusions that with a careful design it is possible to navigate and employ several applications on a device with the form factor of a watch and present data in a timely fashion in spite of the limited screen size and input capabilities. The most difficult factor seemed to be the size of screen. It caused some problems both expressing information and for lack of selecting space.

They also planned external i/o devices to their watch work with IrDA or Bluetooth. For example one could bring the watch near a Bluetooth connected keyboard to enter information into it. Also voice recognition was studied. These things they did not however test.

NaviPoint: Input for Mobile Information Browsing

Kiyokuni Kawachiya and Hiroshi Ishikawa compared different kinds of input devices in mobile environment. [6] They defined two fundamental operations for information browsing: scrolling and pointing, and studied in which kind of input device these functions could perform best. Based on studies they developed a specialized device for mobile information browsing NaviPoint: a kind of small joystick used by only one finger.

Developing user interface to mobile environment they set three requirements. Because of limited screen size the pointer has to be scrolled in arbitrary direction. In a mobile environment, it may be difficult to watch the screen closely or continuously, and that's why the pointing and selecting operations have to be quick. Device should as well be possible to use with one hand.

Kawachiya and Ishikawa set three requirements to scrolling and pointing operations in a mobile environment: arbitrary scrolling, quick pointing and one-handed operation. Comparison of existing operation methods they noticed that none of them completely satisfy the three requirements for mobile information browsing. Based to these requirements they developed a new input device named NaviPoint. It consists of a stick with a micro-switch and a ring-shaped two-dimensional stress sensor around the stick. The stick can be depressed by applying a certain pressure, and can also be tilted in an arbitrary direction inside the ring. Tilting the stick constantly causes scroll of the document, tilting and returning it several times causes move of the highlight to the target item and depressing selects the highlighted item.

After constructing a prototype Kawachiya and Ishikawa experimented and compared it to an existing ”mouse and scroll bar” system. As a result they concluded that NaviPoint is suitable for browsing hypermedia documents with arbitrary scrolling on a small screen in a mobile environment. If scrolling was performed in horizontal direction NaviPoint can be manipulated with an overhead of less than 50% on the usual mouse operation. Considering the advantages of NaviPoint in a mobile environment, this overhead is acceptable for practical use. Kawachiya and Ishikawa also noticed several factors, which could improve functioning of the device.

Context Compass

Riku Suomela and Juha Lehikoinen from Nokia Research Center, Tampere, Finland presents an easy and natural method for accessing contextual data shown on an electronic map in a wearable computer. [7] They have developed a device called Context Compass. It is an access system to contextual data based on a compass metaphor. Basically it is an extension to WalkMap, a navigation application for a walking user which is also developed in Nokia Research Center.

In their system the user has compass attached to his headgear providing information on the absolute orientation of the user’s head. The display is a see-through head-worn display and a linear compass is on top of the field of view. This linear compass shows the accessible objects around the user and he can choose any object which he want simply by watching it.

Suomela and Lehikoinen set four targets for the user interface of Context Compass [7]:

- it has to be unobtrusive so it does not interfere with the users everyday tasks
- user’s orientation has to be shown
- virtual objects are easy to access
- it has to help in navigation

After define the targets they designed a system and came out well in their project. One thing, which might however cause problems, is the head-worn display used as a device. Especially problems may cause if the user has to see large content of information at the same time when he or she is doing something else. The system is designed so that the user has to pause his or her real world activities while looking at that kind of contents.

DEMANDS FOR FUTURE USER INTERFACE OF WEARABLE COMPUTER

Context awareness plays a central role in wearable computers. Many of the new techniques enable the use of surrounding data. Even now for instance GPS technique can be used by telling location of user. After bluetooth and many other techniques become available the devices which are using the context data will have an opportunity to become considerably new use. The new techniques enables completely novel applications with new segment of users.

There are some significant factors, which will have a great influence on success of wearable computers. The devices have to be small enough. Small size makes device imperceptible and practical to use. When we follow the development of mobile phones we can see that the manufacturers have always been attempt to do smaller and smaller products. It seems that a small size is some kind of end in itself. In any case suitable size gives some notable advantages to the product.

Hands-free feature might be the most notable when we are talking about wearable computers. If the device comes to daily use it can’t be all the time in the users hands. I believe that only very few people want to be all the time aware of computer, which they carry with them. The device has to be like that user ignores it in a daily life but he can get it to use right away if necessary. Hands-free feature gives that kind of benefit to the device.

The feature, which becomes more and more important when the amount of users increases is usability of user interface. One good way to make user interface of the new device easier to learn is the use of models familiar from before. Then user has schemas already ready for learning more.

Multimodal user interface of wearable computer

I have formed a model of user interface of wearable computer on the basis of previous studies. It has turned out that none of existing user interfaces is absolutely better than the others. Therefore it might be reasonable to combine various techniques to support each other’s. Base for user interface could be voice interaction system. The user could have instant and constant contact to computer through it so that he could pick relevant information by giving commands regarding context around him. Also the feedback of computer could partly came through speech synthesis system. That kind of user interface could be hands-free and resembles mobile phone. The functions could also be partly similar as in mobile phones.

There is however a lot of information which is much more effective to express in a figurative form. It should also be connected about size of 5*8 centimetre high-resolution touchscreen fixed to the users wrist in the system. The size is still so small that the device is effortless. Fixed to the wrist it is partly hands-free too. Both input and output information could be given through it.

A head-worn display could also be used in some cases. Especially if the amount of information is not very large the head-worn display could be easy to use. The system would be even more effective if the extra devices were installed for users needs. So could for example the tactile system tell the user the way to the target by giving pulses to that arm where he or she has to turn. This kind of system is designed at least in the Institute of Technology in Massachusetts by the researchers Sevgi Ertan, Clare Lee, Abigail Willets, Hong Tan and Alex Pentland. [9]

Figure 1. An ordinary headset of mobile phone can be used in multimodal user interface.

Examples of using multimodal user interface

The idea of multimodal using interface of wearable computer is based on the use of the most suitable interaction technique in every situation. This holds well both to input and output. The aim is that user specifies what types of objects he or she is interested in by giving commands through the microphone, the touchscreen or the head-worn display and the computer presents the information in the most suitable way.

If the user is interested for example in what kind of restaurants is near his present location, he gives a simply command by saying to the microphone ”restaurant” or types it to the touchscreen. Typing is however not the best way because of the size of the screen. The computer gets information from the database to make use of location of the user. The results are shown in the map, which appears to the touchscreen. The user can still select one of restaurants by touching the point in the map and more information comes out. There is also the current location of the user shown in the map and if he wants the computer draws the route to the destination. There is back-button always available in the touchscreen so the user can return whenever he or she likes. This reduces getting lost in the user interface.

If the user wants to have information from some object near him, he can select it either giving command by saying or selecting it through his head-worn display or touchscreen. In that kind of situations Context Compass-like system shown before could perform well. The feedback however could in many cases be more illustrative presented by high-resolution touchscreen than in the head-worn display.

An example of handling context data is how the user gets information about timetables of buses. When he comes near the bus stop the speech synthesis system of computer informs about that. It appears to the touchscreen the map of local area. The user selects the destination point by pressing the right point of the map and computer tells which busses goes there and when will they come. If the busses of that bus stop don’t go to that direction computer shows where is the nearest right bus stop. All feedback comes through the touchscreen.

DISCUSSION

The benefit of voice interaction is hands-free feature. It’s also simply to use and fairly invisible. After speech synthesis technology has developed much enough the voice interaction will be very efficient way to use many tasks. It doesn’t as well take too much attention of the user so he can actively pay attention to surroundings. Completely new dimension in voice interaction will be reached after so-called direction hearing will be utilized.

There has been a lot of discussion about dangerous situations in traffic caused by use of mobile phone during drive. So it is expected that the use of hands-free devices and ear buttons will increase in driving a car. The researches and conversation about health risks of mobile phones might also arouse interest towards hands-free devices. Therefore the habits of consumer in the use of mobile phones might also have an influence on development of devices in wearable computer environment.

There are a lot of things which might be very hard or which can't be express at all by voice interaction. That's why the voice interface has to be supplemented by another device. C. Narayanaswami and M. T. Raghunath [5] have very good experiences about touch screen devices suitable for wrist watch. The largest weakness they found in their researches was small size of display. In the future interface of wearable computer the size of touch screen display could be for example 2.5 times bigger than in an ordinary wrist watch. Being this size it still isn't too big to wear. Planar Inc. have developed the display which is being used both the handheld and bodyworn systems the 1000 lines per inch display in it. [8] Even quite small device could be very informative by using that grade of display. Fasten to users wrist it will also be partly hands-free. When we are dealing with a touch screen a part of interaction can be used straight by using display so separate buttons are not needed. In addition to partly hands-free display the use of wrist watch is fair familiar to ordinary people which supports acceptance of the new product

A head-worn display could be used in some cases with a voice interaction system. A benefit of it is that it's purely hands-free. It is also very comprehensive system because an eye is very dominant sense. Even more than 80 per cent of stimulus comes through eyes. That is the strength of head-worn system but the weakness at the same time too. Using so dominant system the user might stay partly outside of the real world. That may cause disastrous consequences for example in traffic. A head-worn display has the best benefit while user has time to go into context data in detail. So it could be used like a kind of accessory.

CONCLUSION

I have planned a future user interface for wearable computer to be multimodal system with fluent collaboration between different devices. That kind of system gives user an easy and effective access to context data. It's also designed so that an ordinary people don't regard it as a computer. That is a thing, which may help user to be more familiar with the devices.

In the multimodal user interface, which I have planned, has some aspects, which can't be constructed yet. Especially the speech recognition system is far from the target. The touchscreen system however has already workable solutions. That will be the next issue, which I'm going to search. Especially I'm going to think how could a small screen use efficiently and illustratively by expressing contextual data.

ACKNOWLEDGMENTS

The author like to thank the teacher of the course "New Interaction Techniques" professor of computer science Roope Raisamo of being flexible with course schedule.

REFERENCES

1. Blair MacIntyre and Steven Feiner *Future Multimedia User Interfaces.* Columbia University 1996
2. Jie Yang, William Holtz, Weiyi Yang and Minh Tue Vo 1998. *An Adaptive Multimodal Interface for Wireless Applications* International Symposium on Wearable Computers IEEE ISWC '98, pages 172-173
3. Gregory D. Abowd, Anind K. Dey, Robert Orr, and Jason Brotherton 1997. *Context-Awareness in Wearable and Ubiquitous Computing* International Symposium on Wearable Computers IEEE ISWC '97, pages 179-180
4. A.K. Dey, L.D. Catledge, G.D. Abowd & C. Potts. *Developing voice-only applications in the absence of speech recognition technology.* Technical Report GTI-GVU-97-06, GVU Center, Georgia Tech, March 1997. http://www.cc.gatech.edu/fce/savoir/pubs/savoir.html
5. Chandra Narayanaswami and M.T. Raghunath 2000. *Application Design for a Smart Watch with a High Resolution Display* IEEE ISWC '00, pages 7-14
6. Kiyokuni Kawachiyta and Hiroshi Ishikawa 1998. *NaviPoint: An Input Device for Mobile Information Browsing* (CHI 98), ACM Press, 1998, 1-8
7. Riku Suomela and Juha Lehikoinen 2000. *Context Compass* International Symposium on Wearable Computers IEEE ISWC '00, pages xx
8. Steven A. Lewis, Gary D. Havey and Brian Hanzal 1998. *Handheld and Bodyworn Graphical Displays* International Symposium on Wearable Computers IEEE ISWC '98, pages 102-107
9. Sevgi Ertan, Clare Lee, Abigail Willets, Hong Tan and Alex Pentland 1998. *A Wearable Haptic Navigation Guidance System* International Symposium on Wearable Computers IEEE ISWC '98, pages 164-165

Simple Operation email composing system without keyboard and mouse: With Multi-modality Input Methods

June Miyazaki
Department of Computer Science
University of Tampere
P.O. Box 607 (Kehruukoulunkatu 1) FIN-33101 Tampere, Finland
E-mail: *june.miyazaki@uta.fi*

ABSTRACT
The popularity of the internet connection make possible for using electronic mail system as wide spread communication tool. However, current email software is based on keyboard and mouse input design for composing a message. It would be hard for a computer illiteracy to input text from a keyboard and operating a mouse. The design of this software is focused on basic email function with supplemental alternative device without a keyboard and mouse input, which it had not found in current commercial products. This paper discusses the potential demands of user interface and describes this software design.

KEYWORDS: multi-modality, simple email system, simple operation, touch screen, designed simply enough for novices.

INTRODUCTION
Email system had been booming among PC user for in these days. Most of non-technical background people use PC because of Internet connection and composing email. Even an end user, who is totally computer illiteracy motivated to purchase a PC just because them. To consider that fact, email system is the most wide spread software application and must be fast and easy access design.

The fundamental function of email program consists of composing, sending, saving, receiving a message and address book. Focusing on the fast and simple operation for novice, speech recognition and feet device feature is added in addition to touch screen input.

Speech recognition and feet device are used for alternatively with simple operation. Both devices are usually not required to practice to use for simple task and help fast text input. Reducing hands fatigue is the one of negative impact of touch screen technology.

The touch screen input takes place main text input method because it is considered widely applied for novice system, e.g. kiosk system in general. It removes stigma for novice, which fear of using keyboard and mouse.

This study focus on the text input part without keyboard and mouse only in this application. Composition of message plays a main role of email system. Entering an application point assumes to be carried out normal mouse operation.

PREVIOUS WORK
The touch screen with email system are carried out information kiosk by data-sphere (this case is SMS) and i-table!® public browser interface (PBI). That software is part of touch screen hard ware that is built in application, but it is not independent software for ordinary PC.

i-table!®

The i-table!® product is an all-in-one touch screen computer software package including Internet connection because of characteristic of kiosk information system. It consists of five applications such as web browser, email, newsgroups, word processor and telnet. The smallest kiosk information system that has targeted on public space such as Coffee and Espresso Shops, Taverns, Juice and Soup Bars, Hotel and Airports, Restaurants, Hospital waiting rooms, Bookstores, Libraries and anywhere there is a payphone.

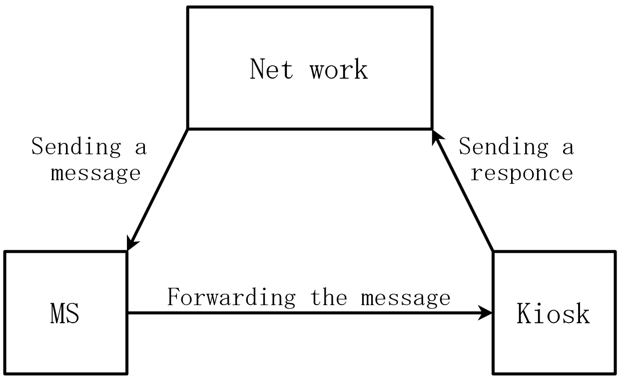
The BasicWedge™ models make very small footprint Internet Portals. These are complete Windows® computers, not just the touch screen like some other industrial systems.

They designed to get attention to public because of their unique shape and small size. The advanced touch screen interface is fun and easy to use for first time web surfers. As a result, these models do not require a mouse, and are easier to learn for first-timer users.

If BasicWedge™ models is purchased with i-table!® PBI software, Advanced Internet Access LLC, an independent spin off company from Evolution Design Engineering LLC (EDE), will install, integrate, and test out the entire system prior to shipment. No keyboard is necessary for most kiosk applications. However, units that use i-table!® PBI software for email and/or data gathering, should have a keyboard attached. In this sense, it just ordinary mailer software.

Data-sphere

Data-sphere's SMS service is forwarding the message and print out them. The user receives a message, they need to approach an SMS kiosk and forward the message to the telephone number given on the kiosk. This kiosk can be located anywhere the advertiser or service provider chooses. One receipt of the forwarded message, the kiosk checks with the network control center whether the message is legitimated and if it is, prints out her relevant paper media. As apart of this checking individual user can be identified and their response to the campaign accurately recorded. Printing a bar code on the paper media can enhance this information.



The text message is just the start, once the user is at the kiosk a whole range of multimedia applications can be used to entertain and to inform them. Touch screen interfaces, access to a businesses website, product information all can be used to enhance the relationship between a company and the members of their SMS customers.

MOTIVATION

Email Position in the society

It is necessary to discuss the importance of email software for the public use for these days. Various academic, government, and industrial organizations to provide access to software and related resources over the past several years have developed network-accessible repositories. Allowing distributed maintenance of these repositories while enabling users to access resources from multiple repositories via a single interface has brought about the need for interoperation. I believe strongly in the value of electronic mail in both corporate and personal domains. Email is cheaper and faster than a letter, less intrusive than a phone call, less hassle than a FAX. Using email, differences in location and time zone are less of an obstacle to communication. There is also evidence that email leads to a more egalitarian information structure. Because of these advantages, email use is exploding. By 1998, 30% of adults in the United States and Canada had come on-line.

Electronic communication, because of its speed and broadcasting ability, is fundamentally different from paper-based communication. Because the turnaround time can be so fast, email is more conversational than traditional paper-based media.

In a paper document, it is absolutely essential to make everything completely clear and unambiguous because your audience may not have a chance to ask for clarification. With email documents, your recipient can ask questions immediately. Email thus tends, like conversational speech, to be sloppier than communications on paper.

This is not always bad. It makes little sense to slave over a message for hours, making sure that your spelling is faultless, your words eloquent, and your grammar beyond reproach, if the point of the message is to tell your co-worker that you are ready to go to lunch.

However, your correspondent also won't have normal status cues such as dress, diction, or dialect, so may make assumptions based on your name, address, and - above all - facility with language. You need to be aware of when you can be sloppy and when you have to be meticulous.

Email also does not convey emotions nearly as well as face-to-face or even telephone conversations. It lacks vocal inflection, gestures, and a shared environment. Your correspondent may have difficulty telling if you are serious or kidding, happy or sad, frustrated or euphoric. Sarcasm is *particularly* dangerous to use in email.

Another difference between email and older media is that what the sender sees when composing a message might not look like what the reader sees. Your vocal cords make sound waves that are perceived basically the same by both your ears as your audience's. The paper that you write your love note on is the same paper that the object of your affection sees. But with email, the software and hardware that you use for composing, sending, storing, downloading, and reading may be completely different from what your correspondent uses. Your message's visual qualities may be quite different by the time it gets to someone else's screen.

Thus your email compositions should be different from both your paper compositions and your speech. Computational email -- the embedding of programs within electronic mail messages -- is proposed as a technology that may help to solve some of the key problems in deploying successful applications for computer-supported cooperative work. In particular, computational email promises to alleviate the problem of remote installation at separately-administered sites, the problem of getting users to "buy in" to new applications, and the problem of extremely heterogeneous user interaction environments. In order for computational email to be practical, however, key problems of security and portability must be addressed, problems for which this research offers new solutions. further work and application development are needed.

Email Overloard

With the growing use of electronic mail come challenges in how to aid users in handling ever-increasing volumes of email. Firsthand experience as well as the systematic study of users' email patterns suggest that a large fraction of messages are parts of larger transactions. It could be implemented an approach toward structuring messages that is intended to help users carry out some of these transactions. The approach has been to make message structure both general-purpose and optional. It could support the applicability and the acceptance of the messaging model. The system, which provides capabilities that are largely absent from conventional email systems, has been deployed in an internal trial.

Desigin Facts

There are some studies done comparing two different designs for a spoken language interface to email. It compares a mixed-initiative dialogue style, in which users can flexibly control the dialogue, to a system-initiative dialogue style, in which the system controls the dialogue. The results show that even though the mixed-initiative system is more efficient, as measured by number of turns, or elapsed time to complete a set of email tasks, users prefer the system-initiative interface. It has been posited that these preferences arise from the fact that the system initiative interface is easier to learn and more predictable.

Becaouse of that, we could consider two factors about user interface design. First, the day of the GUI is drawing to a close. Second, many visionaries have argued that the new user interface will be a direct and delegate interface. But that's wrong.

The coming interface is one in which the user collaborates with the computer. The computer understands what the user is doing, can take part in those activities and is able to respond conversationally to the user's activities. This requires an interface that not only understands the user's individual utterances but also can participate in a conversation. Because conversations are fundamentally about the purposes for which people participate in the conversation, this new interface will also require that the machine understand and model purposes behind conversation.

During this talk we will demonstrate new interfaces, some with speech, that participate with users in collaborations about doing email. We will use these demonstrations to illustrate how conversation and tasks can play a role in user interfaces. We will also demonstrate instances where spoken conversational interaction is more efficient than GUI interaction. For example, Email is still the main communications software outside research labs.

Conventional wisdom inside human factors circles says that the integration of user interface design processes into the software development cycle is the best way to improve the usability of software products. While there is no problem convincing human factors practitioners of this, frequently there is still a need to demonstrate the effectiveness of user interface processes to product development teams and management. Mayhew (1992) suggests that it is not enough to be able to apply human factors knowledge. Successful user interface design must include buy-in from outside of the user interface organization. To demonstrate the effectiveness of a user interface design program, data from usability tests on three versions of a product were analyzed. The oldest version of the product was developed without the inclusion of any user interface design processes. The second version of the product had minimal involvement of user interface practitioners late in the development cycle. The newest version of the product was developed with the user interface design processes fully integrated into the software development cycle. The data indicate that user interface design processes do impact usability, as measured by speed, accuracy, and subjective measures. Furthermore, user interface processes which are part of the software development cycle, as opposed to just a side effort by user interface practitioners, seem to have a much greater impact on usability.

Universal usability is currently impeded by system complexity and poorly-crafted interfaces which lead to confusion, frustration, and failure. One of the key challenges is the gap between what users know and what they need to know. This paper describes and presents early results from three related research projects designed to identify and close this gap and to examine how users might learn what they need to know.

Learning disabilities involve a disruption to one or more aspects of cognitive processing. Because CHI relies so heavily on cognitive processes for interface designs, the learning disabilities area presents challenges to the interface designer. Individuals with learning disabilities score in the bottom 5th percentile on one or more cognitive dimension involving perception, memory, concentration, problem solving, functional integration, etc.

Computer software generally, and personal productivity tools in particular, take on a special significance for individuals with disabilities. Appropriately designed software can reduce the impact of a disability, assisting with activities which other people take for granted.

When writing or reading on paper, we usually have a robust perception of the text as a spatial object with inherent structure. By a quick visual inspection of a book in our hands, and by flipping the pages for a few seconds, we get a preliminary feel for the size, structure and content of the text material. Not only are we guided by those physical cues in the process of approaching a new text, they also enable us to remember the text by its appearance and spatial arrangement.

In contrast, during on-screen writing and reading with a word processor, users often lack a global perspective of the text. In fact, the use of word processors has been shown to cause problems for writers in reading and evaluating long documents on the screen. The word processor is usually used on a small screen, showing only a very restricted part of the text at a time. Moreover, when the user makes revisions or shifts position in the text, the location of the text relative to the screen window varies. This contributes to writers lacking an adequate "sense of the text" when writing a long document.

A History of Wordprocessor

The digital computer is often characterized as a "symbol manipulator." This definition, which emerged early in the history of computing, applies well to the most important uses of the machine from its invention in the 1940s into the 1980s. At first the symbols that the computer manipulated were numbers. But as early as the 1950s, business and government began to use the machine to store and retrieve names, dates addresses, and so on. In the 1950s, too, the artificial intelligence (AI) movement began, and, although the movement did not achieve its stated goals, it did make an effective case for the paradigm of symbol manipulation. AI investigators (Simon, McCarthy, Minsky, and others) insisted that all important knowledge could be represented and generated through a calculus of discrete symbols. By writing programs to solve problems, prove theorems, and process natural language, they broadened our understanding of what computers can do. The great popularization of this technology came of course in the 1980s with the personal computer, the word processor, and the electronic spreadsheet. Word processing in particular is trivial symbol manipulation, yet it has been perhaps the single most influential application. Word processing has made the computer indispensable for any organization and for most individuals who write. Furthermore, word processing is now combining with textual databases, communication networks, and hypertext to create a more challenging symbolic environment.

User design in Word processor

End-user programming involves the end user building new tools, not simply using an application. Hence, word processing is not an example of end-user programming while building style sheets for a word processor would be. Using communication software is not, writing a script for the communication software is. Using someone else's spreadsheet is not, building your own spreadsheet is. Using someone else's HyperCard stack is not, building your own is. Running someone else's cognitive model is not, building a cognitive model that fits your theory is. Studies of people learning to use contemporary word-processing equipment suggest that effective learning is often "active," proceeding by self-initiated problem solving. The instructional manuals that accompany current word-processing systems often penalize and impede active learning.

Throughout the history of human-computer interface development, one aspect has remained constant: output from computers has been almost entirely visual.

A good deal of research in cognitive psychology has demonstrated that, although learners can solve problems that are just like the ones they have been trained on, they often have great difficulty solving new types of problems. People also have difficulty trying to understand instructions or training materials that try to teach a procedure at a level that is general enough to apply to many different kinds of cases. These two findings lead to a quandary for people designing instructions for procedural tasks such as operating computer software: Instructions should be written with a good deal of specificity so that new users can understand and use them right away, but at the same time the user will have great difficulty generalizing what they have learned in novel cases.

Simple operation is important

Inconsistent user interfaces force users to learn different operations to accomplish common user goals. Such inconsistent knowledge is represented as two or more rules with common conditions and different actions. Let ⅰ) represent the common condition and ⅲ) and ⅳ) represent different actions. The rules can be described as ⅰ)−ⅱ) and ⅰ)−ⅲ) which conform to a classical interference paradigm.

Sixty subjects were trained to perform seven utility tasks on a popular, stand-alone, menu-based word processor. Subjects were brought back one day after the training session for a retention test. The retention test involved retraining subjects on the same set of utility tasks.

Production rule models were written for all tasks. Each rule represents a step in one of the seven tasks. The rules were classified as ⅰ)− ⅱ), ⅰ)−ⅲ), or ⅲ)−ⅳ) rules. ⅲ)−ⅳ) rules are rules with a unique condition and action. The rule that appeared more frequently was predicted to interfere with the retention of the less frequent version of the rule.

Fear to mouse and keyboard

In recent years, a body of literature has developed which shows that users' perceptions of software are a key element in its ultimate acceptance and use. It focuses on how the interaction style and prior experience with similar software affect users' perceptions of software packages. In previous studies’ experiment, direct manipulation, menu-driven and command-driven interfaces were investigated. It studied users' perceptions of the software in two hands-on training sessions. In the first session of that study, novice users were given initial training with word-processing software, and in the second session the users were trained on a word processor which was functionally equivalent to the prior one, but had a different interaction style. In the initial training session, it has been found that the interaction style had a reliable but small effect on learners' perceptions of ease of use. The direct manipulation interface was judged easier to use than the command style. The interaction style, however, did not affect learners' perceptions of the usefulness of the software. In the second training session, subjects who had used a direct manipulation interface in the first session learned either the menu-based or command-based software. The perceptions of these users were compared to those of learners, who had used the menu or command software in the initial training session. It found that both interaction style and the prior experience with a direct manipulation interface affected perceptions of ease of use.

Kiosk system as simple operation

A multimedia tool developed for scaffolding constructive conversation and sharing information by means of a public kiosk. The Multimedia Forum Kiosk (MFK) provides an environment where users communicate asynchronously with video, audio, and text. Unlike unstructured media such as entail, the interface provides multiple representations of the structure of the discourse which aid in understanding the previous discussion, eliciting and refining new ideas, and developing a sense of community with other users. The software has undergone evaluation, testing, and revision as a tool for an education research community. Preliminary results indicate that users learn the interface unproblematically without training, and that they successfully explore and contribute to the discussions. The MFK has introduced as a tool for collaborative discussion and learning, and discuss several potential uses for the tool, both pedagogical and utilitarian. However, more formal testing plan to evaluate the software and interface design is underway.

The HyperView, an authoring system which involves users in the creation process directly by providing coaching tools from within the Help facility to support their development of interactive learning modules for business and educational instruction. The argument of Help system in the tool permits users to understand system options and to see the instructive possibilities of computer-aided learning (CAL) applied to the subject matter itself. Users of CAL materials become authors and participate in the development of the modules from which they are learning. It had been developed a series of help options which provide information, examples, instructions, tutorials and, finally, coaching to permit users, from novices to experts, to design and modify materials which can then become parts of the next user's learning experience. In this sense, the courseware and authoring system are quite different from conventional 'kiosk' models, instructional sets or testing courseware. HyperView permits authors to develop materials under all three of these structures, but it is a more comprehensive, user-centered system, adaptable to the various learning strategies of its operators.

Simple userbility - Acess to ease

In spite of this, a greater proportion of texts were composed directly on the screen, without a manuscript, and the general tendency is that less prepared manuscripts are used. This can be seen as a potential conflict in computer-based writing which poses a challenge for system design as well as for writers in their choice of strategies.

In learning to use software, people spend at least 30% of their time on dealing with errors. It could therefore be desirable to exploit users' errors rather than to avoid them.

Touchscreen operation

We can describes an approach to design user friendly interfaces for computer based systems, especially for public use. The iterative prototyping process may not be genuinely new, however some of previous studies presents our experience with this approach. The task was to design a user-interface for a self-service terminal for train tickets, with an underlying fare system which was very complex. The system was to use a touchscreen as input media. The design process was divided into the steps problem analysis, identification of archetypes and prototyping. Prototyping was further divided into the three stages: Archetypes, screen layout and a functional model of the complete system. User tests in all prototyping phases showed that known ways of interaction are preferred, that a colourful screen is accepted better by users, and that a linear path through a program is initially better for untrained users, but is judged to be clumsy and slow for repetitive use or for experienced users. In general, the approach with small steps, involving users at several stages with tests has shown its advantages. The results of the tests are easier to interpret, as they are embedded in the whole process of development.

Touchscreen as simple operation

This study presents evidence that a prototype touch interface technology emulating basic interaction techniques of a mouse-pointing device is comparable in overall usability to a conventional mouse for a direct manipulation, graphical windowing software environment. The touch technology prototype involves using either a stylus or finger, with an overlay sensitive to changes in capacitance. Users practiced each technique (mouse, stylus, finger, keyboard with no mouse), in the context of carrying out office-related tasks on the first of a two-day study, and then eight similar test tasks on the second day, in a completely within-subject design. Significant effects for time on task were found for Techniques and Tasks for five practice tasks on the second day of the study. The clearest significant effect was that the stylus technique was faster than the keyboard. A qualitative analysis of errors indicates that there were problems with the precision of pointing using the finger, and to a lesser extent the stylus and mouse. User comments and ratings indicate that the stylus and mouse were preferred comparably, and were preferred to the finger and keyboard techniques.

Touchscreen as multimodality design

A Thanks to recent scientific advances, it is now possible to design multimodal interfaces allowing the use of speech and gestures on a touchscreen. However, present speech recognizers and natural language interpreters cannot yet process spontaneous speech accurately. These limitations make it necessary to impose constraints on users' speech inputs. Thus, ergonomic studies are needed to provide user interface designers with efficient guidelines for the definition of usable speech constraints.

We evolved a method for designing oral and multimodal (speech + 2D gestures) command languages, which could be interpreted reliably by present systems, and easy to learn through human-computer interaction (HCI). The empirical study presented here contributes to assessing the usability of such artificial languages in a realistic software environment. Analyses of the multimodal protocols collected indicate that all subjects were able to assimilate rapidly the given expression constraints, mainly while executing simple interactive tasks; in addition, these constraints, which had no noticeable effect on the subjects' activities, had a limited influence on their use of modalities.

These results contribute to the validation of the method we propose for the design of tractable and usable multimodal command languages.

Using speech recognition

A number of human-computer interfaces have been developed and adapted into an auditory form, based on the use of synthetic speech. However, for modern interfaces that use more complex displays, synthetic speech is not sufficient.

Two studies were performed to test the efficacy of using three different automated speech recognition devices in parallel to obtain speech recognition accuracies better than those produced by each of the individual systems alone. The first experiment compared the recognition accuracy of each of the three individual systems with the accuracy obtained by combining the data from all three systems using a simple "Majority Rules" algorithm. The second experiment made the same comparison, but used a more sophisticated algorithm developed using the performance data obtained from experiment 1. Results from the first experiment revealed a modest increase in speech recognition accuracy using all three systems in concert along with the Simple Majority Rules (SMR) algorithm. Results from the second experiment showed an even greater improvement in recognition accuracy using the three systems in concert and an Enhanced Majority Rules (EMR) algorithm. The implications of using intelligent software and multiple speech recognition devices to improve speech recognition accuracy are discussed.

THE SOFTEARE DESIGN

Because of the previous discussio of this paper, the software is designed by using touch screen as a main text imput and speech recognizer and foot device as a alternative input assisstance.

The Tablet Look

As Figure 1, there is a input tablet on the screen. It goes and recurrents next character selection by pressing left foot or saying “Next” to microphone. (For example, if the character set is alphabet, it goes A-I to J-R to S-Z)

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	⊔	**Clr**	↑	→
J	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**		**Op**	↓	←

J	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	⊔	**Clr**	↑	→
S	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**			**Op**	↓	←

Figure 1:

As Figure 2, it goes and recurrents next character set by pressing right foot or saying “Change” to the microphone. (For example, if the character set is alphabet , it goese A-I to 0-9)

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	⊔	**Clr**	↑	→
J	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**		**Op**	↓	←

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	⊔	**Clr**	↑	→
										Op	↓	←

Figure 2:

The Clr represents ‘Clear’ and equal function as ‘Back space’ in normal keyboard. The ‘Op’ represents ‘Option’ to send a message, retrieve etc.

The Screen Look

As Figure 3, it turns blue when a character or a tablet area has been selected. The cursor position can move to left, right, up and down direction by pressing each corresponding tablet. Single press causes one space movement and long press cause continues movement.

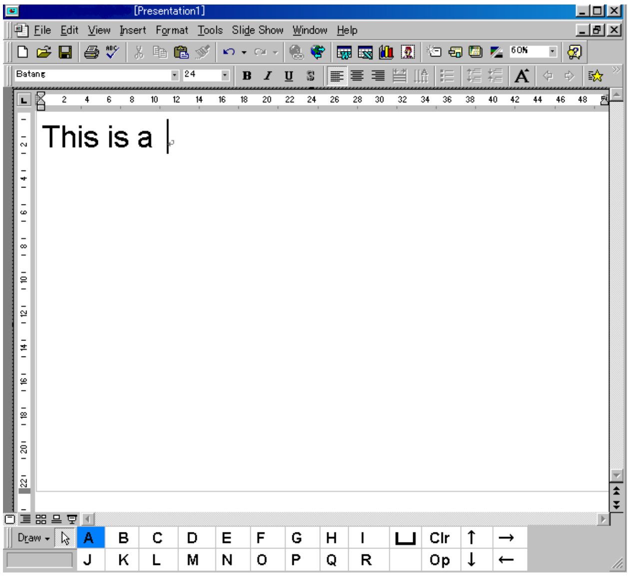


Figure 3:

CONCLUSIONS AND FUTURE WORK

We have discussed the design of touchscreen with alternative suplument textinput, whcich is speech recognition or feet device, no mouse and keybpard email software. But that it is difficult to use and there are significant research questions still to be addressed. For example, chagnig the size of table in a option menu is one erea. It delays the speed of response to calculate net coordinagte positoning rather than in calculating the position in the same There is two directions for appling to no mouse and keybord text input method. The one is applying for land-off selection method and develop a standard word processor (e.g. MS word) for PC. The other are mobible devices to keep the characteristic of simple software.

For the word processor development, the cost of hardware, touchscreen disply is the key issue. If the cost of diplay will be less expensive enough for individuals, it would be a big potential to use speech recognizer and feet device as alternative suplumetal device to help complicated task in those software. It could be used for take-off strategy to provide continous feedback about cursor location, allowing the user to postioon the cursor before a selection is made by lifting the finger for high resolution tasks in those software.

For mobile devices, there are several option such as Personal Digital Assistance (PDA) and mobile phones. It will be used new device as well in PC based software; however, the characteritic of mobile devices simple software requiredment makes no need of suplumental input device. It focuses on the simple operation and fast feedback rather than enrichied functions in device. The first-contact strategy is the best candidate to provide fastest speed by first contact selection with any target.

ACKNOWLEDGMENTS

The authors would like to acknowledge the contributions of many previous editors in the writing and formatting of this document. This document is based on the CHI'94 formatting guidelines.

REFERENCES

1. Brems, D. J., Rabin, M. D., Waggett, J. L., Using natural language conventions in the user interface design of automatic speech recognition systems. Human Factors 37(2), pp265 - 282, June 1995.

2. Bruce, I., Voice guidance for interactive speech systems. Voice systems worldwide 1990. Official proceedings of the international voice applications conference and exhibition, pp115 - 119, 30 May - 1 June 1990.

3. Clark, L., The use of scenarios by user interface designers. HCI'91 conference proceedings, 20-23 August, Diaper D. and Hammond, N. (eds.), Cambridge: Cambridge University Press, pp103-115, ISBN 0-521-416949-9, 1991.

4. Clarke, A. M., Allison, G., Carter, C. J., Graham, R., Maguire, M. C., Parker, C. G., Pomfrett, S. M. and Stapleton, L. J. (eds.), *Human Factors Guidelines for Designers of Telecommunication Services for Non-Expert Users. Volumes 1 and* 2, ISBN 1 901267 00 8, HUSAT Research Institute, 1996.

5. Daly-Jones, O., Bevan N. and Thomas, C., *Handbook of User-Centred Desig*n, EC Deliverable 6.2.1, Telematics Applications Programme, Project IE 2016 INUSE, NPL Usability Services, National Physical Laboratory, Queens Road, Teddington, Middlesex, TW11 0LW, UK, 1997.

6. Daniel, F. and Krueger, H., Designing the look, INTERCHI'93 (adjunct), pp 47-48, 1993. Eason, K. D., A task-tool analysis of the manager-computer interaction. In B. Shackel (ed.) Man-computer Interaction, Amsterdam: Sijthoff and Noordhoff, 1981.

7. Gagnoulet, C. (1989), Voice replaces dial in new public phones. International Voice Systems Review 1(1), pp1-39 Jan/Feb, 1989

8. Gill, J., Access Prohibited, Information for Designers of Public Access Terminals, *Royal National Institute for the Blin*d, 224 Great Portland Street, London W1N 644, UK, ISBN 1 86048 014 1, 1997.

9. Gould, J. D., Boies, S. J., Levy, S., Richards, J. T. and Schoonard, J., The Olympic Message System: a test of behavioural principles of system design'. *Communications of the AC*M, 30,9, pp758-769, 1987.

10. Helander, H. G., Design of visual displays. In G. Salvendy (ed.): *Handbook of HumanFactors.* New York: McGraw-Hill, 1987.

11. Lang, Kathy and Mack, Robert A Benchmark Comparison of Mouse and Touch Interface Techniques for an Intelligent Workstation Windowing Environment Computer Systems: Input Device Comparisons Human Factors Perspectives on Human-Computer Interaction: Selections from the Human Factors & Ergonomics Society Annual Meetings 1983-1994 / Gary Perlman / Georgia K. Green / Michael S. Wogalter 1995 p.172-176 Santa Monica, California Human Factors and Ergonomics Society

12. Maguire, M. C., ISLA — an experimental local information system for the general public, Information design journal, vol. 3/3, pp 198- 205, 1983.

13. Maguire, M. C., TIER 2 Evaluation Report, EC Telematics Applications Programme PD Web project, WP08, HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leics. LE11 1RG, UK. September 1997

14. Mamdani, S., Barton, G., Poulson, D, Allison, G., Ashby, M., and Maguire, M., ATTACH interface guidelines, Internal document, EC TAP ATTACH project internal document, 1997.

15. May, A. J., Development of generic user requirements for interactive public information systems, Master of Science thesis in Ergonomics, Faculty of Science, University College, London, 1993.

16. Noirhomme-Fraiture, M. and Vanderdonckt, J. M., Screen usability guidelines for persons' with disabilities, INTERCHI'93 (adjunct), pp 25-26, 1993.

17. Peltu, M., Power to the people, Computing, 9th June 1994.

18. RACE Project 1065 - ISSUE Usability Guidelines, Volume III: Human Factors Guidelinesfor Multimedia, HUSAT Research Institute, Loughborough, 1992.

19. RNIB, 1996, Smart cards - Interfaces for people with disabilities, SATURN project brochure, (ed.) Gill, J., Royal National Institute for the Blind, 224 Great Portland Street, London W1N 6AA, UK. ISBN 1 86048 007 1.

Conducting a Wizard of Oz test on Ubiquitous Computing system Ovimies

Kaj Mäkelä
Human Computer Interaction Unit
Department of Computer and Information Sciences
University of Tampere
FIN-33014 University of Tampere
+358 3 215 8873
kaj@cs.uta.fi

Esa-Pekka Salonen
Human Computer Interaction Unit
Department of Computer and Information Sciences
University of Tampere
FIN-33014 University of Tampere
+358 3 215 8873
Esa-Pekka.Salonen@uta.fi

ABSTRACT
This paper describes the Wizard of Oz test conducted on Ubiquitous Computing system 'Ovimies'. Some experiences and viewpoints on Wizard of Oz testing are presented.

Keywords
Wizard of Oz, ubiquitous computing, natural language dialogue. Evaluation, speech user interfaces

INTRODUCTION
The ubiquitous computing applications have broken a traditional desktop paradigm of computing. The users utilises daily, not one but several different computers distributed to their every day environment. The applications are used in every day situations in different circumstances. The equipment and applications can be mobile or in the environment.

The use of the computer services is not restricted by the limitations of a desktop computing and the context of use is therefore more informal and unpredictable. The main characteristic of ubiquitous computing application is that it is invisible to the user. One way to make this possible is that the user does not need to give the system explicit instructions, but the system has initiative and it is able to recognize the needs of the user from the context of the action. [6].

Because of the nature of the ubiquitous computing, the evaluation of the ubiquitous computing systems cannot be done in laboratory environment. Testing has to be done in the actual scene of action with real life problems, because the situation is created by the context.

The Wizard of Oz (WoZ) is an experimental method for testing computer user interfaces to support design process and to evaluate the interface [1]. It has been used to test natural language dialogue systems [2] and multimodal systems [5]. This paper describes a Wizard of Oz test conducted on the speech based Ubiquitous computing system called Ovimies.

The paper is constructed as follows. First, the description of Wizard of Oz testing technique is provided. The following sections will give information on the Ovimies system and the setting of the test conducted for the system. The results of the test are given and discussed. Lastly, some concluding remarks are provided.

WIZARD OF OZ TECHNIQUE
The Wizard of Oz testing is an experimental user interface evaluation method in which the user of the system is made to believe that he or she is interacting with a fully implemented system, thought the whole or part of the interaction of the system is controlled by human, a wizard. The interaction is logged and/or recorded for further analysis. [1]

The Wizard of Oz testing is used to evaluate interaction design and natural language models before or after they are actually implemented. The testing can therefore also support iterative design process.

The Wizard of Oz testing is suitable for relatively narrow and well-defined application domains in which the application is performing behaviour that can be performed better by a human. [1]

Why to use Wizard of Oz Technique?
The human-human communication differs from human-computer communication. [1] The findings received from human-human communication research cannot be directly applied to the human-computer communication. Therefore, to gain reliable information about human-computer communication it is relevant to observe the human behaviour while interacting with the computer system. It is important that the user thinks she is communicating with the system, not a human [2].

It has been proven that the human adapt their way of communication to the characteristics of the receiver. This means that the person changes the way they speak by adjusting their communicational behaviour to match with the responder's behaviour, e.g. intonation and prosody or nonverbal gestures. This has also been proven to happen with computer systems. [2]

The Wizard of Oz technique can be applied in development stage of the system to get information on the functionality of the interface design. The method has been applied in the research of natural language dialogue [4] and multi-modal [3] systems.

DESCRIPTION OF THE SYSTEM

'Ovimies' is a Ubiquitous Computing system developed in the University of Tampere in the Department of Computer and Information Sciences in Human-Computer Interaction Unit (TAUCHI) to help the members of the staff and the visitors of TAUCHI in they communicational tasks and everyday lives. This is done by automatically:

- opening the door to identified staff members and visitors, if the target of their visit is recognised,
- guiding visitors in TAUCHI premises to the person or the place they are seeking and
- giving the staff members information about interval organisational messages, their personal communication, e.g. e-mail, icq, phone calls or the visitors.

'Ovimies' uses natural language to communicate with the users. Speech recognition is used as an input and speech synthesis as an output method. The target of the visitors visit is recognised from their speech. The staff members are identified by recognising their name.

'Ovimies' uses also pointing gesture together with synthesised speech output when guiding the visitor to the target of the visit. The guidance is given by using an anthropomorphic robot pointing the direction the user should go to find the target.

The functions of the system can be divided to following stages:

1a) Recognition of the staff member or

1b) Recognition of the target of the visit,

2) Opening the door and

3a) Giving information about new e-mail to the staff member or

3b) Guiding the visitor to the target of his/her visit which can be a person or a room.

The Ovimies system is based on a distributed architecture called 'JASPIS' developed in TAUCHI. The JASPIS was originally designed for speech applications, but has been expanded to consist features, which support also a development of Ubiquitous Computing applications.

The JASPIS easily enables a Wizard of Oz testing because of its modular manager and agent based structure. Each of the components of the JASPIS architecture can easily be replaced with a wizard.

Dialogue model

In the current dialogue model the system prompts are formed to guide the user to answer shortly by stating the name of the person or the room he or she is searching. It is assumed that the staff members will not push the doorbell but say a greeting and their name.

The structure of the dialogue is presented below.

A. Staff Members

1 STAFF: (doorbell button is pushed or go to 3)

2 OVIMIES: "What is the name of the person or the room you are searching?" // Target request

3 STAFF: "John Doe here, hello. Could you open the door?" // Name and greeting

4 If the name is recognised, //confirm

OVIMIES: "Good morning, John Doe. I will open the door for you."

If the name is not recognised, go to C1

5 (Door is opened)

6 OVIMIES: (inside) "Good morning, John Doe."

B. Visitors

1. VISITOR: (doorbell button is pushed or go to 3)

2. OVIMIES: "What is the name of the person or the room you are searching?" // Target request

3. VISITOR: "John Doe" or "usability lab" // Name of the target

4. If the name is recognised,

OVIMIES: "Good morning. I will open the door for you."

If the name is not recognized,

go to C 1.

5. Door is opened.

6a. If the target is a person,

OVIMIES: "Good morning. The person you are searching is in room 444. To find there…."

6b. If the target is a room,

OVIMIES: "Good morning. To find your way to the usability lab, which is the room number 412, go…"

C. Errors

1 OVIMIES: "I am sorry, I did not understand. Say the name of the person or the room you are searching for."

2 VISITOR: --- // statement that cannot be recognised

Following errors are responded:

3 OVIMIES: "Say the name of the person or the room you are searching for."

Stage 3 is repeated three times. After this:

4 OVIMIES: “I am sorry, I cannot open the door. Use the key or push the doorbell button within 15 seconds.”

DESCRIPTION OF THE TEST

The aim of the study was to test and analyse the natural language and multi-modal dialogue model designed for the system before constructing the actual speech recognisers.

The study consisted of two parts:

1) The use of natural language in stating the target of the visit. The interesting point was to find out what kind of language users actually use when talking to a computer system and how the form of the speech output affected on the language used to answer to the system. This is useful information for the design process of the vocabulary and the grammar of the dialogue.
2) Combining synthesised speech and pointing gestures in route guiding. The interesting point was to find out how the route to the target of the visit should be given to the user so that the user would find the way to the target. This contains prosody and timing of the speech and synchronisation of the movements and instructions.

The aim of the test was to recognise the actual behaviour of the user and the problems occurring in following situations:

- the user understanding the question given by using synthesised speech,
- the visitor responding to the question and stating the target of the visit,
- the staff member declaring his/her identity,
- the behaviour of the user while entering the premises and
- the visitor understanding and responding the guidance given by the system.

METHODS AND TOOLS

The Wizard of Oz test was conducted by replacing the forthcoming speech recognition module of the ‘Ovimies’ system with a text input device used manually by the test group member observing the situation. The reactions and the speech of the user were recorded and the system tasks were logged for analysing.

The test group member, ‘the wizard’ listened constantly to the voice input gathered with the microphones installed to the premises. When a user pressed the doorbell button or took the initiative by talking, the wizard interpreted and conveyed the input to the system.

User interface of the testing tool

We implemented a tool for the wizard to give speech recognition information manually to the system (Figure 1). The testing tool was designed as simple as possible to enable short response times and minimise the possibility for error. The wizard control interface was implemented to Java applet form in order to enable flexible changes in test settings.

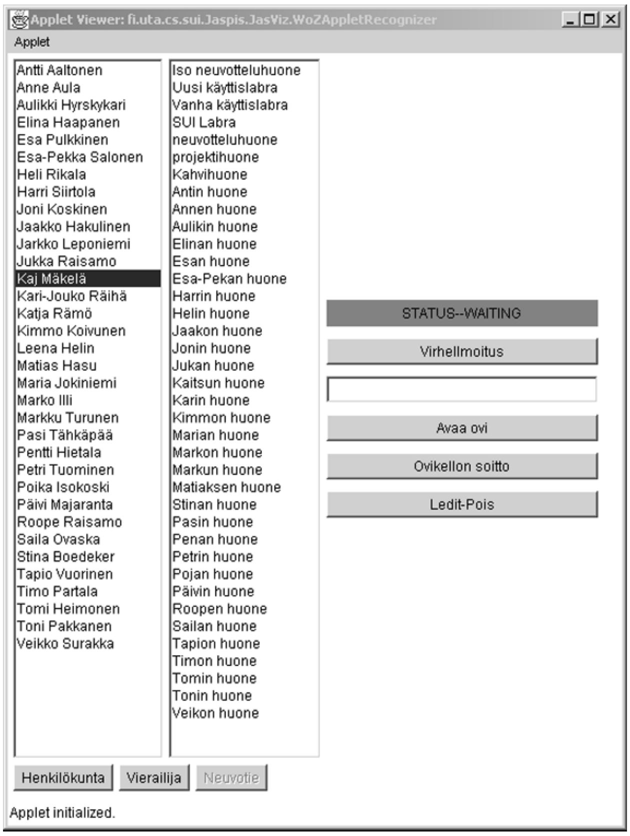


Figure 1. The Interface of the Wizard tool.

The tool provides a simple list based interface consisting all the possible options of the speech recognition results. The tool converts the chosen option to the string form and sends it to the system. Basically there are three kinds of inputs: the identity of the staff member, the target of the visitor’s visit and the recognition errors.

The interface contains two single-selection lists, seven buttons and one indicator. The user’s identity or the target of the visit is chosen from the lists. There is one list consisting the staff member names and another consisting the room names. The role of the user is chosen with a button. If the user is a visitor, either a person or a room name is chosen. If a person is chosen, a visitor button is pushed to tell the system, that the name is a target of the visit. If a room is chosen, a button for route guiding is pushed. If the user is a staff member, the person name is chosen and the staff button is pushed.

There is also a button, which sends the system a recognition error. This is used when the speech of the user do not contain the information expected. For unpredictable situations wizard can use a button to ring the doorbell or a button to open the door.

There is a indicator giving information about the mode of the system. The indicator is red if the system is waiting for input from the wizard and green if the system is handling the given input. The wizard is also given information of the state of the door and the actual doorbell button via the feedback sounds played at the door. The wizard system can also be disabled via the applet by pushing a button

The testing issues

The users were informed of the system with posters next to the system. They were told that the system uses speech recognition and that it can be bypassed by pushing button three times successively. The staff members were informed via e-mail. The standby mode of the system is indicated with led-light next to the doorbell, so that the users know when to use the system.

The Ovimies system opens the door for the users. The door can be opened also with traditional mechanical keys or with electronic key cards. To make the users to use system they were asked to use the system instead.

Because of the Finnish law it is compulsory to inform the users that their speech and actions are recorded for the research purposes. The permissions of the users were gathered in written form. Because of this, one person of the test group was in the hallway gathering permissions from visitors. The permissions of the staff were gathered before starting the test.

The users were not told that the system was controlled by a human wizard, because the information could have been spread and affected on the results.

The wizard had a possibility to listen the audio input gathered with microphone, synthesised speech generated by the system and the doorbell signal. The same audio information was also recorded for analysis. The testing equipment was situated in the separate room, where the sound was recorded and the wizard system was operated.

Wizard rules

To keep the behaviour of the system consistent and credible we formed a set of rules for the Wizard operating the system. One of the main problems in Wizard of Oz testing is that the wizard has superior knowledge and skills compared to the system and he or she has to reduce skills and knowledge [1]. These rules can help also this problem.

The speech of the user should contain any name of staff member or room in TAUCHI premises, otherwise it will cause a recognition error. However, the structure of the sentence should be commonly used. There should also be only one person speaking at the time.

The staff members are supposed to say also something else than their own name, for example, a greeting of some kind or a request to open the door so that they would be recognised as staff. This rule was made because the visitors express the person they are searching by saying only the name of the person. Therefore there was a need to differentiate the ways to identify and to search a person. It was decided that the person saying the name alone would be identified as a visitor.

Wizard should react to everything that is said at the door, even to speech that is not necessarily targeted to the system. This is to give expression of continuous speech recognition based on the frequency of sound.

RESULTS

Behaviour of the users

There were found three groups of users: the visitors, the students and other staff members and the TAUCHI staff members.

The visitors were coming to TAUCHI to meet someone or to some room in TAUCHI premises, for example, to the meeting room or to the usability laboratory. They did not have key or key card to the premises and therefore they normally used doorbell. The visitors were given a possibility to bypass the system and play the doorbell by pushing doorbell button rapidly three times successively. There were few situations where the visitor did not want to use the system.

All of the visitors came to meet a person, they were not searching for the room. For example, a visitor coming to a meeting held in a big meeting room stated that they were coming to meet a staff member.

Students and other university staff members have a key or a key card they can use to access TAUCHI premises. Therefore they had a possibility to pass the system and they mostly used the possibility despite the request to use Ovimies system. They also know the premises and need no assistance in finding a person or a room. They had not been told about the system before they came to the door, which may have been affected their behaviour.

The staff members of the TAUCHI were told about the system by e-mail. They had also been asked a permission to gather voice samples and log information for their personal profile.

The staff members were not given much instructions on using the system. They were only told to greet and introduce themselves to the system. With this we wanted to see which way they would behave without detailed instructions.

We found out that staff can be divided to three groups from the system use point of view. One group is those using system often, most of the times when coming inside. We call them active users. They were mostly people visiting often outside the premises, for example, for smoking. It is also accustomary to help the fellow researches in their research by voluntarily assisting them to gather data.

TAUCHI is an expert organisation, where all the employees are experts of some area of usability and interfaces. This was also shown in their behaviour. A Part of the active users was testing constantly the ability of the speech recognition by using complex impressions and several users at the same time. However, it was shown that these users learned the restrictions of the system and learned to use the system.

The second group were those who tried the system only once or few times. They were interested to know how the system works, but lost their interest quite soon.

The third group was those who did not use the system at all, but instead used their keys or key cards or the other door to get inside the premises.

In the dialogue model it was assumed, that the staff members are not willing to push the doorbell button in order to get inside. This is why they were given a possibility to introduce them selves in any time the system was in standby mode. However, some of the members of the staff used the doorbell button.

There were several reasons for not using the system. There is a possibility to use keys and key cards to enter the premises. The members of the staff are accustomed to use keys and key cards and often used them, but came back to use the system. There is also another entrance to the premises, which is used by those staff members who have their office on the other side of the premises. The system speech outputs were quite long and there are also delays in the system. Therefore some of the users rather used other methods to enter. The users were also informed that the system is gathering data of their behaviour and they may have been avoiding the system because of this.

Problems and remarks

Although the system was not fully implemented, only a few errors in the system logic were found during the testing. Fortunately they did not have a major effect on the test.

The aim of the system is to serve all the users in some way, at least by asking external help to the situation. However, during the tests the system failed to serve the user in two cases. In one situation the user did not speak to the system at all and did not use the possibility to bypass the system. Instead of using the system the user banged the door.

In the other situation the person the visitor was searching for was not a staff member. In this case the user got frustrated when his speech was not recognized and used his cellular phone to contact the person he was supposed to meet. In this case the user tried three times and stopped after this. The system is programmed so that after the fourth try it will go to a state where the user can push the doorbell button to actually ring the bell inside. This observation leads us to assumption that the system should go to the manual mode after three or already after two failed recognitions.

The other problem in this case was that the system is able to guide the visitor only to the persons listed in the system, who are at the moment the members of the staff. The usability laboratory is however used also by students and other organisations. The test person coming to the test will then be searching for the person conducting the usability test and the system is unable to recognise the person searched. It may also be that the person coming to the test does not even know the name of the person conducting the test or the name of the room the test is held in.

The delays in system response were found irritating and the users having a key or a key card often chose to use one instead of waiting the system to react. This was partly because the user was not sure if the system is processing the input. There was not any indicator showing the current state of the process. A Part of the delays were because of the wizard using the control tool. There are delays also in the system function because of the heavy system event logging and limited hardware resources.

Also the length of the sentences spoken by the system annoyed the users. Especially the users using the system on a regular basis were irritated, because the speech delayed their entrance.

The guide robot was often passed without listening the instructions. The reason was mostly that the person knew already where he or she was going. The other reason was that the robot started the guiding too late and the visitor had already passed. The guidance given by the guide robot was also found too long and slow. The timing, the length of the prompt and the tempo altogether caused that the visitors to ignored the guidance. The length made it also hard to remember the guided route.

Some of the staff members started the dialogue by greeting the system and waited the system to respond before stating their name. This may have been because they wanted to make sure that they are heard and to make sure that the connection with the system is established.

It was shown that the people using the system on more regular basis changed their way of speaking to the system by their former experiences. They learned from their mistakes and adapted their interaction to the system. This is in consistence with the observation made by Tennant (reported in [2]).

The users told in informal conversations that the speech synthesis was unclear and therefore sometimes hard to understand. It has been shown that listening to synthesized speech requires more processing capacity than listening to natural speech before human has encoded the synthesized speech. Therefore it could be assumed that people who had difficulties in understanding synthesized speech simply were not accustomed to hear it (in [6], p.190-191).

Analysis

The testing sessions were recorded and analysed afterwards. The system was used 74 times, of which 22 were visitors and 52 were staff members. In 68.2 percent of the cases the visitors used the system so that they responded to the first prompt in the way they were expected. In 4.5 percent of the cases the user succeeded on the second try and 4.5 percent after on the third try. In 13.6 percent of the cases the user bypassed the system by pushing doorbell three times. 9.1 percents were not able to get in by using the system.

The result shows that the system prompt was formed so that the most of the visitor cases (77.2 percents) the users answered in the way they were expected.

In the dialogue design it was assumed that the staff members would not use the doorbell. In 19.2 percent of the cases the staff members did however use the doorbell and therefore heard the prompt formed for the needs of the visitors.

CONCLUSIONS

The amount of data gathered for this research was quite small and therefore the results cannot be used to make generalisations. More data should be gathered to gain more reliable and significant results. However, the results give some guidelines to following iterations of the design process.

The results show that some consideration should be focused on the form of the system speech output and the system delays. We noticed that the system part of the dialogue is too long. Speech outputs need to be shortened and reformed to avoid frustration and confusion.

To gain more results on the use of the system, the user activity should be increased to have more use cases. This could be accomplished by running the test day and night. However, this would call for a wizard controlling the process all the time. Also another problem is the gathering of permissions.

ACKNOWLEDGMENTS

We thank Markku Turunen and Jaakko Hakulinen for constant support during the writing and testing process. We would also like to thank Dr. Roope Raisamo for guidance. This paper is part of the project ‘User Interfaces for Ubiquitous Computing’ project funded by the Academy of Finland.

REFERENCES

1. Bernsen, N.O., Dybkjær, H., Dybkjær, L. *Designing Interactive Speech Systems*. Springer-Verlag, London. 1999. pp 127-160.
2. Dahlback, N., Jonsson, A., and Ahrenberg, L. Wizard of Oz Studies -- Why and How, *in Proceedings of the 1993 International Workshop on Intelligent User Interfaces.* 1993. pp. 193-200.
3. Gustafson, J., Bell, L., Beskow, J., Boye, J., Carlson, R., Edlund, J., Granstrom, B., House, D., Wiren, M. AdApt-A Multimodal Conversational Dialogue System in an Apartment Domain. In *Proceedings of International Conference on Spoken Language Processing* (ICSLP 2000), Peking, China. 2000.
4. Johnsen, M., Svendsen, T., Amble, T., Holter, T., Harborg, E. TABOR - A Norwegian Spoken Dialogue System for Bus Travel Information. In *Proceedings of International Conference on Spoken Language Processing* (ICSLP 2000), Peking, China. 2000.
5. Salber, D., and Coutaz, J. Applying the Wizard of Oz Technique to the Study of Multimodal Systems Empirical Studies, in *Proceedings of the EWHCI'93*. 1993. p. 55-67.
6. Weinschenk, S., Barker, D.T. *Designing Effective Speech Interfaces*. John Wiley & sons. 2000.
7. Weiser, M. The Computer for the 21st century. *Scientific American, 265(3*). September 1991. p. 94-104.

Space Alignment Tool: A New Manipulation Tool for Graph Editing

Jukka Raisamo, Tommi Järvinen
University of Tampere
Department of Computer and Information Sciences
TAUCHI, Pinninkatu 53B
FIN-33014
Tampere, Finland
Tel: +358 3 215 8882, +358 400 578 640
E-mail: jr@cs.uta.fi, *tj58156@uta.fi*

ABSTRACT
Graph drawing is an important application area in the field of computer science. Several editors have been developed for drawing and creating graphs of a different kind, but often the layout of a graph needs to be able to adjust after the graph is being made. However, only a few currently available applications provide methods for adjusting only the wanted parts of the diagram. All the current layout-adjusting methods have the disadvantages of their own. They are either indirectly controlled, so the user cannot know how the operation will affect the graph, or direct, but do not contain enough functionality to offer a pleasant use. In this paper, we introduce a new direct manipulation method for adjusting only the selected parts of a graph. The tool offers an easy way to select and align only the objects the user wants to. It aligns the objects against its surface and makes the distances between them equal at the same time.

KEYWORDS: Graph drawing, drawing tools, alignment stick, direct manipulation, graphical user interfaces.

INTRODUCTION

Graph drawing addresses the problem of constructing the geometric representations of graphs and has important applications to the most important computer technologies, such as software engineering, database systems, visual interfaces, and computer-aided design [6, 20].

However, the current graph drawing programs provide only a few ways to adjust the alignment of finished diagrams. The most common way to do the adjusting operation is to use command based alignment tools from a menu, palette or dialog. Other widely used ways to manipulate diagram layouts are to use automatic layout algorithms and use of direct positioning using an input device such as mouse or touchpad.

The first way to accomplish the goal is interactive, but needs a lot of thinking and concentrating, because it requires thinking about the result of the alignment operation before it is done. Instead, automatic layout algorithms are easy and powerful to use, but they do not respect the mental map the user has formed of the diagram [9]; static algorithms rearrange the screen totally and destroy the user's mental map of the model [7]. Direct positioning is a rather slow but usually accurate way to manipulate diagrams.

Eades et al. [7] pointed out that there is a need for methods that rearrange the screen to focus on a particular region, while not disturbing the layout so much as to lose the mental map. Misue et al. [9] also discuss some layout adjusting methods and the preservation of the mental map of the diagram. They covered two layout-adjusting problems, ensuring node disjointness and providing whole and detailed views.

Diagrams used in different occasions look visually almost alike. They always consist at least of objects (entities) and relations (edges) between those. When drawing diagrams with a computer application, one of the most retarding barriers for the smoothness of the modeling process is the manipulation of the layout of the diagram already drawn.

We think the adjust and manipulation of the diagrams should be a direct and an interactive process, in which the user's editing operations are shown on the screen in real time. In practice, this means a need for diagram manipulation methods based on direct manipulation.

The concept of direct manipulation was first introduced by

Ben Shneiderman [18, 19]. He suggested that user interfaces should be direct to manipulate, meaning the continuous representation of the objects and tools of interest visually, have rapid, complementary and reversible commands as physical actions, and display the results of the actions immediately. In our tool, we have also used two-handed interaction for making the direct manipulation even more direct. William Buxton and Brad Myers [3], for example, have proved that two-handed interaction has manual and cognitive benefits compared with the traditional one-handed interaction. Roope Raisamo [13] found in his empirical study that two-handed interaction received the best subjective ratings from the users and the best results in accuracy, the number of operations and the length of operations.

The tool presented in this paper is based on the idea of the alignment tool introduced by Roope Raisamo and Kari-Jouko Räihä [14]. Roope Raisamo and Tapio Niemi [16] suggested that applying novel direct manipulation techniques [15] in conceptual schema drawing would make modifying the diagram layout easier and more understandable.

In this paper, we introduce a new graph layout adjustment tool that is handy to use, accurate, and is based on the concept of direct manipulation [18, 19, 22]. The tool is developed in a Java-based object oriented drawing program called R2Java [12]. Our tool enables the user to edit selected parts of the diagram in real time. The tool aligns the selected objects against its surface and makes the distances between them equal at the same time. The tool is operated with two hands, using mouse in the dominant hand and keyboard in the non-dominant hand.

The rest of this is paper is organized as follows. First, we take a look at the previous work done. After that, we describe some issues in the graph drawing related to our work. Then we take a closer look at the current alignment methods available, after which we bring up arguments about why the alignment stick should be adapted to graph editing tasks. The design process section describes the design of the tool and the section after that introduces the tool in action. Then we compare our tool to other alignment methods, until in the last two sections the work we have done is being discussed and concluded.

PREVIOUS WORK

Snap-Dragging [1, 2] developed by Eric Bier and Maureen Stone is a method which attend the functionality of the conventionally dragging and dropping of entities. Snap-Dragging uses the ruler and compass metaphor to help the user place his next point with precision, and uses heuristics to automatically place guiding lines and circles that are likely to help the user construct each shape. Snap-Dragging also provides translation, rotation, and scaling operations that take advantage of the precision placement capability. The purpose of the method is mainly to create layouts, not to adjust or manipulate them after the creation.

Kathy Ryall et al. [17] introduced the GLIDE system (Graph Layout Interactive Diagram Editor), which improves the general constraint-based systems by providing a special set of macro constraints, called Visual Organization Features (VOFs). These allow the user to manipulate the diagram interactively with the computer. The animation helps the user in understanding how to achieve a desired layout. The authors claim that with these features, graph-drawing editor is superior in many ways to those based on more general and powerful constraint-satisfaction methods.

Many commercial products have a menu, a palette or a dialog, which is used to align and adjust the layout of a graph. In this method, the alignment procedure is divided in two parts: first, the direct selection of the entities to be aligned and second, the selection of an appropriate alignment command. These methods are commonly called *command based alignment tools*.

Roope Raisamo and Kari-Jouko Räihä [14] introduced a kind of new direct manipulation tool, called the *alignment stick*. It was developed to align drawing objects in a simple object oriented drawing program called *R2-Draw*. The tool uses a ruler metaphor that connects it close to the real world. The basic idea of the tool is to select the objects to be aligned by touching them with the stick cursor, after which the selected objects are aligned against the tool surface. The tool removes one step in the alignment process compared with the command-based tools, so the whole alignment process is retained direct. The developers found out that alignment stick gave two major benefits compared to command based alignment tools: it is intuitive to use, and it has added alignment functionality in the tool. This means great advantage compared with the traditional command based techniques and the traditional drag and drop technique.

The study of alignment stick was continued in extensive usability testing, where the alignment stick was compared with two traditional ways to align drawing objects, menu and palette commands [15]. The study pointed out that even the most novice users managed to use the alignment stick rather well. Moreover, the results of this study show that the alignment tool is a useful tool also in diagram editors, but does need further development to succeed in more complex alignment operations.

Roope Raisamo and Tapio Niemi [16] applied alignment stick and a new tool called the alignment arc as tools for modifying schema layouts. The alignment arc functions similar to the alignment stick; it aligns the objects along with its round shape. They suggested that applying novel direct manipulation techniques introduced by Roope Raisamo and Kari-Jouko Räihä [15] in conceptual schema drawing would make modifying of the layout easier and more understandable compared with using only a layout algorithm.

ISSUES IN GRAPH DRAWING

In this section, we will introduce several issues in graph drawing, mainly related to the visualization point of view.

Graphs are conventionally drawn with entities represented by boxes, which may contain some text, and edges represented by lines between the boxes. The usefulness of the graph representations depends strongly on the quality of the layout of the graphs [5, 9]. The better layout a graph has the better it is capable of conveying the meaning of the diagram quickly and clearly to the user. This sets a requirement for having readable diagrams.

The clarity and outlining of the layout are essential criteria for the readability of different diagrams. The readability issues of the graphs are expressed by means of *aesthetics*. They can usually be formulated as optimization goals for the drawing algorithms [5].

Aesthetics

The aesthetics is important in creating a clear and coherent whole of a diagram. Di Battista et al. [6] state that aesthetics specifies the graphic properties of the drawing as much as possible to achieve readability. Tamassia [21] say aesthetics is subjective and may sometimes need to be tailored to suit one's personal preferences, applications, traditions and culture. The most fundamental and classical aesthetics are the minimization of crossings between edges, the avoidance of bends on edges, proper spatial distances between entities and display of symmetries [5, 6, 10, 20].

Minimization of crossings. Sugiyama et al. [20] presented that the greatest difficulty in tracing paths is line crossings. Di Battista et al. [6] say it is desirable to keep the number of times that lines cross to a minimum. They continue saying that ideally, we would like to have a planar drawing, but not every graph admits one. Purchase at al. [10] made an experimental study where they found out that minimizing arc crossings is an important aid to human understanding of graph drawings. In another study Purchase [11] proved that reducing the edge crosses is by far the most important aesthetic.

Avoidance of bends. Tamassia [21] state edges with more than two or three bends may be difficult to follow for the eye. Sugiyama et al. [20] say it is easy to follow straight lines. Purchase et al. [10] and Purchase [11] point out that minimizing the number of bends on edges has a significant effect on the human understanding of graph drawings. Di Battista et al. [6] state also that the minimization of a total number of bends along the edges is especially important for orthogonal drawings, while it is trivially satisfied by straight line drawings.

Display of symmetries. Di Battista et al. [5] claim that display of symmetries is desirable in all graphic standards. Purchase [11] proved that maximizing symmetry has a significant effect on the human understanding of graph drawings. The symmetrical outfit eases the information processing of human being and makes it easier to adapt the model the graph stands for. Entities closely related to each other should be placed symmetrically close to each other respecting the hierarchy.

Distances of entities. Sugiyama et al. [20] say also that close layout of entities connected to each other is desirable, and it is desirable that edges are short. It is important that entities closely related to each other should also be located close together. The distances between the entities provide additional information about the relations of the entities. In the same way, the length of the edges should be minimized, thought not at the cost of the symmetry or clarity.

Relevant aesthetics for human understanding

Tamassia [21] say that there are infinitely many drawings for a graph, and in drawing a graph, we would like to take into account a variety of properties, for example, planarity and the display of symmetries are highly desirable in visualization applications.

Purchase et al. [10] studied three popular aesthetic qualities: minimize edge crosses, minimize edge bends and maximize symmetries. They aimed to validate those using paper-based experiments. They found out that minimize crossings and minimize bends are both important aids to human understanding, but the third aesthetic, maximize symmetry, did not, however, seem to be a very important aid.

In another study, Purchase [11] considered five aesthetics to place a priority order on the relative importance of the aesthetics. The influence of each aesthetic was measured in time and in the number of errors made. This study pointed out that minimize of edge crosses is by far the most important aesthetic, when minimize of edge bends and maximize of symmetry have a lesser, but significant effect. The effects of maximizing the minimum angle between edges leaving a node and of fixing edges and nodes to an orthogonal grid are not statistically significant.

Problems in graph drawing

Eades et al. [7] and Misue et al. [9] addressed a need for layout adjusting methods, separate from layout creation methods. Both references state that the traditional layout algorithms concentrate only on creating the layout; they cannot be used to adjust just a selected part of the graph.

Misue at al. [9] covered two layout-adjusting problems, ensuring node disjointness and providing whole and detailed views. They stated that the most critical issue involved in layout adjustment was preserving the mental map of a diagram. To achieve this, they described three mathematical models of the mental map: orthogonal ordering, proximity and topology. The first model is the most basic one, where the entities are laid out orthogonal. The proximity model follows the idea that the entities, which are close together, should also stay close together. The topology model suggests that the visual representation of the diagram has the same topology as the transformed visual representation.

Tamassia [21] claims that usually, graph drawing problems can be formalized as multi-objective optimization problems, where we would like to construct a drawing with, for example, minimum area and minimum number of crossings edges. When, confronting this problem we are often obligated to compromise in somewhere. Automatic layout algorithms do this usually by the rules predefined by the designer. We took another approach on this matter and decided to give as much control as possible to the user.

We thought that our tool should primarily align the entities and secondary pay attention to the more advanced aesthetics. This is explained more closely in the new tool section.

CURRENT ALIGNMENT METHODS

In this section, the three major methods of rearranging and manipulating entities in current graphical diagram editors are introduced.

Command based tools

The most widely used method is command based alignment tools, which use a two-step procedure in alignment. The first step demands user to choose the entities to be aligned, usually by pointing and clicking them with mouse. The second step is to issue an appropriate alignment command from the menu, palette or dialog. Command based alignment tools are not direct due to the second step, neither are they very handy. The main reason the manufacturers have held to these must be that the users have accustomed to use their old applications.

Automatic layout algorithms

Automatic layout algorithms are often taken as an easy and carefree way to keep the graph arranged. However, they are often awkward to use and rearrange diagrams by the aesthetic criteria predefined by the programmer, not the user himself. Because of that, automatic layout algorithms cannot pay almost any attention to the users' individual needs to arrange the diagram visually and conceptually pleasant form neither can they rearrange only a part of the diagram.

Eades et al. [7] stated that, when the graph has been changed, the reapplication of a static layout algorithm may not be very helpful, because it may totally rearrange the screen destroying the user's mental map of the model. They pointed out that there is a need for methods that rearrange the screen to focus on a particular region, while not disturbing the layout so much as to lose the mental map.

These disadvantages addressed have been observed by the algorithm developers, and evolutionary and dynamic algorithms of many kinds have been developed [4, 5, 8]. Nevertheless, these have not a completely answered to the conflicts between the user's goals and the response of the application. As Roope Raisamo and Tapio Niemi [16] stated: "The automatic layout algorithms cannot take into account semantic information in the graph."

Direct positioning

Direct positioning, or manual dragging, of the entities is a very time consuming and rather irritating process. When selecting multiple entities, this method can be used only to move them from a place to another. However, an inaccurate fine adjustment of the diagram layout is often handy and fast done by dragging the single entities. Without a doubt, the direct positioning of the entities is the most intuitive and easy way to adjust a graph, when applying it just to a single entity.

ADAPTING ALIGNMENT STICK TO GRAPH EDITING

This section declares why and how to adapt the alignment stick to the graph drawing tasks.

Eades et al. [7] and Misue et al. [9] made a difference between layout creation and adjusting methods. They addressed a need for layout adjusting methods of making possible to manipulate only the specific parts of the graph.

Raisamo and Niemi [16] suggested that the new direct manipulation techniques, for example alignment stick [14, 15], could successfully be adapted for the diagram manipulation tasks. They thought the same interactive tool could make the both large-scale and fine-grained layout changes. They continue saying that the rough layout can first be done with a layout algorithm, after which the user can tune the layout with the direct alignment tools.

The use of both layout algorithm and direct manipulation tool is sensible in the case of manipulating large diagrams. Nevertheless, the algorithm should do only a rather suggestive layout and do not try to optimise it too much, so the tool can be used to adjust the layout. The use of the algorithm is not, however, necessary; the whole layout can be made using only the tool.

Direct manipulation tools like an alignment stick make a great advance to the diagram manipulation. By using these tools, user can easily rearrange the selected parts of the diagram interactively, so the user can see the impacts of alignment operations in the real time. Because of these benefits, it is easy for the user to compose the diagram piece by piece to the pleasant form using the tool.

The greatest benefits of the new direct manipulation tools are just the weakness we addressed for the current alignment methods; the possibility to easily and directly manipulate the diagram part by part to the outfit the user wants to. With these tools users can select multiple entities for the alignment operation and fine adjust the layout to the form they like to.

THE DESIGN PROCESS

This section tells how the new diagram manipulation tool was designed and which topics had to be compromised.

In the process of developing the alignment stick to answer better to the requirements the graph layout editing places, we decided to use the existing Java-code of the R2Java program as much as it would be possible to. After trying to manipulate diagrams with the existing alignment stick, we began to consider, what would be the requirements we are going to set for our tool.

Requirements for the tool

The design process was started by thinking about the requirements for functionality we would like the tool for having in graph manipulation tasks.

First of all, with the new tool it should be possible to adjust the distances between chosen entities in addition to align their surfaces against the tool shape. Second, the adding of new entities into the alignment operation should be easy and logical. Third, we found it important to somehow visualize the functionality and the state of action of the tool, rather than just show the frame of it in the screen.

All these requirements should be implemented simple enough so that it would not affect too much to the smoothness of the manipulation operation; the use of the tool must remain direct.

The Aesthetics concerned

The aesthetics introduced in the issues in the graph drawing section was taken into an account depending on various reasons. We have not paid attention to the aesthetics we did not consider important, such as the minimization of the area or minimization of the aspect ratio of the graph.

The minimization of edge crossings was have to be left to the shoulders of a user, because of the basic idea of our new tool: it should work logically and interactively all the time. The involving of the computer would cause some unexpected and sudden changes to the layout. In practise, the user decides alone while adjusting the graph layout, if crossings are remained.

The avoidance of bends is satisfied trivially in our applications, because all the edges are drawn as straight lines. Currently, the tool does not pay any special attention to the edges connecting the entities being aligned, so it may not be directly adapted to work with the lines with bends in them.

The display of symmetries is an essential feature for our tool. Aligning the selected entities on the same line and setting their distances equal satisfies this aesthetic. The tool can be used to align both high-level and low-level entities, so the whole layout of the diagram can be adjusted with the tool.

The distances of entities is another aesthetic that is satisfied due to the basic functionality of our tool. The user can adjust the distances and decide how closely the entities are related to each other. This is a feature whereby the mental map of the user can be maintained and even improved.

Answers to the requirements

The greatest problem of the design process was to implement the mathematical formula for the tool that makes able the alignment in an angle other than horizontal or vertical. The current implementation is not a perfect one, but at least makes it able to test the tool at the most of the angles.

The first requirement of adjusting the distances between the chosen entities was completed. Notwithstanding, we managed to distinguish the ends of the tool from the distances of the entities aligned. Because of this minor setback, both ends of the tool are unnecessarily long when only a few entities are being aligned with a long stick. This should, however, be moderate easy to improve so that the ends are being handled separately from the distances of the entities.

The second designed goal, the adding of new entities into the alignment operation easily and logically, was reached successfully when using the tool in orthogonal angles. The free-angle functionality can be added to this by making a formula for the intermediate angles.

The third design requirement, the appropriate visualization of the tool and its functionality, was achieved properly. The visualization of the tool is not maybe the most beautiful one, but it satisfies all the requirements we set for it.

THE NEW TOOL

This section introduces the space alignment stick describing the visualization, functionality and control of the tool.

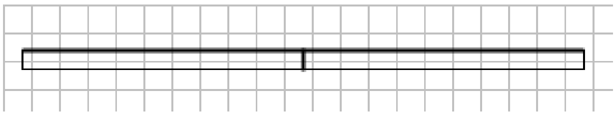
In many cases, the user wants to adjust the layout of a graph so that the distances between the entities at the same level of hierarchy are equal. However, current diagram editors do not offer a decent way to do this. We have implemented a new direct manipulation tool to better meet the demands of the user.

We named the new tool as a *space alignment tool*, because it aligns and adjusts the distance between the selected objects. In addition to aligning the objects selected just like the alignment stick, the tool makes the distances between the objects equal. These distances can then be adjusted by resizing the tool.

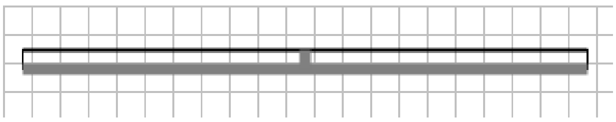
Visualization

The tool is visualised in two different ways. First, the active state of the tool is visualized by drawing the grey line underside of the tool shape. An inactive tool appears as a plain frame. Second, the grey lines are painted from the stick to the centre of the aligned entities.

By default, when the stick is not activated, the tool is seen as a rectangle, where the other long side is a little thicker than the other and a fine cross line in the middle of it (picture 1a). After the tool is being activated by pressing the left mouse button, the activation is visualized by a grey colour on the underside of the tool (picture 1b).

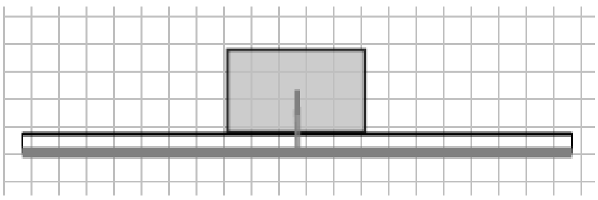


Picture 1a: The tool is in an inactive state.



Picture 1b: The tool is in an active state.

When an entity is added to the alignment operation, a grey line is drawn from the tool to the centre of the entity to visualize of the functionality of the tool (picture 2).

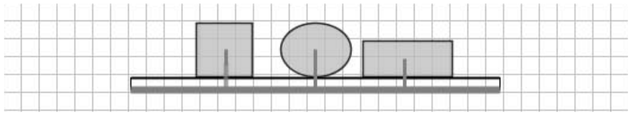


Picture 2: The visualization of the tool in use.

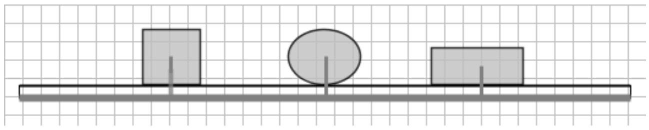
Functionality

In many cases, a user wants to adjust the layout of a graph so that the distances between the entities at the same level of hierarchy are equal. The space alignment tool is designed to align entities, which should be aligned on the same plane. The tool enables the user to select the entities to be aligned and adjust the mutual distances of their centres.

The pictures 3a and 3b show how the tool can be used in graph adjusting. The user wants to align the distances of the three selected entities, after which he wants to increase the distances.



Picture 3a: The entities are aligned with the tool.



Picture 3b: The distances between the entities have been increased.

Control

Roope Raisamo [13] points out that two-handed mouse and trackball setup was the most preferred and accurate input method in the control of direct manipulation alignment tools [14, 15]. In the study, he also found that trackball was clearly better suited for the non-dominant hand than the keyboard, and that the mouse was clearly best suited for the dominant hand.

Before starting the implementation process of the tool we thought that the trackball could be a good solution to use for the non-dominant hand to control the length and the angle of the tool. However, we did not have time enough to implement our tool a trackball compatible, so we chose a numpad of the keyboard for the non-dominant input device.

The tool is meant to be operated by using two hands, mouse at the dominant hand and keyboard at the non-dominant hand, but it can also be controlled by using a three-button mouse only. The tool is activated by pressing the left mouse button and deactivated by releasing it. While the tool is activated, user can add new objects to the alignment operation by hitting them with the stick. The length and the angle of the tool can be adjusted in spite of if the tool is activated or not.

If using the keyboard as a secondary input device, the length of the tool can be adjusted with the “8” and “2” buttons from the numpad, and the angle of the tool can be adjusted with the “7” and “9” keys. If using just a mouse, the length can be adjusted by pressing the right mouse button down and moving the mouse along the stick orientation, and the angle can be adjusted by pressing down the middle mouse button and dragging the mouse horizontally.

COMPARING THE TOOL TO OTHER METHODS

In this section, we tentatively compare the space alignment stick with the methods we introduced in the current alignment methods section. In that way, we try to clarify the strengths and weaknesses of our tool.

Command based tools. The command based alignment tools require two steps to make the alignment operation. Our tool does that also, the selection and adjustment, but these steps are both direct ones. Besides this benefit, our tool enables the user to adjust the distances of the selected entities in the real time, when the command based tools always require at least the trimming of the distance and the execution of a new operation.

Automatic layout algorithms. Static automatic algorithms can rearrange the whole diagram when only a minor change is being made to the layout. This is in many cases disastrous for the user mental map of the diagram. Our tool does the adjustments only to the selected parts of the diagram. These changes are adaptable and retail compared with the sudden and major changes most algorithms do.

However, there are some dynamic algorithms that keep an account of the changes the user has done and makes constricted rearrangement on the basis of these [4, 5]. Some evolutionary algorithms can “learn” the user’s pleasures of the layout by offering alternative guidelines to user to choose of [8]. Some systems also animate the changes the algorithm makes.

Nevertheless, none of these algorithms can take into account the semantic information the graph contains. In addition to this the computation time most of the algorithms need to satisfy the aesthetics is usually rather long. This is where the direct manipulation tool is the most superior in comparison with the algorithms; it makes possible for the user to do all the adjusting interactively and without any delay.

Direct positioning. The traditional direct positioning is comfortable and an excellent method of moving single entities. However, it is quite a tricky method of making perfectly aligned the totality of several entities. With our tool it may be a little bit slower to handle single entities, but when there is a need to arrange two or more entities our tool is much more efficient.

DISCUSSION

The current graph drawing programs provide only a few ways to adjust the alignment of finished diagrams. There are command based alignment tools, automatic layout algorithms and direct positioning using an input device such as a mouse or touchpad. Command based alignment tools are interactive, but they need a lot of thinking and concentrating. Instead, automatic layout algorithms are easy and powerful to use, but especially the static algorithms totally re-arrange the screen destroying the user’s “mental map” of the model. Direct positioning is a rather slow but accurate way to manipulate diagrams.

We introduced a new direct manipulation based alignment tool, called a space alignment tool, which aligns the selected objects both against the stick-shaped tool and their mutual distances. With the tool, the user can adjust the graph part by part, and in that way maintain and even improve the mental map of the graph.

The tool aligns the distances of objects on the basis of their centers. This may cause some problems, particularly if the user wants to align objects of different width or height and have the distances of the bounds of the objects to be equal. Although we thought it is more important to align on the basis of the center points, it could still be useful to implement the bound alignment functionality as well. This might be better to be allocated into a new tool in order that there would not be too much functionality included for a single tool.

Direct manipulation tools like an alignment stick and space alignment tool make a great advance to the diagram manipulation. Compared with widely used align methods, command based tools, algorithms and drag and drop, our tool has several advantages. First, it affects only the entities selected for the alignment operation and makes it that way possible to adjust only the specific parts of the diagram. Second, it is genuinely direct to manipulate in both select and adjust phases. Third, it makes the adjustments for more than just one entity at the same time.

The tool is not developed to produce highly optimised layouts like the most automatic alignment algorithms do. It is developed rather to make able to adjust and create a graph layout that is as pleasant and readable as possible. We have concentrated on developing an adjustable, easy to use and versatile direct manipulation tool for graph editing.

The tool does automatically satisfy the aesthetic of symmetry, but maintaining the other relevant aesthetics (the edge crossings and distances of the entities) is left for the responsibility of a user. When operating with moderate large diagrams (sixty or more entities), users may not want to adjust the whole graph layout manually. In these cases, it may be better to first apply a coarse layout algorithm for the graph and fine-adjust it after that with a direct manipulation tool.

The control of the stick is not completely ready yet, because we did not have enough resources to implement the trackball as an input device for the non-dominant hand. However, the functionality of the stick is ready and should be evaluated more accurately in the future.

The most severe shortage of the R2Java program is that it is not scalable in any particular way. The drawing area can, of course, be scrolled, but there is no zoom option to change the scale of the view. This lack must be improved before the tool can be evaluated thoroughly.

CONCLUSIONS

The new tool described in this paper is about to offer a new and efficient way to adjust diagrams. We found several advantages in our tool compared with the existing alignment methods. However, the appropriate usability tests should be carried out to properly evaluate the usefulness of our tool in practice.

ACKNOWLEDGMENTS

The authors would like to acknowledge the former designers of the R2Java application, Juhani Ränkimies and Tero Kukola, to make able to implement our tool. We also thank the course leader Roope Raisamo for the support and many valuable comments during this project.

REFERENCES

1. Bier, Eric A. Snap-Dragging: Interactive Geometric Design in Two and Three Dimensions. XEROX CORPORATION. Palo Alto (CA US) - EDL-89-2 - 09/1989.

2. Bier, Eric A. and Stone, Maureen C.: Snap-Dragging. Proceedings of SIGGRAPH'86, ACM Computer Graphics, 20 (4), 1986, pp.233-240.

3. Buxton, William and Myers, Brad A. A study in two-handed input. *Human Factors in Computing Systems, CHI'86 Conference Proceedings*, ACM Press, pp. 321-326.

4. Cohen, Robert F., Di Battista, Giuseppe, Tamassia, Roberto, Tollis, Ioannis G. and Bertolazzi, Paola. A Framework for Dynamic Graph Drawing. In *Proceedings of ACM Symposium on Computational Geometry*, 1992, pp. 261-270.

5. Di Battista, Giuseppe, Eades, Peter, Tamassia, Roberto and Tollis, Ioannis G. Algorithms for Drawing Graphs: an Annotated Bibliography. *Computational Geometry Theory and Applications*, vol.4, 1994, pp. 235-282. Available also at *ftp://wilma.cs.brown.edu/pub/papers/compgeo/gdbiblio.ps.Z* (Checked 4.6.2000).

6. Di Battista, Giuseppe, Eades, Peter, Tamassia, Roberto and Tollis, Ioannis G. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice-Hall Inc., Upper Saddle River, New Jersey, 1999. ISBN: 0-13-301615-3

7. Eades, Peter, Lai, Wei, Misue, Kazuo and Sugiyama, Kozo. Preserving the mental map of a diagram. *Proceedings of the International Conference on Computational Graphics and Visualization Techniques* (COMPUGRAPHICS'91), Portugal, September 1991, pp. 34-43.

8. Masui, Toshiyuki. Evolutionary Learning of Graph Layout Constraints from Examples. In *Proceedings of ACM UIST ´94 Symposium on User Interface Software and Technology*, ACM Press, 1994, pp. 103-108.

9. Misue, Kazuo, Eades, Peter, Lai, Wei and Sugiyama, Kozo. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, vol.6, no.2, pp. 195-209, 1995.

10. Purchase, Helen C., Cohen, Robert F. and James, Murray I. An Experimental Study of the Basis for Graph Drawing Algorithms. *ACM Journal of Experimental Algorithms Vol. 2, art. 4, 1997.* Available also at http://www.jea.acm.org/1997/PurchaseDrawing/ (Checked 5.6.2001)

11. Purchase, Helen C. Which Aesthetic Has the Greatest Effect on Human Understanding? *In Giuseppe Di Battista (ed.), Graph Drawing*, volume 1353 of Lecture Notes in Computer Science. Springer-Verlag, 1997, pp. 248-261.

12. R2Java. http://www.cs.uta.fi/research/hci/interact/twohands/

13. Raisamo, Roope. An Empirical Study On How to Use Input Devices to Control Tools in Drawing Programs. In *Proceedings of OzCHI'99, Conference of the Computer Human Interaction Special Interest Group of the Ergonomics Society of Australia*, Charles Stuart University, 1999, pp. 71-77.

14. Raisamo, Roope and Räihä, Kari-Jouko. A New Direct Manipulation Technique for Aligning Objects in Drawing programs. In *Proceedings of ACM UIST´96, Symposium on User Interface Software and Technology*, ACM Press, 1996, pp. 157-164.

15. Raisamo, Roope and Räihä, Kari-Jouko. Design and evaluation of the alignment stick. In *Interacting with computers*, vol. 12, 2000, pp. 483-506.

16. Raisamo, Roope and Niemi, Tapio. Interactive Layout Techniques for Conceptual Schema Editors. In *CD*

Proceedings of NordiCHI 2000, STIMDI (The Swedish Interdisciplinary Interest group for Human-Computer Interaction), 2000.

17. Ryall, Kathy, Marks, Joe and Shieber, Stuart. An Interactive Constraint-Based System for Drawing Graphs. . In *Proceedings of ACM UIST´97, Symposium on User Interface Software and Technology*, ACM Press, 1997, pp. 97-104.

18. Shneiderman, Ben. The Future of Interactive Systems and the Emergency of Direct Manipulation. *Behaviour and Information Technology*, vol.1, 1982, pp. 237-256.

19. Shneiderman, Ben. Direct manipulation: A step beyond programming languages. *IEEE Computer*, August 1983, pp. 57-69.

20. Sugiyama, Kozo, Tagawa, Shojiro and Toda, Mitsuhiko. Methods for Visual Understanding of Hierarchical System Structures. In *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 2, February 1981, pp. 109 - 125.

21. Tamassia, Roberto. Graph drawing. In *Handbook of Discrete and Computational Geometry,* J. E. Goodman and J. O'Rourke (eds.), chapter 44, pages 815-832. CRC Press LLC, Boca Raton, FL, 1997.

22. Ziegler, Jürgen E. and Fähnrich, Klaus-Peter. Direct Manipulation. *Handbook of Human-Computer Interaction*, M. Helander (Ed.), Elsevier Science Publishers B.V., 1988, pp. 123-133.

Combined Voluntary Gaze Direction and Facial Muscle Activity as a New Hands-free Technique for Human-Computer Interaction

Marko Illi[1], Poika Isokoski[1], Veikko Surakka[1,2]

[1]Human-Computer Interaction Group
Department of Computer and Information Sciences
FIN-33014 University of Tampere
Finland
Tel: +358-3-215 8565
E-mail: marko.illi@uta.fi

[2]Department of Clinical Neurophysiology
Tampere University Hospital
P.O. Box 2000
FIN-33521 Tampere, Finland
Tel: +358-3-215 8551

ABSTRACT

The present aim was to explore a new alternative pointing technique for human-computer interaction. We also analyzed the applicability of Fitts' law to the new pointing technique. The new technique combined the use of two modalities: gaze direction for object pointing and facial muscle activation for object selection. Gaze direction data was monitored with an eyetracker. Facial muscle activity data was monitored by recording the voluntarily produced changes in the level of electrical activity of a facial muscle corrugator supercilii (the muscle that lowers the brows). A piece of computer software was written for using both the data for real time object pointing and selection in a graphical user interface. The results from the pointing task times showed, that for short target distances the mouse was significantly faster to use than the new technique. Although the new technique was faster than the mouse at long distances there were no significant differences for medium and long distances. The results showed that the Fitts' law was applicable for both techniques. We conclude that the new pointing technique was a very promising new interaction technique. The new technique can be used both by abled and disabled people, however, for disabled persons the new technique may prove to be especially useful.

KEYWORDS: Gaze, Facial EMG, interaction techniques, pointing techniques, Fitts' law.

INTRODUCTION

We use computers and other technology in our everyday life to perform different kind of tasks and operations. We use the computers for work and for fun. It can be expected that the time we spend with the computers will increase in future. The increasing use of computers will affect the demands we have for them. People should be able to communicate with the computers naturally and without an extra effort.

Graphical user interface (GUI) became popular in 1984 with the Apple *Macintosh* and they still are the most commonly used interfaces. Typical GUI activity involves pointing and selecting objects on the screen. Although, trackballs, joysticks and touchpads are available, the most commonly used pointing device still is the mouse. Recently, there have been many attempts to develop alternative human-computer interaction techniques. Some of the most promising alternatives to the mouse have been eye-based techniques. It is natural for people to look at objects and perform other tasks at the same time; eye movements require little conscious effort and people naturally look at the objects of interest [11]. Combining the eye gaze interaction with other input techniques requires only small amount of additional effort and users can learn to integrate gaze input with other modalities quickly and naturally [11].

There have been many studies on gaze-based interaction techniques. In most of the studies gaze-based techniques have been compared to the mouse. The most commonly asked question has been, how fast and accurate the gaze-based techniques are compared to the mouse. The research results have usually showed that the gaze-based techniques are at least equally fast or even faster to use than the mouse [e.g. 5,6,11,14]. However, the gaze-based techniques have usually been less accurate than the mouse [e.g. 6,11,14]. Inaccuracy usually results from the inherent noise in eye-tracking equipment, and the dissociation between the gaze point and the user's visual attention; the high-acuity area of vision covers approximately one degree of visual arc and the objects outside of the high-acuity area are seen less clearly [11,14]. Gaze has been used for object selection separately (unimodal techniques) and in conjunction with other modalities (multimodal techniques). One hard to solve problem has been the difficulty to find a natural counterpart for the mouse button press to activate the controls in GUI [5]. This problem is present in both the unimodal and the multimodal gaze-based techniques.

When the object selection is made unimodally by using gaze for both pointing and selection, there is so called 'Midas Touch' problem. The Midas Touch problem results that the user cannot look any objects without issuing a command [5]. A dwell time protocol has been used as a solution for the Midas Touch problem [e.g. 9,11,14]. In the dwell time protocol, the system waits a predetermined time before it issues a command. The length of the dwell time depends on the task. For some tasks it is possible to use as short as 150 ms dwell time [5]. Ware and Mikaelian [14] found that the use of hardware button was faster for all target sizes than a dwell time of 0.4 seconds, but the dwell time method was more accurate for all target sizes. As a rule, when the tasks become more difficult and more information processing is needed, the dwell time should be longer [12]. The dwell time protocol is problematic because it is slow due the extra selection time, and because the optimal dwell time is hard to define: the optimal dwell time for one task and for one person may not be optimal for another task and another person. A manual hardware button has been another widely used method for solving the Midas Touch problem. In this method gaze is used for object pointing and manual hardware button press is used for object selection. The use of hardware button still requires manual control and the hands-free advantage is lost. While most gaze-based systems replace the functionality of other inputs with that of gaze, gaze-added interfaces offer an alternative approach: gaze functionality is simply added with that of other inputs, and the user can employ it if and when desired [10,15]. The results have shown that the gaze functionality, when available, is usually employed to perform easy tasks and raw movements [10,15]. Gaze-added techniques allow the user to perform fine movements with manual device (e.g. with mouse or trackball), and have usually been more accurate than pure gaze techniques [10,15]. Unfortunately, the dwell time or manual button press is still needed.

A new alternative selection technique for hardware button press and for dwell time is the use of voluntary facial actions as an input device [6]. Although many facial actions and expressions are activated spontaneously in human-human communication, they can be used also voluntarily [13]. Facial actions and expressions result from muscle contractions caused by electric muscle action potentials. These changes in electrical activity can be registered and monitored with electromyography (EMG). Because the facial muscle potentials can be produced voluntarily and they can be monitored in real time, they offer a tempting alternative for present human-computer input techniques [6].

In this paper, we explored the idea of using combined voluntarily gaze direction and facial muscle activity for object pointing. The original idea of using gaze for pointing and facial muscle activity for selection was first presented by Partala, Aula and Surakka [6]. They tracked the user's gaze direction with an eyetracker and measured the voluntary activation of a facial muscle *corrugator supercilii* with EMG. Both the eyetracker data and EMG data were analyzed offline [see 6]. Their results showed that the new pointing technique was significantly faster to use than the mouse at medium and at long distances. The distance seemed to have only small effect for the task times in the new technique condition. In the mouse condition, however, there was a significant effect of distance. Because the offline results were encouraging, we wanted to investigate how well the new technique works in real-time. We were also interested in investigating how well the Fitts' law applies to the new pointing technique.

Fitts' Law

Fitts' law is a model of human movement. It predicts that the time to move to a target is linearly proportional to $\log_2(2A/W)$, where A is the distance or amplitude to move and W is the width or tolerance of the region within which the move terminate [8,9]. In other words, targets that are smaller or farther away take longer to acquire. According to Fitts' law, the task's *index of difficulty* (*ID*) can be quantified by using the following equation

$$ID = \log_2(2A/W)$$

The Shannon formulation for the index of task difficulty is

$$ID = \log_2(A/W + 1)$$

and is preferred because it always gives a positive rating for the index of task difficulty. It also provides a slightly better fit with observations [8].

The task time prediction model then becomes

$$MT = a + b \log_2(A/W + 1)$$

where constant a and b are empirically determined through linear regression. Constant a is the intercept coefficient and b is the slope of the line. *The index of performance* (*IP*) can be calculated using direct division of mean scores ($IP = ID/MT$). However, more preferred choice for model building is to compute the *IP*s through linear regression ($IP = 1 / b$, from $MT = a + b\ ID$), because linear regression produces the best fit for prediction line [8]. The higher the *IP* (term "bandwidth" is also used) the higher the rate of human performance and more information (bits) is articulated per seconds [8]. Previous research has shown that the Fitts' law is applicable to the use of the mouse [e.g. 9,11]. However, it is less clear if gaze-based techniques follow Fitts' law; Results from previous studies have been discrepant [e.g. 9,11,14].

METHODS

Apparatus

A regular PC mouse with cursor speed set up to medium from Windows 98 control panel was used. A 15" Nokia 500 Xa LCD monitor in 1024 x 768 resolution mode was used as a display. In the new technique condition, an Applied

Science Laboratories Model 4000 corneal reflection eye tracker was used to measure the gaze coordinates from the user's right eye at a sampling rate of 50 Hz. For a bipolar EMG recording, we used disposable surface electrodes. In order to achieve lower than 10 kΩ interelectrode impedance, the skin was cleaned with ethanol and slightly abraded with electrode paste and cotton sticks. The EMG activity of the user's *corrugator supercilii* was recorded with a Grass® Model15™ 8-channel differential amplifier connected to a National Instruments 16 bit AD converter board. The sampling rate was 1000 Hz. Gaze data from the eyetracker computer was send trough serial port and *corrugator* data from EMG computer was send through an isolated 10 Mbps Ethernet segment port to Pentium III 500 MHz PC computer running Windows 98. A piece of software was written to use both data in real time. Our software rectified EMG signals and was set up to produce click whenever the signal crossed certain threshold. If the user had difficulties to produce clicks or if he/she felt that clicks happened unintentionally, threshold and weight levels could be adjust. The experimental tasks were run with the same software.

Subjects

Nineteen right handed voluntary subjects participated and the results from five females and nine males were analysed. Five subjects were omitted from data analysis because of the problems with the eye tracker system or because they made too many errors. All subjects were familiar with the mouse, but unfamiliar with the new technique. All subjects had normal or corrected to normal vision; Contact lenses were allowed, but eyeglasses were not.

Experimental tasks

The experiment was a within-subjects 2×3×3 factorial design with two pointing techniques, three pointing distances and three target sizes. The experiment was counterbalanced so that six of the subjects started with the mouse condition and eight started with the new technique condition.

The actual pointing tasks were similar to those of Douglas and Mithal [1] with minor differences. In both conditions, the subjects were presented two objects simultaneously: a home square and a target circle. Circles were chosen as targets to avoid complications due to the angle of approach [see 8]. Three target diameters (25, 30 or 40 mm), and three target distances (60, 120 or 180 mm) were used. Diameter of the home square was always 30 mm. The targets appeared in one of eight different angles (four orthogonal and four diagonal directions) around the home square. As there were three different distances, three different sizes and eight different angles, there were in sum 72 different trials. Each condition was used twice, resulting in total of 144 trials. All 144 tasks were presented in a fully randomised order for every subject with both interaction techniques.

Procedure

When the subject arrived in the laboratory, the equipments and the test room were presented to subject to made him/her feel more comfortable. The subject was told that the purpose of the experiment was to investigate a new interaction technique. The subject was informed that the test would last overall approximately an hour and both the new technique and a mouse technique would be tested. If the subject started with the new technique condition, the use of the new technique was first explained to the subject in detail. The subject was explained that in the new technique condition gaze was used for object pointing (objects are pointed simply by looking at them) and voluntary facial muscle activity for object selection. EMG electrodes were placed on the region of *corrugator supercilii* on the left side of the face according to the guidelines by Fridlund and Cacioppo [3]. The experimenter showed an example of how to voluntarily activate the facial muscle *corrugator supercilii*. Because we wanted to ensure that the subject was able to voluntary activate the facial muscle, we used a training method where the subject saw the EMG graph on screen. The subject was instructed that a vertical blue line would appear in graph to indicate successful clicks. Visual feedback appeared to be a valid method for training, the subjects learned quickly and easily to use voluntary facial muscle activity as a counterpart to the mouse button press. The eyetracker calibration procedure was explained and trained next. When the subject was able to click without an extra effort and was familiar with the eyetracker calibration procedure, he/she was explained the task. If the subject started with the mouse, the procedure was the same, but the use of the new technique was explained to him/her after finishing the mouse technique condition.

There was a short practice (48 trials) before the actual experiment and it proceeded as follows. First, the home square and the target circle appeared simultaneously. The subject was instructed to point inside the home square (the square turned blue when the user's gaze or the mouse cursor was inside the square, depending on condition) and then click. The home square disappeared after a successful click. Then the subject was instructed to point inside the target circle and click as fast and accurately as possible. The target circle disappeared after a successful click. There was a pause of two seconds after the circle disappeared and the home square and target circle appeared again. Recalibration of the system was done during the practice when necessary. The time between two the clicks were measured as a task times.

After finishing the practice the subjects were asked if they had understood the task. There was a short relaxation period before the actual experiment during which the subject was explained that the actual experiment would be similar to the practice, but it would be longer. When the relaxation period was over, the eyetracker was calibrated and the actual experiment started. Eyetracker was recalibrated during the experiment when necessary.

Data analysis

It was expected that the highly visual feedback would decrease clicks outside the target area. However, in 17.5% and 3.3%, respectively to the new technique and the mouse, of all trials, there were one or more clicks outside the target circle. We excluded these trials from pointing task time analyses. We also excluded trials, in which the pointing task time differed more than two standard deviations from the subjects mean task times. In all, 20.7% of the new technique trials and 6.5% of the mouse trials were excluded from the pointing task time analyses. The same data was used for Fitts's law's analyses. Because only correct trials were included in Fitts' law's analyses, error correction was not needed. The task times were averaged over the different pointing directions. Then the mean task times and error percentages were calculated separately for all size/distance combinations (three sizes × three distances). For error percentage analyses we used those trials in which were one or more clicks missed the target circle. Repeated measures analyses of variance with Greehouse-Geisser corrected degrees of freedom were used for the statistical analyses. Bonferroni corrected paired t-tests were used for post hoc tests.

RESULTS

Pointing task time analyses

The comparison of the two pointing techniques revealed that at short and medium distances the mouse was faster to use than the new technique, but at long distances the new technique was faster to use than the mouse. A 2×3×3 (interaction technique × distance × size) three way ANOVA showed a significant main effect of distance $F(2, 26)=147.6$, $p<0.001$, and a significant main effect of size $F(2, 26)=43.2$, $p<0.001$. The interaction effect of the technique and distance was also significant $F(2, 26)=23.9$, $p<0.001$. However, the interaction effect of the technique and size was not statistically significant $F(2, 26)= 3.4$, $p>0.05$.

Post hoc tests showed that at short distances the mouse was significantly faster to use than the new technique $t'=4.3$, df=13, $p<0.01$. The new technique was slightly faster to use for long distances than mouse. However, there were no significant differences between the two techniques at medium and long distances. Because the interaction effect between the technique and distance was significant, one way ANOVAs were performed for both interaction techniques separately. One way repeated measures ANOVA showed a significant effect of distance for the new technique $F(2, 26)=15.6$, $p<0.001$, and for the mouse $F(2, 26)=215.9$, $p<0.001$. In the new technique condition, post hoc tests showed that the time between clicks was significantly shorter for short than medium distances $t'=3.4$, df=13, $p<0.05$, and significantly shorter for medium than long distance $t'=2.8$, df=13, $p<0.05$. In the mouse condition, post hoc tests showed that the time between clicks was significantly shorter for short than medium distances $t'=13.9$, df=13, $p<0.01$, and significantly shorter for medium than long distance $t'=10.4$, df=13, $p<0.01$.

Error percentage analyses

More errors were made in the new technique condition compared to the mouse for all distances and for all sizes. Three way repeated measures ANOVA showed a significant main effect of technique $F(1,13)=41.0$, $p<0.001$, a significant main effect size $F(2, 26)=29.8$, $p<0.001$ and a significant main effect distance $F(2,26)=5.4$, $p<0.05$. The interaction effects between technique and size $F(2, 26)=23.7$, $p<0.001$ and between technique and distance $F(2, 26)=5.4$, $p<0.05$ were also significant.

Post hoc tests showed that there were significantly more errors for all distances when using the new technique compared to mouse $t'=7.0$, df=13, $p<0.01$, $t'=4.6$, df=13, $p<0.01$, $t'=5.8$, df=13, $p<0.01$, respectively to 60, 120 and 180 mm. Post hoc tests showed that there were also significantly more errors for all sizes in the new technique compared to mouse $t'=8.5$, df=13, $p<0.01$, $t'=4.5$, df=13, $p<0.01$, $t'=4.3$, df=13, $p<0.01$, respectively to 25, 30 and 40 mm. Because the interaction effect of technique and distance was significant, one way repeated measures ANOVAs were performed separately for both techniques. ANOVA showed significant effect of distance for the new technique $F(2, 26)=5.5$, $p<0.05$ and for the mouse $F(2, 26)=4.8$, $p<0.05$. Post hoc tests showed that in the new technique condition there were significantly more errors for short than long distances $t'=2.9$, df=13, $p<0.05$. In the mouse condition, post hoc tests showed that there were more errors between short and medium distances $t'=3.0$, df=13, $p<0.05$. One way repeated measures ANOVAs were also performed separately for both techniques to test the effect of size. There was significant effect of size for the new technique $F(2, 26)=33.8$, $p<0.001$ and for the mouse $F(2, 26)=4.0$, $p<0.05$. Post hoc tests showed that in the new technique condition there were significantly more errors between small and medium $t'=4.1$, df=13, $p<0.01$ and between medium and large sizes $t'=3.3$ df=13, $p<0.05$. In the mouse condition there were no significant differences between sizes.

Fitts' law analyses

The pointing task times were also analysed to determine how well the Fitts' law equation applied to both techniques. The regression equation derived for the new technique was

$$MT = 501 + 79\ ID \text{ ms}; (r^2 = .988, p<0.001)$$

and the regression equation for the mouse was

$$MT = 180 + 198\ ID \text{ ms}; (r^2 = .990, p<0.001)$$

Regression equations for the new technique and for the mouse indicate that Fitts' law applies to both techniques. The new technique had an index of performance (*IP*) of 12.7 bits/s and the mouse had an *IP* of 5.1 bits/s. *IP*s were calculated through linear regression ($IP = 1 / b$, from $MT = a + b\ ID$).

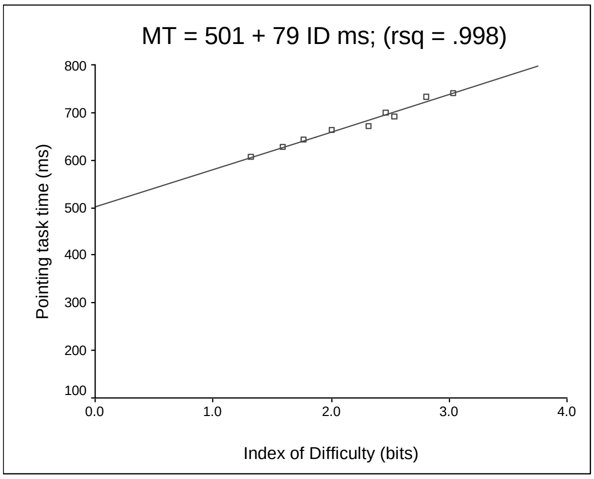


Figure 1. Regression slope for the new technique.

DISCUSSION

Our results showed that the new technique worked well in real time. Although the mouse was significantly faster to use than the new technique at short distances, at medium and long distances there were no significant differences. The new technique proved to be even slightly faster than the mouse at long distances. Equations from linear regression indicated that Fitts' law applied to both techniques. The new technique was quite inaccurate at small target sizes (26.3% errors at 25 mm), but become more accurate when targets were larger (9.5% errors at 40 mm).

Fitts' law assumes that both the target size and the target distance affect to movement task times. Previous research has shown that mouse is a Fitts' law device and our results supported this conclusion. The slope and the correlation coefficient derived from linear regression were high for the mouse. The results from previous research have been more discrepant considering gaze-based techniques. Our results were similar to those of Miniotas [9] and Ware and Mikaelian [14], and indicated that the Fitts' law applied relatively well to the new technique. The slope and the correlation coefficient derived from linear regression were similar for the new technique than for the mouse. Even though the slope for the new technique was lower than for the mouse, it was still clearly ascending. Lower slope for the new technique was due to the significant interaction effect of distance and technique. Thus increase in target distance had less effect to pointing task times in the new technique condition than in the mouse condition. Our results showed that effect of size was similar for both techniques, since the interaction effect of size and technique was not significant. In our study, the regression line intercept was quite large for the new technique. MacKenzie [8] has argued, that the large intercept may be a result of a consistent, additive, component of the task (for example button push). Users may have felt it a little strange to use facial muscle for clicking, and this may have resulted larger intercept in our study for the new technique than for the mouse. The users learned to use their facial muscle for clicking quit easily and quickly. It is still reasonable to assume that more practice would have improved the users performance. The improved performance most likely would have decreased the regression line intercept and the pointing task times. Practice might also have helped the user to perform more accurately. Because the users were familiar with the mouse, we suggest that the practising effect would be higher for the new technique than for the mouse.

In considering the accuracy of the new technique we suggest, that objects smaller than 30 mm should be avoided, when utilizing the new technique. Because of inherently jittery eye movements and immature eyetracking technology it is difficult to point objects smaller than 30 mm. Further, objects should be placed so that the distance between two objects is long enough.

A usual comment from the users was that clicking with the new technique was simple. Also the computer corresponded to users intention. Sometimes, however, the subjects told that the object seemed to disappear before he/she had clicked it intentionally. This may be resulted from the reason that the subject's facial muscle was activated even though he/she was not aware of it. Because the test was quite long, the users overall activity in a region *corrugator supercilii* may have increased during the test. Increased activity may have caused the users to sometimes activate the facial muscle unconsciously. Unintentional clicks may also have resulted from the reason that the trigger value may have been too low. In order to avoid unintentional clicks and individually adjust the system, threshold and weight adjustment were available. If the subject felt that the system was not responding or it responded too easily, threshold and weight level were adjusted at the beginning of the experiment.

We conclude, that our software detected facial muscle clicks fast and accurately. The combination of the voluntary gaze direction control and facial muscle activity offers a new promising interaction technique for real time human-computer interaction. Because people usually look at the object of interest, it is natural to use gaze for object pointing [11]. We also conclude that the use of voluntary facial muscle for clicking is a promising counterpart for the mouse button press. More study is needed before we know what the advantages and the weakness of the new technique are. Interesting research question would be to study how well other facial muscles work for clicking. *Corrugator supercilii* was chosen for present study because it is located near the eyes, and was considered to be appropriate to be used in conjunction with the gaze direction [6]. However, the use of *zygomaticus major* (muscle that draws the lip corners upwards, producing a smile) or some other muscle is also possible. It might be useful to combine the use of two or more muscles to enable more extensive command system.

In future, one of the greatest advantages on our new technique may be the opportunity to extract and convey social and emotional meanings from user's pupil and facial muscle activity. Changes in user's pupil diameter and facial muscle activity can be measured and used to adjust the computer to better correspond the user's social and emotional state. Because pupil diameter usually enlarges when people see or look something interesting, the computer could interpret the changes in user's pupil and, for example, offer extra information of that objects or pop up an agent. Similarly the information from facial muscle activity can be employed to enrich the interaction between the user and computer. Because certain facial muscle activity is associated with negative emotions and certain with positive emotions [2], the computer could be programmed to interpret the subject's feelings and alter it's behaviour according the emotional information.

REFERENCES

1. Douglas, S. A. and Mithael, A. K. The effect or reducing homing time on the speed of a finger-controlled isometric pointing device. In *Proceedings of CHI'94 Human Factors in Computing System*. ACM Press, New York, 1994, 411-416.

2. Ekman, P. and Friesen, W. V. *Facial Action Coding System (FACS): A Technique for the Measurement of Facial Action.* Palo Alto. CA.: Consulting Psychologists Press, 1978.

3. Fridlund, A. J. and Cacioppo, J. T. Guidelines for human electromyographic research. *Psychophysiology*, 23, 5 (1986), 567-589.

4. Gillan, D. J., Holden, K., Adam, S., Rudisill, M., and Magee, L. How does Fitt's law fit pointing and dragging? *Proceedings of CHI'1990 Human Factors in Computing Systems*, ACM Press, New York, 2000, 227-234.

5. Jacob, R. J. K. The use of eye movements in human computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems* 9, 3 (Apr. 1991), 152-169.

6. Partala, T., Aula, A. and Surakka, V. Combined voluntary gaze direction and facial muscle activity as a new pointing technique. Accepted for publication in *Proceedings of INTERACT 2001, Tokyo, Japan, July 2001.*

7. MacKenzie, I. S., Kauppinen, T., and Silfverberg, M. Accuracy measures for evaluating computer pointing techniques. To appear in *Proceedings of CHI'2001 Human Factors in Computing Systems*, ACM Press, New York, 2001.

8. MacKenzie, I. S. (1995) Movement time predictions in human-computer interfaces. In R. M. Baecker, W. A. S. Buxton, J. Grudin, & S. Greenberg (Eds.), *Readings in human-computer interaction* (2nd ed.) (pp.483-493). Los Altos, CA: Kauffman. [reprint of MacKenzie, 1992].

9. Miniotas, D. Application of Fitt's law to eye gaze interaction. In *Extended Abstracts of CHI'2000 Human Factors in Computing Systems*, ACM Press, New York, 2000, 339-340.

10. Salvucci, D. D. and Anderson, J. R. Intelligen gaze-added interfaces. In *Proceedings of CHI'2000 Human Factors in Computing Systems*, ACM Press, New York, 2000, 273-280.

11. Sibert, L. E. and Jacob, R. J. K. Evaluation of eye gaze interaction. In *proceedings of CHI'2000 Human Factors in Computin Systems,* ACM Press, New York, 2000, 281-288.

12. Stampe, D. M. and Reingold, E. M. Selection by looking: a novel computer interface and its applications to psychological research. *Eye movement research*, Elsevier Science B.V., 1995, 465-478.

13. Surakka, V. and Hietanen, J. K. Facial and emotional reactions to Duchenne and non-Duchenne smiles. *International Journal of Psychophysiology, 29 (1998), 23-33.*

14. Ware, C. and Mikaelian, H. H. An evaluation of an eyetracker as a device for computer input. In *Proceedeings of CHI'87 Human Factors in Computing Systems*, ACM Press, New York, New York, 1987, 183-188.

15. Zhai, S., Morimoto, C., and Ihde, S. Manual And Gaze Input Cascaded (MAGIC) Pointing. In *proceedings of CHI'99 Human Factors in Computin Systems,* ACM Press, New York, 99, 15-20.

Using a Mouse With Tactile Feedback

Elina Haapanen
University of Tampere
Kalevantie 4
33014 Tampereen yliopisto
Finland
E-mail: eh58878@uta.fi

ABSTRACT
The purpose of this paper is to describe how tactile feedback could be used in modal dialogs and target selection situations with a tactile mouse. Twelve subjects were tested in a target selection task with four different feedback conditions: Mouse shakes when the cursor is on the target, mouse shakes when the cursor is near the target so that tremble is more powerful when the mouse is near the target, mouse shakes when the cursor is far from the target so that tremble is more powerful when the mouse is far the target and normal feedback when the mouse does not shake at all. This paper describes a prototype that I made and it includes test results about how people like to use tactile feedback.

KEYWORDS: Tactile feedback, dialog, button, mouse.

INTRODUCTION

A mouse is one of the most common input devices in graphical user-interfaces. Computer interfaces that use tactile feedback have not been available for common user for so many years. Immersion Corporation has developed a technology called TouchSense and Logitech uses that technology in their iFeel mice, which are available for common users. TouchSense-enabled mouse allows users to use their sense of touch to interact with their desktops.

In this paper I present my study about how to use tactile feedback in modal dialogs and button pressing situations so that tactile feedback supports seeing. I was testing which way to use tactile feedback the users like to use in dialogs and target selection situations and what kind of tactile feedback gives best results to find the target. My experiment used a simple target selection task where a subject selected targets as quickly and accurately as possible. The use of a "normal" display served as a base-line condition, and to this I added three different ways to use tactile feedback as additional conditions. I measured selection time and subjects were also asked which of the feedbacks was most likeable and which was the most unlikable. Dialogs were modal because I also tested how tactile feedback can help people to work with modal dialogs. In my test the mouse shakes when the cursor is outside of the dialog and subjects were asked what they think about that feature.

PREVIOUS WORK

Akamatsu, MacKenzie and Hasbrouq [2] have studied target selection task under five different sensory feedback conditions. Those conditions were "normal", auditory, colour, tactile, and combined. In that study tactile feedback means that a pin under the fingertip pressed upward presenting a tactile sensation to the finger while the cursor was over the target. They did not find differences in overall response times, error rates, or bandwidths; however, significant differences found in the final positioning times (from the cursor entering the target to selecting the target). For the latter, tactile feedback was the quickest, normal feedback was the slowest.

Akamatsu and MacKenzie [1] have studied target selection task in four feedback conditions: normal, tactile, force and tactile+force. They found that compared to normal feedback error rate was higher with tactile and tactile+force conditions. Fastest movement time, which includes approach time and selection time, was using tactile+force feedback but tactile feedback decreases the selection time. Selection time includes stopping time and clicking time and they both decrease when using tactile feedback. Akamatsu and MacKenzie found that tactile feedback offers the best potential reduces target selection times and this effect becomes more pronounced as targets get smaller.

HARDWARE AND SOFTWARE

In this section I will tell about TouchSense technology and iFeel mice. When people use a computer, they typically use two senses, sight and hearing. Immersion Corporation [3] has patented a technology called TouchSense that engages a third sense – touch – to deliver more natural experience. Instead of just pointing at elements on screen, the cursor becomes an extension of user's hand.

Immersion Corporation has also developed the Immersion TouchSense software and web development kits. The free Immersion TouchSense Software Development Kit (SDK) helps developers to add touch sensations to Windows software applications. The heart of the kit is the Immersion Foundation Classes (IFC). IFC provides much of the code necessary for creating and managing the force feedback device and effects without the need of COM interfaces required by the DirectX and Immersion TouchSense APIs. In addition, Immersion offers Immersion Studio, an editing tool for touch effects used by many developers, and an IFC Tutorial, designed to take the work out of working with TouchSense technology.

The Immersion TouchSense Web Development Kit (WDK) helps developers to add touch sensations to the Web sites. The Immersion Web Plugin is at the heart of the kit and it allows TouchSense technology to be experienced within Web pages. In Microsoft Internet Explorer, the browser plugin is the Immersion Web ActiveX Control. The control makes much of the IFC functionality available within Internet Explorer (or other ActiveX control containers). In Netscape Navigator the browser plugin is the Immersion Web Netscape Plugin. This Netscape plugin makes use of the underlying ActiveX control.

Logitech Inc. [4] has put on the market new kinds of mice that are cheap enough for normal user to buy. These mice use TouchSense technology.

MY ASSUMPTIONS

In this section I explain what kind of dialog I made and tested. I made three different modal dialogs that use tactile feedback and one dialog that does not use tactile feedback at all. Dialogs are modal because I also tested property where the mouse shakes for warning if the user moves mouse outside the dialog.

All dialogs look the same but they work a little bit different ways. Every dialog has two buttons; a button starts the test and another closes the dialog. When the user presses the start button target appears in the dialog. In the test user must press that target and when user has pressed that target, it moves. That happens twenty times. In test situation a user must press targets as quick as possible. In dialog one, tactile-mouse gives feedback when the user moves the mouse on the target. In dialog two tactile-mouse gives feedback when user moves mouse near the target so that tremble is more powerful when the mouse is near the target. In dialog three, tactile-feedback works contrary to dialog two: When the user moves mouse tremble is more powerful as mouse is far from the target. There are no other features in every dialogs.

METHOD

Subjects and apparatus

Twelve subjects participated in the experiment. All subjects were regular users of mice in their daily work. The experiment was conducted using the Logitech iFeel Mouse.

Procedure

In the experiment users were asked to press the target that is colourful circle in the dialog as quickly and accurately as possible. The first target appeared right after subject pressed the start button (in top-left corner of dialog). Timing also started when the subject pressed the start button. When the subject pressed the target, it disappeared and appeared in other place in the dialog. They had to press circle twenty times in on test case.

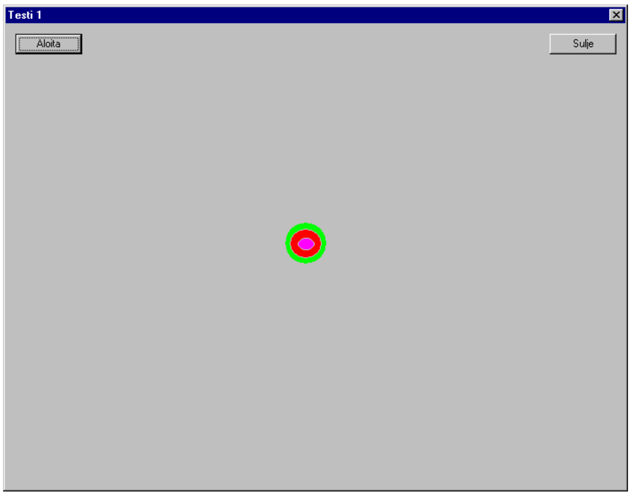


Figure 8. Dialog, 'Start' button in top-left corner, 'Close' button in top-right corner. A target to be selected is in middle of the dialog.

Design

Every subject did every test cases three times after trying out test cases first. There were four different test cases so every subject did test twelve times. To counterbalance for learning affects, every subject made tests in different order. Every subject was also asked which kind of feedback they liked the most and which kind of feedback they liked the less and why they liked or disliked chosen kind of test cases.

The dialog was modal dialog and mouse shaked for warning when the user moved mouse outside the dialog. Subjects were also asked what they like about that effect.

RESULTS AND DISCUSSION

Table 1 shows the effects of the four feedback modalities on the dependent measures of movement time.

Mean	Normal	Tactile feedback on the target	Tactile feedback near the target	Tactile feedback far from the target
15028.1 (ms)	15195.4 (ms)	14867.6 (ms) (-2.2%)	14891 (ms) (-2.0%)	15158.5 (ms) (-0.2%)
Values in parentheses are percent change relative to normal feedback				

Table 1. Average movement times for each test case.

The mean movement time for all test cases was 15028.1 ms. The conditions from fastest to slowest were tactile feedback on the target (14867.6 ms), tactile feedback near the target (14891 ms), tactile feedback far from the target (15158.5 ms), and normal (no tactile feedback) (521 ms).

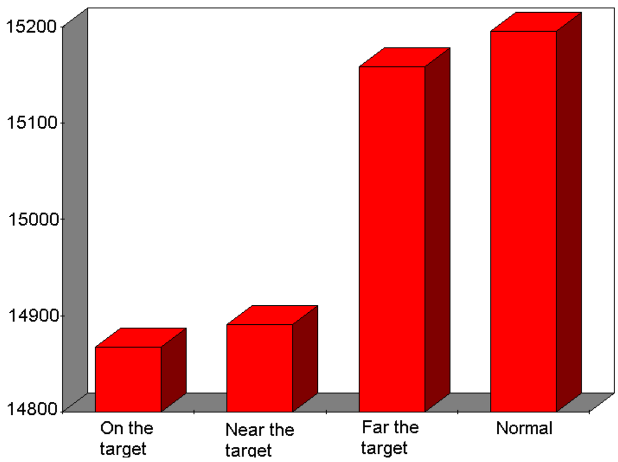


Figure 2. Average movement time for every test case.

Figure 2 shows mean movement time for every test case. It looks like that cases where tactile feedback were given on the target or near the target were fastest and cases where tactile feedback were not given at all or were given only far from the target were slowest.

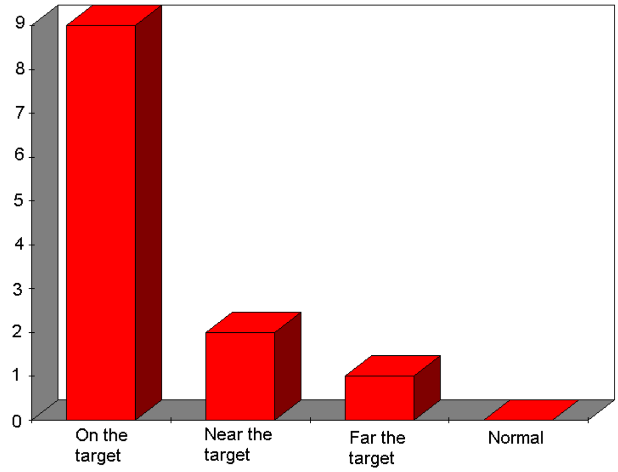


Figure 3 most likeable feedbacks

Subjects were asked what kind of feedback they liked most and what kind of feedback they disliked most. Figure 3 shows what kind of feedback subjects liked most. Almost every on (nine out of twelve) liked feedback where mouse shakes on the target most. No one liked most of the normal feedback.

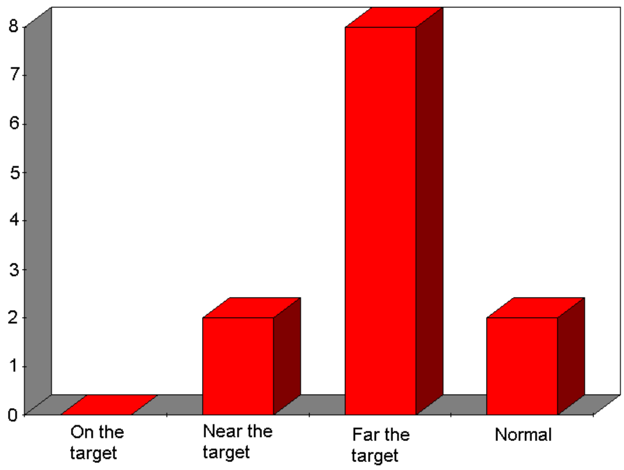


Figure 4 most unlikable feedbacks.

Figure 4 shows what kind of feedback subjects disliked most. Most of the subjects (eight out of twelve) chose feedback where mouse shakes far of the target for the most unlikable feedback.

Subjects were also asked what they like when mouse shakes outside of dialog. They thought that it was quite good feature when it is used in right places, for example in modal dialogs and situations where a user easily misses a dialog. Subjects also thought that it could be useful if a mouse shakes in different ways when the mouse is on different non-active dialog.

It could be said that tactile feedback was best when the mouse shakes on the dialog. In the future I am going to study more about that kind of tactile feedback. In this test the target was quite big (46*46 pixels) so in the future work I will test how tactile feedback helps selection when target is smaller.

ACKNOWLEDGMENTS

I would like to thank every people who helped me to do this study and participated in the experiment.

REFERENCES

1. Akamatsu, M., & MacKenzie, I. S. (1996). Movement characteristics using a mouse with tactile and force feedback. International Journal of Human-Computer Studies. Available: http://www.yorku.ca/mack/IJHCS2.html (25.3.2001).
2. Akamastu, M., MacKenzie, I. S., & Hasbrouq, T. (1995). A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics*, *38*, 816-827. Available: http://www.yorku.ca/mack/Ergonomics.html (25.3.2001).
4. Immersion Corporation, http://www.immersion.com/developer/cool.html (25.3.2001)
5. Logitech http://www.logitech.com/cf/products/index.cfm (25.3.2001)

Foot as a supportive input technique for navigation tasks

Jiri Hlusi
University of Tampere, Finland
jiri.hlusi@uta.fi

ABSTRACT

In this paper, we describe the project, which aims on using feet as an alternative input technique in conjunction with a CAD-like application. The intended task to be performed this with feet is scrolling of the workspace.

Keywords

Foot control, scrolling, 2D, project summary

INTRODUCTION

In the past, several researches have been carried on evaluating foot as an alternative input technique for computer systems. Deploying this idea, indeed, might bring at least an extraordinary point of view for utilizing feet and in the same time offer some benefits for the user. For instance, combining foot and hands in text typing tasks would reduce the hands' load by releasing them from scrolling. Not having to switch between keyboard and a standard mouse would definitely increase the performance when creating text. Going further on, a foot-controlled device (*FootMouse*) could also be used for scrolling in 2D space. Just let's imagine drawing program with enough large workspace, where the user uses foot for scrolling the workspace whereas the standard mouse is used for choosing tools from the provided palette and for performing the actual drawing. Yet another place, where foot might be used in some reasonable way. In this context also a question of more dimensions (e.g. 3D) rises up. Although this is not a scope of the hereby described project work, from poorly technical point of view, the 4D input might be realized with two FootMouse devices, where each operates in the 2D space of logical vectors

However, despite of outlined benefits, there are several things about to be considered in advance. First, as confirmed during previous research works, feet are not very precise in performing intended actions. The other thing is that people quite naturally tend to express emotions by pushing the foot mouse with different accent. This puts some demands on the mechanical construction of the input device. In addition, it is quite obvious that a proper visual feedback has to be provided for every action in order to make users self confident with the input device.

In other words, provided physical device should be reactive under any circumstance and solid in the same time. On the other hand, it shouldn't be too sensitive because small leg movements might be unintentional in many cases.

THE FOOTCAD PROJECT

One of important issues in designing and evaluating any user input technique is finding the proper application. In many cases a purpose-tailored application is created and the tests are carried using this facility. As the main idea of our project was to evaluate foot input in the 2D environment, a graphic creating application with enough large workspace was needed. Out of available options, the CAD-like application was chosen (this denotes the project name – FootCAD).

However, standing before decision whether rather create a prototype of a new CAD-like application or to try to use some available product, the second option was chosen. The chosen application then was the *ArchiCAD*, an application used for architecture work in building industry. In other words, the entire project turned to be an integration task between the *FootMouse* and the *ArchiCAD*.

The desired software architecture

As the ArchiCAD is an application for Windows platform, from the software point of view the entire project turned to a driver, which communicates with the FootMouse and sends appropriate commands to the ArchiCAD application. In order to implement this idea, three different approaches are available on the Windows platform:

1. *A Windows driver*

   This approach is the best one from the input device integration point of view (the FootMouse would be seen a standard Windows device). On the other hand a high level of knowledge on internal Windows architecture is required.
2. *Checking the FootMouse's status and sending standard Windows massages into the ArchiCAD application*

   This approach is less demanding on knowledge related to Windows on itself, on the other hand some spy-like investigation is necessary about the ArchiCAD in order to discover which message should be sent where.
3. *Windows direct input*

   Using this approach, mouse clicks are emulated by inserting appropriate events into the standard Windows input queue (via the function *SendInput*). In other words, operations are emulated in the same way, as the user would perform it alone.

Implementation and necessary adjustments

For implementing the project idea, a Microsoft Visual C++ together with Windows Platform SDK has been chosen. This choice has been done mainly due to the potentially smooth integration with the Windows platform and also due to performance reason (i.e. Java as an alternative solution would be perhaps much slower).

In the first stage we have tried to create a prototype of both essential parts of the "driver" (actually, it was a console-type Windows application, not a real driver). One operating with the input device, the other one sending messages to the ArchiCAD.

Working with the FootMouse:

Unfortunately, the FootMouse (a real device composed of two pedals – one of them operating in 2D) has been found broken and an attempt to fix it didn't lead into acceptable result. So, finally there was a need for changing the actual input device – the big enough track ball. The only constraint is that it doesn't look to be solid enough for a safe controlling by a foot (risk of breaking it).

The other problem was separating the track ball from the standard mouse, since in the Windows environment all the pointing input devices are by default controlling the mouse cursor. Trying to communicate with the track ball directly turned to be quite a tough task and we didn't succeed. The solution for this problem was to use two separated computers, both running Windows. On the first machine a track ball was used for moving the mouse cursor and the observed motions have been sent to the other computer, where the receiving party controlled the ArchiCAD application in accordance to received information (e.g. scroll left, right, up and down). The communication channel was IP/UDP.

Interaction with the ArchiCAD

Even this part was not that easy as expected. We have started with trying to send standard Windows messages (WM_SCROLL) to the application. However, for that an exact knowledge into which window (controller) what kind of information has to be sent. Perhaps due to the reason that the actual drawing is in *ArchiCAD* displayed in several windows (e.g. 2D and 3D view to the same architectural work), finding the right destination for the messages as well as discovering the proper message types simply failed. After that another approach came to the question. This involved the function SendInput, which allows a user actions to be emulated (e.g. "Move mouse", "Click", "Move back"). Although this method is quite reliable, it brings several side effects. First, since the clicking on scroll bars is about to be emulated, there is a need for knowing the exact location of the scroll bar. One has also to remember that the actual workspace (working window) can be resized and/or moved. Thus, for every emulate click, the entire window hierarchy has to be examined in order to perform the right click on the right location. In addition, the coordinates for the SendInput function have to be provided in scale 0-65535. Mapping functions for the real screen resolution (e.g. 1024x768) are needed. Second, since the real mouse action is emulated, no other objects (e.g. toolbars) may be placed above the scroll bar. Otherwise the emulated mouse click would initiate unexpected action. And finally, all mouse buttons should be released when the emulated mouse click is about to be performed. Otherwise the first operation in sequence (moving above to scroll bar) would result into the drag operation (moving with a mouse button pressed). But even with those restrictions, the evaluation of the scrolling with foot is still feasible.

Experience gained during solution usability tests

Once having both modules running, usability tests have taken its place. During this we have discovered several aspects:

1. It seems to be very important that the controlling foot is in a natural position (different from person to person). Feeling inconvenient makes the input technique almost unuseful. From this point of view, having the real FootMouse would be much more valuable than the track ball.
2. As known from previous researches, foot is not very precise. Looking from the other view angle, when sitting people quite often move the foot without any intention of performing a particular action.
3. Even after some training, the fact that a foot is fixed to be located above the input device (whatever it is) is quite inconvenient after some longer time. From this point of view, fixing both feet (i.e. having two pedals) would be almost impossible, indeed.

Suggestion for further research

It is clear that the actual performance of the foot input technique is critically dependent on the quality and design of the actual input device. Although not verified yet, it seems that ideal input device would be a pressure sensitive shoe (wireless) or a sensitive floor. This would allow user to change his/her position up to time and would be adoptive to different types of user. Sensing for the level of pressure and the gradient of change might even bring more input information related to the emotional expression of intentions performed by the user. Actually, as we know, some people are even capable of playing organs with their feet. Let's believe that controlling in the same way computers is not too far future.

ACKNOWLEDGMENTS

I would like to dedicate my special thanks to *Poika Isokoski* for his excellent guidance during the project, consultancy and the ability of keeping me motivated. Also taking this opportunity, I'd like to thank to my friends working for the company *ACE Design* for their consultancy in certain Windows issues.

Autocomplete in Text Input – Comparison between Finnish and English

Juha Jääskeläinen
Department of Computer and Information Sciences
University of Tampere
+358 50 366 1864
jj61466@uta.fi

Sami Pekkola
Department of Computer and Information Sciences
University of Tampere
+358 40 849 1156
sp61411@uta.fi

ABSTRACT
In this paper, we describe our research project about Autocomplete in Text Input – Comparision between Finnish and English.

Keywords
Autocomplete, text input, vocabulary, personalization.

INTRODUCTION

Autocomplete in classical text input with keyboard and PC is not so essential because it is usually faster to write all characters of words than find matches with autocomplete. Also usability point of view is important, user might be irrated by autocomplete's continuous suggestions. In new text input methods e.g. mobile phones or wearable computing autocomplete is more competent because inputting text is much slower and more difficult than with classic keyboard and two hands. Main purpose of autocomplete is to reduce number of needed key presses and by this make text input more effective.

Keyboard in new tehnical devices can differs lot of from classical keyboards because there is new arrangement to be done to reach higher usuability, e.g. different order for keys optimized for one hand use. Autocomplete doesn't respond this but there is a need for accept command. The accept command needs a new key or definition for acceptance key or key pressing sequence, like ctrl + a. For autocomplete there can be need for several new keys to reach more sophisticated autocomplication. More sophisticated autocomplete could include e.g. a accept key for some part of suggestion. Which user can use to accept parts of suggestion and re-write end of suggestion to achieve result user is aiming for.

For autocomplete there is some settings to done beforehand to make autocomplete as effective as possible for user. These settings can be for example adjust the feature that defines after how many key presses for this appropriate word autocomplete begins to make suggestion. Other adjustable thing could be that autocomplete adds space after accepted word. In our autocomplete tests application there is possible to adjust this thing but we found that adding space after accepted word is more effective. Our autocomplete simulator adds space automatically after accepted word.

COMPARISION BETWEEN FINNISH AND ENGLISH

For English there are several applications which includes autocomplete feature, for Finnish there are also some (for example Microsoft Word). But they are not very popular because lack of benefits users get by using autocomplete feature. Finnish and English differs a lot when comparing languages structure. In English there is lots of small words (like a, the, of and so on) to give meaning for words. In Finnish this is given usually by adding something to end of the word and usually this isn't even enough.

Example text in Finnish

Haluan kiittää Teitä, rouva presidentti, kutsusta saapua viralliselle valtiovierailulle Latviaan. Puolisolleni ja minulle on suuri ilo vierailla Latviassa myös näin virallisesti. Erityistä arvoa annan sille, että vierailu toteutuu Latvian uudelleen itsenäistymisen 10-vuotisjuhlavuonna ja Riian kaupungin perustamisen 800-vuotisjuhlan merkkivuonna.

- Number of words: 38
- Number of character with spaces: 404

Example text in English

I want to thank you, Madam President, for your invitation to pay a state visit to Latvia. My husband and I are delighted to visit your country also in an official capacity. We especially value the fact that our trip is taking place on the tenth anniversary of the reinstatement of Latvian independence and as Riga celebrates the 800th anniversary of its foundation.

- Number of words: 63
- Number of characters with spaces:435

SIMULATION

Our research's aim was to find out differences of autocomplete benefits in text input between Finnish and English. We had same text in Finnish and English. Both texts was compined from Tarja Halonen's (President of Finland) official speeches which can be founded from www.finland.fi/news. For both texts we had a complete vocabulary (in other words every word in texts exists in the vocabulary). We programmed a simulator which uses two text files for input; first one for the vocabulary and second one for the actual text. When the simulator is executed it reads the actual text word by word and finds suggestions (after two characters) for words from the vocabulary. Simulator counts characters from word so long than it finds a definite match for the current word. When it finds definite match for the word, we can think that user accepts that word by pressing acceptance key, so there is one more key press added to total keypresses and the simulator moves on to the next word. When whole actual text has been gone throw we will have a result that tell us how many key presses there would be needed from user to write that text with autocomplete feature. We executed same procudere for both languages Finnish and English.

Statistical data about both textes

Finnish:

- Words in text: 18202
- Words in vocabulary: 6971
- Number of characters: 160171
- Number of needed key presses: 126783
- Characters autocomplete ratio: 20%

English:

- Words in text: 26069
- Words in vocabulary: 3589
- Number of characters: 157535
- Number of needed key presses: 130324
- Characters autocomplete ratio: 17%

Analysis of statistics

As we can see from the statictical data there is a significant differencies between these two language's structure even actual text's content is same. If Finnish there is much less words in actual text (18202 vs 26069) but number of characters in Finnish and English text are almost equal (160171 vs 157535). In vocabularies there is also big difference between these two languages, in Finnish there is almost two times more words in vocabulary than in English. Characters autocomplete ratio tells how much characters from text were replaced from the vocabulary. So we can come to a conclusion that with complete vocabulary (in other words every word in texts can be founded from the vocabulary) there is small advantage for Finnish. This because in English there is lots of short words (1 or 2 characters) which are always written manyally (not from vocabulary).

There is much more words in Finnish and words are also much longer in Finnish than in English. Main reason for this is that in English there is used only few prepositions for all words (e.g. a, in, the) and these prepositions are located before words. In Finnish there is much more these prepositions and they can be founded at the end of words (e.g -lle, -lla, -ssa). In Finnish there is also used finite forms which make biggest difference between there two languages. Here is simple example about finite forms between Finnish and English where it is seen that Finnish needs larger vocaburaly:

- I go, you go, he/she goes, we go, you go, they go (8 words needed in vocabulary)
- Minä menen, sinä menet, hän menee, me menemme, te menette, he menevät (12 words needed in vocabulary)

SUMMARY

At the beginning of research we believed that English is much adaptable for autocomplete because languages structure. With the result of our simulations shows after all that Finnish is more adaptable for autocomplete. This is caused by complete vocabulary. The complete vocabulary is not effective for Finnish if it is really complete and includes all Finnish words, in English this might work better. For results of simulations we arrived at the result that with sophistically optimized vocabulary autocomplete can accomplish more effectivety to text input for both languages when written text is limited to a given area. For Finnish limited finite forms reduces a lot of number of needed key presses when e.g. writing verbs.

New interaction technique for flexible text copying

Teemu Tanila
E-mail: teemu.tanila@uta.fi

ABSTRACT
This technical paper presents a programming project, which aims to demonstrate the use of more flexible copy feature than can currently be found from applications.

KEYWORDS: Copy, paste, text.

INTRODUCTION
The traditional way of copying text into the clipboard is familiar to all of us. Select the text that you want to copy and use a copy command to copy the selected text into the clipboard. Then paste the selection from clipboard into the desired document. If you want to make another copy/paste operation in the same context, repeat the procedure.

What if you would like to copy several different parts of the document? You also have to make several different copy/paste operations, since currently it's not possible to select individual parts of the document at the same time.

Quite recently Microsoft offered an improved copy feature in their Office 2000 product. Now it's possible for several individual objects to exist in the 'clipboard' at the same time. You still have to copy each selection individually into the clipboard, but all the objects can be pasted at once if desired. Unfortunately this feature is available only in Office products.

FreeCopy demonstrates how a user can select different parts of the document and copy them into the clipboard as a single block in only one copy operation.

FREECOPY
I have implemented a small program that can be used for demonstrating the use of this technique. The program is basically a lightweight Notepad. It can be used for opening and displaying *.txt files, making a selection of text and for copying selected parts of the text into the clipboard. Implementation was made with C++ builder4.

INTERACTION TECHNIQUE
The idea behind this interaction technique is quite simple and so is the implementation. When the user makes a selection in the FreeCopy, it's automatically copied, into a local buffer. When the user makes another selection, it's appended into the same buffer. In this way, the user can perform number of selections desired. When the user is ready, the content of the buffer can be copied into the clipboard in a single copy command. At the same time the buffer is flushed (Figure 2).

This technique can coexist in perfect harmony with the current implementation of the copy/paste functionality. In my opinion, it should be thought as an extension of the copy/paste operation.

Unfortunately, at least to my understanding, it's impossible to implement a general solution for this, since it seems to be impossible to get a hold of selections of stand-alone applications. The 'FreeCopy' functionality must be implemented separately into each application that wishes to use it.

USING FREECOPY

The user interface of the FreeCopy (Figure 2) contains only one button related to the new copy functionality. The same button is used for copying the buffer content into the clipboard and for flushing the buffer.

Only new thing user has to learn is the usage of a single button. It is important to notice that the selection of the text has to be done with the mouse in order to use the new copy functionality.

Let's take an example. User wants to copy several different parts of the text into a clipboard using the new copy functionality. User selects desired parts of the text with the mouse. Every selection is automatically copied into a local buffer. When a user is done with the selections, he clicks the button marked with an 'i' in the user interface. Local buffer contents are copied into the clipboard and the local buffer is flushed. User can now paste the contents of the clipboard into a desired application as usual.

User can now repeat the procedure. User can also use the same button to clear the buffer if he is not happy with the selections he made. It should be noted that the content of the buffer is first copied into the clipboard. This is acceptable because at this point it doesn't matter what is the content of the clipboard. User is going to copy a new object into the clipboard anyway.

EVALUATION

Copying techique presented in here doesn't interfere with the existing technique, i.e. it doesn't limit the functionality of the traditional technique. It does, however, bring some advantages over it. If a user wants to copy several different parts of the text into the same place, this new technique improves the efficiency of working. It is very clear that some time is saved when it is not compulsory to copy each selection separately.

This is only a small improvement over the functionality provided by the Office 2000, but a major one over the traditional technique.

User might experience some difficulties when trying to remember the contents of the buffer. Currently FreeCopy doesn't have the functionality to signal the state of the buffer to the user.

The possibility of partial deletion of the buffer was considered but it didn't seem necessary since it is probably rare for the buffer to hold a lots of content at once. Also the functionality should be as simple as possible. In this case, less seems to be more.

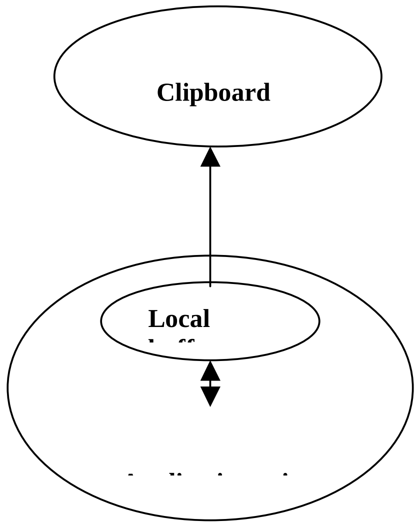


Figure 1: Structure of the FreeCopy

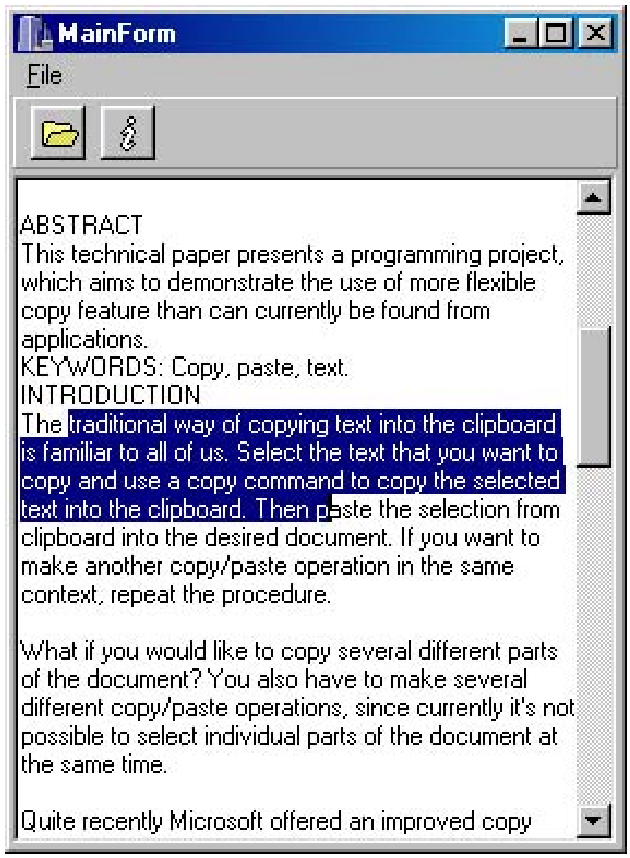


Figure 2: FreeCopy UI

Wearable E-Mail Client

Mikael Rinnetmäki
Tampereen Yliopisto
www.uta.fi/~mr60305/
mikael.rinnetmaki@uta.fi

ABSTRACT

This paper includes a description of a wearable e-mail client. The application is controlled using N-fingers wearable input device and voice output.

Some suggestions for enhancing a very limited interaction channel are presented.

Keywords

N-fingers, input devices.

INTRODUCTION

The aim of this work was to study how the N-fingers input device, which only allows for a very narrow interaction channel, can be used to control applications with some more complexity.

Postimies E-mail reader was chosen as the application to be controlled.

Previous Work

N-fingers Input Device

N-fingers is a wearable input device containing five contacts on a minimal glove as illustrated on Figure 1. Four contacts serve as input keys. Input commands are triggered by tapping the four contacts with the one contact attached to users thumb.

N-fingers is developed at Nokia Research Center.

Postimies E-mail reader

Postimies (Mailman) is a multilingual speech-only e-mail client based on an adaptive speech application framework [1]. It is normally controlled via voice input and speech recognition.

Postimies retrieves the e-mail headers from a user-specified mailbox, and sorts them to folders. Each folder can contain the maximum of six messages. This is to ensure that the navigation between the messages does not exceed the limits of cognitive capabilities of the users.

Postimies is developed by the Adaptive Speech User Interfaces project of Human-Computer Interaction Group at University of Tampere.

Using N-fingers Input Device with a command based application and Voice Output

Features

With the application, it is possible to do everything Postimies is currently capable of. This includes listing, browsing and removing mail folders, listing, browsing and managing individual messages, and naturally reading (listening to) messages. Also a limited help feature is provided.

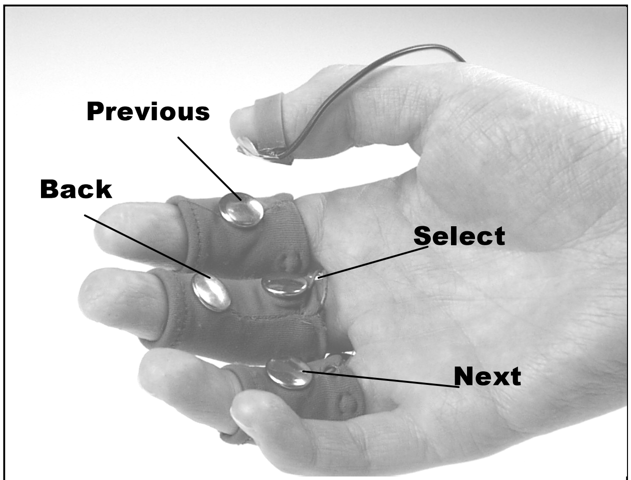


Figure 1. *The contacts on the N-fingers input device and their naming.*

For the application to be as easy to use as possible, each input key always triggers same kind of action. For instance, a click on the pointing finger always selects the previous item (a folder, a message, or an option).

For output the application uses three kind of audio. The information on the folders and messages is provided by Postimies with synthesized speech. Help feature uses sampled speech. Response to user's inputs and information on the state of the application is provided by short sampled sounds.

The design process

The main problem when designing the application was the limited amount of input keys, compared to the amount of features the application was supposed to be capable of executing.

Postimies application has a vocabulary of 21 words and a small grammar telling how the words can be ordered. This kind of language can be considered quite limited for a speech-controlled system, but from the point of view of an input device with four inputs it seems quite large.

Generating multiple inputs with only four keys

Although N-fingers only has four inputs, more different inputs can be generated with it. The device can be seen as a chord keyboard, with the exception that the key combinations have to be sequential, as there is no way of touching multiple contacts with the thumb at the same time. The drawback of key combinations is that they are hard to learn and producing them requires more time and effort than single key presses.

When using N-fingers with Postimies, key combinations are used in starting and closing the application. These actions take place only once per session, so the lack of efficiency compared to simpler input methods is no problem. Also, this kind of limited number of key combinations should not be too difficult to learn and to remember. Furthermore, a complex method for starting the application is actually needed to protect this from occurring by accident. Inputs occurring within 1 second are considered a key combination.

In addition to key combinations, another way to generate more inputs with the contacts of N-fingers input device is time-dependant inputs. E.g. a short input may be interpreted as a click, a longer one as a press.

The application uses time-dependant inputs for a help feature. When the duration of an input signal is less than 0.8 seconds it is considered an action. When the duration exceeds the threshold of 0.8 seconds the application offers help to the user by telling which action would be executed by a shorter input signal (a click).

As we have seen, with only four keys we can generate multiple inputs to an application. However, the cognitive load for the user grows with every special key combination or timeout. This means that the amount of different input methods should be minimized, and the use of special methods should be consistent.

Application state model

In addition to different inputs, more features can be provided by dividing the application to internal states.

The application was divided into two main states: browsing the folders, and browsing the messages. In addition to the main states there are two options states, which are used to manage (remove and list) folders and messages.

In folders state, the user can browse the folders (*previous*, *next*). By selecting a folder the user triggers a transition to messages state. In messages state, the user can browse the messages (*previous*, *next*). User can read (listen to) a message by selecting it.

From the main states, the user can get to options states by selecting *back*. In options states, it is possible to list the items (folders or messages) by selecting *previous*, and to delete items by pressing *next*. From the options states, the user can trigger transition to main state by *select*, and to the previous state by *back* command.

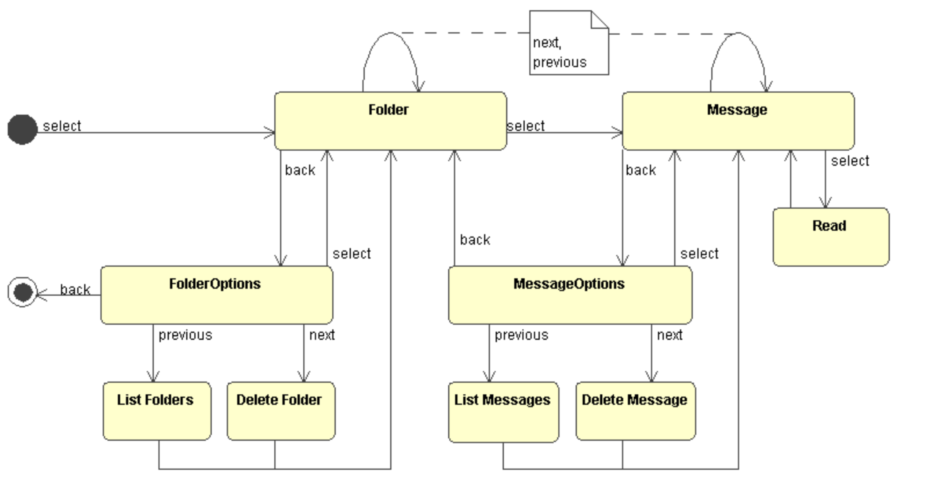


Figure 2. *States of the application. The same input triggers different actions depending on the state of the application. Additional Options states allow for more actions.*

Discussion

The application should be evaluated for its usability. The timeouts for complex input methods are merely estimates of what could be a useful threshold and would need to be tested and adjusted.

Conclusions

When designing an application with limited amount of input keys and multiple features, you can make use of key combinations, timeouts, and internal states.

ACKNOWLEDGMENTS

Markku Turunen and Jaakko Hakulinen from TAUCHI have been of great help in integrating the application to the Postimies system.

REFERENCES

1. Postimies http://www.cs.uta.fi/hci/SUI/Postimies/Mailman.html.