Proceedings of

COGAIN 2007

‘Gaze-based Creativity, Interacting with Games and On-line Communities’

COGAIN NoE is funded by the EU IST 6th framework program.

Welcome

Welcome to COGAIN 2007!

This is the third COGAIN camp and conference and I am delighted to welcome you to De Montfort University and Leicester. The theme of the conference marks how the subject of gaze-based communication has evolved to meet the changing ways in which computer systems are being used. Recent years have seen a massive increase in the use of computer games (both networked and stand-alone) as well as in online virtual communities and 3D environments, such as Second Life. These offer users with disabilities opportunities to interact with others in that environment or game on a similar footing to able-bodied users. To date, most of the work in gaze-based interaction has focused on 2D desktop applications and rightly so. Much has been achieved to produce truly usable systems and applications. It is now time to look towards 3D interaction with a real-time component to it. To be successful in this, we need to understand and support a greater variety of interaction techniques. Furthermore these need to be fast if a disabled user is to participate in collaborative activities on equivalent terms with able bodied users. This is the first conference to focus on this area specifically.

The conference also has sessions on broader aspects of interaction techniques using gaze and on technical aspects of eye measurement, reflecting the real breadth of work being carried out in COGAIN. Nearing the end of its third year, COGAIN has been a really successful venture in research collaboration and using the collective energies of its partners to further awareness and to promote the valuable work carried out.

So welcome to Leicester! It's a modern vibrant multicultural city with a rich historic past. Enjoy the conference and enjoy being in Leicester.

Howell Istance
COGAIN 2007 Conference Chair

COGAIN 2007 Program

Monday 3 Sept 2007: Academic sessions

Centre for Excellence in Performance Arts (CEPA)
De Montfort University, Leicester, UK

8:30-9:00	Registration
09:00-10:30	**Keynote:** **Modes and Transitions - Charting a Course for Creative Interaction** Dr. Andrew Duchowski, Clemson University, South Carolina, USA
10:30-11:00	Refreshments - Centre for Excellence in Performance Arts
11:00-12:30	**Session 1: 'Games and attention'** **Gamepad and Eye Tracker Input in FPS Games: Data for the First 50 Minutes** Poika Isokoski[1], Aulikki Hyrskykari[1], Sanna Kotkaluoto[1], Benoît Martin[2] [1]University of Tampere, Finland; [2]University of Paul Verlaine, Metz, France **Gaze beats mouse: a case study** Michael Dorr, Martin Böhme, Thomas Martinetz, and Erhardt Barth Institute for Neuro- and Bioinformatics, University of Luebeck, Germany **Eye Trackers: Are They Game?** Javier San Agustin, Jakob Schantz, and John Paulin Hansen IT University of Copenhagen, Denmark **Game play experience based on a gaze tracking system** M. Perreira Da Silva, V. Courboulay, A. Prigent L3i, Université de La Rochelle, La Rochelle, France **Dwell time reveals a narrowing of active options during selection in multi-element arrays** Mackenzie G. Glaholt, Eyal M. Reingold University of Toronto at Mississauga, Department of Psychology, Mississauga, Ontario, Canada

12:30-14:00	Lunch – DMU campus centre
14:00-15:30	**Session 2: 'Technology and Environments'** **Improved Low Cost Gaze Tracker** Detlev Droege, Thorsten Geier, Dietrich Paulus University of Koblenz-Landau, Koblenz, Germany **Magic Eye Control** Luis Figueiredo, Ana Isabel Gomes Escola Superior de Tecnologia e Gestão da Guarda, Guarda, Portugal **3D head orientation estimation and expression influence elimination using characteristic points of face** Donatas Dervinis, Gintautas Daunys Siauliai University, Vilniaus str., Siauliai, Lithuania **Environmental Control by Remote Eye Tracking** Fangmin Shi, Alastair G Gale Applied Vision Research Centre (AVRC), Loughborough University, Loughborough, Leicestershire, UK **Hands Free Interaction with Virtual Information in a Real Environment** Susanna Nilsson[1], Torbjörn Gustafsson[2], Per Carleberg[2] [1]Department of Computer and Information Science, University of Linköping, Sweden; [2]Division of Sensor Technology, Swedish Defence Research Agency, Linköping, Sweden
15:30-16:00	Refreshments - Centre for Excellence in Performance Arts
16:00-17:45	**Session 3: 'Interaction and Input'** **Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures** Jacob O. Wobbrock[1], James Rubinstein[2], Michael Sawyer[3], and Andrew T. Duchowski[4] [1]The Information School, University of Washington, Seattle, USA; [2]Department of Psychology; [3]Department of Industrial Engineering; [4]School of Computing: Clemson University, Clemson, USA **Dwell time free eye typing approaches** Mario H. Urbina, Anke Huckauf Bauhaus-University Weimar, Weimar, Germany **Using Face Position for Low Cost Input, Long Range and Oculomotor Impaired Users** G. Julian Lepinski, Roel Vertegaal Human Media Lab, Queen's University, Kingston, Canada

Accessible Web Surfing through gaze interaction

Emiliano Castellina, Fulvio Corno
Politecnico di Torino, Torino, Italy

The Exploration of Large Image Spaces by Gaze Control

Nick Adams, Mark Witkowski and Robert Spence
Department of Electrical and Electronic Engineering, Imperial College London, U.K.

Comparing two Gaze-Interaction Interfaces: A Usability Study with Locked-in Patients

Markus Joos[1]. Susann Malischke[1], Sebastian Pannasch[1], Alexander Storch[2] and Boris M. Velichkovsky[1]
[1]Dresden University of Technology, Germany; [2]Department of Neurology, Dresden University of Technology, Germany

17:45	Close

Centre for Excellence in Performance Arts

DMU campus centre

COGAIN 2007 Papers Index

COGAIN 2007 Keynote by Dr. Andrew T. Duchowski 7

Modes and Transitions - Charting a Course for Creative Interaction 7

Session 1: Games and Attention 9

Gamepad and Eye Tracker Input in FPS Games: Data for the First 50 Minutes 11

Gaze beats mouse: a case study 16

Eye Trackers: Are They Game? 20

Gameplay experience based on a gaze tracking system 25

Dwell time reveals a narrowing of active options during selection in multi-element arrays ... 29

Session 2: Technology and Environments 35

Improved Low Cost Gaze Tracker 37

Magic Eye Control 41

3D head orientation estimation and expression influence elimination using characteristic points of face 45

Environmental Control by Remote Eye Tracking 49

Hands Free Interaction with Virtual Information in a Real Environment 53

Session 3: Interaction and Input 59

Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures 61

Dwell time free eye typing approaches 65

Using Face Position for Low Cost Input, Long Range and Oculomotor Impaired Users 71

Accessible Web Surfing through gaze interaction 74

The Exploration of Large Image Spaces by Gaze Control 78

Comparing two Gaze-Interaction Interfaces: A Usability Study with Locked-in Patients 82

COGAIN 2007 Keynote by Dr. Andrew T. Duchowski

Modes and Transitions - Charting a Course for Creative Interaction

Motivated by recent real-time eye movement analysis research, the notions of modes and transitions are examined, particularly in the context of interactivity. Conscious of the theme of gaze-based creativity, modes and transitions are reviewed in terms of their use in games and virtual environments. Modes pertain to the automatic estimation of the user's intent or location of attentive deployment. Considering dwell time as a specific type of modality, this interactive style is endemic in eye tracking work spanning the last decade, especially when instantiated as a form of selection (other dwell-based interactive styles are contrasted with selection). Eye movement transitions, pertaining to gaze point switching among display regions, in comparison, have received less attention. Developments exploiting transitions are highlighted, indicating this topic's recent gain in popularity. Of the eye tracking publication sample considered in the last decade, a quarter is on transitions and has been published in the latter quarter of this time span. The relationship between publication quantity and its time of publication suggests emerging potential for innovation based on eye movement transition and a shift away from the dependence on dwell time.

Biography

Andrew T. Duchowski,
Clemson University, 100 McAdams Hall, Clemson, SC, USA 29634
Email. duchowski@acm.org
Tel. +1 864 656 7677

Andrew is Associate Professor at the Department of Computer Science, Clemson University. His Research & Teaching Interests include Visual perception and human-computer interaction, Computer graphics, eye tracking, virtual environments, and, Computer vision and digital imaging. He has recently written a book on gaze tracking: "Eye Tracking Methodology: Theory and Practice, 2nd edition", ISBN: 978-1-84628-808-7, and is a leader in the field of eye tracking having authored many papers on the subject.

Session 1: Games and Attention

Gamepad and Eye Tracker Input in FPS Games: Data for the First 50 Minutes

Poika Isokoski, Aulikki Hyrskykari, Sanna Kotkaluoto
Department of Computer Sciences
FIN-33014 University of Tampere, Finland
poika@cs.uta.fi

Benoît Martin
LITA, University of Paul Verlaine Metz
Île du Saulcy
57045 METZ CEDEX 1
benoit.martin@univ-metz.fr

Keywords
First person shooter, aiming, navigation, efficiency, game, eye tracker, input device, gamepad

Introduction

We report results on the use of eye tracker input in first person shooter (FPS) games. Six participants participated in 10 identical sessions. Each session consisted of three five-minute blocks with different input device configurations.

Earlier results have shown that input device configurations where the aiming is done with a mouse are more efficient in FPS games than using a gamepad (Isokoski & Martin, 2007). The results of Isokoski and Martin (2006) further suggested that using the mouse is also more efficient than using the gaze. Based on these results we focused on the comparison of purely gamepad-controlled input and two levels of eye control combined with the gamepad input. The earlier work did not provide enough data to differentiate between the efficiency of these techniques. This is why we considered this issue worth investigating.

In FPS games the 3D game world is shown from the player's point of view. The main needs for controlling the game are moving in the virtual world and aiming at objects to shoot at them. There are numerous ways to move the point of view in 3D virtual environments (Hand, 1997). Apart from special circumstances, such as walls that block the movement and jumping over things, the player's movement in FPS games follows the vertical changes in the terrain automatically. Consequently, moving around requires controlling only two degrees of freedom (2 DOF). Similarly, 2 DOF are needed for controlling the orientation of the view since the rotation that would tilt the view is usually not used. One commonly used mapping of controls is to use keys on the keyboard to control the movement of the game character with buttons for moving forward, backward, and sideways. The mouse is usually used to control the game characters gaze (i.e. the orientation of the "camera" projection).

In FPS games aiming the weapon is commonly connected to the view. A reticule for aiming the weapon is displayed in the centre of the player's view. By turning the view the target is aligned with the reticule before firing the weapon.

It is possible to use an eye tracker for one or both of the 2DOF controls: i.e. to control the moving in the environment or to control the aiming. Aiming seems to be a natural way of using the eye input. Shooting at the point of gaze makes the step of fine-tuning the orientation of the view before shooting unnecessary. This was an anticipated advantage of using eye tracker input in FPS games. The inaccuracy of eye tracking was an anticipated disadvantage.

We are not the first to explore eye tracker use in FPS games. However, Jönsson (2005) as well as Smith and Graham (2006) took a more general view on eye trackers in games. Controlling FPS games were only a small part of their studies. This precluded the kind of detailed exploration of the design space of the controls that we are aiming for.

The Game-Like Environment Used in the Experiment

We conducted an experiment in a simplified game-like pointing device testing environment. The environment consisted of a 3D model of a 1000x1000 unit terrain with trees and grass among which round targets with a penguin logo on them moved on random trajectories. The task of the participant was to navigate on the terrain and shoot as many targets as possible in 5 minutes. Whenever a target was hit, it disappeared and a new one appeared at a random location. There were 10 targets on the terrain at all times. A scene from the game is shown in Figure 1.

Figure 1. A scene from the FPS-like environment that we used in the experiments.

The Design of the Experiment

The experiment was designed to compare three ways of controlling the game and to record the learning curve for the first 50 minutes of play in each condition. Six[1] participants, 4 female and 2 male, ages varying from 25 to 60 (average being 31 years), were recruited for the experiment. Each participant completed 10 sessions. Each session consisted of three sub-sessions of five minutes of play. The three sub-sessions corresponded to the three input device configurations. The within-session order was balanced between participants. Two sessions were completed on one visit to the lab to minimize the number of times that a participant had to visit. The five double-sessions were scheduled with at least one hour, but no more than three days in between.

1 One of the participants reported nausea after a couple of sessions. His data was discarded and another participant was recruited to replace him. The effect of eye control to the onset of such symptoms may be worth further study.

The three conditions we compared were: (1) a traditionally used gamepad controller (*XBox360*), (2) the combination of gamepad controlled moving and aiming with eyes (*XBox360AE*), and (3) the XBox controller used only for moving forward and both the aiming of the weapon and steering of the movement were done by eyes (*XBox360ASE*).

In the Xbox360 method the left joystick of the controller was used for moving. The right stick was used for turning the view. The right shoulder button of the controller fired the weapon at the center of the screen. Thus, to hit a target the player needed to turn the view so that the target was at the center of the display.

In the second condition the left and right sticks and the right shoulder button were still used for moving and shooting the same way than in the first condition. However, aiming was done by eyes, i.e. the weapon pointed to where the player was looking. A small red reticle gave the player feedback on the tracker's interpretation of the gaze position.

In the third condition the sticks were not used at all. Instead the eyes were used to steer the movement and the left trigger of the controller was used to control the player's velocity in the game world. The further the trigger was pressed the faster the player moved to the direction of his or her gaze. The aiming reticle followed the player's gaze as it did in the second condition. The view did not follow the player's gaze when the trigger was not pressed. This way it was possible to stop and shoot at different parts of the display while the view stayed motionless. In summary, the player needed only two buttons to play the game: the left trigger to control the speed of moving and the right shoulder button to fire the weapon. Everything else was done with the eyes.

Results

There wasn't much difference in the number of target hits between the input methods as seen in Figure 2. A Session by method (10×3) repeated measures ANOVA showed a statistically significant effect only for the session ($F_{9,45}$=11.7, $p<0.001$) confirming the trivial finding that training increased the number of hits. However, moving more control to eye movements does not seem to change the players' ability to find and shoot the targets.

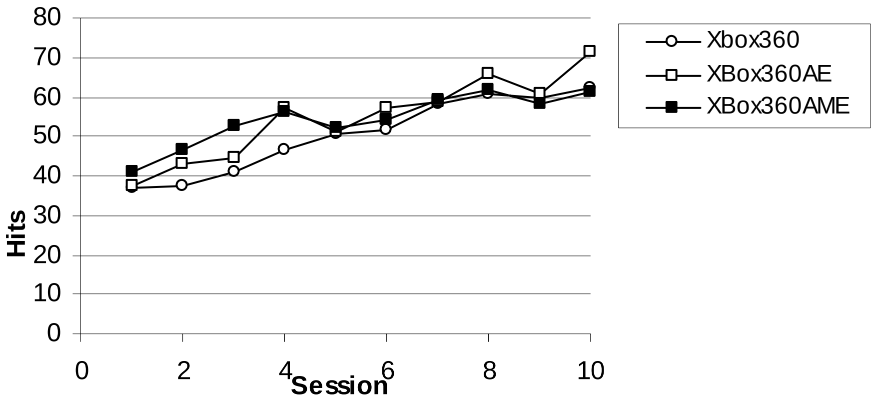


Figure 2. The average number of shots that hit a target in different input conditions.

The number of shots that missed the target, shown in Figure 3. A repeated measures ANOVA showed a statistically significant main effect of the input method ($F_{2,10}$=7.4, $p<0.05$) and session ($F_{9,45}$=2.7, $p<0.05$)), but no statistically significant interaction. With the help of Figure 3 we can interpret these results as the participants achieving the competitive number of hits the eye tracking conditions through more liberal use of ammunition. In both conditions with eye tracker based aiming they fired over twice as many shots that missed the targets as in the pure gamepad condition. The explanation for this behavior is that we did not filter the eye tracker data in any way. Consequently, the aiming reticle that followed the user's gaze was jittery. The users felt that it jumped around randomly, and adopted the strategy of looking at a target and shooting repeatedly until the target disappeared. When this is done for distant targets, several shots may be needed before the reticle happens to jump on the target at the same time with a round being fired.

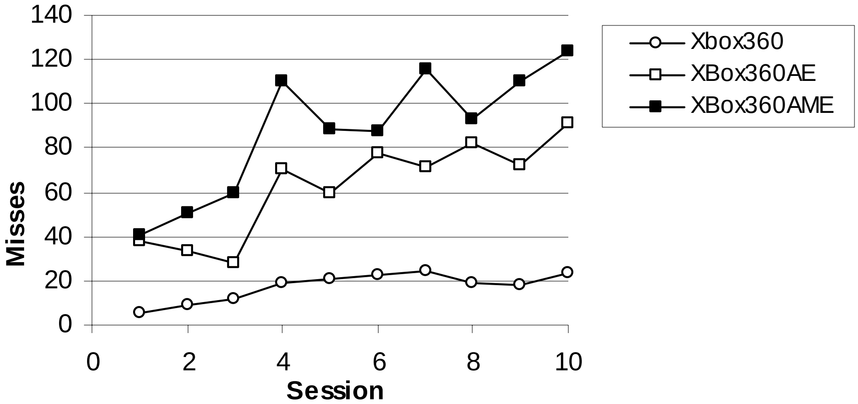


Figure 3. The average number of shots that missed all targets in different input conditions.

In an earlier study with the same environment (Isokoski & Martin, 2007) we observed that when an accurate pointing device (mouse) is used for aiming, players tend to shoot from further away than with a less accurate device (gamepad with analog joysticks). Our results on this issue are mixed. There is no main effect of input method ($F_{2,9}$=2.4, p=0.13), but there is a significant interaction of session and method ($F_{18,90}$=1.9, p<0.05) indicating that the shooting distance develops differently with different input methods under training.

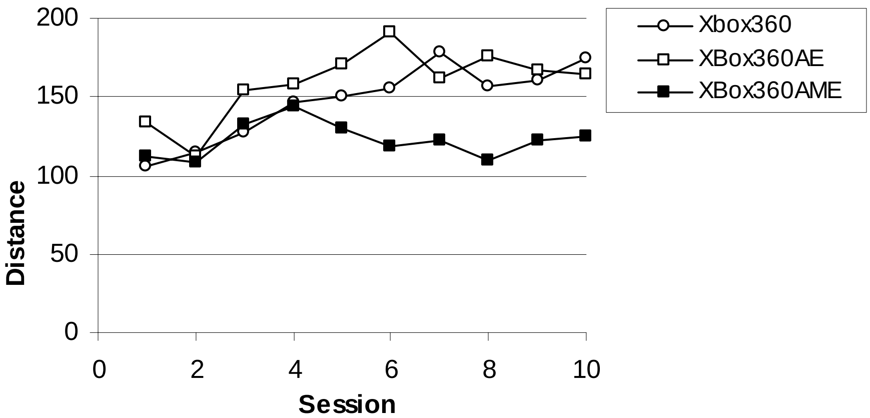


Figure 4. The average distance from the player to the target at the time of a hit under different input conditions.

With the help of Figure 4, we can in interpret the interaction as smaller effect of training on the shooting distance in the input condition with eye-tracker aiming and steering (Xbox360AME). One possible explanation for this is that approaching the targets was easy in the AME condition. The players only needed to look at the target while pressing the trigger that controls their velocity.

Conclusion

Based on our data we can confirm that eye tracker input can compete in killing efficiency with gamepad input in FPS games. However, eye tracker input tends to lead to a higher proportion of missed shots which can be a problem in games with restricted ammunition.

Our ultimate goal is to minimize the use of hands in FPS gaming to make this hobby accessible to those without the fine manual dexterity required for gamepad, mouse, and keyboard use. We are currently working on mapping head movements to the velocity control to eliminate first of the two buttons that remained hand controlled in the experiment reported above. The trigger problem seems harder to solve. Trigger presses need to be fast and well timed. We have not yet found an efficient way to produce such events with an eye tracker.

Acknowledgments

We thank Laurent Gomilla and Maurice Svay for their excellent work on the FPS-like testing environment and the participants for their effort.

References

Hand, C. (1997) A Survey of 3D Interaction Techniques. *Computer Graphics forum*, 16 (5), Blackwell Publishers, Oxford MA, pp. 269-281.

Isokoski. P., and Martin, B. (2006) Eye tracker input in first person shooter games. *In Proceedings of the 2nd Conference on Communication by Gaze Interaction*, Communication by Gaze Interaction (COGAIN), pp. 76-79.

Isokoski, P., and Matin, B. (2007) Performance of Input Devices in FPS Target Acquisition, *In Proceedings of the International Conference on Advances in Computer Entertainment Technology*, ACM Press, pp. 240-241.

Jönsson, E., *If Looks Could Kill – An Evaluation of Eye Tracking in Computer Games*, Master's Thesis, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, 2005.

Smith, J. D., Graham, T.C.N. (2006). "Use of Eye Movements for Video Game Control ". *In Proceedings of ACM Conference on Advancements in Computer Entertainment Technology (ACE 2006)*, ACM Press, Article no. 20.

Gaze beats mouse: a case study

Michael Dorr, Martin Böhme, Thomas Martinetz, and Erhardt Barth
Institute for Neuro- and Bioinformatics, University of Luebeck
Ratzeburger Allee 160, D-23538 Luebeck, Germany
{dorr, boehme, martinetz, barth}@inb.uni-luebeck.de

Keywords
Gaming with gaze; human-computer interaction; alternative input devices

Introduction

Eye tracking has become cheaper and more robust over the last years. Soon it will be feasible to deploy eye tracking in the mass market. One application area in which an average consumer might benefit from eye tracking is in computer games, where gaze direction can add another dimension of input. Progress in this direction will also be of high relevance to disabled users who lack the dexterity to control the input modalities traditionally used in computer games. Not only could gaming with gaze be enjoyable in itself, but the virtual world of multi-player games might also be one arena where disabled users could meet non-disableds on an equal footing.

However, for a satisfactory gaming experience, it does not suffice to simply replace the mouse with a gaze cursor; usually, changes to the game play will also have to be made.

In this paper, we will present an open-source game that we adapted so that it can be controlled by either a mouse or by gaze direction. We will show results from a small tournament that indicate that gaze is an equal if not superior input modality for this game.

Breakout

Breakout was one of the first commercially available video games when it was released in 1976 (Kent, 2001). Its game play was based on *Pong*, where the player has to move a paddle horizontally to hit a ball that is reflected at the borders of the game area. *Breakout* now extended this concept by putting bricks in the upper part of the game area which dissolved upon contact with the ball; the goal of the game was no longer to keep the ball in the game as long as possible, but to destroy all bricks (see Fig. 1 for a screenshot). This simple, easy-to-understand game play makes *Breakout* still appealing today, more than 30 years after it was first sold. Countless clones have been published for various computer platforms, with better graphics and extras that are released on explosion of bricks and need to be collected with the paddle. The one-dimensional nature of paddle control in *Breakout* and *Pong* also makes these games suitable for input modalities other than a joystick or mouse, e.g. brain-computer interfaces (Krepki, Blankertz, Curio, & Müller, 2007) or pitch of voice (the Sony SingStar console game). In the following, we will describe our version of *Breakout* which was adapted to be controlled by gaze.

Figure 1. Screenshot of *LBreakout2*.

Implementation

Our gaze-controlled version of *Breakout* is based on the open-source game *LBreakout2*.[1] *LBreakout2* is published under the GNU General Public License[2] (GPL), so that the game can be freely modified under the condition that the modifications will only be released under the GPL as well.[3] This open source approach is especially appropriate for such (currently) small markets as that for games geared towards those with severe motor impairments.

LBreakout2 is written in C and uses the Simple Media Layer[4] for graphics, sound, and network functionality. We have modified it to work with SensoMotoric Instruments eye trackers, which use an ASCII network protocol sent over a UDP link, so that no additional libraries are required. The major change to the source code was to implement a function that waits on a UDP socket for samples from the eye tracker and decodes them; instead of the paddle position being shifted by mouse movements, it is now set in absolute coordinates to the gaze position of the user.

Strictly speaking, it is not even necessary to calibrate the tracker from inside the game. A first version of the game used an external tool to calibrate the tracker to the screen before the game was started; especially for demonstration purposes, where several players take turns, the constant need to shut down and restart the game led us to implement a calibration procedure that can be started by a key press from inside the game.

Adaptation of game play

To prevent the ball from going out of play, the paddle needs to be at the same horizontal position as the ball when the ball reaches the lower end of the screen. When the paddle is controlled by gaze, this means that, in principle, the player only needs to look at the position where the ball will meet the paddle. That this is very intuitive might be demonstrated by the following anecdote: During the CeBit trade fair show,

[1] See http://lgames.sourceforge.net.

[2] See http://www.fsf.org/licensing/licenses/gpl.html.

[3] The source code of our modifications is available on request; we would like to receive feedback and/or incorporate changes made by the community.

[4] See http://www.libsdl.org

we presented our game to a visitor who claimed to have had no experience with computer games at all. After calibration, she started playing and performed very well until, about 2 minutes into the game, she asked when "the whole thing would actually start". Apparently, she had just constantly looked at the ball (and therefore always hit it with the paddle) without even realizing that the paddle followed her gaze!

Although playing with gaze is very intuitive, players naturally face other challenges in a gaze-controlled setting. A well-known problem for gaze-based user interfaces is how a user should confirm an action (the equivalent of a mouse click). In *Breakout*, a mouse click normally is needed to start the game and release the ball from the paddle. We solved this problem by releasing the ball automatically after 5 seconds when the game is played with gaze.

Another problem is that even the best eye trackers today still have calibration errors, so that the paddle position might be slightly shifted from the "true" gaze position. This can be highly irritating and must be consciously compensated for by the player, even though there are no fixation targets at the location that needs to be fixated. Also, by carefully adjusting on which side of the paddle the ball is deflected, the player can control the direction in which the ball is sent off again. Due to tracking noise, this is much harder with gaze than with a mouse, but it seems that gaze players get better at this with some training.

In the *Breakout* version we have adapted, bricks that are destroyed sometimes release "extras" that fall towards the bottom of the screen. Once they are collected with the paddle, they alter the game by, for example, increasing the speed of the ball or making the ball explosive (so that several bricks can be destroyed at once). Some of these extras require a reaction by clicking the mouse, so we removed them from the game using the integrated level editor. Other extras should not be collected by the player because they have a negative impact; carefully avoiding to look at something in a dynamic environment takes a conscious effort and some training by the player. One extra that is particularly enjoyable in gaze-playing mode, though, is the extra ball. Because of the much higher speed at which the eye can travel compared to the hand, it is possible to keep several balls in play simultaneously. Keeping track of a number of dynamic objects while still maintaining fixation on the ball that is going to reach the bottom of the screen next was found to be highly entertaining by our test subjects.

Pitting gaze against mouse

LBreakout2 also offers a multi-player mode with one paddle at the bottom and one at the top of the screen. The goal is to play the balls in such a way that the opponent cannot return them. To make the game more lively, every player can fire up to 3 balls so that up to 6 balls are in the game simultaneously.

To test how well our gaze-based interface fared against the mouse, we set up a little tournament in which pairs of players took turns playing against each other. First one player controlled the game with gaze and the other with the mouse, then the roles changed.

20 undergraduate and graduate students from our department volunteered. 4 had been involved in writing or presenting the game before; the other 16 had had little or no eye-tracking experience and had not played the game before. To ensure a fair game, we also matched pairs by their general computer game experience. Eye movements were recorded with a SensoMotoric Instruments iViewX Hi-Speed tracker running at 240 Hz. We also have successfully played the game with a 50 Hz SMI RED-X remote tracker, which obviously is better suited for gaming because it does not require the player's head to be fixed.

After calibration, the gaze player was able to try out the gaze control for about 15 to 30 seconds. Then, the match started. Each match lasted 5 rounds. For every ball that the opponent could not return with their paddle, a player scored 1 point; a round was won by the first player to win 10 points. Every round was set on a different background, i.e. the layout of ball-deflecting bricks in between the players changed. Such bricks close to the player's baseline are a slight disadvantage for the gaze player because it is easier to aim shots exactly with the mouse (see above). The results are shown in Fig. 2.

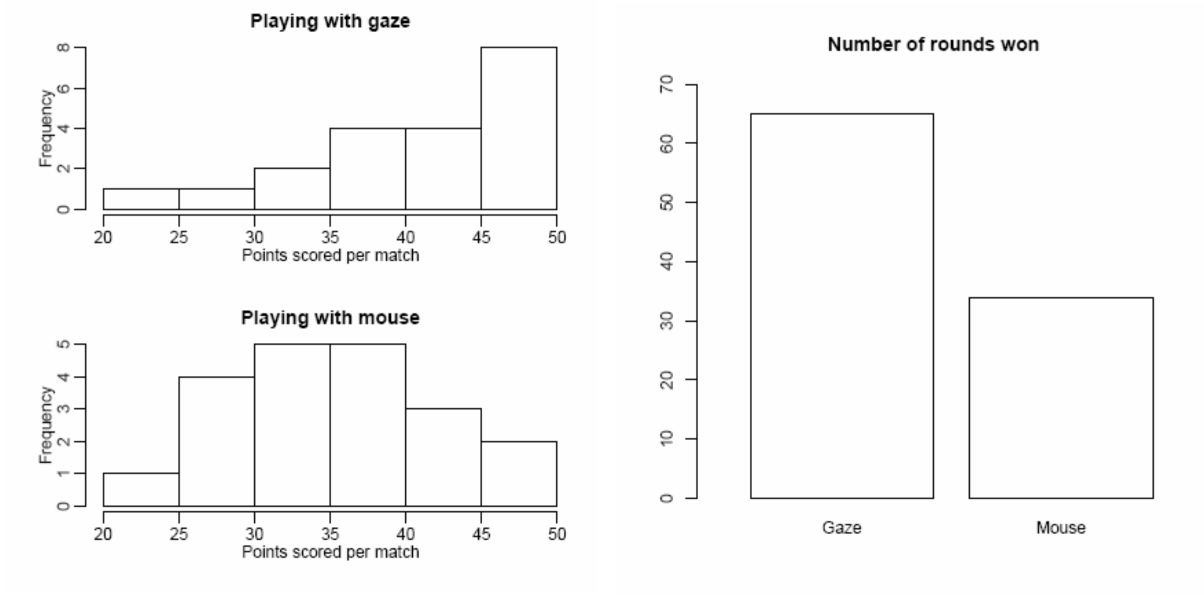


Figure 2. Left: Distribution of scores. Right: Number of rounds won.

Clearly, playing with gaze yielded a higher score on average (41.95 vs. 36.25). Almost two thirds of all rounds (65 out of 99, one data set had to be discarded because the tracker had lost the pupil temporarily) were won by the gaze player. Gaze control thus was a statistically significant advantage ($p < 0.0015$).

Conclusion

We have presented modifications to the open-source game LBreakout2 that allow the game to be controlled with gaze. Even though both the graphics and the game play of LBreakout2 are very simple, our test subjects found “playing with eyes” highly enjoyable. More importantly, we also have presented results that show that gaze-based interfaces can be superior to traditional input modalities even for users that have had no previous training with such interfaces.

Acknowledgements

Our research has received funding from the European Commission within the project GazeCom (contract no. IST-C-033816) and the Network of Excellence COGAIN (contract no. IST-2003-511598) of the 6th Framework Programme. All views expressed herein are those of the authors alone; the European Community is not liable for any use made of the information.

References

Kent, S. (2001). *The ultimate history of video games: From Pong to Pokemon and beyond.* Prima Life.

Krepki, R., Blankertz, B., Curio, G., & Müller, K.-R. (2007). The Berlin Brain-Computer Interface (BBCI): towards a new communication channel for online control in gaming applications. Journal *of Multimedia Tools and Applications.* (in print)

Eye Trackers: Are They Game?

Javier San Agustin, Jakob Schantz, John Paulin Hansen
IT University of Copenhagen
Rued Langgaards Vej 7, 2300 Copenhagen S, Denmark
{javier, whyweb, paulin}@itu.dk

Keywords
Eye tracker performance, videogames, gaze tracking

Introduction

Videogames are a constantly growing market. A revolutionary way of controlling games was introduced last year by Nintendo with their console Wii. Potentially, gaze represents a fast and natural input method that could also be exploited in game interaction. If we can find use for eye trackers in computer games it may increase the availability of hardware and decrease the price (c.f. Hansen et al. (2005)).

Videogames usually require performing two different tasks: target selection, i.e. pointing at a specified object on the screen plus selecting it; and target tracking, where the user is required to track a moving target with the pointer, for instance to shoot it. While recent studies (e.g. Sibert and Jacob, 2000) have evaluated target selection with gaze interaction under the Fitts' Law framework, we are not aware of any studies of eye tracking systems used for smooth pursuit target tracking. The last kind of studies would be fundamental to the development of a branch of interesting gaze controlled computer games, for instance first person shooters.

In this paper, we evaluate and compare the performance of six different input devices, namely mouse, touch-screen, joystick, head tracker and two eye trackers, Tobii 1750 and QuickGlance v3 on a game-like target selection and -tracking task. The differences between the two eye trackers are of special interest to our study, since it may indicate if games can be used to compare eye tracking systems, and thus serve as a benchmark test that is motivating to use, even during long-lasting learning experiments.

Previous work

Many studies have been carried out to evaluate the performance of different input devices in target selection tasks. The most common metrics used to quantify the performance are accuracy and speed. Accuracy is given as the error rate of selections with the pointer outside the target. Speed is reported in its reciprocal form, movement time (MT). Both measures are combined in the performance measurement used in the ISO standard for evaluating pointing devices (ISO 9241-9), *throughput*.

$$Throughput = \frac{ID_e}{MT} = \frac{\log_2\left(A/W_e + 1\right)}{MT}$$

ID_e is the effective Index of Difficulty expressed in bits. It depends on the distance to the target, or amplitude A, and the effective width of the target measured along the axis of movement W_e. The Index of Difficulty measures how difficult it is to hit the target. The validity of throughput as a measure of device performance for selection tasks is discussed in Douglas et al (1999). MacKenzie et al (2001) propose new accuracy measures in target selection tasks that help discriminating between different devices.

There are few studies on the performance of input devices on target tracking tasks. The obvious metric to measure the accuracy of a device is the time on target, i.e. the percentage of time that the pointer is positioned on the target. Klochek and MacKenzie (2006) compared the performance of a mouse and a gamepad in a three-dimensional tracking task. They introduced several metrics to measure the accuracy and smoothness of each device. In there experiment, most of the information was conveyed on the time on target metric, though. No experiments have been found regarding eye trackers and target tracking tasks.

Tasks and performance metrics

Target selection tasks require the user to point on a target and activate a button to select it. In our study, 16 targets are presented sequentially as proposed in ISO 9241-9. The layout circle has a radius of 250 pixels, while the targets can have two different sizes, 75 and 150 pixels (diameter). The nominal indexes of difficulty are thus 2.3 and 1.6 bps. The performance metric used in this task is *throughput*.
Tracking a target requires the user to keep the pointer on the target while this moves on the screen. A target might move with a constant velocity or have some acceleration. In this study, targets move with a constant velocity of 90 pixels/second, and they move from their original starting point to the centre of the screen. Two possible ways to alert the user when the pointer is not on target are sound feedback, which alerts the user by emitting a sound, and movement feedback, which alerts the user by stopping the target. In our experiment, we tested sound feedback, and a combination of sound and movement feedback. The metric used to evaluate the performance is *time on target* (TOT).

Method

Six participants took part in the experiment. All of them were regular mouse users and had previous experience with joystick devices. Three of them had previous experience with eye trackers and one of them with head trackers.

The software used was programmed in C# under the XNA framework, and run at a constant frame rate of 30 fps. The input devices tested were mouse (Logitech optical mouse), finger on touch screen (Dell E157FPT), joystick (Logitech Attack 3), head tracker (NaturalPoint), and two eye trackers (Tobii 1750 and Quick Glance v3, both set with the minimum possible smoothing between images on estimated cursor position).

The experiment was a 6x2x2 within-subjects factorial design, with factors the factors **device** [mouse, finger, joystick, head tracker, Tobii, Quick Glance], **feedback** [sound, sound + movement] and target **size** [75 pixel or 150 pixel].

Each participant conducted the tests with all the devices, starting always with the mouse. The order of the other five devices was counter-balanced across participants using a balanced Latin square. Prior to starting the experiment participants familiarized themselves with the task in a warm-up trial using the mouse. All blocks were performed in one day and the total experiment lasted about two hours with short breaks between each device.

In each block, 16 targets were displayed consecutively in a random order. Targets were arranged on a circular layout with a radius of 250 pixels. Once the participant selected the target, it started moving towards the centre of the screen with a constant velocity of 90 pixels/second. The target disappeared when reaching the centre, and an X appeared in its place. The participant had to point at the X to release the next target. This ensured that for every target the starting position of the pointer was the centre of the screen. For each device participants run 4 blocks with different target sizes and feedbacks in a balanced order.

Results

Data analysis was performed as a 3-factor ANOVA, with **device** (mouse, finger, head, joystick, Tobii or Quick Glance), **target size** (75 pixels or 150 pixels) and **feedback** (sound or sound plus movement) as the independent variables (subjects were treated as repetitions). Throughput and time on target (TOT) were analyzed as the dependent variables. An average of the 16 trials conducted under each block were calculated for each subject. All data were included.

The grand mean of throughput was 1.85 and there was a significant effect from device on throughput $F(5, 143) = 13.92$, $p < 0.0001$. Finger had the highest throughput (mean = 2.42) and it was significantly different ($p<0.01$, Scheffe post hoc test) from the head tracker (mean = 1.34) and the joystick (mean = 1.09). Mouse (mean = 2.21) was also different from head tracking and joystick. Both of the eye trackers (Tobii mean = 1.91 and Quick Glance mean = 2.11) were different ($p<0.005$) from the joystick and Quick Glance was also different from the head tracker ($p<0.05$). The eye trackers did not differ significantly. Figure 1.a) shows the throughput of the different devices for each target size.

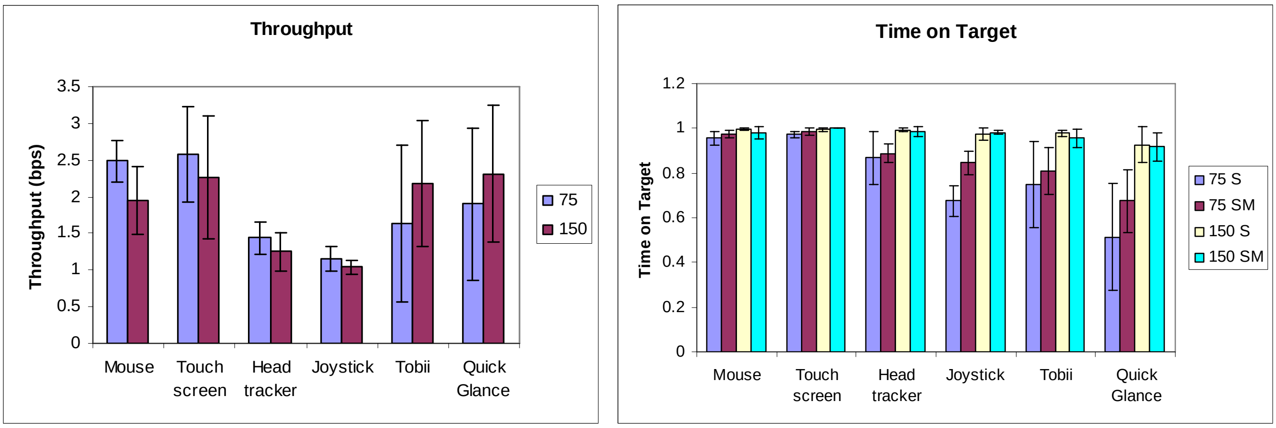


Figure 1. a) Mean throughput for each device and target size. b) Mean TOT per device, target size and feedback.

The grand mean of TOT was 0.90 and there was a significant effect from device on TOT $F(5,143) = 25.60$, $p<0.0001$. Target size also had an significant effect on TOT, $F(1,143) = 115.45$, $p<0.0001$. The mean of TOT was 0.83 on small targets and 0.97 on big targets. Feedback had a significant effect on TOT $F(1,143) = 6.02$, $p<0.05$. The mean TOT with sound feedback was 0.88 while adding movement to the sound feedback increased the TOT to 0.92. Figure 1.b) shows the mean TOT for each device, target size and type of feedback.

The interaction between size and device was significant on TOT $F(5,143) = 12.75$, $p<0.0001$, c.f. Figure 2. The post hoc test showed that the difference between the QuickGlance device and the other 5 devices was significant for the small 75-pixel target ($p<0.001$). The Tobii tracker was different from the mouse and the finger ($p<0.0001$), but not from the joystick or head tracker under this condition. None of the devices differed under the large 150-pixel target condition.

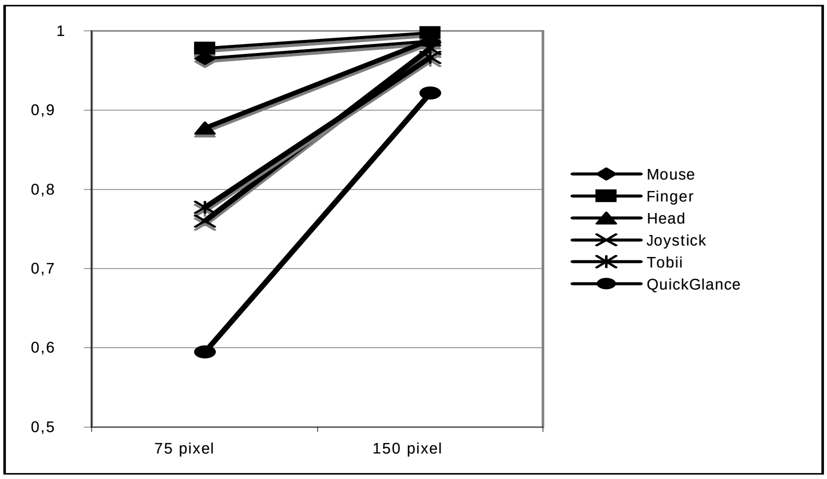


Figure 2. TOT for a tracking task with 6 different control devices on two different target sizes, N = 6.

Discussion

The results show potentials for eye trackers to be used in videogames. Although we did not find a higher throughput of the gaze compared to the mouse - like Sibert and Jacob (2000) did in their study - the gaze throughput in our experiment was higher than the throughput of a joystick used in many games. QuickGlance also performed better than a head tracker that is frequently used in games and as an alternative input device for disabled people.

Unlike the other devices, both eye trackers improved the throughput when target size increased. Bigger targets compensate for miss-calibrations and possible offsets in the estimated cursor position. Interfaces designed especially for gaze-based interaction should preferably be designed with big target areas to benefit an eye tracker. The visual part of a target need not be as big as the targets functional hit-area, so a gaze controlled game may well contain tiny targets that are difficult to discover – but easy to hit, once they are detected.

The time-on-target performance for small targets was relatively poor for both eye trackers, and QuickGlance was especially low under this condition. Maintaining the pointer on the target can be challenging if the eye tracker is not accurate enough. In some of the popular shooting games it is equally important to aim as quickly as possible as it is to track a target while it's moving. In these games players with an eye tracker are likely to lose, unless the targets are wide. We did not study the effect of target speed in our experiment, but it would be interesting to see, for instance, if gaze could outperform other input modes when following high-speed targets.

The only significant difference we found between the two eye trackers showed up in the time on target measure when tracing small targets. However, we consider the ability to differ between two gaze trackers as a promising starting point for the development of a more advanced testing procedure that might include several different target sizes and probably also different target speeds. The 75 pixel target was especially difficult to trace with one of the trackers, and 75 pixels is actually a rather big target when compared to the interactive elements on a standard Windows interface, that sometimes span less than 20 pixels. So there are good reasons for including even smaller targets in a test than what we did.

Our subjects only tried each device four times, while real gamers will play over and over again before they master a new controller. We only consider the present study to be preliminary to a more conclusive long-lasting learning experiment. The code used in our experiments could relatively easy be converted it into an arcade game with persuasive graphics and sound effects, where the player would have to fulfil goals at different levels. A game score could be calculated from the throughput and time on target performance metrics every time the user plays the game. This score would be sent to the game's website, where the user could see a ranking showing how well they have performed against other players with different eye trackers. Data collected in this distributed and collaborative way could then be used for evaluation and comparison purposes.

References

Douglas, S. A., Kirkpatrick, A. E., and MacKenzie, I. S. (1999). Testing pointing device performance and user assessment with the ISO9241, Part 9 standard. *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*, pp. 215-222. New York: ACM.

Hansen, John Paulin; Hansen, Dan Witzner; Johansen, Anders Sewerin; Elvesjö John (2005). Mainstreaming Gaze Interaction Towards a Mass Market for the Benefit of All. *Proceedings of 3rd international conference on universal access in human-computer interaction.*

Kolcheck, C., and MacKenzie, I. S. (2006). Performance Measures of Game Controllers in a Three-Dimensional Environment. In GI '06: Proceedings of the 2006 conference on Graphics interface, pages 73–79, Toronto, Ont., Canada, Canada, 2006.

MacKenzie, I. S., Kauppinen, T., and Silfverberg, M. (2001). Accuracy measures for evaluating computer pointing devices. *Proceedings of ACM CHI 2001, 9-16.*

Sibert, L. E. and Jacob, R. J. 2000. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM Press, New York, NY, 281-288.

Gameplay experience based on a gaze tracking system

M. Perreira Da Silva, V. Courboulay, A. Prigent
L3i, Université de La Rochelle,
Av Michel Crépeau, 17042
La Rochelle, France
{mperreir, vcourbou, aprigent}@univ-lr.fr

Keywords
Gaze-based interaction, game play, low cost gaze tracking systems

Introduction

Human-computer interaction is an important point in the development of software. In the field of games, it is a way to increase immersion in the virtual world for the player. Classical interactions with mouse, keyboard or gamepad, are limited in comparison with the reality of graphics displayed. Indeed, a big interest is to concentrate on new kinds of interfaces between the player and the virtual world. For example, some approaches are using a headpiece device to detect head movements in order to change the game camera direction. In this paper, we concentrate on using gaze direction as a new kind of game play. We intend to interact with narrative elements of the game without using any intrusive equipment for tracking. We present a prototype that has been developed in the framework of emergent narrative. We also describe how gaze tracking can be used in the framework of player behaviour detection for game play and adaptive narrative purpose. Indeed, we have developed a 3D adventure game in a virtual environment that represents our laboratory of computer science. An extension of this game will be to use methods that have been described above to allow the player to control the game by a gaze tracking approach. The other extension will be to use gaze detection to observe the player behaviour (for example stress, attention...) and adapt the game scenario dynamically.

An adaptive adventure game

Usually video games are based on linear narration which reduces significantly the field of interactions between the player and the environment he evolves in. In this case, we talk about linear games. The challenge is then to increase the player's freedom.

This is one of the reasons that explain why there is an increasing interest in narration in video games. Narration creates the drama and develops interest by creating challenges the player has to overcome. Narration should adapt the game unfolding in order to take into account the various levels of emotions (stress, frustration, rewards, etc.) needed to maintain the player's attention.

In this context, we have developed an adventure game based on the visit of a virtual world that represents the computer science laboratory (L3i) of the University of La Rochelle. The player has to explore the laboratory by opening doors that are closed. The figure 1 is a screen capture of the prototype that has been developed with the Unreal Engine editor.

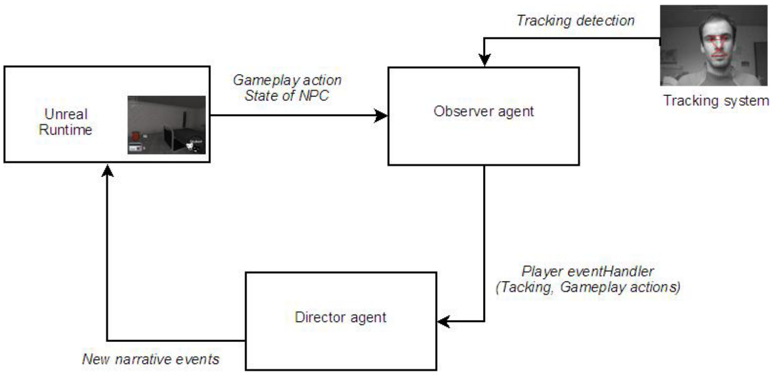


Figure 1. Left: A screen capture of the prototype. Right: Overview of a narrative based game architecture.

The game concept is the following one. The player is a student that has a fixed delay to give a work to his teacher. He is in direct competition with an *evil* student that tries to prevent the player from reaching his goal and with a *little pest* that tries to steal his work and give her own work first to the teacher.

We propose to give to the player a maximum amount of interactivity while keeping a robust and interesting narrative framework. The approach of emergent narrative consists of a particular architecture that increases player actions freedom and produces a dynamic control of narrative quality. A challenge is, for example, to detect the player's behaviour in order to modify dynamically the scenario.

An interactive architecture

Game architecture

In (Champagnat, Prigent, & Estraillier, 2005) we have proposed an architecture for interactive storytelling. Like (Magerko & Laird, 2003; Young & Saretto, 2003), this architecture is made up of a process simulation and a story director (or planner). Figure 1 gives an overview of the architecture.

This architecture defines a set of agents that catches player's inputs, analyses the game unfolding and computes an adaptive execution of narrative (as a feedback to player's inputs).

The narrative controller proposes a consistent unfolding. According to the story director's analysis, it can enable or disable parts of the story. The narrative controller catches actions by means of an observation agent. This observer is being configured by the narrative agent (it gives a set of expected actions) at each execution step.

Let us give, in the sequel, a brief presentation of the agents that will interact with de gaze tracking system:

- *the observation agent* is in charge of capturing the player's behaviour. For example, the player chooses to open a door by clicking on it or by pressing the space bar key on its keyboard. The observation agent interprets this explicit action on controls (*i.e.* the procedures of the game) and translates it in a player action (in the sense of an action of the narration) corresponding to the player's choice. In our new gaze based interaction architecture, the observation agent also receives informations from the gaze tracking system (head pose, gaze direction, etc.).
- *the narrative agent* performs a supervisory control of the storytelling. It receives the players's actions from the observation agent, and determines the set of possible game events. It is also in charge with defining the parameters of the observation agent.

A low cost, robust gaze tracking system

The tracking algorithm we have developed is built upon three modules which interoperate together in order to provide a fast and robust eyes and face tracking system (see Figure 2).

- *The face detection module* is responsible for checking whether a face is present or not in front of the camera. In the case a face is present, it must also give a raw estimate of the face and face features (eyebrows, eyes, nostrils and mouth) 2D position in the image.
- *The face features localization module* finds the exact features position. When all features position are known, we use the method derived from (Kaminski & Shavit, 2006) to estimate the 3D position and orientation of the face. Gaze direction is processing by combining face orientation estimation and a raw estimate of eyeball orientation processed from the iris centre position in the eyes (Zhu & Yang, 2002).
- *The features position prediction module* processes the position of each feature for the next frame. This estimate is built using Kalman filtering on the 3D positions of each feature. The estimated 3D positions are then back projected to the 2D camera plane in order to predict the pixel positions of all the features. Then, these 2D positions are sent to the *face features localization module* to help it process the next frame.

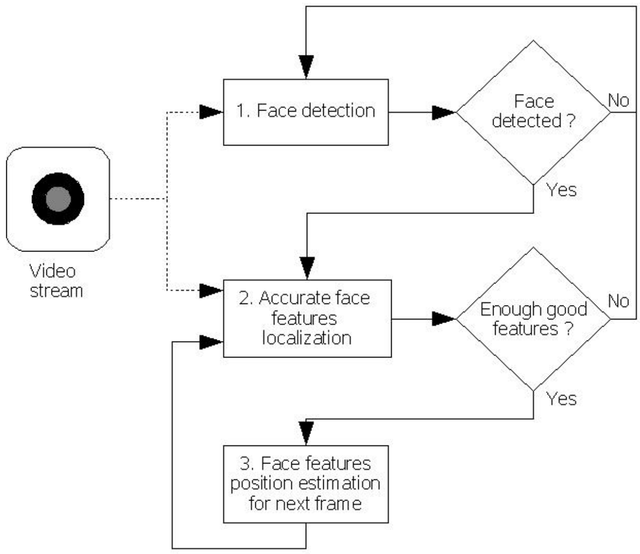


Figure 2. General architecture of the face and eye/gaze tracking algorithm.

During the system development, we focused on robustness. As a consequence, the system accuracy is lower than commercial systems like *Facelab*, but our solution is much less expensive and allows gaze tracking in a broader range of head poses.

Game / Gaze interaction

During our first experiments, we only took into account a few explicit player's behaviour:

- Firstly, we have focused on the interaction with the non player character of the *little pest*. She tries to steal the work of the player. The player can interact with the girl by doing a wink to the camera when he is in front of the girl. Then, the girl will give his work back to the player if she has stolen it to him.

- Secondly, we allow the player to protect himself against the *evil* student by looking down. Indeed, if the *evil student* arrives near the player, this one will put his head down and then the *evil* student will go on without stealing his work.

We have shown how gaze detection can be used for gameplay purpose. We now explain that implicit behaviour can be detected and how the game can dynamically adapt the scenario following these observations. We are currently working on the integration of more complex kinds of behaviour into our interactive game framework:

- First, we will use gaze tracking to observe the level of attention of the player. For example, if the player stops watching at the screen, a particular game action can be launched to refocus his attention. In L3i Life for example, in this kind of situation, the adaptive architecture can modify the unfolding of events, and make the evil student run after the player to steal his work. It is a stressing action that can bring some interest back to the player.
- Another possible observation is the stress of the player. There are a lot of possibilities for detecting stress. The scenario we have chosen to detect is the following one: the player keeps his head near the screen without any movement. We will interpret this behaviour as a big stress situation. During game level design, the main purpose is to guarantee a variation of stress during game execution. Using our stress observation method will allow to detect the moment when the player is stressed and give him some easy action to perform (no more enemies for a while) until the stress decreases.

Conclusion and perspectives

During our first experiments we have observed that this new kind of interaction improves the players immersion in the virtual world of the game. Moreover, it also increases its interest for the game because the gameplay is richer and the game more *fun* to play. These observations will certainly be confirmed once our framework will include the observation of implicit behaviour as we will have more real-time feedback on the user's gaming experience.In this paper, we have presented a system which reacts in a dynamic way thanks to the observation and analysis of the player's behaviour. We described the architecture of a general framework for game adaptation using gaze as one of its inputs. The principle of the interaction is based on the extraction of information according to the spatial and temporal context. The observation and the analysis of behaviour consists in determining the behaviour of players from their actions, by taking into account the scenario. It has proven to improve the gaming experience for adventure games. However, more behaviours are needed to improve and validate the proposed models and architecture.

References

Champagnat, R., Prigent, A., & Estraillier, P. (2005, June). Scenario building based on formal methods and adaptative execution. In Isaga. Atlanta (USA).

Kaminski, M. K. D., J.Y. Teicher, & Shavit, A. (2006). Head orientation and gaze detection from a single image. In International conference of computer vision theory and applications.

Magerko, B., & Laird, J. (2003, March 24 26). Building an interactive drama architecture. In 1st international conference on technologies for interactive digital storytelling and entertainment, tidse. Darmstadt, Germany.

Young, M. B. M. J. A. M. R., R.M. Riedl, & Saretto, C. (2003). An architecture for integrating plan-based behavior generation with interactive game environments. Journal of Game Development.

Zhu, J., & Yang, J. (2002). Subpixel eye gaze tracking. Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 131.

Dwell time reveals a narrowing of active options during selection in multi-element arrays

Mackenzie G. Glaholt, Eyal M. Reingold
University of Toronto at Mississauga
Department of Psychology
3359 Mississauga Road N. RM 2037B
Mississauga, Ontario, Canada L5L 1C6
{mackenzie, reingold}@psych.utoronto.ca

Keywords
Gaze bias, decision support, visual attention, preference, selection

Introduction

There is ample evidence from basic research that eye movements are preceded by an attentional shift to the saccadic target, and consequently the spatial distribution of eye fixations is a good indirect measure of the distribution of visual attention (e.g., Deubel & Schneider, 1996; Henderson, 1993; Hoffman, 1998; Remington, 1980; Rizzolatti, Riggio & Sheliga, 1994; Schneider & Deubel, 1995; Shepherd, Findlay & Hockey, 1986). Given the tight coupling between attention and eye movements, it is not surprising that measures of gaze duration and location have proven to be invaluable in the context of human factors, usability engineering and marketing research (see Duchowski, 2002 for a recent review).

In many real world and computer based applications the user is confronted with a cluttered, dynamically changing visual environment in which objects and locations are serially and often repeatedly selected for detailed or attentive processing. The cost of examining items on a visual display becomes readily apparent in a situation where the user is confronted with a large array of possible alternatives and must select one according to their preference (e.g. online shopping). In the context of such a task, the distribution and duration of eye fixations would potentially provide an excellent measure of the observers interests and preferences. Eye movement information may be used as decision support in a preference or selection task by directly identifying the preferred or selected item or by reducing a very large set of available items down into a smaller, relevant set based on the users' manifest preferences.

Several recent studies suggest that such an effort may be feasible and promising. Shimojo, Simion, Shimojo & Scheier (2003) reported a gaze bias that exists during preference decision between 2 visually presented items. Gaze was shown on average to dwell longer on the face that was later selected. Based on this finding, Bee, Prendinger, André & Ishizuka (2006) demonstrated the feasibility of predicting the visual preference decisions of users in real-time, for the purpose of designing applications that would automatically detect users' visual preferences solely based on eye movement in a two-alternative forced choice (2-AFC) setting. These authors reported that in a pilot study involving the selection of neckties, their system correctly classified subjects' choices with an average accuracy of 81% (with 50% constituting chance performance).

The goal of the present research was to examine the usefulness of gaze information in predicting choice, and narrowing the set of potential alternatives, during preference decisions in a multi-element array. Each participant was asked to select the most attractive stimulus in visual arrays of 8 faces (group 1) or arrays of 8 company logos (group 2) (see the 8-AFC task, Figure 1). We analyzed the dwell time on each of the stimulus items in the beginning and end of the trial. Our results show that by the end of the trial, participants' eye gaze is biased towards a subset of the stimuli that contain the item to-be-selected. This indicates that dwell time may be used to select a subset of relevant options from a large array.

Method

Participants

One group of five participants took part in the face version of the experiment and another group of five participants took part in the logo version of the experiment. All participants were students at the University of Toronto at Mississauga, and each of them received $10 compensation.

Apparatus

The eyetracker employed in this research was the SR Research Ltd. EyeLink 1000 system. This system has high spatial resolution (0.005º) and a sampling rate of 1000 Hz (1-msec temporal resolution). In general, the average error in the computation of gaze position was less than 0.5º of visual angle. The participant used a chinrest with a head support to minimize head movement.

Stimuli

Faces were constructed as unique combinations of 3 stimulus dimensions (eyes, nose, and mouth) with 8 possible features (i.e., exemplars) in each dimension. The features were stored as bitmaps and assembled into faces on each trial (see Figure 1). The assembled faces were centered within black outline boxes measuring 200x200 pixels, each spanning 6.25° of visual angle. Of the 512 possible faces in our feature space, we selected a set of 64 in which all pairs of features occur once, and all individual features occur eight times.

The logo version of the experiment was analogous to the face version. We constructed logos by combining features from each of three stimulus dimensions: font, shape, and texture (see Figure 1). All the logos were portrayed as possible logos for a fictional company with the initials 'TEK'.

Procedure

One group of participants was given the face version of the experiment, and another group was given the logo version. Each participant was given instructions prior to completing an eight-alternative forced-choice (8-AFC) task with 128 trials. All 64 faces (see Stimuli) were presented 8 times, across 64 stimulus displays, where each display contained a unique combination of 8 stimuli (see Figure 1). At the beginning of each trial the display appeared and the participant decided which of the eight stimuli (faces or logos) was the most attractive. Once a decision was reached, the participant looked at a grey dot located at the center of the display. Having fixated within the center region for 500 ms, the dot turned green to indicate that the selection-by-looking tool is active. The participant then looked at the preferred stimulus, terminating the trial.

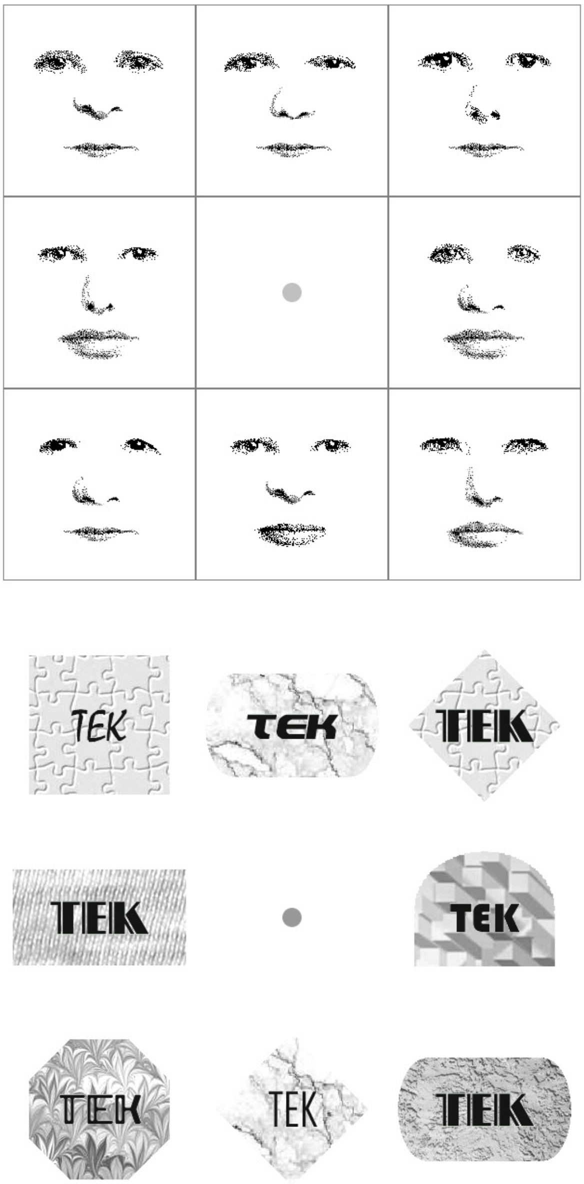


Figure 1. Sample stimulus displays for a single trial of the face (top) and logo (bottom) preference tasks. Having reached a decision, participants looked to the centre circle, which turned green, enabling them to select their choice by looking at it.

Results

We divided the eye-gaze data into “dwells”, where each dwell is defined as one or more consequtive fixations within one of the eight stimulus regions. Dwells were typically made up of two or three fixations. Trials with less than eight dwells were excluded, leaving rougly 90% of the trials for analysis. We focused on the first 4 and last 4 dwells, in order to detect evidence of selectivity and convergence. For each segment we computed several variables: number of different items examined (to a maximum of four), number of visits to the chosen item (to a maximum of two), mean dwell duration, total dwell time on the chosen item, proportion of trials in which the chosen item had the longest dwell, and the proportion of trials in which it was one of the two longest dwells. The mean for each measure, across participants, is displayed in Table 1.

Condition	Dwell Set	# of items visited (of 4)	# of visits to Chosen item	Mean dwell time (ms)	Chosen dwell time (ms)	Chosen is top dwell (prop'n)	Chosen within top 2 dwells (prop'n)
Faces	First 4	3.71	0.55	365	244	0.24	0.48
	Last 4	3.6	1.12	486	763	0.56	0.87
Logos	First 4	3.71	0.6	376	301	0.27	0.53
	Last 4	3.59	1.07	482	691	0.49	0.84

Table 1. Mean gaze data for the first 4 and last 4 dwells, for the face and logo preference tasks.

Our results suggest a change in gaze behaviour across the trial. Both faces and logos show the same trends, though in each case the Face condition shows a slightly stronger contrast between the First 4 and Last 4 dwells. The number of items visited differed only slightly between the First 4 and Last 4 dwells; the number of returns to a previously viewed item increases only slightly by the end of the trial. However, the number of visits to the chosen item rises sharply by the end of the trial. Dwell time measures, on the other hand, strongly differentiate the beginning and end of the trial. Mean dwell time increased sharply over the trial, and particularly, dwell time on the item-to-be-chosen increased substantially. The chosen item tended to have the longest dwell time. Our data indicate that dwell time could be used to identify the item that would be selected on roughly half the trials, where chance is 0.125. Furthermore, the chosen item was in the top two dwell times on a very high proportion of the trials (~0.85), indicating that dwell time may be especially useful in narrowing the set of alternatives down to a smaller set that contain the most desirable choices.

Conclusion

As might be expected based on previous findings (Shimojo et al., 2003; Bee et al., 2006), dwell time on the chosen item increased sharply by the end of the trial, and the chosen item had the longest dwell time on a high proportion of the trials. While dwell times evidently can be used to predict the item that a participant will select with appreciable accuracy, they appear to be even more powerful for identifying an active, relevant subset. By the end of the trial, gaze is biased towards a small number of items, one of which will eventually be selected. This is shown here to hold for two very different, but practically relevant, stimulus domains.

The narrowing of the 'active' set of items is a dynamic process that occurs over the trial, but the analysis presented here is post-hoc, in that the dwell times are computed after selection has occurred. Future work could seek to employ dwell times to narrow a large set of items within-trial, to see if it allows for savings in decision time. Additionally, it will be informative to compare the efficacy of dwell time in estimating selection, and in narrowing selection, for decisions other than preference.

References

Bee, N., Prendinger, H., André, E. and Ishizuka, M. (2006) Automatic Preference Detection by Analyzing the Gaze 'Cascade Effect'. In *Proceedings of the 2nd Conference on Communication by Gaze Interaction*, Turin, Italy, Sep 4-5, 2006.

Deubel, H. and Schneider, W. X. (1996) Saccade target selection and object recognition: Evidence for a common attentional mechanism. *Vision Research* 6, 1827-1837.

Duchowski, A. T. (2002) A breadth-first survey of eye-tracking applications. *Behaviour Research Methods, Instruments & Computers* 34(4), 455-470.

Henderson, J. M. (1993) Visual attention and saccadic eye movements. In *Perception and cognition: Advances in eye movement research*, G. D'Ydewalle and J. van Rensbergen, Eds. Elsevier, North-Holland, 301-303.

Hoffman, J. E. (1998) Visual attention and eye movements. In *Attention*, H. Pashler, Ed. Psychology Press, East Sussex, UK, 119-153.

Remington, R. W. (1980) Attention and saccadic eye movements. *Journal of Experimental Psychology: Human Perception and Performance* 6(4), 726-744.

Rizzolatti, G., Riggio, L. and Sheliga, B. (1994) Space and selective attention. In *Attention and Performance XV*, C. Umilta and M. Moscovitch, Eds. MIT Press, Cambridge, MA, 231-265.

Schneider, W. X. AND Deubel, H. (1995) Visual attention and saccadic eye movements: Evidence for obligatory and selective spatial coupling. In *Eye movement research: Mechanisms, processes and applications*, J. M. Findlay, R. Walker and R. W. Kentridge, Eds. Elsevier, North-Holland, 317-324.

Shepherd, M., Findlay, J. M. and Hockey, R. J. (1986) The relationship between eye movements and spatial attention, *The Quarterly Journal of Experimental Psychology* 38A, 475-491.

Shimojo, S., Simion, C., Shimojo, E. and Scheier, C. (2003) Gaze bias both reflects and influence preference, *Nature Neuroscience* 6(12), 1317-1322.

Session 2: Technology and Environments

Improved Low Cost Gaze Tracker

Detlev Droege, Thorsten Geier, Dietrich Paulus
University of Koblenz-Landau Universitatsstr. 1
56070 Koblenz
{droege, smartfx, paulus}@uni-koblenz.de

Keywords
Gaze tracking, low cost, multi platform

Introduction

Gaze tracking may be used to replace the use of a keyboard when a person cannot use fingers for typing. A gaze tracking system will analyze eye movements and the characters which the person wants to type will be displayed on the screen. This kind of interaction with the computer is of particular help for handicapped persons.

Based on previous work, described in (Fritzer, 2005) and published in (Droege et al., 2005) and (Droege et al., 2006), we improved our system to work even more stable and to be installed on different computers. We also extended the interfaces to backends, i.e. those pieces of software, that are responsible for linguistic inference, word completion, etc.

Given the high costs for commercial gaze tracking systems, a second focus of our work is to establish a system with inexpensive parts. While it does not yet work entirely with *commercial of the shelf (COTS)* parts, most of them fulfil this aim.

Improved Approach

In general, the same hardware setup is used as in our previous approach described in (Fritzer, 2005). It consists of a high-sensitivity b/w camera (Sony EXView HAD CCD chip), equipped with a simple NIR-filter letting only NIR wavelengths pass and a set of IR-LEDs to produce a corneal reflection on the user's cornea. In contrast to the previous setup, the IR-LEDs are now positioned below instead of besides the camera. This avoids shadowing the opposite eye by the user's nose and thus supports the usage of reflections in both eyes. The setup is shown in Figure 1 and Figure 2. To test different distances between the camera and the user, the optical devices were mounted on a rack. In most cases, only three of the nine IR-LEDs mounted on the rack are used, as they already provide sufficient light intensity to produce a reliably detectable reflection on the cornea.

Our new implementation of the system uses the OpenCV library[1] which is available for Windows™ and Linux platforms. All machine dependent parts are encapsulated so that the program can be compiled and run on both systems.

[1] http://opencvlibrary.sourceforge.net

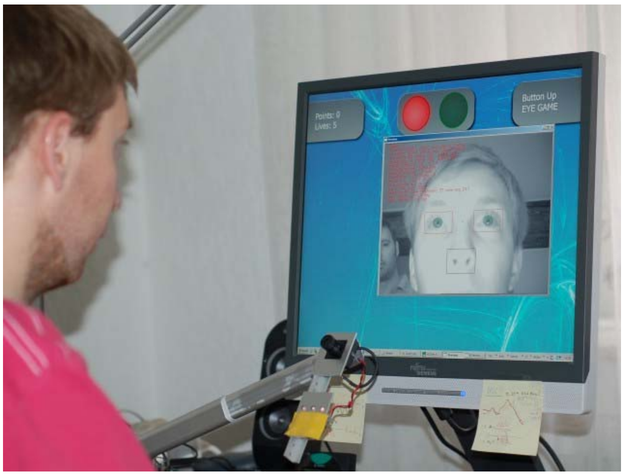


Figure 1. Complete system, showing monitoring window

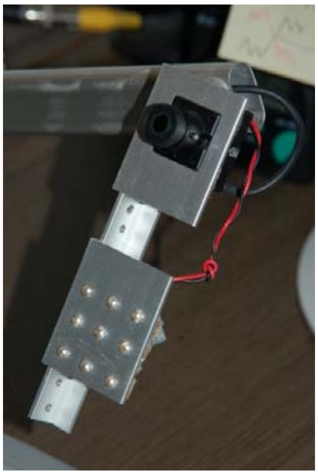


Figure 2. Camera rack with IR-LEDs

Operation

First, if no previous eye position from preceding frames is known, the input image is scanned for possible circles, using an appropriately adapted Hough algorithm. To speed up operation, an image of reduced size is used in this step. Limiting the Hough parameters (notably the radius) to a reasonable range provides additional speedup. Next, the detected candidates are checked against further constraints like a suitable distance of the pupils and a realistic roll angle between them. If no matching pair of pupils is found, the image is discarded. For successfully matched pairs of pupils, subimages around the estimated pupil center are extracted for further processing. An example is shown in Figure 3.

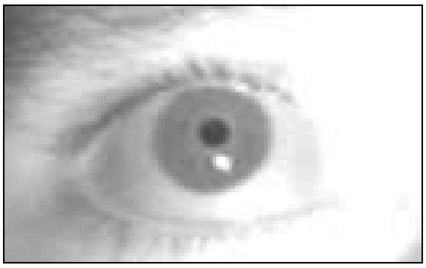


Figure 3. Extracted image of a detected eye

Especially due to interlace effects, but also caused by other influences the pupil center coordinates, as found by the initial Hough algorithm are not sufficiently accurate for further processing. For exact calculation of the gaze direction however, this coordinate must be as accurate as possible. Two approaches for pupil center estimation seem reasonable: finding the center of the pupil or finding the center of the iris. While the iris provides a larger structure and thus higher stability for the estimation, it is often partly covered by the eye lid and thus not entirely visible. Also, its outer bound does not always have a high contrast to the surrounding parts of the image. The pupil however can be easily spotted as the darkest region of the (sub-)image. Unfortunately, it can become very small in bright environments. Also, it might be partly covered by the corneal reflection.

Using the center of the Hough-circle as a base, the surrounding dark pixels are collected to form the pupil region. The center of gravity for all pupil pixels is calculated and considered to be the exact eye position. This value also forms the starting point for the next cycle. If the lids are detected to be closed during this step, again the image is discarded.

The radius of the iris is now estimated by looking for its outer bound. This radius later limits the search area for glints. An additional subimage is extracted from the eye image, centered around the pupil center an slightly larger than the iris. This image is checked for the corneal reflection using a simple pattern matching approach. If no reflection is found, again the image is discarded. Otherwise, the optical eye center is estimated and the gaze direction gets calculated. It is then intersected with the monitor plane to calculate the estimated viewing point.

These calculations are done for both eyes independently. Depending on the viewing direction, the per eye results differ significantly, the average value of both results however gives quite good results. The estimated viewing point can then be used for further processing. It can be handed to the window management system as mouse coordinates, thus providing an easy way to connect the system to existing software.

During development, a collection of monitoring windows can be displayed to give immediate feedback to the developer as shown in Figure 1. These windows are of course hidden during normal operation.

Results

Since the determined viewing points heavily depend on the distance to and position of the actual screen, which is not fixed with respect to the camera position, an appropriate registration has to be performed.

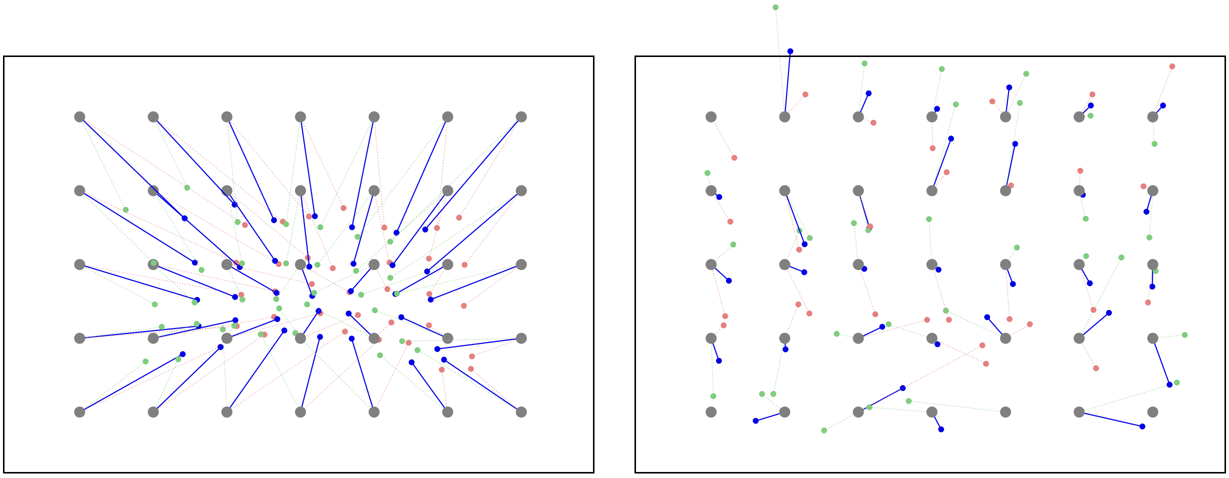


Figure 4. Gaze position estimation unregistered (left) and registered (right). Grey dotes denote the position to look at, blue dots the average measured position, green and red dots the left and right eye positions respectively

Results of the estimation of gaze positions before and after the registration process are shown in Figure 4. The images show the estimated viewing points (blue), the expected points (gray) and the different values for the left and right eye (green and red).

While giving good results in the center of activity, the plots show growing deviation with increasing distance from the mid point. Several influences may cause these effects. First, when looking to the lower middle of the screen, the pupil is partly covered by the corneal reflection. This leads to severe inaccuracies when determining the pupil center. Second, the reduced eye model that is used to do the geometrical calculation might be oversimplified. It uses a sphere model to represent the cornea and its surface. This sphere model might not be suitable for the outer regions of the cornea, where it bends towards the eye body. Furthermore, the tear film often is not homogeneous, e.g. due to dust particles, possibly causing an additional error for the glint detection.

The system has been connected successfully to typing programs like Dasher (Ward et al., 2000), UKO–II (Kuhn & Garbe, 2001) and GazeTalk (Hansen et al., 2003) and showed satisfactory results. Furthermore, sample programs for the usage of eye tracking, including a small game, have been written for demonstration purposes.

While a grid of approximately 5 × 7 fields usually can be detected and distinguished, the affine transformation model currently used for registration however seems not to be sufficient to account for the distortions introduced towards the outer regions of the screen. Also, several parameters like the distance between the right and the left pupil, the radius of the cornea sphere etc. are currently hard coded into the system, leading to additional inaccuracies if different users use the system.

Conclusion

Based on previous work by (Fritzer, 2005) an improved low cost gaze tracking system has been developed. It shows satisfactory results for different users, while improving the stability with respect to its predecessor, notably by using information from both eyes instead of only one. The integration with existing eye typing systems was easily possible, providing the possibility for performance tests to compare with other systems.

While the detection of pupils and glints gives good results in most cases, the dynamic adaption of the geometric parameters for viewing point calculation needs more investigation.

Unfortunately, we still have not found a source for highly sensitive mini cameras using a USB interface. The currently used device requires an analog video input (to be found as input in inexpensive TV tuner cards) and an external power supply. The IR illumination still is a custom built, though meanwhile USB powered solution. As soon as such cameras are available, the whole system can be really seen as a low cost but high quality solution to gaze tracking.

References

Droege, D., Fritzer, F., & Paulus, D. (2005). Gaze tracking using a webcam (poster, abstract). In M. Groner, R. Groner, R. Müri, K. Koga, S. Raess, & P. Sury (Eds.), *Proc. thirteenth european conference on eye movements ecem13* (p. 130). University of Bern, Switzerland.

Droege, D., Fritzer, F., & Paulus, D. (2006). Goldengaze: An inexpensive real-time gaze tracking system. In H. Istance & L. Farinetti (Eds.), *Cogain 2006 – gazing into the future* (p. 29-30). (short paper)

Fritzer, F. (2005). Texteingabe per Augenbewegung. Unpublished master's thesis, Universität Koblenz–Landau, Campus Koblenz, Fachbereich 4 Informatik, Institut für Computervisualistik.

Hansen, J. P., Johansen, A. S., Hansen, D. W., Ito, K., & Mashino, S. (2003). Command without a click: Dwell time typing by mouse and gaze selections. In M. Rauterberg, M. Menozzi, & J. Wesson (Eds.), *Human-computer interaction interact '03: Ifip tc13 international conference on human-computer interaction, 1st-5th september 2003, Zurich, Switzerland.* IOS Press.

Kühn, M., & Garbe, J. (2001). Predictive and highly ambiguous typing for a severely speech and motion impaired user. In C. Stephanidis (Ed.), *Universal access in hci: Towards an information society for all* (p. 933-937). Lawrence Erlbaum.

Ward, D. J., Blackwell, A. F., & MacKay, D. J. C. (2000). Dasher – a data entry interface using continuous gestures and language models. In *Proceedings of user interface software and technology 2000* (p. 129-137).

Magic Eye Control

Luis Figueiredo, Ana Isabel Gomes
Escola Superior de Tecnologia e Gestão da Guarda
Guarda, Portugal
{Luis.figueiredo, Ana.isabel.gomes}@ipg.pt

Keywords
Low cost eye gaze system

Introduction

The growing processing capacity of personal computers, together with the new characteristics of digital cameras equipped with a CMOS sensor, permits the development of new gaze detection systems, based on fast and efficient algorithms that permit mouse control with great precision, speed and stability, at low costs.

This paper, more than effecting a comparative study of the solutions already in the marketplace, which are known to all the interested parties in this theme, is intended solely to convey the characteristics of the newly-developed system, leaving for succeeding works and/or authors a comparative analysis. For this reason, there are no bibliographical references.

Physical Description

Figure 2 shows the system, composed of a portable computer, an auxiliary display, and an attached video camera with a BW CMOS 1280*1024 resolution sensor. The camera support and display are integrated, thereby allowing easy adjustment of the camera aim through three axes.

The display is specially adapted, with 4 infrared 840nm LEDs, powered by the display itself. The total power of the 4 LEDs is less than 0.5W, corresponding to a luminous intensity less than 10mW/cm2, at a distance of 50cm from the display.

All the elements of this system are assembled in a special support that allows a bed-ridden person to use it from a horizontal position.

A desktop computer can substitute the portable computer, where only one monitor is used. This solution however, hinders a second person viewing the monitor when the user is horizontal.

Figure 2: Magic Eye equipment

Software Description

Initially, the application acquires images through the USB camera connection at a resolution of 1280*1024, and 50fps. During this phase the camera employs a characteristic that allows the elimination of one line and one column respectively, for each two lines or two columns. This characteristic reduces the size of the objects in the image by 50%, thus transmitting 640*512 pixels in each image, but retaining all the image's information.

After detecting the user's eye coordinates, the camera is reprogrammed to acquire the maximum resolution images of a region of interest centred in the centre of the user's eye, with a resolution of 320*320, and a frequency of 100fps. This region of interest is updated in line with the user's head movement, so as to consistently maintain the tracking of his/her eye. In situations where eye is lost for a period longer than an involuntarily blink, the application goes back to the initial phase to detect the position of the eye again.

For each unit of 100fps, the application detects not only the centre of the pupil, but a minimum of three of the four reflections of the infrared LEDs in the eye. With these data, a mapping of the points is constructed in the coordinates system of the display. Although the variation of the centre of the eye in relation to the four infrared reflections ranges from 30 to 35 pixels, the developed algorithms allows the placement of the cursor in the display with enough precision to close a Windows© window (X box) with a graphic resolution of 1024*768.

Each user needs to perform a system configuration, which is then saved for subsequent sessions. During this configuration the user is requested to look at a ball that moves to 9 positions in the display. This operation takes about 20 seconds.

The user can move his/her head horizontally by approximately 12cm, and vertically by about 10cm, without such movements interfering with the detection of the eye's orientation, or with the initial calibration. As to backing away from, or approaching the camera, these movements are limited by the focus capacity of the lens, which is not equipped with auto-focus.

It is possible to perform all types of mouse clicks in two ways: by closing the eye for a certain duration, or by maintaining the cursor in an area for a certain duration. The right-click, the double click, or dragging can be performed via a pre-selection of a popup menu.

Using a portable computer with a Pentium M processor at 1700MHz as a reference, this application uses about 25% of the CPU's capacity, including the acquisition and processing the 100fps.

The cost of this application is in the range of 1750€, including the software, the adaptation of the infrared LEDs in the display, the high definition camera, lens and its respective support, not including the support for bed-ridden users. A significant part of the above-mentioned value is the cost of the high definition camera and its lens. One can infer that the large-scale production of this equipment may significantly reduce the hardware prices, and consequently the price of the final product.

Results

System trials have been performed by persons unhindered by physical problems, and have amply demonstrated the capacities of the system. The lack of access to real-world conditions has been hindering the accomplishment of trials with "authentic" users. Only two trials have been performed. In the initial phase, the system was tested on a 10-year old child from the Azores. The child suffers from a genetic disease that impedes speech, or performing any controlled movement. She was unable to perform the system calibration; the calibration was performed by a third party, which hindered the entirety of the work. In spite of this difficulty, it was possible for her to select one of the nine displayed pictures. The great distance between continental Portugal and the Azores is an impediment to the accomplishment of a second round of trials. However, we would like to mention that since those first trials, the application has undergone great improvements.

The second test was undertaken by a 61-year old user with Amyotrophic Lateral Sclerosis in its final phase. In this case, no significant results were obtained, as the user was unable to open the eye sufficiently for the detection of the pupil.

The use of this system in conditions with a high infrared level—namely with a lot of natural light, or with halogen lamps illumination—impedes operation. The same is true if the user wears eye-glasses, as the lenses cause high reflection of the infrared LEDs.

Necessary Improvements

As this work is a recent development, it is essential that it be tested under real conditions, so that the most important aspects needing improvement can be identified. In spite of the lack of these tests, it has been possible to identify some aspects requiring improvement:

- Use a camera with greater resolution to increase the range of head movements without losing the resolution of the eye image.
- Increase the frequency of images sampling.
- Find solutions where the detection of only one infrared LED reflection is necessary.

Conclusions

The developed system has a group of characteristics that they can represent a more value:

- Reduced costs with the possibility of even greater cost-cutting.

- Precision and stability of the mouse movement that allows closure of a Windows© window with a graphic resolution of 1024*768.
- Possibility to perform a range of clicks using alternatives that suit the user
- Sampling rate of 100fps that allows a reasonable synchronization of the gaze direction with the position of the mouse.
- Possibility of head user movements.
- Evolution technology.

3D head orientation estimation and expression influence elimination using characteristic points of face

Donatas Dervinis, Gintautas Daunys
Siauliai University
Vilniaus str. 141-413, Siauliai, Lithuania
{ddonatas, g.daunys}@cr.su.lt

Keywords
Head pose estimation, face expression, characteristic points of face

Introduction

The most robust to user's head movement are remote gaze tracking systems that are using an algorithm of pupil centre and several corneal reflections for estimation of user's gaze direction (Villanueva et al 2007). The additional infrared light sources are used to obtain corneal reflections. The cameras, used in such systems, must be sensitive to infrared light. The light sources must be close to a user that to form a detectable glint at eye image. All these features are disadvantages of the system.

Gaze tracking in visible light is an alternative approach for low cost gaze trackers (Daunys et al 2006). User's head orientation in space must be estimated in such case. It is accepted to divide head tracking methods into two types: appearance-based (Rae 1998) and model based (Stiefelhagen 1996).

Previously we proposed a model based method (Dervinis 2006, Daunys et al 2006) for 3D head orientation estimation from a single monocular camera. Coordinates tracking of several characteristic facial points (facial features) is used in the method. The coordinates of facial features obtain the shifts not only after head translation or rotation but also after a change of face expression. The way to minimise influence of facial expression is proposed in the current paper. The method was examined using computer simulation and analysis of the images from CMU PIE database (Sim et al 2003).

Computer simulation of head pose estimation errors caused by a face expression

Initially, we need to select characteristic facial points for an algorithm implementation. We analysed 18 points–candidates, which can be detected automatically. All of them coordinates shift with a face expression (Dervinis 2005). The goal is to find a minimal number of the points, which are the most stable versus different expression and ensure small head angles estimation errors.

A geometrical model of 18 points arrangement on head was build. Further, we refer to it as a 3D head model. A head rotation center can be chosen arbitrary. We chose a point on the midline between user's eyes because it is seen on the acquired images. The facial points, which coordinates have minimal shifts versus different expression and are easy detectable, were selected. Firstly, the points having the longest distance from the center of rotation were included into set. The points were inluded by the next order: a nose tip, mouth corners, outer corners of eyes, inner corners of eyes. Then we simulated angle estimation errors versus a number of characteristics face points (Fig.1).

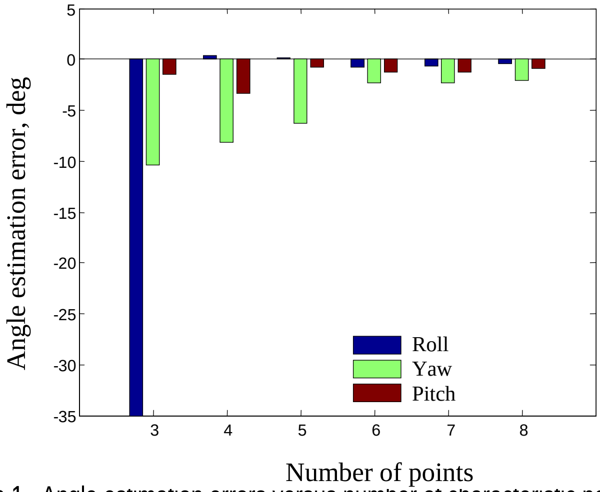


Figure 1. Angle estimation errors versus number of characteristic points of face

We defined from the plot in Fig. 1 that the optimal number of points is 6 – 8. In such case angle estimation error is less than 3 degree. The face characteristic points, which coordinates least change versus expression, are shown in Fig.2 .

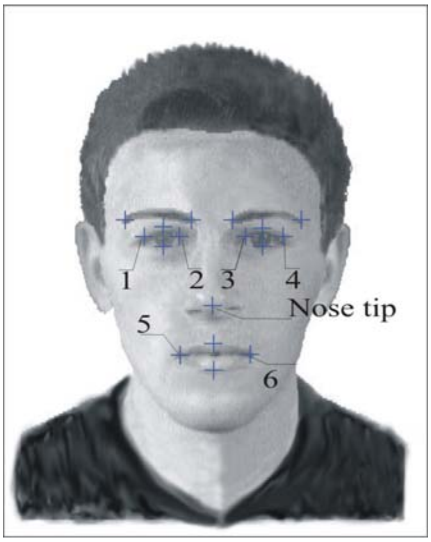


Figure 2. Location of characteristic points, which least changes versus different expression

Previously (Dervinis 2005) we defined the coordinates shifts of 6 characteristic points in five different face expressions: angry (1), happy (2), neutral (3), sadness (4) and surprise (5). Now, simulation of errors based on 3D head model was done. Rotations of head by 10 degree around all axis were simulated. This yielded new coordinates of selected face points. Random shift values according to expression were added to all coordinates before rotation. Afterward, head rotation angles were calculated from obtained coordinates by our suggested method. The differences between estimated by method values and initial rotation value (10 deg.) are angle estimation errors, caused by face expression. The bar plot of mean angle estimation error versus different expression is show in Fig. 3. Similar simulation results are shown in Fig. 4. Only now, after expression caused random shifts were added, the mean shifts values for expression were subtracted. The situation was simulated, when facial points coordinates shifts were compensated by recognized expression mean shifts. We can see that errors in Fig. 4 are some times smaller than in Fig. 3. Error ranges are presented in Table 1.

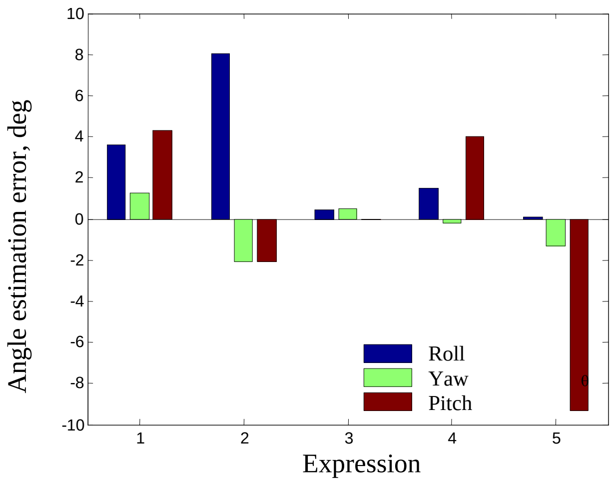


Figure 3. Mean angle estimation error without expression compensation after head rotation about three axis by 10 degrees

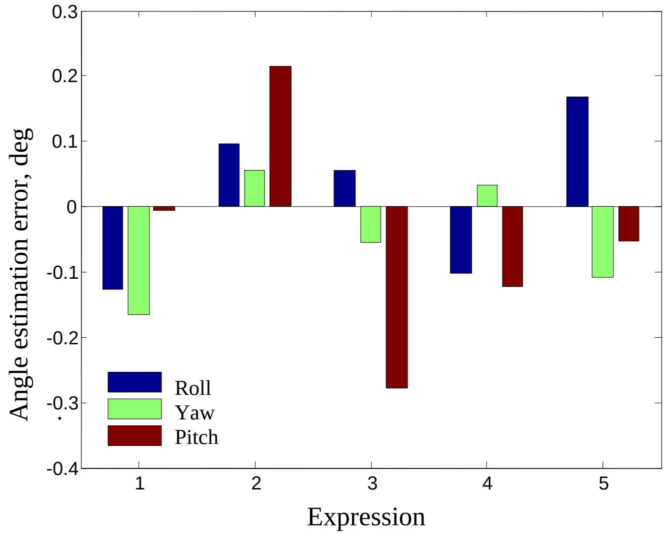


Figure 4. Mean angle estimation error with expression compensation after head rotation about three axis by 10 degrees

Table 1. Intervals of head orientation angles estimation errors with expression compensation

Expression	Roll, deg	Yaw, deg	Pitch, deg
1. Angry	[–1.77 ; 1.26]	[–1.94 ; 1.28]	[–2.60 ; 2.58]
2. Happy	[–1.33 ; 1.72]	[–1.84 ; 2.06]	[–1.69 ; 2.55]
3. Neutral	[–1.55 ; 1.78]	[–1.22 ; 1.00]	[–2.73 ; 1.61]
4. Unhappy	[–2.02 ; 1.61]	[–1.36 ; 1.49]	[–2.36 ; 1.87]
5. Surprise	[–1.23 ; 1.91]	[–1.45 ; 1.02]	[–2.01 ; 1.79]

Head pose estimation errors obtained from database images

It is impossible to rotate a head accurately by a desired angle. Consequently we used CMU PIE database (Sim et al 2003) of head images acquired from several cameras simultaneously. Because the cameras looked at face with different angles, heads in images seem as rotated by different angles. In addition, the faces in database were acquired with different expressions. Our expression elimination method gives angle estimation errors, presented in Fig. 5.

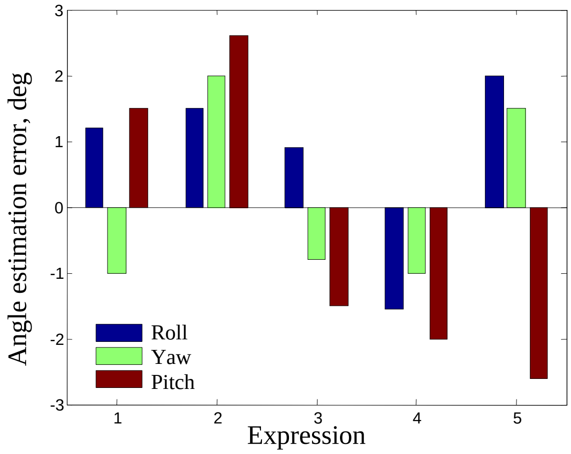


Figure 5. Mean head angle estimation error with expression compensation for CMU PIE images

Conclusion

The optimal number of facial features for tracking is 6-8. The points are: a nose tip, mouth left and right corners, both eyes outer and inner corners. A computer simulation gives that angle estimation error without expression compensation after head rotation around three axis by 10 degrees could reach maximal value13 degree (in surprise expression) and mean value of error is in range 6-8 degree. A proposed compensation method significantly reduces angle estimation error for tested expressions.

References

Daunys, G. et al. (2006) D5.2 Report on New Approaches to Eye Tracking. *IST-2003-511598 NoE report*, available through COGAIN pages at http://www.cogain.org.

Dervinis D. (2005) Automatic Face Expression Detection. *Electronics and Electrical Engineering*, No.8(64), Kaunas: Technologija, pp. 79–82 (in Lithuanian).

Dervinis, D. (2006) Head orientation estimation using characteristic points of face. *Electronics and Electrical Engineering*, No. 8(72), Kaunas: Technologija, pp. 61–64.

Rae, R., Ritter, H. J. (1998) Recognition of human head orientation based on artificial neural networks. *IEEE Transactions on Neural Networks*, No. 9, IEEE, pp. 257 – 265.

Sim, T., Baker, S., Bsat, M. (2003) The CMU Pose, Illumination, and Expression Database. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, IEEE, pp.1615 – 1618.

Stiefelhagen, R., Yang, J., Waibel, A. (1996) A model – based gaze tracking system. *IEEE International Joint Symposia on Intelligence and Systems*, IEEE, pp. 304 –310.

Villanueva, A. et al. (2007) A Geometric Approach to Remote Eye Tracking. *Universal Access in the Information Society* (in press), Springer.

Environmental Control by Remote Eye Tracking

Fangmin Shi, Alastair Gale
Applied Vision Research Centre (AVRC)
Loughborough University, Loughborough,
Leicestershire, UK
{F.Shi, a.g.gale}@lboro.ac.uk

Keywords
Eye-tracking, environmental control, object recognition, disabled people

Introduction

Eye movement interfacing can be found in some specially designed environmental control systems (ECSs) for people with severe disability. Typically this requires the user to sit in front of a computer monitor and their eye gaze direction is then detected which controls the cursor position on the screen. The ECS screen usually consists of a number of icons representing different controllable devices and an eye fixation landing within a pre-defined icon area then activates a selection for control. Such systems are widely used in homes, offices, schools, hospitals, and long-term care facilities.

Wellings and Unsworth (1997) demonstrated that a user-friendly interface design is the weak link in ECS technology, in particular for severely disabled people. Disabled individuals need straightforward control of their immediate surroundings and so making a detailed menu selection by techniques, such as eye-screen interaction, can be a difficult and tedious process for some individuals. This situation can be exasperated by real-world issues such as eye tracking systems which do not tolerate user's head movement.

This paper presents a different approach to environmental control using eye gaze selection, in which the control options applicable to a given device are automatically pre-selected by means of the user directly looking at the device in their environment. This intuitive method therefore minimises the amount of navigation that the user must perform. To date, two main methods have been employed to achieve this direct eye-device control. The initial development using a head-mounted eye tracker was previously reported (Shi et al., 2006). This current paper describes subsequent development of the system (Shi et al., 2007) using a remote eye tracker which is simply situated before the user with no need for any attachment to them.

Remote Eye Tracker

The Smart Eye tracker is used which does not need the user to wear any attachment. Instead, three cameras are placed in front of the user (cameras ①-③ as shown in Figure 3) which can track eye movements over a field of view of up to 170°/60° (horizontally/vertically) by accommodating a range of both head and eye movements.

Figure 3. Smart Eye system setting up

A user's eyes and head are tracked by means of reference to pre-defined templates, which are marked facial features present in a number of snapshots of the user, taken from the three cameras beforehand. These are used to build a model of the head and to create a personal profile for each user. The advantages of the Smart Eye system are accompanied by the need for some additional preparatory steps in setting it up, which include:

1) Eye camera calibration

   A chessboard needs to be placed into the common fields of view of all the three eye cameras to enable the system to calculate the relative positions of the cameras.

2) Definition of a World Coordinate System (WCS)

   To relate the measurements to the real world, again the chessboard needs to be seen by all three cameras and its centre can be selected as the origin of the coordinate system.

3) Creation of a personal profile

   A set of snapshots at different poses with or without head movement are taken. Facial features are manually marked and a virtual 3D head model is generated.

4) Gaze calibration

   By looking at some pre-known devices, the system calculates the difference between the visual and the optical axis of the eye.

Once the preparatory steps have been carried out, the unit can be set working in real time mode and outputs a number of parameters such as head position and eye line of gaze. These measurements are in three dimensions with reference to the pre-defined WCS. Figure 4 is a snapshot of the tracking process with imposed feature points as well as the eye gaze directions. The output can also be projected onto a two dimensional plane in the WCS, which is an approach adopted in this paper.

In this implementation, the Smart Eye unit is run through a combination of its own interface software, which remains open in the background throughout, and an SDK which interfaces with our bespoke software. During development, the latest Smart Eye software (version 4) has achieved a gaze accuracy of 1.49°/1.97° along horizontal/vertical axes.

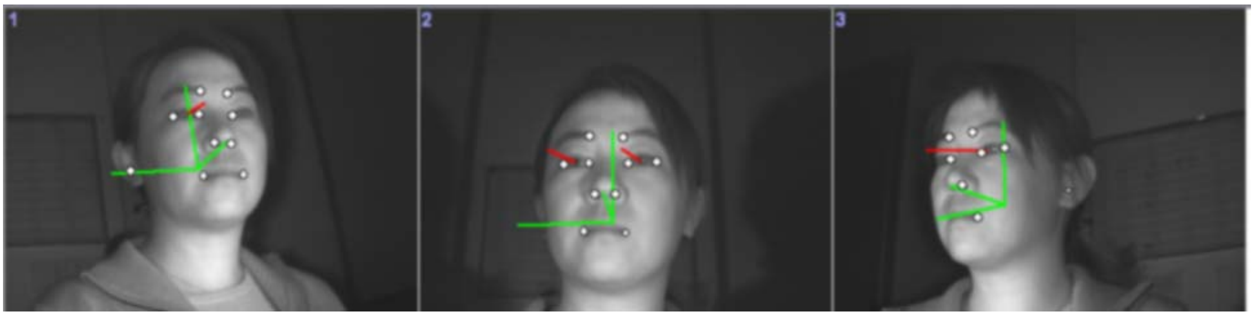


Figure 4. Smart Eye system setting up

System development

A high resolution USB camera (number 4 in Figure 3) provides more information to the system's object recognition algorithms than would a standard video camera. It is positioned beside the user's head such that it achieves a similar field of view, and is mounted on a common support with the eye cameras such that they can all move about together whilst maintaining their relative association, say with a wheelchair.

To relate the eye gaze coordinate system with the scene camera information, another calibration similar to that of a head-mounted system needs to be performed (Shi et al., 2007). This is done by setting some target points before the scene camera, obtaining their corresponding gaze coordinate measures at the same time and then determining the projection matrix of the two coordinate systems.

The whole system operation is driven by a central Matlab interface which integrates all the functions from calibrations to real time tracking. The program alters at a couple of stages, for instance, when an eye fixation is obtained, when the process of comparing the potential device of interest with references in the pre-saved database finishes, and when the control options for an identified controllable device are enabled. Due to the use of Matlab software for development purposes, the system does not yet run to its full potential speed and currently requires the user to gaze at a device for 1s or more, and it then takes about 3~20s to identify the gazed device.

Laboratory pilot trail and discussion

Currently the whole prototype system is mounted around a wheelchair. Four normal household appliances, i.e. a fan, a lamp, an e-curtain and a tv, are used as controllable devices. Six able-bodied persons, of different heights (155cm~177cm), ages (25+~50+ years old), sex, and with/without eyeglasses, have had profiles created and then been trialled. The system processes have proved successful and have demonstrated the overall research concept. The main observations from the initial user trials are as follows:

- Preparations before real time tracking do not all need to be done in the presence of a user. The eye camera calibration and the WCS definition can be done prior to the user's arrival. This can reduce the demands on the user and save time during trials.
- It takes less than a minute to take some snapshots of the participant sitting with different head poses to form the user's profile. However, facial features in all the snapshots must be manually marked to form the head model, and this can take around 10-15 minutes. Smart Eye has plans to make this process faster and fully automatic.
- To obtain a fixation on a device requires a user gaze of 1s, with further time taken to perform object recognition and provide appropriate control options; the former time can be altered as required and the latter times can be reduced by programming in visual C++.

More user trials are underway. To fully test the efficacy of the system with real target users, to gain understanding of how they perceive their experience of the system, and to arrive at optimum values for various system settings, we shall try it with many more people including severely disabled individuals. This is also in agreement with the steps proposed by Craig (Craig et al., 2004) for providing an environmental control system for people with severe disabilities.

Conclusion

This paper has presented an eye gaze based environmental control system for severely disabled individuals. The employment of the Smart Eye tracker completely releases the user from wearing any attachment. It features three dimensional head/gaze outputs, which allows a great deal of user head movement. The paper has also discussed a number of usability issues using the remote system and indicated that the main drawback lies in the need for more time to set up a participant initially. The pilot trial with some able-bodied persons under laboratory conditions has proved the functionalities of the system. More tests with both able-bodied people and the severely disabled will be the main focus of our next stage of work.

Acknowledgements

This research is funded by the UK ESRC PACCIT Programme and is also part of the COGAIN network.

References

Craig A., Tran Y., Mclsaac P. and Boord P. (2004) The efficacy and benefits of environmental control systems for the severely disabled, *Medical Science Monitor*, vol. 11, pp. 32-39.

Shi F., Gale A.G. & Purdy K. (2006) Direct gaze-based environmental controls, *Communication by Gaze Interaction, COGAIN 2006*, Turin, Italy, 2-7 September 2006, pp 34-39.

Shi F., Gale A.G. & Purdy K. (2007) Exploring eye responsive control – from a head mounted to a remote system, *Annual Conference of Ergonomics Society*, Nottingham, UK, 17-19 April, pp.

Wellings, D.J. and Unsworth, J. (1997) Environmental control systems for people with a disability: an update. *British Medical Journal*, 315, pp. 280-287.

Hands Free Interaction with Virtual Information in a Real Environment

Susanna Nilsson
Department of Computer and Information Science
University of Linköping
SE-583 83 Linköping, Sweden
susni@ida.liu.se

Torbjörn Gustafsson and Per Carleberg
Division of Sensor Technology
Swedish Defence Research Agency
PO Box 1165, SE-581 11 Linköping, Sweden
{torgus, percar}@foi.se

Keywords
Gaze controlled interaction, Augmented Reality, user driven instructions

Introduction

In augmented reality (AR) systems, real and virtual objects are merged and aligned relative a real environment, and presented in the field of view of a user. AR applications that give hierarchical instructions to users often require some feedback or acknowledgement from the user in order to move to the next step in the instructions. This feedback should be possible to give quickly and without interruption from the ongoing task. Many different types of interaction techniques have been used in the domain of AR; there are numerous examples of systems that use manual input, gestures and/or speech interfaces (Nilsson & Johansson 2006, Billlinghurst et al 2001, Gandy et al 2005 and Henrysson et al 2007). However, there are situations where speech and gesture may not be appropriate. For instance, during surgical procedures in an operating room the surgeon may have difficulties manually interacting with technical devices because of the need to keep her/his hands sterile. Voice interaction with a system may also not be appropriate due to surrounding noise or filtering problems. There is one modality that can overcome the issues of noisy environments, keeping hands sterile and the need to work with both hands while at the same time trying to interact with a computer or an AR system, and that is the visual modality. The aim of this paper is to present an AR system with an integrated gaze tracker, allowing quick feedback from the user to the system, as well as analysis of the users gaze behaviour.

Augmented Reality

Azuma (1997) mentions three criteria that have to be fulfilled for a system to be classified as an AR system: they all combine the real and the virtual, they are supposedly interactive in real time (meaning that the user can interact with the system and get response from it without delay), and they are registered and aligned in three dimensions. AR applications can be found in diverse domains, such as medicine, military applications, entertainment, technical support and industry applications, distance operation and geographic applications.

Technically, there are two different solutions for merging reality and virtuality in real time today – video see-through (VST) and optic see-through (OST), which is at first glance the most preferable solution but it has some technical and practical difficulties (Azuma 1997 and 2001, Kiyokawa 2007). A way to overcome the problems with OST is by using a camera in front of the users' eyes, and then projecting the camera image on a small display in front of the users' eyes (VST) (Azuma 1997, Gustafsson 2004, Kiyokawa 2007). The virtual images are added to the real image before it is projected which solves the OST problem of surrounding light as well as gives control over where the virtual objects are placed. This method however, has other problems such as the lag, determined by the camera image update frequency, which can have effect on the user experience of the system, such as simulator sickness.

A gaze controlled AR system

In interactive AR applications that require responses from the user, there must be an efficient and non-interruptive way to deliver the responses to the AR system.The main goal of implementing gaze control into an AR system is to make the interaction between the system and the user easier and more efficient. Implementing gaze detection to the AR system could also be a way to predict the user's intentions and to anticipate what actions will be requested of the system (Hyrskykari et al 2005,Vertegaal 2002).

For a gaze based interaction AR system to be useful, the gaze detection process should be implemented in a way that does not interfere with the user's normal behavior (Oviatt & Cohen, 2000). A gaze tracker for AR applications should be able to integrate with micro displays and must function in varying conditions of illumination. The following sections present a fully functional helmet mounted AR system with an integrated gaze tracker, which can be used both for monitoring of the users gaze behaviour as well as for interaction.

HMD and integrated gaze tracker

We have developed a head mounted video-see-through AR system with an integrated gaze tracker (see fig 1 and 2). The integrated head mounted display, black/white gaze tracker camera (640x480 pixel resolution) and VST camera is an in-house construction and the different components used are shown in the schematic in figure 1. The systems camera view is bi-ocular meaning that the camera view is presented to both eyes independently, while the virtual objects are bin-ocular. The displays have a resolution of 800 x 600 pixels and a field of view of 37 x 28 degrees. The gaze tracker camera and the micro display are integrated and have co-aligned optic axis to facilitate future studies of vergence movements controlled systems (Gustafsson et al 2005).

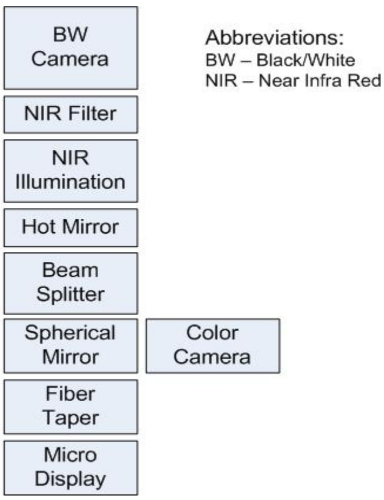


Figure 1: A schematic overview of the integrated HMD, gaze tracker and VST camera.

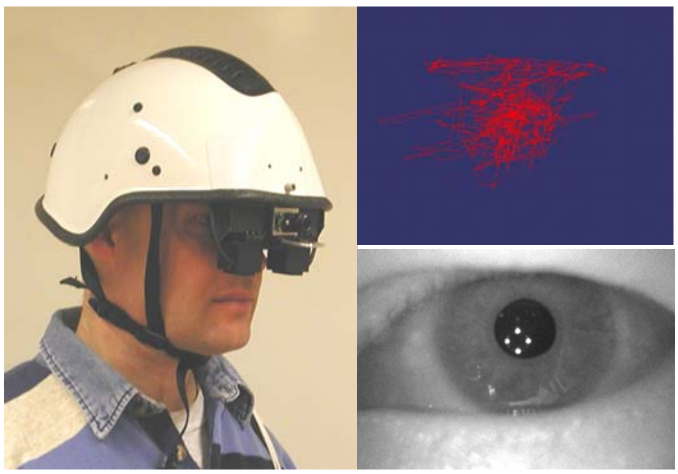


Figure 2: To the left a head mounted gaze controlled AR system, top right a gaze pattern of a user working with the gaze interaction dialog seen in figure 3. The bottom right image shows the gaze trackers view of the user's eye.

The gaze tracker camera detects the pupil and it's reflections by filtering and thresholding the image information. The position of the pupil and the positions of the reflections on the cornea caused by near infra red illumination are calculated. The technical solution utilizes the dark pupil principle, which implies that the NIR (Near Infra Red) illumination is placed by the side of the camera optical axis. The IR light source is integrated fully into the system and is not a separate device. Four illumination sources are used, however, only two reflections are needed for the calculation of the gaze (see figure 1). The system can choose the between these four different reflections which increases the robustness in the system.

MR software

Many MR systems in research today use marker-based tracking techniques where the MR system detects markers placed in the surrounding. These markers inform the system of where to place the virtual objects which can be in the form of text, 3D models, sound, images, videos, animations and volumetric models. The MR system described here, uses a hybrid-tracking technology, basically a marker detection system (based on ARToolKit (HITLAB webpage) ARToolKit Plus (Schmalstieg 2005) and ARTag (Fiala 2005) but with the addition of a 3DOF inertial tracker (InterSense (isense.com) and Xsens (xsens.com).

Special software has been developed with the aim to permit an application developer to define a scenario file in XML syntax. For gaze controlled interaction, the application designer can define the layout of the gaze control dialog areas, as well as gaze action specifications. With this tool, the application developer can experiment, compare and verify the functionality of different gaze controlled interaction schemes (see figures 3 and 4).

Figure 3: Gaze interaction dialog. The user can respond to the question "Is the patient pregnant?" by looking at the answers "no", "acknowledged" and "yes".

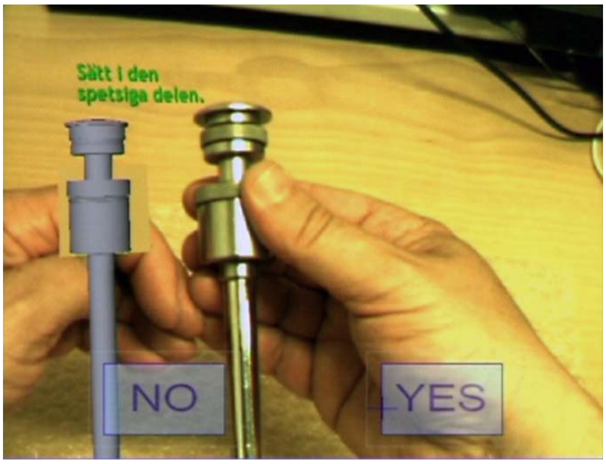


Figure 4: Gaze interaction dialog. An alternative layout of the gaze interaction dialogue. This example is from another AR application instructional sequence.

In the developed system, eye gaze interaction can be restricted both temporally and spatially - certain parts of the display will have the function, and only when there is a need for gaze interaction. The interaction areas are defined in the application scenario XML file, as well as eye gaze dwell times and command actions. The gaze dialog area positions can either be fixed or dynamic, relative to detected marker position which allows flexible design of the application. The areas in which the gaze interaction is active are represented by transparent images as can be seen in figures 3 and 4. The interaction area can also be non-transparent, if useful for the application. Transparency, color, image and placement of the interaction areas are set in XML syntax in the scenario file.

Preliminary test of the system

The gaze controlled MR system has been tested in a laboratory setting and for the trials of the gaze controlled MR system an instructional application was used where the user completes a set of instructions, and receives the next after acknowledging to the system that the previous instruction has been completed. The task has previously been used in a user study investigating usability of AR instructions in the medical domain (Nilsson & Johansson 2006). The main goal of the instructions is to activate a diathermy apparatus and prepare it for operation. The instructions were given as statements and questions that had to be confirmed or denied via the input device, in this case the gaze interaction dialog where the user can choose to look at 'yes', 'no', and 'ack' (short for 'acknowledged'), see figure 3. The dwell time used in the test series was set to 1 second.

The experience so far is that the gaze controlled interaction is equally fast and as distinct as pressing an ordinary keyboard button. This is in accordance with earlier research and the results of Ware & Mikaelian (1986) who illustrated that gaze interaction may even be faster since the time it takes to shift position of the cursor manually slows down the speed of interaction in traditional mouse pointing tasks.

Discussion and further research

Gaze interaction allows the user to work freely with her/his hands while stepping through an instruction hierarchy. This freedom of movement is of value in many situations, such as in the application described above as well as other applications involving maintenance and repair tasks. The conclusion therefore is that the system can be used as an alternative to traditional manual interaction. This is especially of interest for applications where hands or speech as input devices are not appropriate or possible. The experience from the limited test runs are important for the further development of the system and have clearly indicated that the system is functional. Future tests, including a larger user group, will investigate the robustness of the system as well as give more insight to the speed and accuracy in other applications than the hierarchical instructions used here.

The gaze recognition in the system is not restricted to use for interaction directly, but can also be used for indirect communication with the AR system. Gaze recognition can add a 'user awareness' dimension to the system, which can monitor the user's visual interest and act upon this. The gaze awareness can also allow the system to acknowledge (via the users gaze direction) when and if the user wants to interact with the system. If the user has two or more markers in the field of view, the gaze direction can be used as a an indicator of which marker's virtual information should be displayed, and is thus a relatively easy way to de-clutter the users field of view.

Combining the concepts from AR and gaze recognition and input gives possibilities to create quick and easy interaction in a MR system that allows for natural human communication, such as communicating intention by the use of eye gaze. Cognitive interest may not always be the same as the visual interest, but in many cases visual interest can be an indicator of what the user is focusing on both visually and cognitively, and therefore allowing the system to respond to this, for example by presenting requested virtual information.

The proposed MR system with added gaze awareness and gaze controlled interaction will be further tested in a user application. Another aspect of the gaze aware AR system is its ability to log and analyze the gaze patterns of AR users, possibly allowing further usability studies and evaluations of the AR system. Gaze control in the AR system may also be a useful tool during the development process of the interaction methods with the system and the design of the displays.

References

Azuma, R. A survey of Augmented Reality. (1997) Presence: Teleoperators and Virtual Environments. 6 (4), August. P355-385

Azuma, R, Bailot, Y., Behringer, R. Feiner, S., Simon, J. & MacIntyre, B. (2001) Recent Advances in Augmented Reality, *IEEE Computer Graphics and Applications.* November/December, p 34-47

Billinghurst, M., Kato, H. and Poupyrev, I. (2001) The MagicBook: Moving Seamlessly between Reality and Virtuality. *IEEE Computer Graphics and applications* Volume 21, Issue 3

Fiala, M. (2005) ARTAG Rev2 Fiducial Marker System: Vision based Tracking for AR. *Workshop of Industrial Augmented Reality*, Wienna Austria Oct 4

Gandy. M. MacIntyre, P. Presti, P. Dpw, J. Bolter, B. Yarbrough, and O'Rear, N. (2005) AR Karaoke: Acting in your favourite scenes. *Proceedings of the forth IEEE and ACM International conference on Mixed and Augmented Reality (ISMAR)*, October 5-8, Wienna Austria

Gustafsson, T., Carleberg, P., Nilsson, S., Svensson, P., Sivertun, Å. and LeDuc, M. (2004) *Mixed Reality for technical support.* Technical report ISSN 1650-1942, Swedish Defence Research Agency

Gustafsson, T., Carleberg, P., Svensson, P., Nilsson, S., and Sivertun, Å. (2005) *Mixed Reality Systems for Technical Maintenance and Gaze-controlled interaction.* Technical report ISSN 1650-1942, Swedish Defence Research Agency

Henrysson, A. Ollila, M. and Billinghurst, M. (2007) Mobile Phone Based Augmented Reality. in Haller, M. Billinghurst, M. and Thomas, B. *Emerging technologies of Augmetented Reality. Interfaces and design.* London: Idea Group Publishing

HITLAB, http://www.hitl.washington.edu/artoolkit/ (as of 2007-06-10)

Hyrskykari, A., Majaranta, P. and Räihä, KJ (2005) From Gaze Control to Attentive Interfaces. *Proceedings of HCI 2005*, Las Vegas, NV, July

InterSense: http://www.isense.com/ (as of 2006-04-20)

Kiyokawa, K. (2007) An Introduction to Head Mounted Displays for Augmented Reality. In Haller, M., Billinghurst, M., Thomas, B. (eds): *Emerging Technologies of Augmented Reality: Interfaces and Design*. Idea Group Publishing, London, UK

Nilsson, S. and Johansson, B. (2006) A Cognitive Systems Engineering Perspective on the Design of Mixed Reality Systems. Proceedings of the 13th European Conference on Cognitive Ergonomics September 20-22, Zürich 2006

Oviatt, S. and Cohen, P. (2000) Multimodal Interfaces that Process What Comes Naturally. *Communications of the ACM* vol 43, no 3, March

Schmalstieg, D. (2005) Rapid Prototyping of Augmented Reality Applications with the STUDIERSTUBE Framework, *Workshop of Industrial Augmented Reality (IAR)* Wienna, Austria Oct 4

XSens: http://www.xsens.com/ (as of 2007-06-10)

Vertegaal, R. (2002) Designing Attentive Interfaces. *Proceedings of ETRA'02* ACM

Ware, C. & Mikaelian, H.H. (1986) An evaluation of an eye tracker as a device for computer input. *Proceedings of the SIGCHI/GI conference on Human Factors in computing systems and graphic interfaces*, Toronto, Canada April 05-09. ACM

Session 3: Interaction and Input

Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures

Jacob O. Wobbrock,[1] James Rubinstein,[2] Michael Sawyer,[3] and Andrew T. Duchowski[4]

[1]The Information School
University of Washington
Seattle, WA 98195 USA
wobbrock@u.washington.edu

[2]Department of Psychology
[3]Department of Industrial Engineering
[4]School of Computing
Clemson University
Clemson, SC USA 29634
{jrubins, msawyer, andrewd}@clemson.edu

Keywords: Text input, eye-typing, unistrokes, EdgeWrite, EyeWrite.

Introduction

People with severe motor disabilities often cannot use a conventional keyboard and mouse. One option for these users is to enter text with their eyes using an eye-tracker and on-screen keyboard (Istance et al. 1996). Such keyboards usually require users to stare at keys long enough to trigger them in a process called 'eye-typing' (Majaranta and Räihä 2002). However, eye-typing with on-screen keyboards has many drawbacks, including the reduction of available screen real-estate, the accidental triggering of keys, the need for high eye-tracker accuracy due to small key sizes, and tedium. In contrast, we describe a new system for 'eye-writing' that uses gestures similar to hand-printed letters. Our system, called *EyeWrite*, uses the EdgeWrite unistroke alphabet previously developed for enabling text entry on PDAs, joysticks, trackballs, and other devices (Wobbrock et al. 2003, Wobbrock and Myers 2006). EdgeWrite's adaptation to EyeWrite has many potential advantages, such as reducing the need for eye-tracker accuracy, reducing the screen footprint devoted to text input, and reducing tedium. However, the best interaction design was non-obvious. As a result, EyeWrite required extensive iteration and usability testing. In this paper we describe EyeWrite and its development, and offer initial evidence in favour of this new technique.

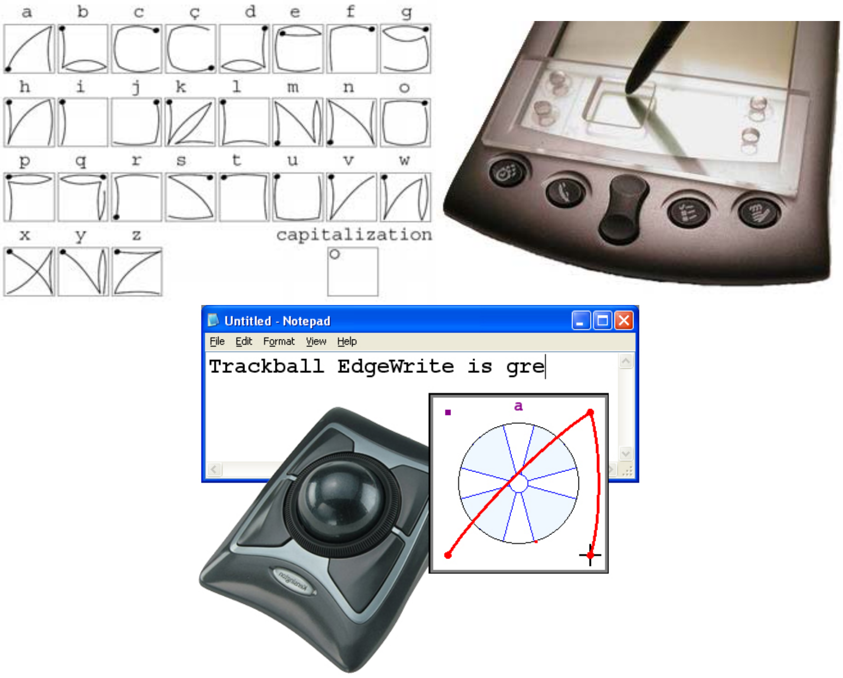


Figure 1. Stylized EdgeWrite letters, Stylus EdgeWrite on a PDA, and Trackball EdgeWrite. Used with permission.

Background and Related Work

EyeWrite is based on the gestural unistroke alphabet used in EdgeWrite (Wobbrock et al. 2003), a text entry method capable of being used on numerous devices (Figure 1). Our decision to use EdgeWrite's alphabet for EyeWrite was not arbitrary; in fact, the alphabet has important properties that make it useful for gaze input. Since the eyes move in saccadic bursts rather than smooth paths, it would be impossible to 'write fluidly' as one does with a pen. Fortunately, EdgeWrite recognizes characters based only on the order in which the four corners of its square input area are hit. This allows EyeWrite to employ four discrete corner targets that help users form their gestures. Another benefit of EdgeWrite's corner-based recognition scheme is that it provides tolerance to tremor in the stroke path, since all that matters is the order in which the corners are hit. This means that eye-tracker jitter is not overly detrimental to EyeWrite's recognition. A third benefit of using EdgeWrite's alphabet is that it has been shown to be very easy to learn, which is important when considering tradeoffs with on-screen keyboards, which are easily comprehended.

Most prior eye-based text entry methods use on-screen keyboards developed for eye-typing (Istance et al. 1996, Lankford 2000, Majaranta and Räihä 2002). To our knowledge, EyeWrite is the first letter-like gestural text entry system for the eyes. Only a few prior systems use eye-based gestures, but these gestures are defined by underlying screen regions, making these systems fancier variants of eye-typing. One system is *Dasher*, which uses expanding letter regions that move toward the user's gaze point (Ward and MacKay 2002). Although Dasher is fast, it can be visually overwhelming for novice users since letter regions 'swarm' toward the user. Other gestural interfaces are the systems developed by Isokoski as part of his exploration of off-screen targets (Isokoski 2000). These designs place letter regions beyond the edges of the screen to solve the Midas Touch problem. A noted issue, however, is that users have difficulty locating the off-screen targets.

The EyeWrite Design

Adapting EdgeWrite for use with the eyes may initially seem straightforward, but the design challenge was considerable. Two key questions were how to translate eye movements into EyeWrite gestures, and how to segment between letters when a letter was finished. This latter issue is the so-called 'segmentation problem.'

Our first design simply mimicked Stylus EdgeWrite (Wobbrock et al. 2003). A literal trace was drawn as the user moved his eyes within EyeWrite's on-screen input area. However, drawing a trace based on the literal eye position created strokes that were jagged and distracting, even with filtering. Since there was no 'stylus lift' signal as in Stylus EdgeWrite, segmentation was first achieved by looking for a cessation of movement. This worked poorly because eye-tracker jitter meant the eye-trace never stopped moving. A second scheme segmented letters after sufficient time had elapsed since the last corner was entered. This time was calculated using the average inter-corner time for the current stroke. However, this resulted in unwanted segmentations when users paused to think. It also meant that the gaze point would remain in the same corner after segmentation, which meant that this corner had to be re-entered before a new letter could be started there.

Our second design utilized a vector-based approach akin to Trackball EdgeWrite (Wobbrock and Myers 2006). In this approach, the absolute position of the eyes was no longer relevant. Instead, the *direction* in which the eyes moved indicated the corner to which the stroke should proceed. When such a vector was indicated, a stylized stroke was drawn from the previous corner to the indicated corner. Although this solved the jitter and distraction of the first design, it momentarily decoupled the stroke corner from the user's gaze point. This resulted in the creation of unwanted vectors and, in turn, unwanted corners.

Another problem was that the user could not look away from EyeWrite to verify their text without indicating a movement vector.

Our third design accommodated lessons from the first two. We returned to a tight coupling between the user's gaze and EyeWrite's input, but instead of drawing a literal eye-trace as in the first design, we drew stylized arcs between corners as in the second design. Instead of vectors, corners were simply hit-tested for the presence of the eyes—when the gaze point entered a new corner, an arc was drawn there. Thus, the gaze point and stroke corner were never decoupled. We also gave users direct control over the segmentation process by segmenting only when the eyes returned to the center of the input area. Users could therefore prevent segmentation and 'pause to think' by simply leaving their gaze in the current corner. Return-to-center segmentation also meant that every new letter would be started from the center. As in the first design, segmentation time was based on the average inter-corner time, but now with a minimum threshold about twice the time of a saccade. This prevented unwanted segmentations when moving among corners. Users could also clear their current stroke by simply glancing away from the EyeWrite square. Finally, to reduce the need to look away between letters to verify the last entry, an incremental recognition result was displayed in the current corner of the EyeWrite square. It was also displayed in the center of the square after segmentation, so users knew exactly what character had been produced. These improvements culminated in the current version of EyeWrite (Figure 2). EyeWrite is implemented in Visual C# using .NET 2.0. We run it on a Tobii 1750 eye-tracking system.

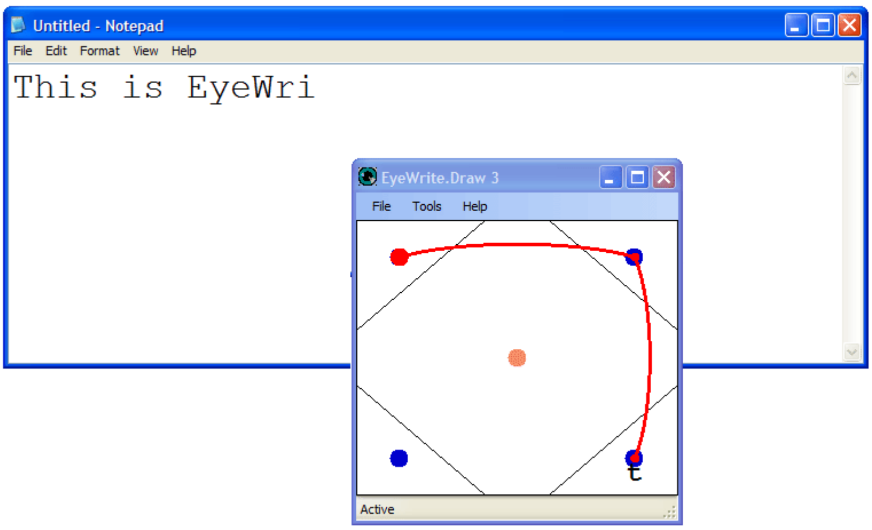


Figure 2. EyeWrite being used with Microsoft Notepad. Up to this point, a 't' has been made, which appears in the bottom-right corner. When the user is ready to segment, he will look at the salmon-coloured dot in the centre.

Conclusion and Future Work

To our knowledge, this paper is the first to describe a letter-like gestural writing system for the eyes. EyeWrite has potential advantages including reduced screen footprint, few large proximate targets, tolerance to eye-tracker jitter, ability to add commands without increased screen consumption, reduced distance between input and output areas, and greater elegance through minimalist design. Entering gestures may also be less tedious and more fun than repeatedly dwelling over keys (Wobbrock and Myers 2006).

Going forward, EyeWrite will be compared to eye-typing in a longitudinal study. Such a study will measure entry, error, and learning rates. Conceptually, the speed comparison amounts to whether fixating on 2-4 large, proximate targets (EyeWrite) is faster than dwelling on one smaller, more distant target (eye-typing). Our initial study results to-date indicate that experienced EyeWrite users can write at about 7.99 WPM with 1.25% uncorrected errors. A full study will allow us to elaborate on these preliminary findings.

References

Isokoski, P. (2000) Text input methods for eye trackers using off-screen targets. In *Proc. ETRA '00*. New York: ACM Press, pp. 15-21.

Istance, H.O., Spinner, C. and Howarth, P.A. (1996) Providing motor impaired users with access to standard Graphical User Interface (GUI) software via eye-based interaction. In *Proc. ECDVRAT '96*. Reading, UK: University of Reading, pp. 109-116.

Lankford, C. (2000) Effective eye-gaze input into Windows. In *Proc. ETRA '00*. New York: ACM Press, pp. 23-27.

Majaranta, P. and Räihä, K.-J. (2002) Twenty years of eye typing: Systems and design issues. In *Proc. ETRA '02.* New York: ACM Press, pp. 15-22.

Ward, D.J. and MacKay, D.J.C. (2002) Fast hands-free writing by gaze direction. *Nature 418*, p. 838.

Wobbrock, J.O., Myers, B.A. and Kembel, J.A. (2003) EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proc. UIST '03*. New York: ACM Press, pp. 61-70.

Wobbrock, J.O. and Myers, B.A. (2006) Trackball text entry for people with motor impairments. In *Proc. CHI '06*. New York: ACM Press, pp. 479-488.

Dwell time free eye typing approaches

Mario H. Urbina, Anke Huckauf
Bauhaus-University Weimar
Bauhausstr. 11
99423 Weimar, Germany
{mario.urbina, anke.huckauf }@medien.uni-weimar.de

Keywords:
Gaze-based interaction, eye typing, user interfaces, input devices

Introduction

Gaze-based text input operates often via selecting characters by dwell time (fixation threshold). Dwell times, however, bear several problems (e.g., Huckauf and Urbina, accepted; Jacob, 1991). With current dwell time based systems, typing speeds of about 6,22 (Marajanta et al., 2004) to 8,5 (Miniotas et al., 2003) words per minute (wpm; MacKenzie, 2003) are achieved. Compared with manual typing speed of about 40 wpm (MacKenzie and Zahng, 1999), this is still very slow – what is certainly also due to the less frequent usage of gaze input. Therefore, promising tools are not only characterized by a high typing speed, but also by a steep learning curve.

Some alternatives to dwell time based eye typing interfaces have been developed; the most prominent among them is certainly *Dasher* (Ward and MacKay, 2002). *Dasher* gets rid of dwell times by presenting the characters moving in columns towards the point of fixation. Each column contains the complete alphabet, and a character is selected by simply gazing towards it. But, since the whole alphabet is always moving, novice users feel often stressed by the interface. In fact, in Dasher, it is difficult to review the written text without deleting parts of it. Besides performance and learnability, another important factor is that users like to work with the interface. The QWERTY and Dvorak keyboards are good examples for the enormous effects of attitude on behaviour: Although users achieve better performance using Dvorak keyboards (e.g., Goettl et al., 2005), these keyboards never got to be popular.

New approaches

We investigated the design space for gaze-based text entry while substituting dwell times by saccades. The duration of saccades varies around 30 ms and depends on the distance. Saccade latencies range between 200 and 400 ms varying with task affordances (Duchowski, 2003). This is much faster than the reported dwell times from 400 (Miniotas et al., 2003) to 1000 ms (Marajanta and Räihä, 2002).

In a first concept, namely *pEYEdit,* we used marking or pie menus, which have already been shown to be powerful menus for mouse or stylus control (Kurtenbach and Buxton, 1993). On this interface, letters are ordered in groups on pie slices. Selecting a group requires to gaze to the outer frame of the slice. Instantaneously a second smaller pie that contains each single letter in one slice is popped up. Selecting a letter, again via gazing in the outer frame of the pie, writes it in the text window. Pie menus can be operated in two modes. Novice users can search the two-layered pie menu serially. But, for experienced users, pie menus offer the possibility of associating a certain motor behaviour with a desired effect (here: a gaze path and typing a letter). Given that users are able to memorize the gaze path, this approach might be comparable to stenographic writing.

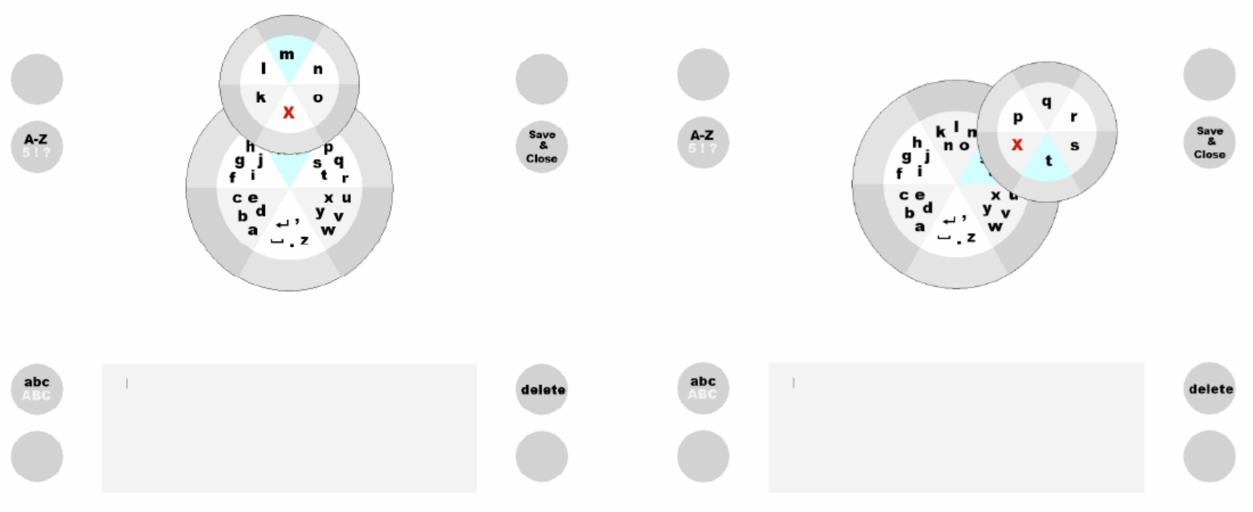


Figure 1. pEYEdit: A gaze path (marked by blue slices). (a) for letter "m" and (b) for the letter "t".

Iwrite is based on screen buttons. We implemented an outer frame as screen button. That is, characters are selected by gazing towards the outer frame of the application. This lets the text window in the middle of the screen for comfortable and safe text review. The order of the characters, parallel to the display borders, should reduce errors like unintentional selection of items placed on the way to the screen button (e.g., Ware and Mikaelian, 1987). The strength of this interface lies on its simplicity of use. Additionally, it takes full advantage of short saccade selection.

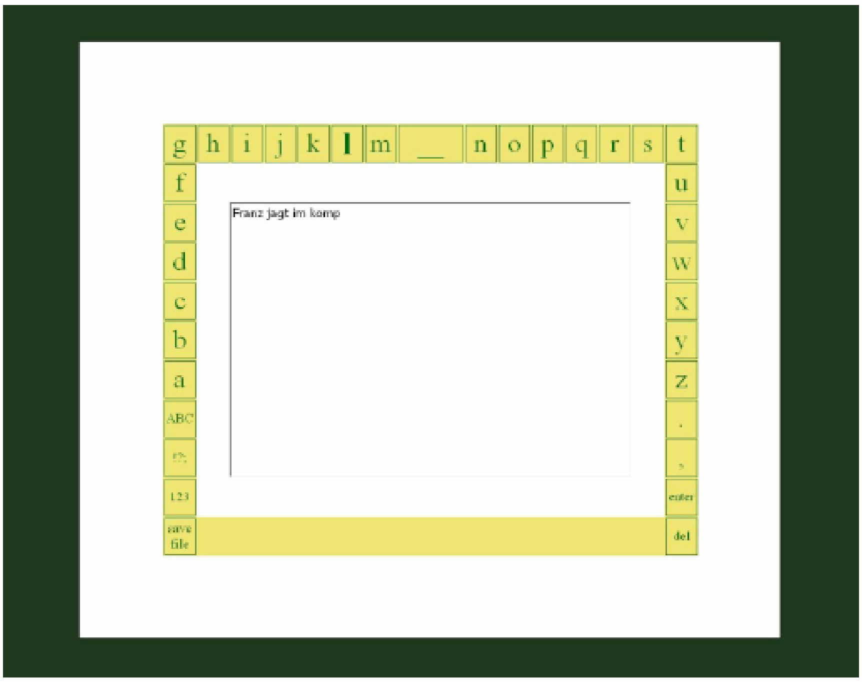


Figure 2. Screen shot of Iwrite; the black surrounding frame is used as screen button to select a character.

In *StarWrtie* letters are typed by dragging them into the text field. This provides instantaneous visual feedback of what is written. When a character is fixated, it and its both neighbours are highlighted and enlarged in order to facilitate the character selection. In order to use both, x- and y-coordinates for target selection, letters were arranged on a half-circle in the upper part of the monitor. The text window appeared in the lower field. Eye typing performance of these newly developed interfaces, namely, *pEYEdit*, *Iwrite*, and *StarWrite*, was compared to typing using a virtual *QWERTY* keyboard and to *Dasher*. In addition to performance measures, also attitudes towards the interfaces were collected within this user study.

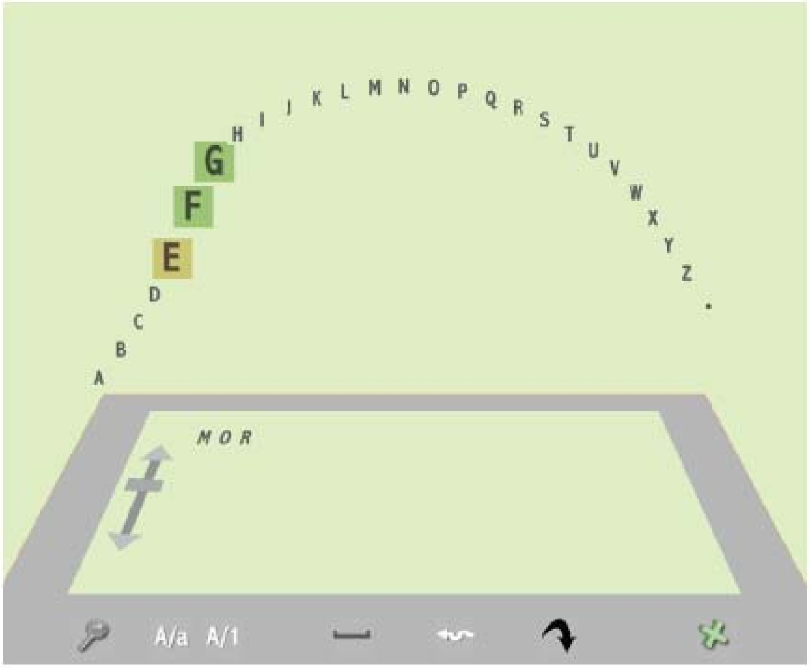


Figure 3. Screen shot of StarWrite; in the targeted area, characters are enlarged.

The user study

Methods

A virtual *QWERTY* keyboard served as a comparison baseline for our new approaches (see Marajanta et al., 2002, for a similar version). Dwell time for selection was set to 500 ms. As a further comparison, we used *Dasher* version 4.2.2 for Windows, with its default settings, but without text completion for comparison. In all our new interfaces, characters were ordered alphabetically. In *pEYEdit*, letters were sorted into 6 slices. Each group consisted of five characters. The second layer thus consisted of five character slices plus one slice used to exit the 2nd pie without writing. This cancel slice was always situated next to the centre of the 1st pie (see Figure 1). In *Iwrite*, characters were ordered starting from the left bottom of the frame, upwards (see Figure 2). In *StarWrite*, characters were ordered on the half-circle above the text entry window (see Figure 3). Delete, space and enter keys were presented below the text window and are operated with a dwell time of 500 ms. In all applications except Dasher, a "click" sound was played as auditive feedback and character highlighting as visual feedback for each selection.

The interfaces were presented on a 21'' CRT-Display. The gaze was trackred by using Eyelink2 (SRResearch). After practicing with one interface, typing of three times two familiar sayings was recorded. The experiment was divided in two days with permuted order of devices. Sixteen volunteers participated. Six of them had already trained in advance with the devices.

Results

We are still in the process of data analysis. Therefore, the reported data stem from three participants; two novices and one advanced user.

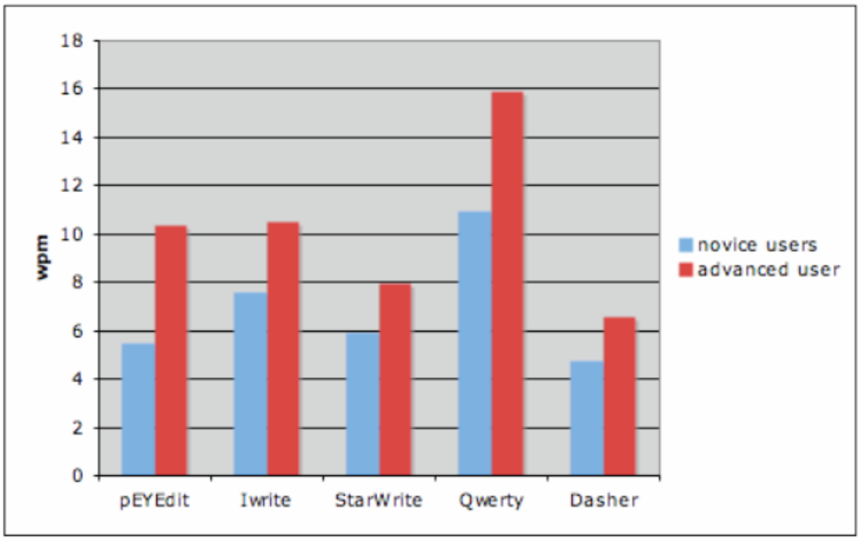


Figure 4. Mean typing speed in words per minutes for two novices (left balk) and one advanced user (right balk).

With the virtual *QWERTY* keyboard, users reached a mean of 10,9 wpm (novices) and 15,8 wpm (advanced). Maximal typing speed was achieved by the experienced user with 16,3 wpm. Using *Dasher,* participants reached a mean speed of 4,7 wpm. Fastest typing of 7,4 wpm was observed with the advanced user. With *pEYEdit*, novices typed 6 wpm (advanced: 10,9 wpm). With *Iwrite*, novices reached a mean speed of 7,6 wpm. The advanced user achieved to type up to 11,4 wpm. Using *StarWrite*, novices operated at 5,9 wpm, the advanced user at 8,4 wpm.

QWERTY and *pEYEdit* revealed the largest increment in performance between novices and the experienced user (4,9 wpm). In *Iwrite,* a difference of 1,3 wpm was observed, and in *StarWrite* of 2,1 wpm. Using *Dasher,* the advanced user typed 2,5 wpm more than the novices.

The three users all prefered the *QWERTY keyboard*. Novices second choice was *Dasher*, which was preference number three for the advanced user. The advanced user put *pEYEdit* on the second place, which was number three for the novice users. Both novices reported that they preferd *QWERTY* because it was familiar and much faster than the other interfaces. Eventhough *Dasher* was their slowest typing device, its dynamics gave novices the feeling of typing fast. The advanced user reported to have noticed his increasing speed with *pEYEdit*, and that searching became increasingly easy.

Discussion and Future Work

Comparing our results to the text entry speed mentioned in the literature (Marajanta et al. 2004, Hansen and Itoh, 2004, Miniotas et al., 2003) users had a well acceptable performance with all our interfaces. Performance with the new approaches was slower than with *QWERTY*, but faster than *Dasher*. Of course, *Dasher* is designed to be operated with word completion algorithms. Then, marks of more than 25 wpm can be achieved (Ward and MacKay, 2002). We have planned solutions to integrate word completion in all our interfaces. It can be integrated to *pEYEdit* either by adding more layers to the pie containing the word candidates, or by splitting the main pie into several menus with the word candidates and their declinations. In Iwrite, we saved the bottom of the frame for word completion candidates. In *StarWrite*, complete words can already be given in the text and be confirmed by an extra key. Remarkable is the fact that in *GazeTalk* (Hansen and Itoh, 2004), which is not faster than our systems, word prediction is already included to speed up text entry. Nevertheless, with our implementations based on letter-level text entry, the speed reported for Dasher can probably scarcely be achieved.

When comparing the three approaches, *pEYEdit* seems to be the most promising interface. Although not the fastest in novices, users liked the handling. Moreover, *pEYEdit* produced a strong learning effect. Probably, users learn the gaze paths of some letters, for example, move straight upwards in order to type the letter “m”, or that “t” is moving to the upper right and then down (see Figure 1). This strongly suggests that, even if *pEYEdit* cannot compete with eye typing performance with dasher in word completion mode, pie menus seem to be a suitable alternative for various tasks in gaze input. For example, augmenting the virtual *QWERTY* keyboard with *pEYEdit* could be a solution for word completion on QWERTY keyboards on gaze control. A similar approach has already been suggested for touch screen devices (Isokoski, 2004). The current implementation of *pEYEdit* is based on the cursor position. An implementation based on the path could avoid the exhibition of the pies to expert users, and could make the exertion of menus possible wherever the current gaze is located.

Although the current findings are preliminary, we can conclude that gaze-based text entry can easily be achieved without dwell time-based applications. Pie menus must be regarded as a powerful tool not only for mouse, but also for gaze controlled applications.

Acknowledgements

Thanks to Daniel Bierwirth, Petros Kapakos, Nicole Magiera, David Riedel, and Katja Schöllner for assistance in implementation and data collection.

References

Duchowski, A. T. (2003) Eye Tracking Methodology: Theory and Practice. Springer-Verlag, London.

Goettl, J. S., Brugh, A. W., and Julstrom, B. A. 2005. Call me e-mail: arranging the keyboard with a permutation-coded genetic algorithm. *In Proceedings of the 2005 ACM Symposium on Applied Computing. L. M. Liebrock, Ed. SAC '05*. ACM Press, pp. 947-951.

Hansen, J. P., Tørning, K., Johansen, A. S., Itoh, K., and Aoki, H. 2004. Gaze typing compared with input by head and hand. *In Proceedings of the 2004 Symposium on Eye Tracking Research & Applications ETRA '04.* ACM Press, pp. 131-138.

Huckauf & Urbina (accepted pending revision). Improving object selection in gaze controlled systems. *Transactions of Applied Perception*.

Isokoski, P. (2004). Performance of menu-augmented soft keyboards. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '04.* ACM Press, pp. 423-430.

Jacob. R. J. K. (1991) The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Trans-actions on Information Systems,* 9, ACM Press, pp. 152–169.

Kurtenbach, G. and Buxton, W. (1993) The limits of expert performance using hierarchic marking menus. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '93*. ACM Press, pp. 482-487.

MacKenzie, I. S. (2003). Motor behaviour models for human-computer interaction. *In J. M. Carroll (Ed.) HCI models, theories, and frameworks: Toward a multidisciplinary science*. San Francisco: Morgan Kaufmann, pp. 27-54.

MacKenzie, I. S. and Zhang, S. X. (1999) The design and evaluation of a high-performance soft keyboard. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit*. ACM Press, pp. 25-31.

Marajanta, P. and Räihä, K.-J. (2002) Twenty Years of Eye Typing: Systems and Design Issues. *Proceedings of Eye Tracking Research & Applications, ETRA2002.* ACM Press, pp. 15-22.

Majaranta, P., Aula, A., and Räihä, K.-J. (2004) Effects of Feedback on Eye Typing with a Short Dwell Time. *Proceedings of Eye Tracking Research & Applications, ETRA2004*. ACM Press, pp. 139-146.

Marajanta, P., MacKenzei, I. S., Aula, A., and Räihä, K.-J (2003) Auditory and Visual Feedback During Eye Typing. *CHI 2003*, pp 766-767.

Miniotas, D., Spakov, O. and Evreinov, G. (2003) Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space. INTERACT 2003.

Ward, D. J. and MacKay, D. J. C. (2002). Fast hands-free writing by gaze direction. Nature, 418(6900), pp. 838.

Ware, C. and Mikaelian, H.T. (1987). An Evaluation of an Eye Tracker as a Device for Computer Input, in Human Factors in Computing Systems*: CHI+Gl'87 Conference Proceedings*, Toronto. ACM Press, pp. 183-188.

Using Face Position for Low Cost Input, Long Range and Oculomotor Impaired Users

G. Julian Lepinski, Roel Vertegaal
Human Media Lab
Queen's University
Kingston, Canada
{2gjl, roel}@qlink.queensu.ca

Keywords
Face tracking, long range, low cost, oculomotor impairment

Introduction

While traditional gaze-tracking techniques aim to provide the user with an input method of comparable resolution to classic hand-based input, lower-resolution input may often suffice for a number of computing tasks. Low-res input also provides benefits with respect to cost and ease of use for many users, including users with oculomotor impairments such as Parkinson's, as well as applications for very large displays.

Overview

As discussed in (Card et al., 1991), the neck and head are significantly slower than the eyes, fingers and wrists in terms of input data rate. Many computer-based tasks, such as typing, strongly depend on high data rates and alternative input methods often seek to meet or approach these rates. There exist, however, a number of computer-based tasks which do not require such a high-rate input. There are a number of motivations for using low-resolution input when available, including:

- **Cost** – head-tracking only requires an inexpensive webcam and off-the-shelf computer vision software.
- **Hands-free use** – head-tracking allows the user to provide gross control while using the hands for fine control or other tasks.
- **Eyes-free use** – eye-tracking often impedes a user's ability to scan and peruse screen real estate as they must be conscious of the input they are providing with their gaze.
- **Disabilities** – head-tracking, which is low-resolution by nature, allows users who may have difficulty with fine-motor skills, such as users with Parkinson's disease, to execute tasks they couldn't otherwise complete. A number of other diseases and injuries specifically affect control of eye movement such as those discussed in (Ciuffreda et al., 2007).

We have identified a number of software tasks which, while usually controlled with the keyboard and mouse, often use only a few common actions that do not require pixel-perfect input. These task categories include:

- File and web navigation
- Navigation in virtual worlds/cartography software
- Control of audio/video software

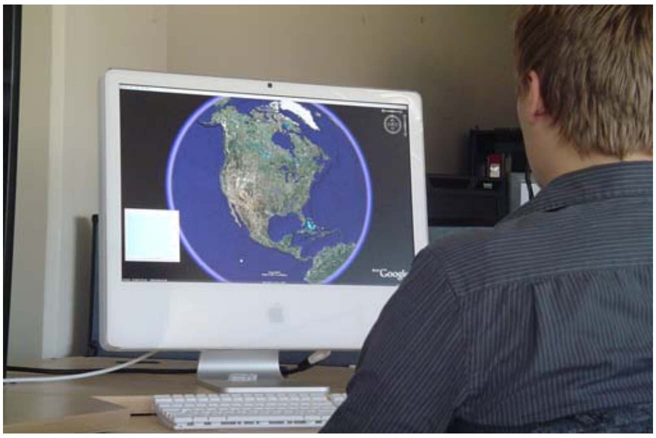


Figure 1. User controlling Google Earth with a standard webcam.

We chose the second category, Navigation, as our exemplary task, and have constructed a controller for Google Earth (GE), shown in use in Figure 1. Our controller takes the metaphor of user head movements in three dimensional space around a fixed, physical globe as a source of inspiration for input gestures. This mapping allows for almost instantaneous learning of controls, even for users unfamiliar with the system. As shown in Figure 2, moving to the left outside of the central zone causes a right rotation in GE, while moving to the right causes a left rotation. Moving upwards or downwards respectively causes the earth to rotate down or up, as if looking over or under the globe. In addition to the location of the face, the software also uses the size of the bounding sphere around the face as an indication of proximity to the camera. This data is used to trigger zoom thresholds, where moving towards the camera triggers a zoom-in, while moving away zooms out.

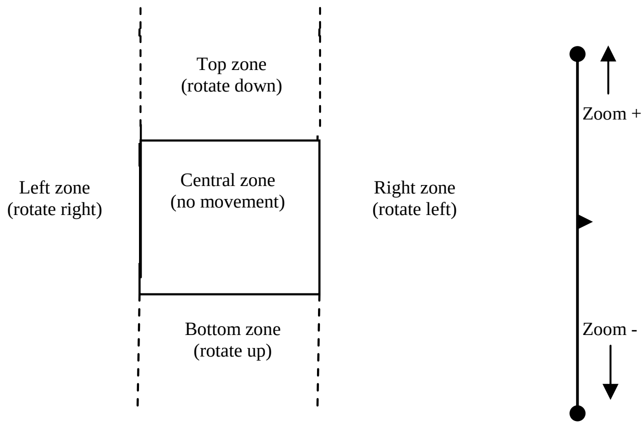


Figure 2. Movement zones and proximity (face size) threshold.

Our hardware consists of a low-cost 640x480 resolution Apple iSight webcam with a frame rate of up to 30fps. Apple has shipped all models of their personal computers with webcams built in above the screens since 2005, and it is reasonable to expect that webcam penetration into the consumer market may soon match that of the personal computer.

Our software uses a standard Haar filter included in the Open Computer Vision Library (http://sourceforge.net/projects/opencvlibrary/) to provide face coordinates and size data to a second Google Earth control module (Figure 3). The control module automatically calibrates to the user's starting position and assigns gesture zones to regions outside of the user's initial calibration zone. The user is able to see their position within the camera's view and with respect to their original calibration zone by using an onscreen display. Providing this feedback has shown in our tests to allow the user much more intuitive control. The software control module will be made available for download at the Human Media Lab website: http://hml.queensu.ca/eyeinthesky. Users with webcams should be able to install Google Earth and the control software and be up and running within minutes.

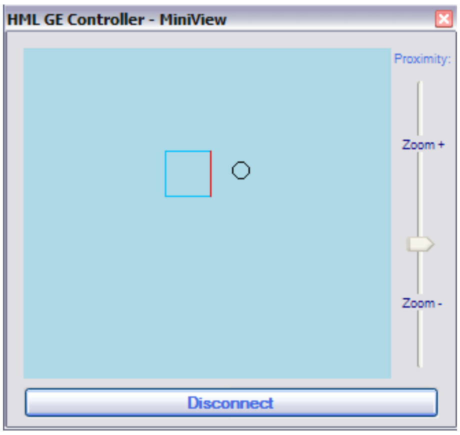


Figure 3. The GE Controller software giving the user feedback on position relative to calibrated centre.

Initial Evaluations

Informal tests indicate that users generally found the interaction techniques easy to learn, as well as use. Users accomplished various tasks, such as navigating to a specific location on the GE globe, consistently and with few errors. We further found that users were able to execute their tasks using the face-tracking controller with their hands occupied, as well as with their eyes looking elsewhere.

The face tracker is able to achieve sufficient accuracy to be used both at seated as well as standing distances, between 0.5m and 3m. The upper limit is likely a limitation imposed by the camera resolution – at higher resolutions we expect the distance to further increase. In seated operation mode the software is capable of tracking small changes in positition, allowing the user to control the application without requiring gross movements.

Future Research

While we have applied our low-resolution control methods to one specific application in this paper, we see a number of tasks which may be sufficiently executed without requiring high-resolution input from the user. In the future, we would like to explore a generalized framework that would allow developers and researchers to map any number of low-resolution controlled tasks to a suitable input device; providing real, workable input applications to users and the research community.

Acknowledgements

We gratefully acknowledge the contributions of Adam Archer, Rob Retchless and Mike Rooke.

References

Card, S. K., Mackinlay, J. D., and Robertson, G. G. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems, 9,* 2 (1991), 99-122.

Ciuffreda, K.J., Kapoor, N., Rutner, D., Suchoff, I.B., Han, M.E., Craig, S. Occurrence of oculomotor dysfunctions in acquired brain injury: a retrospective analysis. *Optometry* (St. Louis, Mo.). 78(4):155-61, 2007 Apr.

Accessible Web Surfing through gaze interaction

Emiliano Castellina, Fulvio Corno
Politecnico di Torino
Corso Duca degli Abruzzi,24
10129 Torino, Italy
{emiliano.castellina, fulvio.corno}@polito.it

Keywords Accessibility, Web Browsing, Eye tracking

Introduction

The Web is an increasingly important resource in many aspects of life: education, employment, government, commerce, health care, recreation, and more. However, not all the people can equally exploit the Web potential, especially people with disabilities. For these people, Web accessibility provides valuable means for perceiving, understanding, navigating, and interacting with the Web, so allowing to actively contribute and participating to the Web.

In 1999 theWorldWildeWeb Consortion (W3C) beganWeb Accessibility Initiative (WAI) to improve accessibility of Web. The WAI has developed a number of guidelines, concerning both Web contents (W3C, 1999; W3C, 2007) and user agents (Web browsers, media players, and assistive technologies) (W3C, 2002), that can help Web designers and developers to make Web sites more accessible, especially from the view of physically disabled people. There are many applications (screen readers, voice control, etc..) that get Web browsing more accessible for blind or deaf people, but only few applications (typically provided with commercial eye tracker) can allow Web navigation for disabled people that need gaze tracker devices. We propose, in according with WAI guidelines, aMozilla Firefox (the most used open source Web browser) extension[1] that implements a novel approach to Web site navigation also suitable for low resolution gaze tracker devices.

Accessible Surfing Extension (ASE)

Many aids developed for eye-tracking based web browsing try to cope with the basic difficulties caused by current web sites. In particular, the three main activities when browsing the web are, in decreasing order of frequency, link selection, page scrolling, form filling. Link selection is a difficult task due to the small font-sizes currently used, that require high pointing precision. In come cases link accessibility is also decreased when client-side scripting is used (e.g., in the case of pop-up menus created in Javascript, or with Flash interfaces) or time-dependent behaviors are programmed (e.g., the user has limited time to select a link before it disappears); such situations are incompatible with the current WAI guidelines for web page creation (WCAG).

Most approaches, such those provided in Erica System[2] and in Mytobii 2.3[3], tend to facilitate link selection by compensating the limited precision that can be attained with eye tracking systems: zooming is

[1] Extensions are installable enhancements to the Mozilla Foundation's projects and add features to the application or allow existing features to be modified
[2] Eye Response Technologies, http://www.eyeresponse.com/
[3] Tobii Technology, http://www.tobii.com

a common feature that increases the size of links near to the fixation point (either by screen magnification, or with widely-distantiated pop-ups) to facilitate their selection in a second fixation step.

In our work we explored a different integration paradigm, that decouples page reading from link selection. In a first phase, when the user is on a new web page, he is mainly interested in reading it, and perhaps he needs scrolling it. When an interesting link is identified, only then the user should be concerned for the mechanism for activating it. If the link is large enough (e.g., a button image), usually no help is needed (and the zoom interface would only interfere with user intentions). If, on the other hand, the link is too small, than a separate method for selecting is available.

At all times, the browsing window is integrated by a side-bar containing a link-selection interface, that is always synchronized with the currently displayed web page. When the user wants to select a link, he may use the sidebar that features large and easily accessible buttons.

This interaction paradigm has been developed as a Mozilla Firefox extension. This browser has been selected instead of others (Internet Explorer, Opera, Konqueror, Safari, etc..) for being open source, cross platform (Windows, Linux and Mac OS X), customizable and expandable and it has a simplified user interface.

The Accessible Surfing Extension (ASE) is a sidebar application inside the browser window(see Figure 1): whenever a new Web page is loaded ASE analyzes its contents, modifies the page layout and refreshes the graphical user interface.

Figure 1. Accessible surfing on Cogain.org

According to user preferences and skills, ASE allows users to navigate Web pages in two modalities:

- **Numeric mode**: each link in the Web page tagged with a consecutive small integer, shown besides the link text or image. Such links are always visible, non intrusive, and usually don't disrupt the page layout. The integer numbers are used by the user to identify uniquely the link he is interested in. At this point, the user turns his attention to the sidebar, where the ASE displays a numeric on-screen keyboard and a selection confirmation button. Users select the link by dialing its number with the on-screen buttons, and then confirming with the selection button. Feedback is continuous: the selection buttons reports the text of the currently dialed link number, and such link is also highlighted in the web page.

- **Browsing mode**: a different, simpler modality, can be activated for users less familiar with web browsing. In such case, page scrolling and link selection were blended in the ASE interface: through a "Next" button, that selects the next group of 5 links in the web page, highlights them in the web page, and updates 5 big buttons to select them, while simultaneously scrolling down the page to the region containing them. Thus, the main focus of the user is now on the ASE, to control web page scrolling. When the user finds an interesting link chances are that it is already present in one of the 5 buttons and can be directly selected with one fixation.

The ASE also allows page zooming, and the number and sizes of the buttons can be customized to be adapted to the specific eye tracking system precision.

ASE architecture

ASE is composed by five modules (see Figure 2):

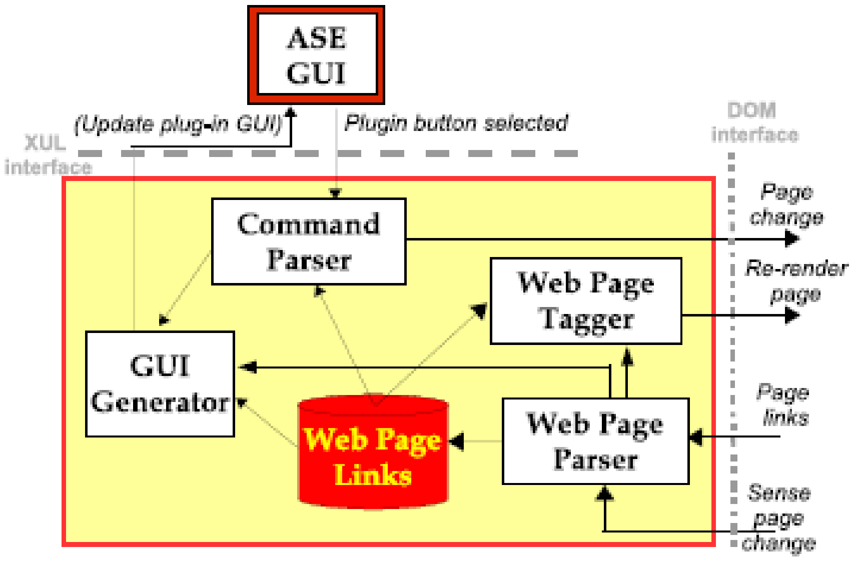


Figure 2. ASE architecture

Web Page Parser When a new Web page is loaded a sense page change event is received from this module that captures theWeb links through the DOM (Document Object Model) interface and stores them into the Web Links Database. The DOM is a platform and language-independent standard object model for representing and interacting with HTML or XML. The Web page parser sends an update message to the **GUI generator** and to the **Web Page Tagger** when it has finished the Web page parsing.

Gui Generator This module retrieves Web links from the database, then prepares and displays the graphical user interface according to the selected navigation modality.

Web Page Tagger It tags each Web page link, retrieved from the database, with progressive numbers and then sends a *re-render page* message to Web browser.

ASE GUI Users can interact with the browser through this XUL[4] graphical interface. When a user presses a button a *command message* is sent to the **Command Parser**.

Command Parser This module translates ASE user commands (i.e. link selection, zoom in, etc.) to Mozilla Firefox action (*page change*) commands.

Conclusions and future works

A preliminary usability experimentation has been conducted on a ALS user with the partnership of "Molinette Hospital of Turin". Mozilla Firefox (version 1.5) and ASE extension have been installed on ERICA eye-gaze system. The Molinette experimentation has involved, at now, twenty people with ALS, yet only one had the opportunity to connect to the Internet so as to test our software. The aim of the Molinette tests is to understand how and how much a communication device like ERICA can improve the life quality of terminally ill patients. ERICA systems has been tested by each patients for a week. Psychological questionnaires have been proposed to the users before and after the trial. This tests prove that the psychic condition is significantly improved after the trials. The user who has tried our software considers it fairly good and comfortable to use. When browsing the Internet, he actually ended up to prefer the ASE (numeric mode) than the ERICA zooming interaction.

The software presented in this paper is still a beta version; a public release is available at http://elite.polito.it/ASE. We plan to integrate the future versions of ASE with the COGAIN gaze tracking standard (Bates & Spakov, 2006).

References

Bates, R., & Spakov, O. (2006). Implementation of cogain gaze tracking standards. Communication by Gaze Interaction (COGAIN).

W3C. (1999). Web content accessibility guidelines 1.0. http://www.w3.org/TR/WAI-WEBCONTENT/.

W3C. (2002). User agent accessibility guidelines 1.0. http://www.w3.org/TR/UAAG10/.

W3C. (2007). Web content accessibility guidelines 2.0. http://www.w3.org/TR/WCAG20/.

[4] The XML User Interface Language, is an XML user interface markup language developed by the Mozilla project for use in its cross-platform applications.

The Exploration of Large Image Spaces by Gaze Control

Nick Adams[1], Mark Witkowski[2] and Robert Spence[2]
Department of Electrical and Electronic Engineering
Imperial College London, U.K.
[1]n8vision@gmail.com, [2]{m.witkowski, r.spence}@imperial.ac.uk

Keywords
Image space navigation, gaze control, Stare-to-Zoom, Head-to-Zoom

Introduction

We describe methods for navigating large variable resolution images primarily using eye gaze control. In many professions a user's task involves inspecting very large image spaces at a level of granularity ranging from an overview mode to study of fine detail. Such images are becoming commonplace, for instance: maps (e.g. Google Maps), earth imaging (Google Earth, NASA World Wind[1]), surveillance footage, architectural plans, astronomical images and medical images. Traditionally the navigation – i.e. pan and zoom – of such images has been achieved by well-established means of interaction such as mouse control. We explore the use of eye-gaze – possibly in conjunction with other forms of interaction – to control the actions of panning and zooming in the context of navigating or exploring very large images. Our exploration of these methods uses Google Earth satellite and aerial imagery to investigate how users can be enabled to traverse a complete virtual image from the broadest to finest levels of detail available. In choosing Google Earth for this purpose we note that this represents a very large image indeed[2], which is made readily and freely available on demand. The data is convenient, as it is both familiar and may be intuitively navigated by anyone with a rudimentary knowledge of geography. We investigate the comparative properties and advantages of two related methods of gaze control – Stare-to-Zoom (STZ) and Head-to-Zoom (HTZ) – and offer some preliminary findings and thereby give some pointers to designers who may wish to adopt these methods. Gaze control has been established for both disabled and able-bodied users for data input (e.g. Majaranta and Räihä, 2002), display inspection (e.g. Starker and Bolt, 1990) and spatial navigation (e.g. Bates and Istance, 2005).

Figure 1
Experimental Setup

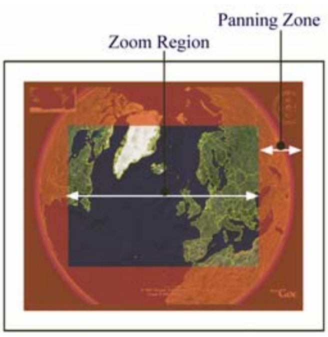


Figure 2
Pan and Zoom Regions

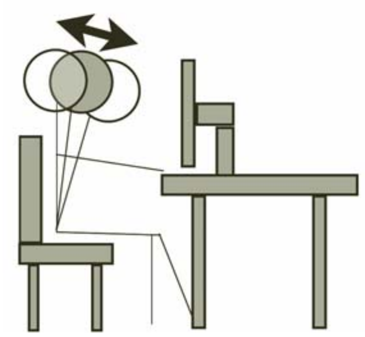


Figure 3
Head-to-Zoom Mode

[1] http://earth.google.com/; http://worldwind.arc.nasa.gov/

[2] The potential size of this "image" is staggering. Were the earth to be imaged at $1m^2$ over its entire surface, the resulting image would have the equivalent of ~$5x10^{14}$ pixels. Of course, civilian imaging does not offer this resolution yet, but the data set is still very impressive.

Pan and zoom by gaze control

We explored two methods of pan and zoom control effected primarily by eye movements. These modes Stare-to-Zoom and Head-to-Zoom are described in this section. Both rely on the determination of a point-of-gaze on a computer screen by appropriate technology. Figure 1 shows an overview of the equipment used.

Stare-to-Zoom

This method represents a single mode user interface. All control of the image traversal uses gaze position and timing alone. Figure 2 illustrates the general strategy. The screen is divided into a central zoom region, surrounded by a pan zone. The width of the pan zone (100 pixels top and bottom, 150 pixels left and right, on a 1024x768 screen) has been established empirically. It allows the user sufficient screen space to achieve uninterrupted panning. The panning rate used (~90 pixels/sec) allows some limited visual search within the outer panning region without causing zooming. Clearly, a faster effective panning rate (i.e. across the image space) can be achieved by zooming out to a lower resolution prior to panning. No zooming takes place while gaze is in the pan zone.

Sustained gaze in the zoom central region causes the image to zoom inwards. Normal saccades and fixations in the zoom region do not cause zooming and the image may be inspected in the usual way. Extended stationary gaze (>420 ms) initiates zooming at a comfortable rate. Zooming continues while the point of gaze remains stationary, as determined by a running calculation of the standard deviation of screen position. For non-central regions zooming is accompanied by panning towards the screen centre. This is inherent in the Google Earth interface. Once the identified feature is at the centre of the screen, zooming is uninterrupted until the maximum resolution is attained, while gaze is sustained on that feature. Zooming outwards is achieved by glancing directly at the camera fixed to the base of the screen (figure 1).

Head-to-Zoom

HTZ mode modifies the STZ mode just described by controlling zoom direction and rate by small movements of the head. The user zooms into the image by moving the head (or leaning) forward slightly and out by moving the head away from the screen by a small amount (~±40 mm), Figure 3. This mode allows the user to inspect any part of the image closely without initiating zoom, however the range over which the head may be moved is restricted (by the equipment properties) and the consequences of this are discussed later.

The position of the cursor remains visible in both modes. It is filtered to give the user the appearance of being centred at the point of gaze. That is, saccadic movements are preserved, but any eye movement "jitter" is suppressed during fixations.

Equipment issues

The system design and investigations described here used LC Technologies (www.eyegaze.com) eyegaze position monitoring equipment. Gaze position on screen is determined by comparison of corneal and retinal reflection from an axially mounted infra-red source on the eye-imaging camera mounted beneath the screen (figure 1). The system requires a brief calibration procedure prior to use by each new user. Accuracy is quoted as 1º (about 15 pixels), readings are made 60 times a second.

Eyegaze software (supplied) and Google Earth run on a single computer. Control of Google Earth is achieved by a combination of the Google Earth COM API and emulation of mouse clicks; direct interaction through the API having been found to be too slow for this type of real time application. The effective field of view of the camera relative to eye position is a volume of 100 mm^3. If the eye position leaves this volume tracking is lost, leading to erratic zooming behaviour in HTZ mode. An eye "icon" can be displayed on screen to assist the user with their head-positioning relative to the camera, although the system, by and large, provides its own feedback in terms of pan and zoom.

Procedure

We have conducted exploratory pilot investigations in a relatively informal manner. The main aims were to gain an insight into the relative merits of the two strategies at this prototype stage and to discover potential improvements for each option through user exposure prior to an extended study under controlled conditions. Seven volunteer participants were each asked to find the University site close to central London twice from a completely "zoomed out earth" manually using the (normal) mouse based interface. This was in order that the test was not influenced by the participant's ability to find the location. Then, using timed runs, participants were asked to zoom into the University site using the mouse, STZ and HTZ modes. Separately, participants were asked to give feedback on the experience, how the two methods compared and to comment on which had the better potential as a method of gaze controlled image inspection. In addition the participants were asked to rate the system on three factors, on a scale of 0 – 10: 1) How they rated their control of the system, 2) How immersed in the system they felt, and 3) How they rated their enjoyment of the system.

Preliminary results

Average time to complete the zoom task were 26 seconds for STZ and 32 for HTZ on the first attempt. Unsurprisingly, practice improved performance. As a control indication, using the mouse "normally" took 17 seconds. Using the scale described, the participants rated the two systems as follows:

Average rating (0 – 10, best)	**HTZ**	**STZ**
Control	6.3	7.3
Enjoyment	6.16	7.25
Immersiveness	6.6	7

While offering no statistical significance, the users consistently rated the STZ mode more highly than HTZ. However, users tended to prefer one method over the other quite strongly, and opinion was mixed. Those participants who valued HTZ over STZ gave the following reasons: a) It was more enjoyable, b) It gave more control because they could choose when to zoom, and c) It was more responsive as there was no gap between deciding to zoom and actually zooming in. Those participants who valued STZ over HTZ cited the following reasons: a) It was more predictable, b) It gave more control because they did not have to worry about their head making erratic and unpredictable motions, c) It required less coordination and cognitive load to operate.

We noted that there was a distinct tendency for participants to drift out of the field of view of the eyegaze camera. This was a particular problem with HTZ, as users often moved their head too close or pulled to far back, resulting in suspension of screen movement while they corrected their position. This detracted from the smooth operation of the system and lead to frustration. We believe that this is partly due to equipment limitations, which might be overcome using alternative technologies.

Discussion

These exploratory investigations have confirmed the value of eyegaze control in this navigation task and provided us with the confidence to proceed with a full-scale experimental investigation, which is now completed and is to be reported separately later. Several refinements of the techniques were discovered and improvements implemented as a result of this study. We have also added a further method (Dual-to-Zoom), combining gaze position input with manual zooming. Such techniques will no doubt find application for the disabled. We also believe that gaze control will be valuable as an auxiliary input mode for interface designers.

References

Bates, R. and Istance, H. (2005) Fly Where You Look: Enhancing Gaze Based Interaction in 3D Environments, Proc. COGAIN-05, pp. 30-32.

Majaranta, P. and Räihä, K.J. (2002) Twenty Years of Eye Typing: Systems and Design Issues, Proc. ETRA-02, pp. 15-22.

Starker, I. and Bolt, R.A. (1990) A Gaze Responsive Self-Disclosing Display, Proc. CHI-90, pp. 3-9.

Comparing two Gaze-Interaction Interfaces: A Usability Study with Locked-in Patients

Markus Joos[1], Susann Malischke[1], Sebastian Pannasch[1], Alexander Storch[2] and Boris M. Velichkovsky[1]

[1]Dresden University of Technology
Mommsenstr. 13, 01067 Dresden, Germany
joos@interactive-minds.de
+49 351 481 965 61

[2]Dresden University of Technology
Department of Neurology
Fetscherstrasse 74
01067 Dresden, Germany

Keywords
Eye typing, Locked-in Syndrome, ALS, Usability, Design of Experiment

Introduction

One of the most severe consequences for people suffering from motor neuron diseases like ALS is the loss of the ability to communicate. In order to overcome these difficulties a subclass of gaze-interaction interfaces called eye-typing systems had been developed. They mainly consist of an onscreen keyboard that is driven by gaze movements. Disabled people can regain their ability to communicate by “typing” with their eyes. Several different types of these eye-typing systems have been proposed. Some of them use hierarchical selection schemes with and without word prediction units; others rely on a single level graphical layout with all possible input possibilities visible at the same time. Obviously, the advantage of hierarchical design is that interfaces do not need especially exact gaze measurement while multi button gaze-sensitive systems require more accuracy. Besides merely technical requirements, usability becomes an important issue in evaluating these systems (see in particular ISO 9241) A general problem of usability testing is however, that it is hardly possible to judge the usability of a system *per se* as the measures have no absolute scale. A reasonable approach is therefore to compare two or more systems using the same measures. This comparative approach can be applied to usability analysis of eye-typing systems. In the same vein, one of the most important aspects of usability is “learnability”. Taking into account a relatively small number of users in need of eye-typing interfaces, a small-sample within-subject (repeated measurement) procedure should be the method of choice. In the present study, we applied this methodology to investigation of two eye-typing systems, Gazetalk (Hansen, 2001) and the Eyegaze System (Cleveland, 1994).

Method

Participants

Three female and one male subject participated in this study. All were diagnosed to have a locked-in syndrome caused by ALS (Amyotrophic Lateral Sclerosis) with the subtype of the bulbar form. All subjects had normal vision, or by glasses corrected to normal vision. None of the subjects had any previous experience with eye tracking. All subjects had German as their mother tongue. The investigation took place in the apartments of the subjects. Their mean age was 59.1 years with a standard deviation of 14.2 years. None of the subjects were able to communicate by voice or by manual signs.

Procedure

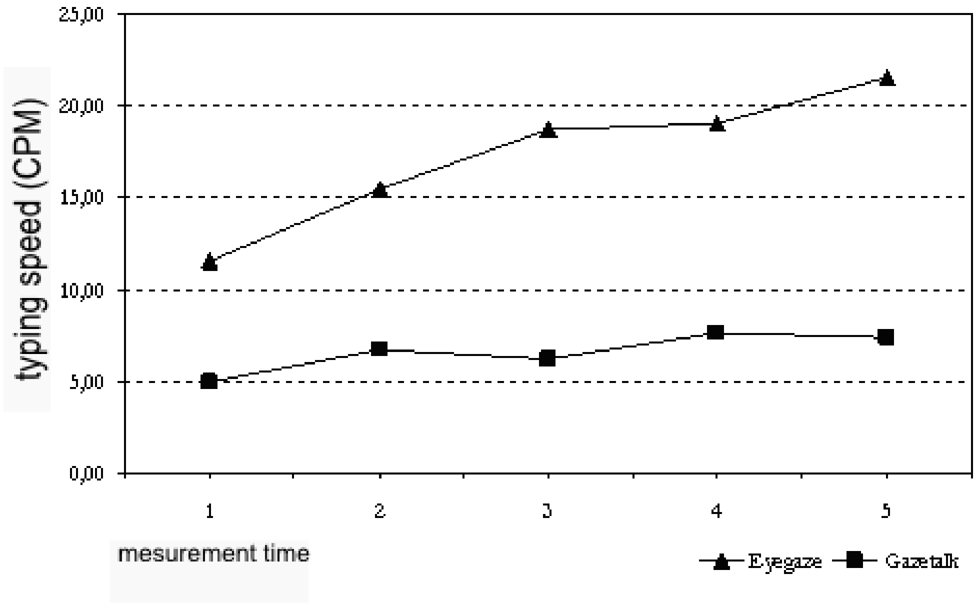
The core task for the subjects was to type 10 blocks of sentences with their gaze over a period of 5 measurement sessions. Each block consisted of 5 sentences with 131 characters per block. The sentences were a German translation of the “Phrase Set” of MacKenzie and Soukoreff (2003), which was specifically designed for experiments with eye typing. Subjects’ task was to type the sentences shown on a laptop computer as fast and as accuarate as possible. Two of the subjects started with Gazetalk and two with Eyegaze. These interface systems were then alternated between the measurement sessions. Before and after each testing session, subjects have to complete a special gaze-aware questionaire in order to assess their concentration, alertness and motivation on a four point Liekert scale. In addition, subjects completed a 12 item depression inventary scale specifically designed for people with ALS (Kübler, Winter, Kaiser, Birbaumer, Hautzinger, 2005) as well as the ISO-Norm 9241/110 usability questionaire (Prümper et al., 2006).

Apparatus

A binocular eye-tracking system from LC Technologies was used in this investigation. Sampling rate was 120 Hz and accuracy of the system was better than 0.5 ° in each session. Since the Eyegaze software uses an internal smoothing algorithm based on 10 samples, we developed a similar eye-mouse program for the Gazetalk system using a moving average smoothing algorithm with a bin size of 10 samples. Therefore, the gaze prediction delays were comparable. The dwell selection time was set to 800 ms for both systems.

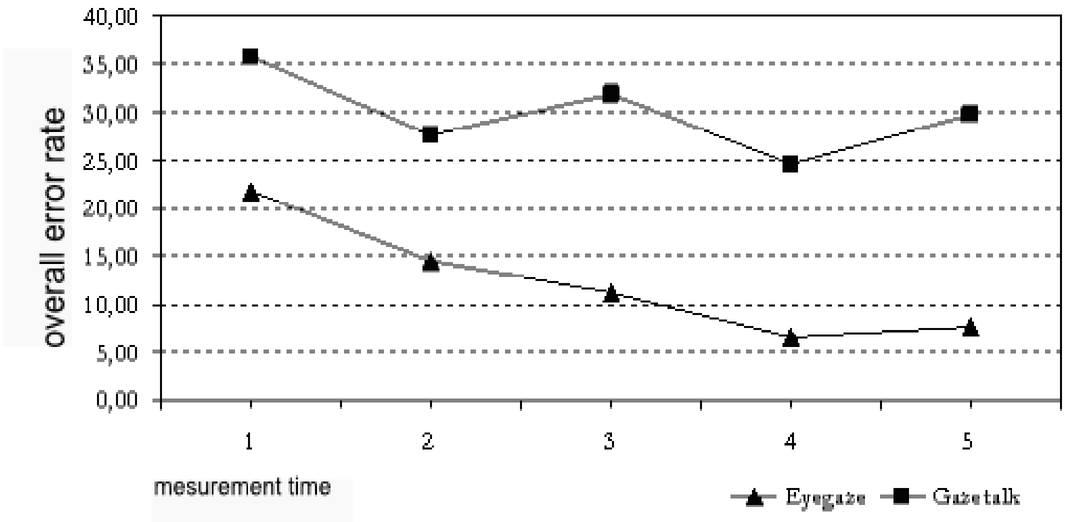
Results

Typing Speed:



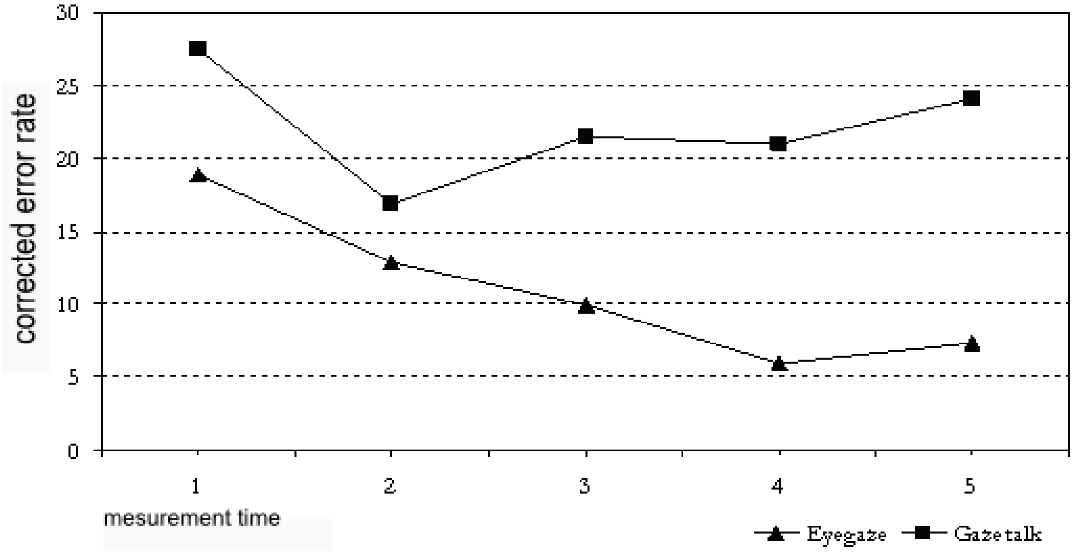
A 2 x 5 factors repeated measures ANOVA showed a significant main effect of eye-typing system, F(1,2)=56.268, *p*=.017, η^2=.966, as well as a significant interaction between eye-typing system and measurement time, F(4,8)=10.245, *p*=.003, η^2=.837 on the parameter of typing speed. As can be seen from the graphical representation of data, typing speed was higher for Eyegaze than for Gazetalk and this difference increased over time.

Overall Error Rate (Soukoreff, MacKenzie, 2003):



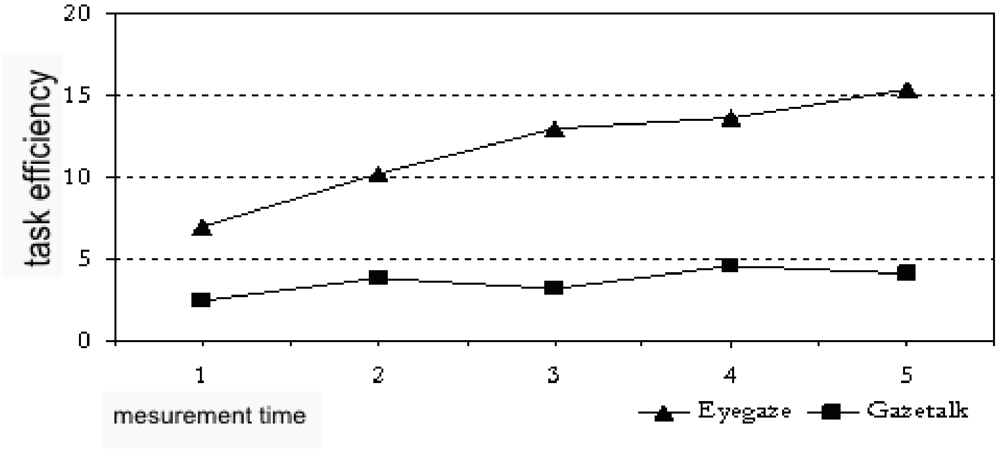
Repeated measures ANOVA for Overall error rate yielded no significant differences for mean effects as well as for the interaction. Only the decrease in the overall error rate over time was significant, $F(4,8)=5.286$, $p=.022$, $\eta^2=.725$.

Corrected Error Rate (Soukoreff, MacKenzie, 2003):



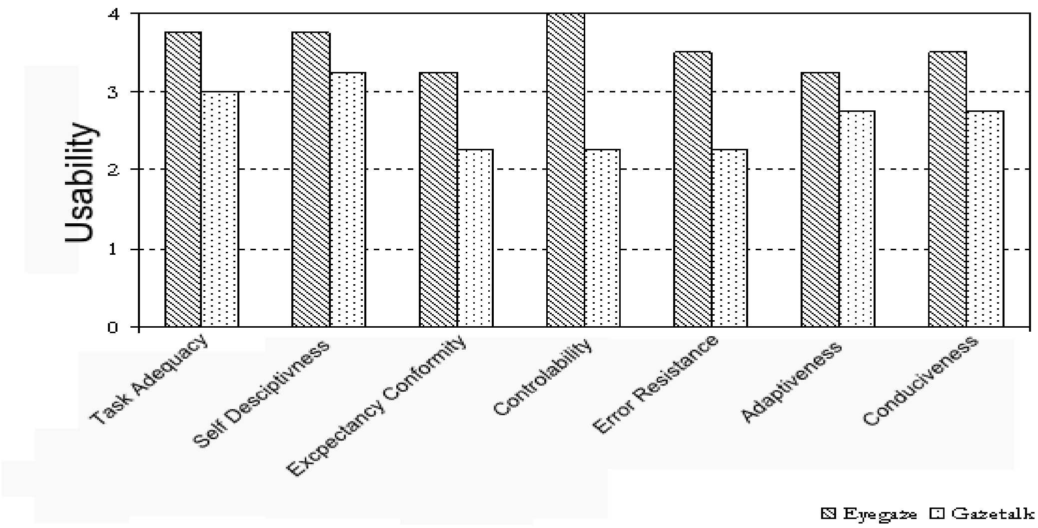
For the corrected errror rate, a 2 x 5 repeated measures ANOVA reveiled a significant difference between Gazetalk and Eyegaze, $F(1,2)=50.258$, $p=.019$. No main effect of measurement time and no interaction effect were found.

Task Efficiency:



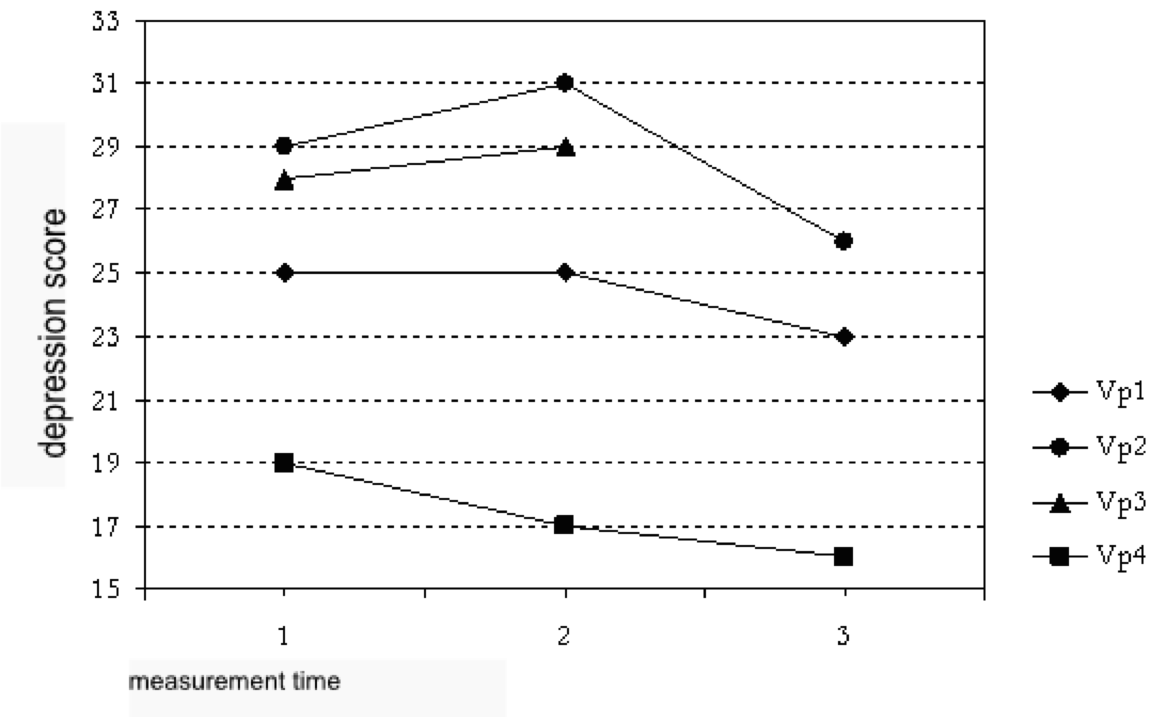
For task efficiency, repeated measures ANOVA showed a significant difference between the two systems, $F(1,2)=48.763$, $p=.020$, $\eta^2=.961$, as well as a significant increase of the efficiency over time, $F(4,8)=11.020$, $p=.002$, $\eta^2 = .846$ and a significant interaction, $F(4,8)=6.744$, $p=.011$, $\eta^2=.771$, demonstrating that the efficiency of dealing with Eyegaze system improved faster.

Subjective Ratings on the Usability Scale:



As the above figure shows, Eyegaze system was jugded more positive on all scales (overall two-sided paired Wilcoxon test T= 0, p<0.02), with an apparent tendency to highest differences in controllability followed by error tolerance. However, none of these specific differences reached significance in separate Wilcoxon tests.

Depression Scale:



The depression inventary was administered at measurement sessions 1, 3 and 5. Although a slight tendency of decreasing depression seems to be present in the data (see the above figure), this tendency was statistically not significant, ANOVA, $F(2,4)=4.923$; $p=.083$.

Conclusion and an Outlook

In this study, we compared two gaze-typing systems, Gazetalk and Eyegaze, in terms of objective performance measures and various subjective usability criteria. The results were rather straightforward. First of all, typing speed was faster for Eyegaze than for Gazetalk (at the first measurement session twice characters per minute, whereby at the last measurement this difference increased up to three times). Although not statistically significant, the Overall error rate was lower for Eyegaze than for Gazetalk. Even more the decrease of error rates from session 1 to session 5 was 66% for Eyegaze in contrast to only 17% for Gazetalk. In terms of uncorrected errors Eygaze was also superior, although this effect might be due to the lower overall error rate for Eyegaze. The most important performance criterion is task efficiency, which includes error rate as well as time for task completion. For this criterion Eyegaze was significantly better than Gazetalk with an efficiency value at the last session being three times higher. These objective criteria are as well reflected in the subjective assessments by the subjects.

Since rather big differences between the two systems were found, one can ask what has lead to these results. To answer the question one has to take into consideration that Gazetalk system was developed with a hierarchical GUI design particularly appropriate for a low spatial resolution eye-tracking device. This however must be disadvantageous due to the need of up to three times more selections for one letter compared to the explicitly flat design of Eyegaze. Although Gazetalk uses word prediction, which theoretically can increase typing speed by approximately 50%, in practical terms this enhancing did not happen. One explanation could be that inexperienced users, like those of the present study, may need more learning time to take full advantage of the Gazetalk word prediction module. Future investigations should therefore extend the temporal frame of the analysis as well as to include a larger number of participants. Indeed, the biggest critique on our study is the small sample size of only four persons, which make the application of elaborated statistical tools somewhat problematic. Nevertheless, we think that comparative usability studies should be preferably approached in small-size within-subject or even longitudinal designs in order to improve the eye-typing systems for the sake of those who need them most: the disabled users.

Acknowledgments

The study benefited from a permanent encouragement on the part of other members of COGAIN consortium. Michael Heubner, Fiona Mulvey, Franziska Schrammel and Sascha Weber contributed to the discussion of the study on all its stages. Last but not least, we would like to thank our subjects and their relatives who made this investigation possible.

References

Nancy R. Cleveland (1994). Eyegaze Human-Computer Interface for People with Disabilities. In *Proceedings of the First Automation Technology and Human Performance Conference*, Washington, DC, April 7-8, 1994.

Hansen, J.P., Hansen D.W., and Johansen, A.S. (2001). Bringing Gaze-based Interaction Back to Basics. *Proceedings of Universal Access in Human-Computer Interaction (UAHCI 2001)*, New Orleans, Louisiana, pp. 325–328.

Kübler, A., Winter, S., Kaiser, J., Birbaumer, N. & Hautzinger, M. (2005). Ein Fragebogen zur Messung von Depression bei degenerativen neurologischen Erkrankungen (amyotrophe Lateralsklerose)

[An inventory to measure depression in neurodegenerative diseases (amyotrophic lateral sclerosis)]. *Zeitschrift für Klinische Psychologie und Psychotherapie, 34* (1), 19-26.

MacKenzie, I. S., & Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems – CHI 2003*, pp. 754-755. New York: ACM.

Pataki, K., Sachse, K, Prümper, J. & Thüring, M. (2006). ISONORM 9241/10-S: Kurzfragebogen zur Software-Evaluation. In F. Lösel (Eds.), *Berichte über den 45. Kongress der Deutschen Gesellschaft für Psychologie* ,pp. 258-259. Lengerich: Pabst Science Publishers.