Technical University of Denmark

COGAIN2009 - "Gaze interaction for those who want it most"

The 5th International Conference on Communication by Gaze Interaction. Under the auspices of The COGAIN Association, May 26, 2009 Proceedings, The Technical University of Denmark, The ITMAN Graduate School, Lyngby, Denmark

Villanueva, Arantxa; Hansen, John Paulin; Ersbøll, Bjarne Kjær

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

COGAIN2009

"Gaze interaction for those who want it most"

The 5th International Conference on
Communication by Gaze Interaction

Under the auspices of:

The COGAIN Association

May 26, 2009

Proceedings

Editors:

Arantxa Villanueva
John Paulin Hansen
Bjarne Kjær Ersbøll

Sponsors:

The Technical University of Denmark
The ITMAN Graduate School

Technical University of Denmark

Informatics and Mathematical Modelling

Building 321, DK-2800 Kgs. Lyngby, Denmark

Phone +45 45253351, Fax +45 45882673

reception@imm.dtu.dk

http://www.imm.dtu.dk

IMM-TECHNICAL REPORT

ISBN: 978-87-643-0475-6

Preface

Welcome to the 5th international COGAIN conference on eye gaze interaction and the first under the auspices of The COGAIN Association. The conference emphasises user needs and future applications of eye tracking technology. Robust gaze interaction methods have been available for some years, with substantial amounts of applications to support communication, learning and entertainment already in use. However, there are still some uncertainties about this new technology amongst communication specialists and funding institutions. The 5th COGAIN conference will focus on spreading the experiences of people using gaze interaction in their daily life to potential users and specialists who have yet to benefit from it.

The theme of the conference is *"Gaze interaction for those who want it most".*

We present a total of 18 papers that have been reviewed and accepted by leading researchers and communication specialists. Several papers address gaze-based access to computer applications and several papers focus on environmental control.

Previous COGAIN conferences have been a most effective launch pad for original new research ideas. Some of them have since found their way into journals and other conferences. The high quality of papers presented at this conference makes us confident that they will also become markers of the pioneering work conducted in our field.

Enjoy the unique collaboration between users, researchers, communications specialists and system developers that has become the hallmark of COGAIN conferences.

We believe that the continuing success of the conference, organised for the first time by The COGAIN Association, strongly indicates the sustainability of the association – thereby taking over the long-term responsibility of the COGAIN Network of Excellence.

John Paulin Hansen & Arantxa Villanueva

COGAIN2009 Conference Chairs

and

Bjarne Kjær Ersbøll

President of The COGAIN Association

Table of Contents

Eye gaze assessment with a person having complex needs
– the benefits of using an assessment method

Margret Buchholz
DART, Queen Silvia Children's Hospital
Kruthusgatan 17, 411 04 Göteborg
margret.buchholz@vgregion.se
Phone +46 707 850276
Fax +46 31 155829

Eva Holmqvist
DART, Queen Silvia Children's Hospital
Kruthusgatan 17, 411 04 Göteborg
eva.holmqvist@vgregion.se
+46 707 850280

Keywords
assessment, assistive technology, disabilities, methodology

Introduction

This paper describes the assessment method used for an adult with severe Cerebral Palsy, which affects speech and motor functions as well as cognitive skills. The user has had experience of several different access and communication methods, but previous assessments throughout the years have never been successful in finding a method for independent communication in daily life. After an initial COGAIN trial, the user participated in an extensive assessment concerning computer access and communication. The presentation will give an understanding of the method in practical use as well as general advice on assessment strategies.

Background

To carry out assessments with persons with disabilities affecting motor performance as well as communication and cognitive skills can be a complex and complicated matter. The fact that the result of this assessment might be the user's only way of independent activity and communication makes high quality assessment necessary.

Compared to other technical aids fewer communication aids are prescribed. Where the user is provided with a technical aid, there are a problems with high rates of abandonment of assistive technology (Copley and Ziviani 2004). There are several reasons for this, for example: the maintaining/adjustments of the system are not carried out, attitude towards alternative methods, lack of training, lack of support and poor fit (Lidström and Zachrisson 2005; Johnson, Inglebret et al. 2006). An assessment process taking all these factors in account is a good start. Though there is literature in this area (for example: Glennen and DeCoste 1997; Braende and Halvorsen 2003; Beukelman and Mirenda 2005; Case-Smith 2005; Blesedell Crepeau, Chon et al. 2008) there has been lack of a more general model on how to carry out a computer assessment for people with complex needs.

An essential prerequisite for carrying out high quality assessments is a transdisciplinary team working towards a common goal, where each team member contributes with his or her specific knowledge and experience in relation to the user, the activity, the environment and the technology .

In AAC-centres throughout Sweden there is consensus about an assessment model documented in a book: *Be active using a computer* (Lidström and Zachrisson 2005) website (Lidström, Borgestig et al. 2005). They cover different kinds of computer access methods including eye gaze systems. The assessment model is based on the International Classification of Functioning and Health (ICF) (WHO 2001), which describes components affecting health from a biological, individual and social perspective with the environment as an important factor. The theoretical framework called the Canadian Model of Occupational Performance (CMOP) (CAOT 1997) is well known for its client centred perspectives.

Fredrik is 34 years old and has severe Cerebral Palsy. This affects speech and motor functions as well as cognitive skills and causes extensive problems with spasticity. During his school years, several assessments were made and Fredrik tried different computer access methods, mainly with different kinds of switches as well as an EEG/EMG-system. Despite extensive efforts, it was never possible to find a method for independent communication in daily life. Nowadays Fredrik works in a day centre specializing in people with severe communication problems.

Prior to this assessment, Fredrik's main means of communication was to answer yes and no by moving his eyes left or right. He also used a bliss chart with eye-pointing, but then only with a few skilled communication partners. Computers had not been in use for several years.

Due to COGAIN, Fredrik was given the opportunity to try an eye gaze system using simple activity grids with large targets. This was partly successful and it was assessed that eye gaze could be a useful access method for him. A thorough communication assessment at their regional AAC-centre was thus initiated.

The assessment

Identify, describe and prioritize issues

The work of identifying, describing and prioritizing occupational performance issues is a task mainly done by the user (if possible). The result is used as a foundation for further assessment. For this, we have to listen to the user and ask the right questions. There are different approaches depending on whether the user is an adult or a child and if there are cognitive or communication problems. An example of a good tool is Talking Mats™ (Murphy and Cameron 2006) which was used to interview Fredrik on his activity goals and preferences concerning the eye gaze system. Fredrik chose several areas for which he would like to use the eye gaze system and then prioritized them in order of importance. He wanted to be able to do the following activities: choose and listen to music, look at pictures, write shopping lists, listen to books/magazines, write with blissymbols, read and write e-mail, use the Internet, be able to participate in planning for his economy, watch sport results and watch movies.

Select theoretical approach

The theoretical approaches can include theories, models or paradigms and aims at finding a good support for further actions, including assessments and implementations. In Fredrik's case, the team uses theoretical framework suitable for his situation and condition. Here the ICF (WHO 2001), client centred models, CMOP (CAOT 1997), laws and praxis models concerning prescription on technical aids are applicable.

The result of the Talking Mats™ interview

Describe the conditions for the activities

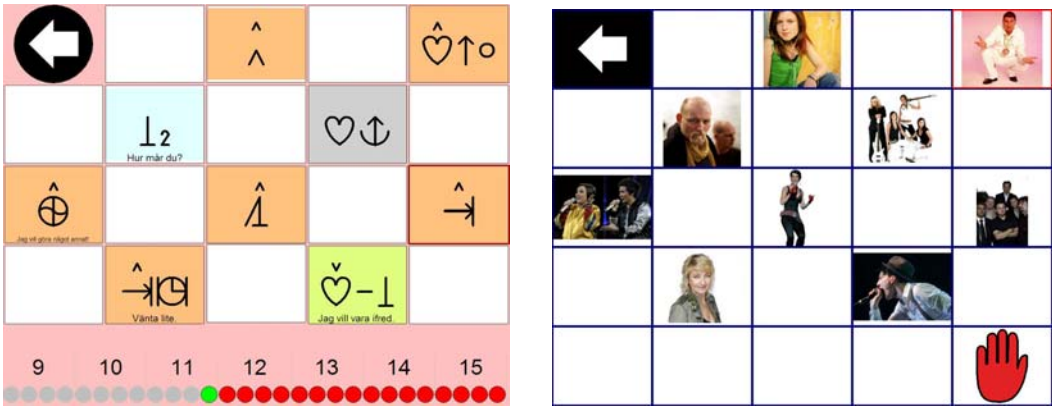
Here the team surveys the user's individual conditions as well as factors in the actual environment for the identified activities. It is advised to look at the strengths and resources of the individual, the environment, the technology and the activity. Diagnoses and status are collected through journals and tests. Examples of questions to be answered: Does the user have cognitive problems? Does this affect the understanding, memory, perception, orientation or concentration? Are there problems with vision or hearing? What motor functions and endurance does the user have? How can the user best make use of them? Which activities should be analyzed? It is also important to evaluate possible technical aids systematically according to how well they meet the needs of the user and the environment.

Describe targeted outcomes and develop an action plan

If the description of issues is clear, the goal setting is easier. It is important that the user and the team agree on the goals and that those are related to the activities the user wants to be able to carry out. The goals should be realistic, understandable and measurable. An action plan is completed, describing where, when, how and who is responsible for the planned measures. In Fredrik's case, the consensus discussion resulted in the main goal that was set to be: That Fredrik can use the eye gaze system for communication. This is a very general goal and hard to measure, so detailed goals were set up based on the list of Fredrik's desired activities, and that he would be able to perform these activities.

Implement plans

This section of the method describes how to plan and carry out an assessment, as well as how to carry out training and education. Fredrik's assessment was planned to include: Three sessions for eye gaze assessment, three sessions for choosing and finding the right strategies for use of communication software, a meeting to inform the team on the assessment results, including prescription of system and communication software. After the equipment was delivered, the team received education and a personalized communication setup was developed by the regional AAC-centre. The equipment was then introduced to Fredrik at the day centre and the staff started to train him in different activities. This was done step by step, after a thorough schedule, so that the training would be at the right level. During this period, the staff had support from the regional AAC-team on methodology as well as technology.



Fredrik's communication setup

Evaluate outcomes

After implementation, the user's occupational performance in the targeted activities is evaluated. It is preferable to use evaluation methods where the user takes an active part. Has the occupational performance changed? Does the system function satisfactorily? Are there new issues? Is there a need for further assessment? Sometimes new issues occur after the initial goals are reached. Technology often gives the user the possibility of continuous development of his or hers occupational performance and a change of life roles. This gives a continuous process with a need for new goal setting following the user's development. In Fredrik's case, the implementation period was carried out at the time of writing this paper. Evaluation will soon take place and the results from this will be part of the presentation.

Conclusion

When undertaking complex assessments, the user and the team benefits from a mutual roadmap, and a consensus on methodology makes it easier to cooperate in the team as well as together with the user and others. Evaluation of the process and results are easier with a documented methodology, which also facilitates tutoring of the rehabilitation team and others who will continue the work on a daily basis.

Acknowledgements

The authors would like to thank Fredrik, his family, his rehabilitation team and the staff at the day centre.

References

Beukelman, D. R. and P. Mirenda (2005). Augmentative and alternative communication. Supporting children and adults with complex communication needs. Baltimore, MD, Paul H Brookes.

Blesedell Crepeau, E., E. S. Chon, et al. (2008). Willard and Spackman's Occupational Therapy Lippincott Williams & Wilkins.

Braende, E. and A. K. Halvorsen (2003). Fra vilje til uttrykk. Metodikk for utprövning av betjeningsutstyr for barn med omfattende motoriske vansker. Oslo, Kommuneforlaget AS.

CAOT, Ed. (1997). Enabling Occupation: An Occupational Therapy Perspective. Canadian Association of Occupational Therapists. Toronto: ON, CAOT Publications ACE.

Case-Smith, J. (2005). Occupational therapy for children, Mosby.

Copley, J. and J. Ziviani (2004). "Barriers to the use of assistive technology for children with multiple disabilities." *Occup Ther Int* **11**(4): 229-43.

Glennen, S. L. and D. C. DeCoste (1997). Handbook of Augmentative and Alternative Communication. San Diego, CA, Singular Publishing Group.

Johnson, J. M., E. Inglebret, et al. (2006). "Perspectives of speech language pathologists regarding success versus abandonment of AAC." *Augment Altern Commun* **22**(2): 85-99.

Lidström, H., M. Borgestig, et al. (2005). "Aktiv med dator." Retrieved February 19, 2009, from http://www.hi.se/aktivmeddator.

Lidström, H. and G. Zachrisson, Eds. (2005). Aktiv med dator - möjligheter för personer med rörelsehinder. Stockholm, Hjälpmedelsintitutet.

Murphy, J. and L. Cameron (2006). Talking Mats. A Resource to Enhance Communication. Stirling, Scottland, University of Stirling.

WHO. (2001). "International Classification of Functioning, Disability and Health." Retrieved February 19, 2009, from http://www.who.int/classifications/icf/en.

Performance Evaluation of a Low-Cost Gaze Tracker for Eye Typing

Maria Barrett
IT University of Copenhagen
Rued Langgaards Vej 7,
2300 Copenhagen S, Denmark
mbarrett@itu.dk

Henrik Skovsgaard
Javier San Agustin
IT University of Copenhagen
Rued Langgaards Vej 7,
2300 Copenhagen S, Denmark
{hhje,javier}@itu.dk

Keywords
low-cost gaze tracking, eye typing, usability, universal access

Introduction

To some people with severe motor disabilities gaze interaction is the only way of communicating in their daily life. The systems must provide means for effective and efficient communication without discomfort. Gaze interaction systems built from off-the-shelf components decrease costs and increase availability for the disabled users as well as for students, researchers and developers.

Prior studies show that multimodal interaction using click activation increases the efficiency of gaze interaction with respect to gaze-based selection such as dwell (Zhang and Mackenzie, 2007). Other studies only examine single modality gaze interaction (Majaranta et al., 2004). Recent studies have shown that the performance of a low-cost gaze tracker compares well to the commercial gaze systems in target-acquisition tasks and eye typing (San Agustin et al., 2009).

The aim of this paper is to evaluate a low cost gaze tracker built from a standard web camera (less than $20) for eye typing. In the experiment, we test two robust, noise-tolerant gaze-typing applications, GazeTalk (Hansen et al., 2004) and StarGazer (Hansen et al., 2008). The activation modes used in both systems were mouse click and gaze selection with adjustable activation time.

Experiment

The two eye typing applications used in the experiment have different strategies for dealing with noise. StarGazer uses a pan/zoom, 3D interface to enlarge the objects on the screen thus making them easier to hit. From the starting position (see Figure 1), the full alphabet is visible and all characters can be activated with equal effort from the user. The user looks at the desired target, which will be enlarged and activated when the pre-defined activation time has elapsed. GazeTalk has large buttons made possible by hierarchical organization of the alphabet (see Figure 2). In GazeTalk the user activates the targets using dwell time activation. Word prediction was deactivated on both systems during the experiments but the built-in letter prediction in GazeTalk was enabled showing the six most likely letters on the main interface. The remaining 22 letters of the Danish alphabet required three keystrokes per activation.

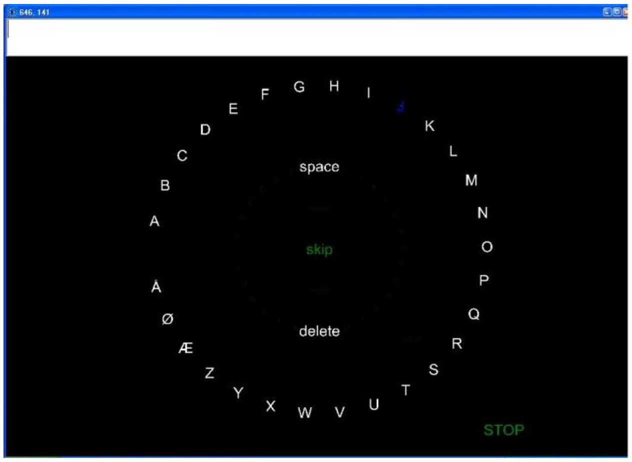


Figure 1. The StarGazer prototype system.

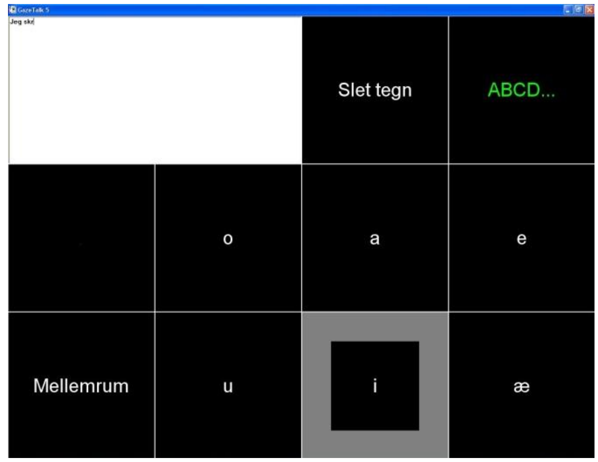


Figure 2. GazeTalk (version 5.2.2).

Participants

7 participants (2 women, 5 men, ranging from 20 to 35 years old), recruited from campus and nearby university, volunteered to participate in the experiment. All had normal or corrected-to-normal vision and 4 of the participants had prior experience with gaze interaction.

Apparatus

A 17" monitor with a resolution on 1280 x 1024 presented the typing interfaces. A 1.86 GHz Intel Dual Core PC with 3 GB RAM running Windows XP was used for the experiment. The mouse was a standard optical mouse and distance to screen was approximately 60 cm.

The gaze tracker is based on a Sandberg Nightvision web camera with a resolution of 320 x 240 pixels and 30 frames/sec. The camera has six built-in infrared lights that create a dark pupil effect. The gaze tracking algorithm tracks one eye and uses the pupil centre to determine the gazed coordinates on the screen. No hardware modifications are required.

The camera is mounted on a piece of balsawood placed between the user's teeth. An armchair was used to provide some support for the head during the test.

Figure 3. The input device. Sandberg web camera mounted on a piece of balsawood.

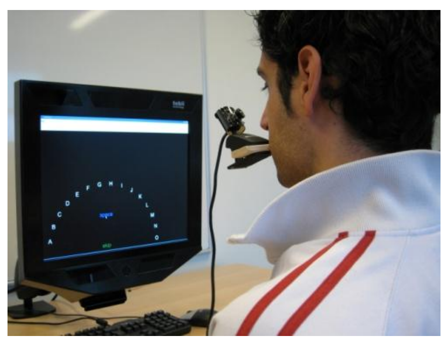


Figure 4. The input device in use. The user bites the balsawood and is depicted typing with the StarGazer application.

Task and procedure

The users were asked to type selected sentences (with an average of 25 characters per sentence) as quickly and accurately as possible. The sentences were Danish translations of the phrase set by Mackenzie and Soukoreff (2003).

No participant entered the same sentence more than once. After completing a sentence, the next was immediately shown to the user and a break was offered after five sentences. Before each block the participants were allowed to adjust the activation time themselves with an average final setting in the last session of 1375 ms for StarGazer with zoom activation and 403 ms for GazeTalk with dwell time activation.

Design

The within-subjects experiment tested the following four conditions and the design was repeated over three sessions: Stargazer with zoom activation, Stargazer with mouse click activation, GazeTalk with dwell time activation and GazeTalk with mouse click activation. A total of 420 sentences were typed (7 participants x 3 sessions x 4 conditions x 1 block x 5 sentences). The conditions were counterbalanced in order to neutralize learning effects.

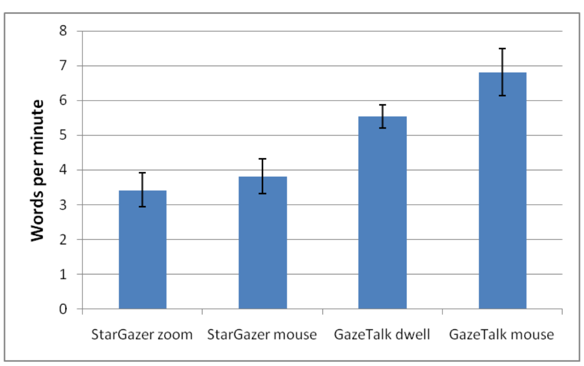
The performance metrics measured were WPM (words per minute) and MSD (minimum string distance). One word is defined as five characters including space. According to Mackenzie and Soukoreff (2002) the MSD should not be reported alone as it only reflects the errors in the final sentence. Therefore, we match it to NDC (number of deleted characters).

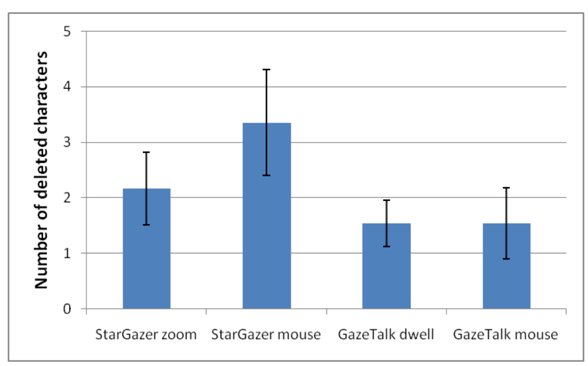
Results

Three one-way ANOVAs were conducted to compare the effects of interaction technique (StarGazer zoom, StarGazer mouse, GazeTalk dwell, GazeTalk mouse) on WPM, MSD and NDC. Figure 5 shows the average WPM, MSD and NDC. Sentences with an MSD value above six (49 out of 420) were considered to be outliers and thus removed prior to analysis.

The grand mean for WPM was 4.90. There was a significant effect from typing method on WPM, $F(3, 18) = 30.67$, $p < 0.05$. An LSD post-hoc analysis showed that WPM with GazeTalk was significantly higher than with StarGazer. No difference was found between click activation and dwell/zoom activation within each typing technique.

Typing method did not have a significant effect on MSD, $F(3, 18) = 0.417$, $p > 0.05$, but it did on NDC, $F(3, 18) = 3.51$, $p < 0.05$. An LSD post-hoc analysis showed that GazeTalk with dwell had a lower number of deleted characters than StarGazer with mouse.





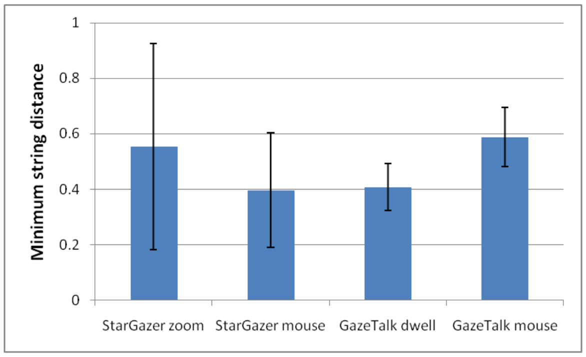


Figure 5. Average WPM, NDC and MSD for each typing method. Error bars show the standard error of the mean.

Discussion

Overall, the low-cost gaze tracker was usable for text entry with both applications. Text-entry speed was lower compared to other studies (e.g. Tuisku et al., 2008). However, in our experiments we did not use word prediction. Contrary to Zhang and Mackenzie (2007) we have not found significant effect of click activation.

GazeTalk obtained significantly higher typing speeds than StarGazer. However, typing speeds are not comparable between the applications due to the effect of letter prediction and difference in activation time. The theoretical upper limit for typing speed with the average activation time used in GazeTalk with dwell time activation in the last session (403 ms) is 29.8 WPM. This does not include reaction time and assumes that all desired letters are among the six most likely. With an activation time of 1375 ms it is not possible to produce writing speeds above 8.7 WPM. This can explain why GazeTalk produced significantly faster typing speeds than StarGazer.

After calibration of the mouth-mounted gaze tracker, even the slightest movement introduced a large offset relative to the screen which participants could compensate for by adjusting the head position in relation to the monitor. These adjustments were easier to perform in GazeTalk's grid-based interface than in StarGazer's zooming interface as the there are no fixed objects in StarGazer to use as reference points.

The weight of the camera and biting the balsawood did not cause any problems once the users got used to the input device. Only in the first session, the users averagely reported discomfort in mouth/jaw and on average no discomfort was reported in the succeeding sessions.

The typing speeds for GazeTalk with dwell time activation found in this experiment compare well to those found by Hansen et al. (2004) on a commercial gaze tracking system. In their experiment GazeTalk with dwell time activation (500 ms) obtained 6.26 words per minute on the second day (of two). On the third session of our experiment the users had approximately the same amount of practice with the system as the second day in Hansen et al.'s experiment due to different session lengths. In our experiment the final typing speed for GazeTalk with dwell time activation was 6.6 words per minute.

In a previous StarGazer study (Hansen et al., 2008), users typed with a fixed activation time resulting in an average speed of 4.7 WPM. The difference to the average typing speed for StarGazer with zoom activation in our experiment (3.4 WPM) can be explained by the use of a short, well known text string as typing task in Hansen et al.'s experiment.

Conclusion

Typing was possible with both eye typing applications, and click activation did not affect performance significantly. The results indicate that low-cost technology holds potential for being used for gaze interaction. The primary reason why StarGazer cannot reach the typing speed of GazeTalk is that it is possible to use shorter activation time with GazeTalk. For typing with applications designed for noisy gaze interaction, the writing speed of the low-cost gaze tracker compares well to a commercial gaze tracker. Comparing typing speeds to previous experiments, GazeTalk with dwell time activation achieves almost similar results (6.6 WPM achieved in this experiment compared to 6.26 WPM) Stargazer with zoom activation (3.4 WPM) does not quite reach the writing speeds found in earlier experiments (4.7 WPM), but the primary cause of the discrepancy is considered to be due to difference in difficulty level of the tasks.

References

Hansen, D. W., Hansen J. P., Skovsgaard H. H. and Møllenbach E. (2008) Noise Tolerant Selection by Gaze-Controlled Pan and Zoom in 3D. *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ACM, pp. 205-212.

Hansen, J. P., Tørning K., Johansen A., Itoh K., and Aoki H. (2004) Gaze Typing Compared to Input by Head and Hand. *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ACM, pp. 131-138.

MacKenzie, I. S. and Soukoreff, R. W. (2002) Text Entry for Mobile Computing: Models and Methods, Theory and Practice. *Human-Computer Interaction*, 17, Lawrence Erlbaum Associates, pp.149-198.

MacKenzie, I. S. and Soukoreff, R. W. (2003) Phrase Sets for Evaluating Text Entry Techniques. *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, ACM, pp. 754-755.

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

San Agustin, J., Skovsgaard H.H., Hansen J. P. and Hansen D. W. (2009) Low-Cost Gaze Interaction: Ready to Deliver the Promises. *CHI '09 Extended Abstracts on Human Factors in Computing Systems*. ACM.

Tuisku, O., Majaranta, P., Isokoski, P. and Räihä, K. Now Dasher! Dash away!: Longitudinal Study of Fast Text Entry by Eye Gaze. *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ACM, pp. 19-26.

Zhang, X. and MacKenzie, I. S. (2007) Evaluating eye tracking with ISO 9241 - Part 9. *Proceedings of HCI International 2007*, Springer Heidelberg, pp. 779-788.

Text Editing by Gaze: Static vs. Dynamic Menus

Päivi Majaranta[1], Niina Majaranta[1], Gintautas Daunys[2]. and Oleg Špakov[1]

[1] Department of Computer Sciences/TAUCHI
33014 University of Tampere, Finland
{forename.surname}@cs.uta.fi

[2] Siauliai University
Vilniaus str. 141-413, Siauliai, Lithuania
g.daunys@cr.su.lt

Keywords
Gaze typing, text entry, gaze input

Introduction

Eye tracking enables communication and computer control by gaze for people with severe disabilities: the gaze direction is used to point at items on a computer screen, for example, to point at a letter on an on-screen keyboard (Majaranta & Räihä, 2007). Dwell time (a prolonged gaze longer than the time needed for normal viewing, typically >500 ms), is used to distinguish the intentional command from casual viewing.

Even though it is easy to point at items by gaze, direct mapping of the eye movements into “eye mouse” coordinates is problematic. The accuracy of the measured point of gaze is usually about 0.5 degrees, after a successful calibration. However, due to drifting, the practical accuracy is often much worse (from 1 degree up to several degrees). This means that selection of small items is hard by gaze alone, without supporting techniques such as zooming (Bates & Istance, 2002; Skovsgaard et al., 2008).

There are several practical implications for the inaccuracy. Since the measured point of gaze has drifted a few pixels off from the actual focus point, it is difficult to place the cursor exactly on the desired location in the body text. Furthermore, a gaze operated on-screen keyboard also requires large buttons that may allocate most of the available screen space. Since the keyboard itself takes a lot of space, there is not much space left for editing commands (such as copy, paste, bold, italics, etc.). Gaze based text entry systems typically provide a backspace or undo key for immediate corrections. However, the editing commands are often hidden in the virtual keyboard’s menu structure.

Using the same modality for both input and output presents another kind of challenge for the interface design. Since the gaze is needed for selecting the button, the user cannot see the effect on the text simultaneously but needs to leave the keyboard area to review the result of the action on the text.

We developed a dynamic pie-like menu that can potentially facilitate the task of text editing by gaze. A pie menu is a pop-up menu that appears at the place of focus. The menu items are placed in a circular pattern around the center of the pie. Pie menus have been proved to be faster than ordinary linear menus in normal mouse-based interaction (Kurtenbach & Buxton, 1994). Our assumption is that by having the editing commands near the focus of interest can facilitate the editing process by gaze. In this paper, we first briefly review related research. We then introduce our prototype of a dynamic pie menu for text editing by gaze and report preliminary results from the first user trial in which we compared the dynamic pie menu with a static editing menu. Finally, we discuss ideas for further improvements and future work.

Related Research

Huckauf and Urbina (2007) developed a pie menu based system “pEYEs” (also called pEYEwrite) for gaze based text entry. The letters were grouped into the sectors of the pie menu. To type a letter, the user first selects the sector that contains the desired letter. It opens a sub-menu that contains a separate sector for each of the letters in the group. No dwell time is needed, since the focused sector is selected by looking at or crossing a special selection area on the outer edge of each sector. The letters are located so that the most common letter of each group is located at the direction of the sector, thus the user can select a letter in a long continuous gaze gesture into that direction. For example, a look up selects the sector containing ‘e’, and another look into the same direction (up) selects the subsector ‘e’ in the sub-menu.

Tien and Atkins (2008) tested different menu layouts for gaze interaction: a layout that resembled a typical drop-down menu, a layout resembling typical gesture based menus found in hand-held devices, and a variation of the gesture based menu adjusted for gaze, with big buttons and near distance. They did not find significant differences in task times between the layouts. After the initial experiment they implemented several improvements for the menu designed specifically for gaze, for example, a “snap-on” feature that fixed the eye mouse cursor to the centre of the button, and a feature that opened the menu with a quick off-screen glance to the left. In the follow-up experiment they found that after memorizing the menu commands, participants were able to perform menu selections using dwell times as short as 150 or 180 ms.

Also Kammerer et al. (2008) experimented with three different designs of multi-level menus operated by gaze. They found that a semi-circle menu was better suited for selection by gaze than a full-circle layout or a linear (conventional) menu design. Based on participants’ subjective experience, Kammerer et al. stated that the major drawback of the full-circle menu was its confusing arrangement (widespread and ungrouped menu items) and long distances between menu items.

Dynamic Pie Menu for Text Editing by Gaze

We implemented a prototype of a gaze operated dynamic pie menu for text editing (illustrated in Figure 1 on the left). The pie menu is shown on the point of the user’s focus when the user fixates on the text for longer than the predefined dwell time (1500 ms). The cursor (caret) is located in the center of the pie. The user can see the text through the central hole and also the menu items are partially transparent.

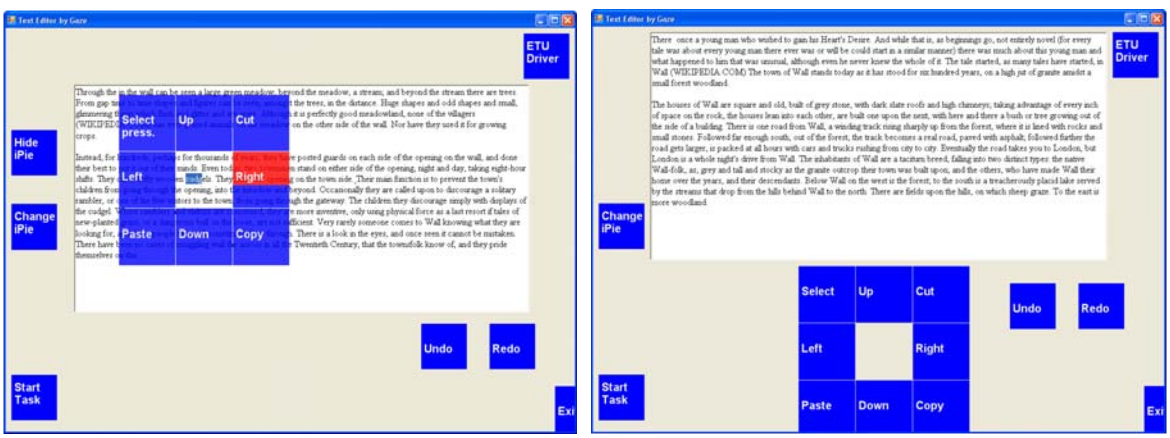


Figure 1. Dynamic pie menu (on the left) and static menu (on the right). The image on the left also illustrates feedback shown on the progression of the dwell time on the ‘right’ menu key.

The user can fine-tune the (often misplaced) cursor position using the left, right, up, and down keys. The pie menu moves along so that the cursor is always located in the center. To select text, the user needs to press (dwell on) the ‘Select’ button, and then move the cursor using the direction keys. An editing command such as ‘Copy’ can then be executed for the selected text. Our experimental prototype included functions for basic text editing (cut, copy, paste) and text formatting (bold, italic, underline). The commands on the pie menu can be changed by using the ‘Change iPie’ button (located on the left side of the application window).

In addition to the dynamic pie menu, we also implemented a static menu that stays at a fixed location on the bottom of the screen (see Figure 1, on the right). The layout and functionality of the keys were the same for both menus. The dwell time for selecting a button was set to 1000 ms for both. If the user kept on looking at the button it started to repeat the click with an interval of 450 ms. Durations are based on pilot tests.

Method

We conducted an experiment with 13 participants (10 males, 3 females, 19-26 years, mean 21) to study the usability of the dynamic pie menu for text editing by gaze. Participants were university students, with good computer skills and average to good text editing skills. All were novices in editing text by gaze but two had some previous experience on gaze control and one had participated in an eye tracking related experiment. The Tobii 1750 eye tracker was used with the COGAIN ETU Driver to track the gaze.

The experiment was a within subjects study with two conditions: dynamic pie menu and static menu (illustrated in Figure 1). The participants were assigned into two groups; participants in the first group started with the dynamic pie menu, and participants in the second group started with the static menu.

The participants were first briefed about gaze interaction and the experiment. They were informed that we wanted to compare two different interfaces for editing text by gaze. They then filled in a pre-questionnaire. Each test started with calibration. Before the actual test, the experimental software was introduced to the participants, starting with the condition that was assigned to the participant, and they had a change to practice using it with two simple tasks and to ask any questions.

During the test, each participant finished six similar tasks with both interfaces. Each task started with the press of the ‘Start Task’ button and ended by selecting ‘End Task’. Participants started with four simple formatting tasks, for example, to select a word and bold it. The last two tasks were text editing tasks, where the participant had to move a word or swap the locations of two words using the cut and paste commands.

After finishing all six tasks with one condition, the participants were interviewed about the first design. The same procedure was then repeated with the second condition, starting with the introduction of the interface and practice, and ending with the interview (the same questions were asked from each user after both conditions). After finishing both conditions, the participants filled in a questionnaire where they had a chance to compare the two designs and we interviewed them.

Preliminary Results

We lost data from several tasks from several participants due to technical problems. A few participants had poor calibration, which affected their performance. In addition, there was a bug in the experimental software that we noticed only after the tests had begun. Therefore, we will not report statistically significant results for the performance measurements; instead, in this paper we focus on reporting initial user reactions and ideas for further improvement.

Despite loosing some of the data, we did look into task completion times (including only successfully finished tasks with no bugs, compared with the corresponding tasks from the other condition for each participant). There seems to be a trend in the task times indicating that the participants performed faster in the simple formatting tasks (tasks 1-4) using the dynamic pie menu (with the average grand total of 38 seconds) compared to the static menu condition (aver. total 47 sec). However, when completing the more complex editing tasks (tasks 5-6), they performed slower using the dynamic pie menu (with the average grand total of 77 seconds) compared to the static menu (aver. total 67 sec).

Out of the 13 participants 8 preferred the static menu over the dynamic pie menu (preferred by 5) if they had to choose only one. If they had a chance to use both, 5 would still prefer using the static design only, 3 would prefer dynamic and 5 would like to use both (especially after further practice, as noted by a couple of participants).

We were interested in the usefulness of having the navigation keys on the dynamic pie menu, since we assumed they would be especially useful for adjusting the location of the cursor in the text, even if all other functions (formatting and editing) were placed on a static menu. We asked the participants if they felt it was easier to use the arrows on the dynamic or the static menu. Five participants felt it would be best to have the arrows on the static menu (to avoid confusion, as some of them commented). Others felt placing the arrows on the dynamic menu was indeed a good idea (4 participants), or probably a good idea (4 participants).

We also asked the participants to rate which one of the designs was faster, easier to use, more comfortable and easier to the eyes. The dynamic and static design both got equal number of votes for being faster (6/6, plus 1 "cannot say" reply). The static design got more votes for all other categories (number of votes for static/dynamic/cannot say): easier (9/3/1), more comfortable (7/5/1), easier to the eyes (7/3/3).

During the interview, we asked what was most difficult in using each of the menus. For the dynamic pie menu, 3 participants felt that the menu disturbed visibility (of the text under it or the interface in general), and 3 felt it was difficult to swap the gaze between the menu buttons and the text (to see the effect of e.g. selection). For the static menu, 4 participants felt that switching between the menu and the text was difficult, and 3 participants felt placing the cursor on the correct place was difficult. Other difficulties observed by more than one participant were related to gaze interaction in general, such as the difficulty of fixating in the same location for long enough, or a feeling of rush when the dwell time was running out, or difficulties related to the implementation of the experimental software. For example, we had implemented a feature that automatically hid the dynamic menu if the user looked at the grey area around the text field for longer than the threshold. However, there was a bug in the implementation since it sometimes caused a disappearance of the dynamic menu in the middle of text editing.

For both designs, a few participants complained that it was hard to select a full word and to remember how many times they had to eye-press the "left" or "right" to select all of the letters in the word. Some participants suggested that there should be an option to select a full word, or an option to define the starting and ending point for the selection (instead of repetitive presses of the left or right arrow). Even though most participants felt that the static menu was easier to use and perceive as it stayed in the familiar location, they also wished it was nearer to the text. Several participants wished they could adjust the transparency level of the dynamic menu's buttons. Other suggestions for improvement for the dynamic menu included replacing the buttons' caption texts with icons, having more options (more buttons or sectors for the pie) or placing the 'Change iPie' button (which swaps between the formatting and editing menus) into the dynamic menu itself for easy access. Some also felt the buttons were unnecessarily big and too far apart.

Discussion and Future Work

Even though most participants preferred the static menu over the dynamic pie menu, we believe there is potential in having the editing commands in a dynamic pie menu. First, we observed a trend in having faster task competition times using the dynamic pie menu for simple formatting tasks. Second, some participants preferred the dynamic pie menu and several more would like to have both options available. It is worth noting that there were more bugs in the dynamic pie menu condition than in the static menu condition, which may have affected the participants' subjective experience (even though we asked them to ignore the bugs in their ratings). The current experiment was very short and participants were novices in gaze interaction, therefore we believe the difficulties related to gaze interaction in general may have also affected the results. More practice would be needed to see the full potential of both solutions. Thus, we plan to organize a longitudinal experiment after improving the design (and correcting the bugs, obviously).

We agree with the participants that the text in the menu buttons should be replaced with icons. Icons would be fast to recognize and would not disturb the visibility of the text as much as the current design which has a partially transparent text buttons over the body text.

In the current implementation, the buttons in the dynamic pie were basically normal dwell time activated buttons. In the future design, we want to test a menu that looks like and operates more like a pie menu: it would be circular with sectors near each other. The sectors could also be selected by simply looking over the outer edge of the sector (similarly as in the pie menu design by Huckauf and Urbina, 2007): as long as the user views the command icon in the sector, nothing would happen but as soon as the gaze crosses the sector's outer edge, the sector would be selected – or a new sub-menu (with sub-sectors) would be opened. For example, the basic layout could have the arrows (as icons) and other sectors for formatting and editing. Those could open a next level of command, for example, activating the formatting sector could show a sub-menu for bold, italics and underline.

Our prototype did not allow using the dynamic pie menu near the edge of the screen and that is why a fairly large empty (grey) area was added around the text field. This problem could be solved by implementing the half-circle layout suggested by Kammerer et al. (2008). Dynamically changing the orientation of the half- (or partial) circle layout could easily compensate for the lack of space in one direction.

To our knowledge, editing text by gaze has not been researched. This pilot study is the first step towards more user friendly text editing by gaze. We believe this area offers a rich set of opportunities for future research and development.

References

Bates, R. & Istance, H.O. (2002) Zooming interfaces! Enhancing the performance of eye controlled pointing devices. *Proceedings ASSETS '02*, ACM Press, 119-126.

Huckauf, A. & Urbina, M. (2007) Gazing with pEYE: new concepts in eye typing. *Proceedings APGV '07*, ACM Press, 141-141.

Kammerer, Y., Scheiter, K., & Beinhauer, W. 2008. Looking my way through the menu: the impact of menu design and multimodal input on gaze-based menu selection. *Proceedings of ETRA '08*, ACM Press, 213-220.

Kurtenbach, G. & Buxton, W. (1994) User learning and performance with marking menus. *Proceedings of CHI'94*, ACM Press, 258-264.

Majaranta, P. & Räihä, K.-J. (2007) Text Entry by Gaze: Utilizing Eye-Tracking. In I.S. MacKenzie & K. Tanaka-Ishii (Eds.), *Text entry systems: Mobility, accessibility, universality*. San Francisco: Morgan Kaufmann, 175-187.

Skovsgaard, H.H.T., Hansen, J.P., & Mateo, J.C. (2008) How can tiny buttons be hit using gaze only? *Proceedings of COGAIN 2008*. CTU Publishing House (ISBN 978-80-01-04151-2), 38-42.

Tien, G. & Atkins, M. S. (2008) Improving hands-free menu selection using eyegaze glances and fixations. *Proceedings of ETRA '08*. ACM Press, 47-50.

Selecting with gaze controlled pie menus

Mario H. Urbina
Bauhaus-Universtät Weimar
Bauhausstr. 11
99423 Weimar, Germany
mario.urbina@uni-weimar.de

Maike Lorenz
Bauhaus-Universtät Weimar
Bauhausstr. 11
99423 Weimar, Germany
lorenzmaike@yahoo.com

Anke Huckauf
Bauhaus-Universtät Weimar
Bauhausstr. 11
99423 Weimar, Germany
anke.huckauf@uni-weimar.de

Keywords
Gaze-based interaction, user interfaces, input devices, pie menu, selection methods.

Introduction

Since the early 90's, it is known that pie menus are promising tools, which could replace in some cases pull-down menus (Hopkins, 1991). Pie menus are two-dimensional circles, centred on the mouse cursor. Their elements are ordered like pie slices. This means, all elements are at the same distance from the cursor, which accelerates selections relative to pull-down menus, where a user needs to find the desired item by iterating downwards through the menu. Moreover, pie menus allow a fluent transition from novice to expert usage: menu items are always placed at the same position. Hence, novice users can search for the item whereas expert users know the item position, just needing to perform a certain movement trajectory without needing to look at the menu. This way of selection is also called "marking ahead" (Kurtenbach and Buxton, 1993).

One important feature responsible for the good usability of pie menus is that the slices increase in size towards their outer border. This facilitates accurate slice selection, even in conditions of low spatial accuracy of the input device - what can enhance performance especially in gaze input. However, although they have lot of advantages, pie menus never became popular. This can be attributed mainly to the exceptionally high familiarity of users with pull-down menus (e.g., Zhai, 2008). Nevertheless, for gaze input, pie menus have already been shown to be a reliable alternative to pull-down menus (Kammerer et al., 2008). As we have shown earlier, they can be effectively and efficiently used even for tasks like eye typing (Urbina and Huckauf, 2007) or desktop navigation (Huckauf and Urbina, 2008a).

The problem of selecting a pie slice by gaze

In manually controlled pie menus, a slice is selected via pressing a key (for mouse input) or via tapping (for stylus input). In gaze controlled pie menus, selection is usually performed via dwell times (e.g., Kammerer et al., 2008, Istance et al., 2008). That is, a fixation longer than a critical duration on a certain object leads to its selection. In favour of dwell time selection, one might assume that for users it is relatively easy to determinate their own point of gaze – although eye movements are usually unconscious. However, this method requires an optimal set of the threshold: dwell times too short will produce lots of unintended selections, whereas those too long require the user to hold the gaze unnaturally long on a certain location, thus slowing performance (e.g., Jacob, 1991). For this reason, for pie menus, we have suggested an alternative method of selecting a slice (Huckauf and Urbina, 2008b): Here, a selection border is used for selection (see Fig. 2). Whenever the gaze crosses the line between the inner pie and its outer frame (i.e., the selection border), the respective slice is selected. That is, selections can be performed via a fixation within the selection frame (behind the selection border) or via any saccade crossing the selection border. Although this method turned out to be effectively usable, its advantages and disadvantages relative to the standard selection by dwell time are unclear.

Providing a detailed comparison between selection by dwell time and by selection border is the aim of the present study.

The main idea of the concept of the selection border is twofold: for novice users, the selection borders do not constrain the fixation durations within the menu. That is, novices can investigate the menu as long as they need to. Experienced users, however, might simply perform slightly longer saccades when navigating through the menu in order to directly select a certain slice. Nevertheless, how effective and efficient this selection method can be realized by the users is unknown, as is the learning rate.

We compared two selection methods (by dwell time and by selection border) with novice users, who repeated the same task five times. The task was chosen from the original work from Kurtenbach and Buxton (1993) in which they evaluated performance for manually controlled pie menus. In the current study, after training, we additionally examined performance in a *marking ahead* trial (i.e., without presenting the pie menus or any visual help on the screen, beside the tasks and the starting point). The purpose of this "blind" trial was to investigate whether users can indeed make advantage of the marking ahead option of pie menus in gaze control. One might assume that a selection method based on fixations (i.e., selection by dwell time) may require longer selection times, but allows for more precision and spatial accuracy by determining the fixation points on marking ahead trials. Or, with other words, whereas the selection border allows for faster selection times, it may be less accurate and thereby less suited for marking ahead.

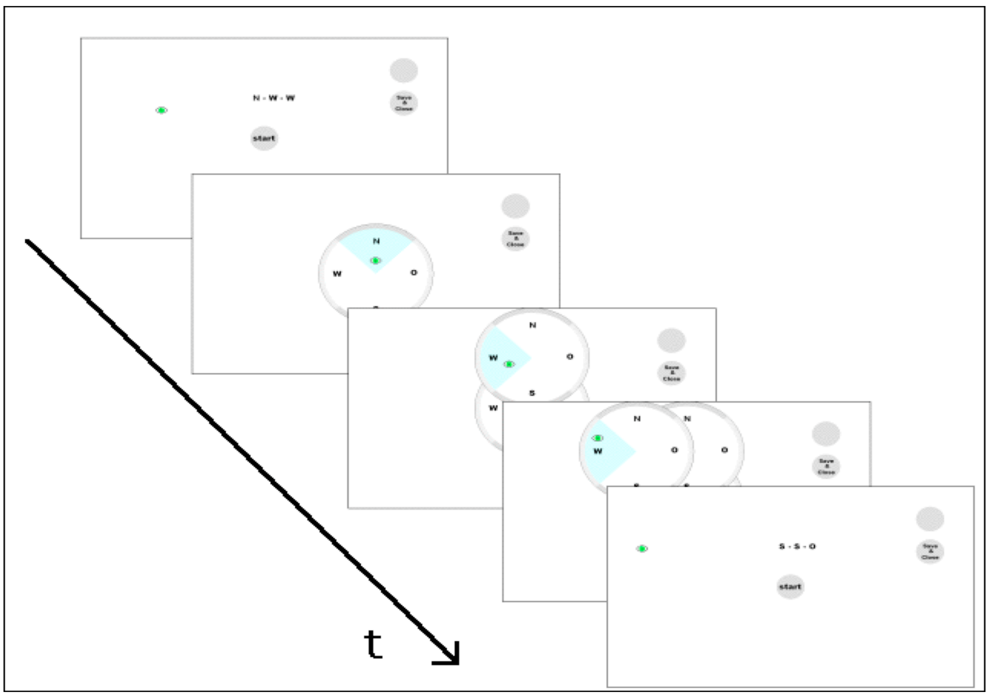


Figure 1. Sequence of events in one task (selecting north – west – west) for selection via selection border. Note that for selection via dwell time, the sequence and visual appearance were the same.

User Study

Methods

Pie menus consisting of four slices were presented in three hierarchical layers. The slices were labelled with four orientations (N - north, O - East ["Ost" in German], S - south, W - west). The task was to select three presented coordinates (e.g., N - W - W, see Fig. 1). As can also be seen in Figure 1, the task-relevant coordinates were displayed on the centre top of the screen. After reading the task coordinates, participants were required to fixate on the central start button (see Fig. 1). Instantly, the pie menu popped up. Each selection was accompanied by a click sound (Majaranta et al. 2004). Immediately, with a selection either the next level pie popped up, or the menus were closed and the start button appeared again.

Pie menus had a radius of 180 pixels (about 7 degrees of visual angle, see Fig. 2) in addition a to an outer frame of 20 pixels, which was used to remark the selection border. Depending on the task, selections were performed either via crossing the selection border (see Fig. 2) or via 400 ms of dwelling on a slice. Five of ten participants started with either condition. For each selection method, five blocks of 32 trials and a sixth marking ahead block were performed. Eye movements were tracked with a head-mounted EyeLinkII (SR-Research).

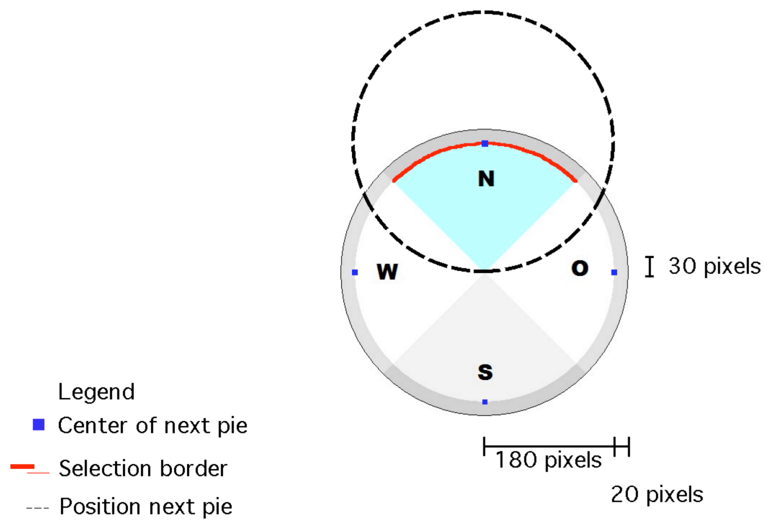


Figure 2. The pie menu implementation in detail. Selection border (in red), position of the upcoming pie (dashed line) and its centre (blue square).

Results

For the first five blocks, task completion time (TCT, measured from onset of the first pie until closure of all pies) was entered in an ANOVA with the repeated measures factors selection method (2) and blocks (5). In mean, selecting via selection border took 569 ms (standard error se= 29 ms), and selection via dwell time took 589 ms (se=14 ms). This difference was not of significance ($F<1$) as was the interaction with blocks ($F(4,36)=1.33$; $p=.29$). Performance differed between blocks ($F(4,36)=7.41$; $p<.01$) which should be ascribed to learning (see Fig. 3).

The corresponding analysis for the error data revealed a significant effect of selection method ($F(1,9)=27.5$, $p<.001$). Selection via selection border resulted in 9.4% of errors (se=2) and selection by dwell time in 21.12% (se=3.45). As in TCTs, blocks differed significantly from each other ($F(4,36)=4.89$; $p<.05$), and there was no interaction between both variables ($F<1$). As depicted in Fig. 4, differences between the first five blocks might be attributed to learning.

Performance in the "marking ahead" block was compared to the fifth block. Of course, TCTs in the marking ahead condition were worse than in the fifth block ($F(1,9)=76$, $p<.001$). Of more interest, neither the selection method nor the interaction with it revealed any significant effect ($F<1$). In the corresponding error analysis, however, not only the block ($F(1,9)=55.8$, $p<.001$), but also the selection method ($F(1,9)=88.5$, $p<.001$) was of significance. Moreover, there was a marginal interaction effect ($F(1,9)=4.4$, $p=.06$) showing that the impediment by presenting no visual information was especially disturbing with dwell time selection.

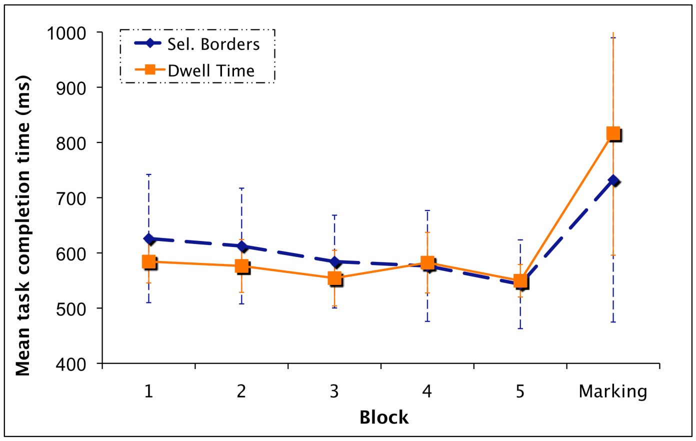


Figure 3. Mean task completion times for each block, separately for the selection methods.

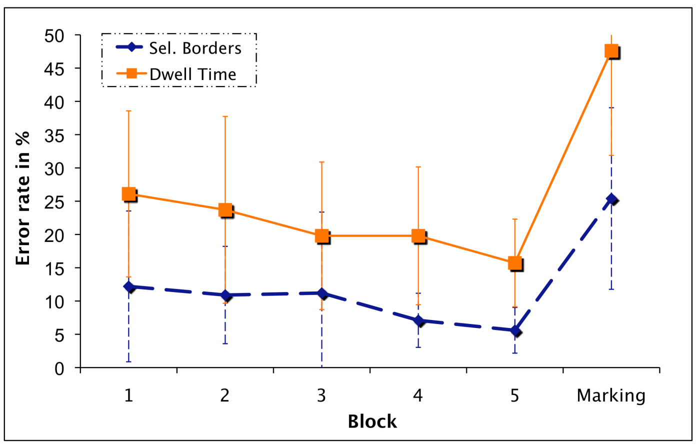


Figure 4. Mean error rates for the blocks, separately for each selection methods.

Discussion

In mean, selecting via selection border turned out to be as fast as dwell time selection after very short training, showing thereby significantly fewer errors. Error rates for selection via dwell time were, even in the fifth block, still larger than those in the first trial achieved with the selection border. The relative high error rate for selection by dwell time suggests that the critical dwell time of 400 ms might have been too short. Extending it, however, would lead to longer TCTs. Hence, selection via selection border seems to be the more promising method.

The current data did not reveal a significant difference between both selection methods in learning rates. However, for longer training durations, and for various tasks, we cannot rule out that both methods differ in their steepness of the learning function. The presented results suggest that selection through selection border can be more benefited from training than selection by dwell time.

The clearest superiority of selection border over dwell time selection was observed in the "marking ahead" block. This was astonishing because during dwell time selection, users had to fixate longer on the pies, which should have facilitated the spatial orientation and navigation after several repetitions. However, one point in favour of selection border is the positioning of the pies: since the upcoming pie is centred always in the selection border (see Fig. 2), the distance between the centre of the pies in different layers had already been trained during the first five blocks. This distance corresponds to the amplitude of the selection movement, which in most of the cases was from the centre of the pie to the middle of the selection border (i.e. the centre of the upcoming menu). It is difficult to estimate this movement in dwell time selection, where the user can select the pie slice anywhere within the slice. This may imply an offset to the next pie's centre, which needs to be corrected in order to perform a gesture. Thus, for the marking ahead performance, and therefore also for expert behaviour, these findings indicate that gestures with constant amplitudes are to be trained in menus, more than certain landing positions. This should be regarded in the design of pie menus. In sum, we can strongly recommend the usage of pie menus, primarily selecting via selection border.

Acknowledgements

This research was partly supported by the "Verein zur Förderung von Studenten der Bauhaus-Universität Weimar e.V.". The authors would like to thank all volunteers who participated in the experiment.

References

Hopkins, D. (1991) The design and implementation of pie menus. In *Dr. Dobb's Journal* 16, 12, pp. 16-26.

Huckauf, A., and Urbina, M. H. (2008a) Gazing with pEYEs: towards a universal input for various applications. *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications. ETRA 2008*, ACM Press, pp. 51-54.

Huckauf, A., and Urbina, M. H. (2008b) On object selection in gaze controlled environments. Journal of Eye Movement Research, 2(4):4, pp. 1-7.

Istance, H., Bates, R., Hyrskykari, A., and Vickers S. (2008) Snap clutch, a moded approach to solving the Midas touch problem. In *Proceedings of the 2008 symposium on Eye tracking research & Applications*, ACM Press, pp. 221-228.

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.

Kammerer, Y., Scheiter, K., and Beinhauer, W. (2008) Looking my way through the menu: the impact of menu design and multimodal input on gaze-based menu selection. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications. ETRA 2008*. ACM Press, pp. 213-220.

Kurtenbach, G., and Buxton, W. (1993) The limits of expert performance using hierarchical marking menus. In *ACM CHI'93*. ACM Press pp. 35-42.

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.

Urbina, M. H., and Huckauf, A. (2007) Dwell time free eye typing approaches. In *Proceedings of the 3rd Conference on Communication by Gaze Interaction - COGAIN 2007*. Leicester, UK, pp. 65-70.

Zhai, S. (2008) On the ease and efficiency of human-computer interfaces. *Proceedings of the 2008 symposium on Eye tracking research & applications ETRA '08*, ACM, pp. 9-10.

Environmental Control Application compliant with Cogain Guidelines

Emiliano Castellina
Politecnico di Torino,
Turin Italy
emiliano.castellina@polito.it

Faisal Razzak
Politecnico di Torino,
Turin Italy
faisal.razzak@polito.it

Fulvio Corno
Politecnico di Torino,
Turin Italy
fulvio.corno@polito.it

Keywords

Environmental Control, Domotics, Eye based interface

Introduction

Domotics Systems, also known as home automation systems, have been in use for several years. They have helped and assisted many people in making their lives better, by making the interaction with household devices and appliances easier and more automated. Current domotic systems suffer from accessibility issues: very few systems provide a control interface accessible through assistive devices, such as switches and eye trackers. In 2007 the Cogain project published the Draft Recommendation for gaze based environmental control (Corno et al., 2007) to overcome this drawback. The Draft Recommendation proposed a software architecture for house gateways and a set of guidelines for developing domotic control interfaces based on gaze interaction. In 2008 several works of e-lite research group of Politecnico di Torino provided an implementation of the software architecture proposed in the draft: the Domotic OSGi Gateway (DOG) (Bonino et al., 2008). This paper presents an ongoing research focused on the design and development of a user interface for **DOG**, based on **EYE** Interaction (DOGEYE) with the purpose of fulfilling the Cogain guidelines requirements and to become a reference implementation of the Recommendations. The application presented in this paper introduces some new gaze interaction techniques which can increase the speed selection and speed up also the error recovering.

User Interface Requirements

Table 1 shows the requirements mentioned in Draft Recommendation for Gaze based environmental control (Corno et al., 2007). These guidelines explain how to develop accessible control applications for home automation system. The primary goal of these guidelines is to promote safety and accessibility. The guidelines are divided in 4 main categories, dealing Control applications safety,Input methods for control application, Control applications operative features and Control applications usability.

The guidelines have two priority levels (PL):

1. Priority Level 1: the guideline has to be implemented by the applications;
2. Priority Level 2: the guideline should be implemented by the applications.

Guideline	Content	PL
1.1	Provide a fast, easy to understand & multi-modal alarm notification	1
1.2	Provide the user only few clear options to handle alarm events	2
1.3	Provide a default safety action to overcome an alarm event	1
1.4	Provide a confirmation request for critical & possibly dangerous operations	1
1.5	Provide a STOP Functionality that interrupts any operation	1
2.1	Provide a connection with the COGAIN-ETU Driver	1
2.2	Support several input methods	2
2.3	Provide re-configurable layouts	2
2.4	Support more input methods at the same time	2
2.5	Manage the loss of input control by providing automated default actions	2
3.1	Respond to environment control events and commands at the right time.	1
3.2	Manage events with different time critical priority	1
3.3	Execute commands with different priority	1
3.4	Provide feedback when automated operations and commands are executing	2
3.5	Manage Scenarios	2
3.6	Communicate the current status of any device and appliances	2
4.1	Provide a clear visualization of what is happening in the house	1
4.2	Provide a graceful and intelligible interface	2
4.3	Provide a visualization of status and location of the house devices	2
4.4	Use colors, icons and text to highlight a change of status.	2
4.5	Provide an easy-to-learn selection method.	2

Table 1. Cogain guidelines summary

Architecture

DOGEYE has been designed according to two main constraints: on one side the application should be able to interact with the DOG platform, on the other side it should be compliant to user interface requirements of the D2.5 guidelines. Figure 1 shows an overview of the DOGEYE architecture where the application communicates with the domotic environment through DOG (XML-RPC protocol, right) and with the ETU-driver layer (COM object integration, left), defined in (Bates & Spakov, 2006), to control Eye-tracking equipment [Guideline 2.1].

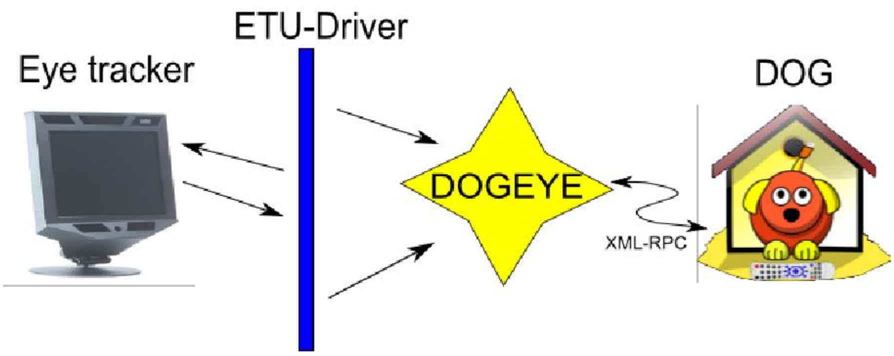


Figure 1. DOGEYE architecture

DOGEYE Interface

The DOGEYE interface is divided into 4 frames (Figure 2):

Main Window : shows the list of house rooms [Guideline 4.1]; when a room is selected it displays its devices with their current statuses [Guideline 4.3, 3.6].

Alarm Window : shows the messages that require most user attention. In case of an emergency, it disables all other windows and provides the user with only few options to choose [Guideline 1.1, 1.2, 1.3].

Status Window : displays notification messages about less important ongoing activities in the house (e.g., scenario completion or usually activated commands) [Guideline 3.4].

General Help Window : provides the user with general context information and properties about the currently highlighted object.

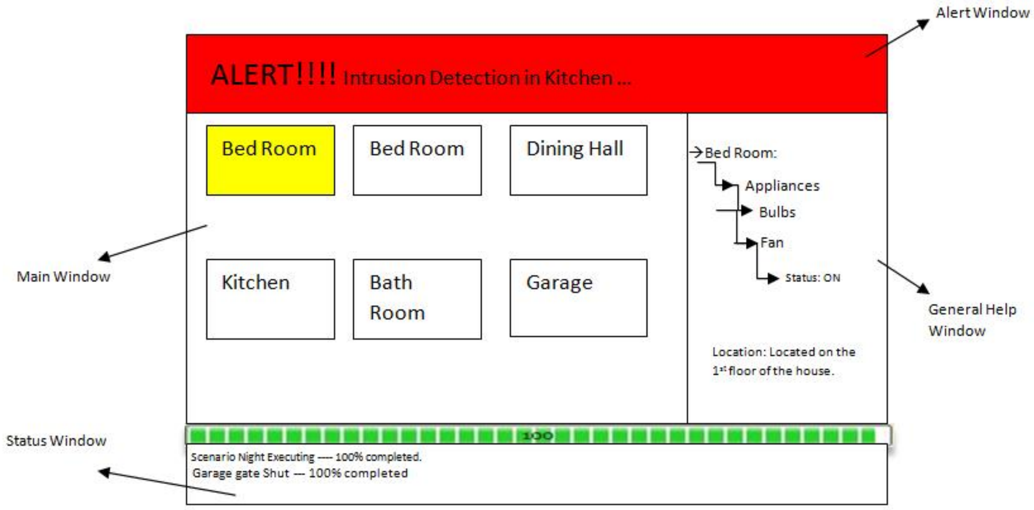


Figure 2. DOGEYE interface

DOGEYE implements two novel methods which may improve the performance and enhance user friendliness on gaze control interaction. These methods are described in the following.

Reverting Back Operation (RB)

The Reverting back Operation allows a faster return to the previous state, if a wrong selection is made by the user [Guideline 4.5]. The typical interaction pattern in dwell time based applications requires at least two selections to recover from a wrong selection: first, the user selects the wrong item, then for reverting the operation he/she has to select the **back** action in the next screen. Therefore selection errors incur in a time penalty equal to twice the dwell time.

To avoid this issue, the DOGEYE interface defines a screen area with a smaller dwell time where the user can gaze immediately to come back to the previous window. Such area is displayed only for a little time after each user selection and then fades away.

Strategy of Separating Selection from Action

In accordance with COGAIN recommendations, DOGEYE aims to provide the user with a graceful and intelligible interface. Interface design techniques include two main interaction patterns, i.e., Command Based interaction pattern and non-Command Based Interaction pattern. The Command-based interaction pattern is one in which user explicitly directs the computer to perform some operation.

The Non-command based interaction pattern is one in which the computer observes and interprets user actions instead of waiting for user commands. Interactions become more natural and easier to use (Tanriverdi & Jacob, 2000). DOGEYE implements a strategy for separating Selection from Action. Before defining the proposed strategy, it is necessary to define some terms:

Object is anything that can be selected and that may have associated options, either properties, commands or notifications;

Selection is the process of choosing such objects and displaying its associated options to the user;

Actions are are operations that perform some task like opening a particular window, acquiring status of a particular object.

The General Help window associated with DOGEYE interface displays the information about any item on the screen immediately as it is gazed by the user, thus providing a non-command based interaction. The Main window provides command based interaction, where a particular action is executed after a certain dwell time. This strategy combines both command based interaction pattern and non-command based interaction pattern [Guideline 4.2].

Results and Conclusions

This research is the first step towards fulfilling the interface design guidelines to give a fully compliant gaze driven domotic system for users and to become a reference implementation of the COGAIN recommendations. It fulfills most of the requirement mentioned as Priority level 1 in design guidelines and also implements some Priority level 2 guidelines. It also focuses on developing new techniques that will enable users to use the interface with comfort and enhance the performance of the overall system. Environmental Control Interface based on gaze is a new area and work must be done for new techniques that focus on user comfort along with increasing performance. In the future we intend to evaluate the DOGEYE with disabled people and also suggest more interface design patterns.

References

Bates, R., & Spakov, O. (2006). *D2.3 implementation of Cogain gaze tracking standards* (Tech. Rep.). COGAIN.

Bonino, D., Castellina, E., & Corno, F. (2008, November). The DOG gateway: enabling ontology-based intelligent domotic environments. *Consumer Electronics, IEEE Transactions on*, *54*(4), 1656-1664.

Corno, F., Castellina, E., Bates, R., Majaranta, P., Istance, H., & Donegan, M. (2007). *D2.5 Draft Standards for Gaze based Environmental Control.* (Tech. Rep.). COGAIN.

Tanriverdi, V., & Jacob, R. J. K. (2000). Interacting with eye movements in virtual environments. In *Chi '00: Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 265–272). New York, NY, USA: ACM.

Home and environment control

Petr Novák
Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University
novakpe@labe.felk.cvut.cz

Olga Štěpánková
Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University
step@labe.felk.cvut.cz

Keywords
Home control, environment control, alternative input, appliances control, universal protocol / format

Introduction

The means for environmental control are extensively studied, now [Aiello, Dustdar, 2008]. There are many professional smart systems for home and environment control. But often, they have severe limitations from the point of the target user: first, they are far too expensive and complex, second only selected appliances can be connected into such systems. Moreover, frequently their input buttons and display panels are not suitable for users with limited abilities who are entirely dependent on special communication means.

There is no easy way how to connect for example common old TV set or radio into such a complex system and control these devices. Most people would like to use environment control for appliances they already have at home, e.g. their common old TV set or radio, the computer they use for internet browsing or mailing. Our aim is to create a flexible and open solution for this problem. The principal idea is to complement common low-cost personal computer with an interface that will mediate communication between simple input devices (connected into the computer through standard input for example USB, RS232, LAN, BlueTooth) and selected household appliances to be controlled. For that purpose we have designed suitable and simple communication protocols for intended input devices (eye tracker, finger switches, movement of arm or head, …) and output devices (TV set, radio receiver, electrical plug). For eye trackers can be used for example ETU Drive providing uniform type of output for various eye-tracking systems, it represents its generalization. If the ETU Driver [Špakov] is used as the input channel the rest of the system becomes independent of the type of the connected input machinery. The input device can be a direct component of the control systém or it can be connected to it by nay standard means (USB or LAN). A remote periphery (e.g. an eye-tracker on a wheelchair [Novak et al.]) can serve as an input device, too. In this case, the connection is ensured using BlueTooth or ZigBee. While the Home Control System informs the input device about the current state and displays possible current choices, the input device replies by indicating the choice (in the form of a series of commands). The universal interface removes the difference among the ways the input is connected (USB, LAN or BlueTooth). Moreover, all the input devices are interchangable from the point of the Home Control System.

Input devices / channels

All over the input device can apply the most sophisticated technology, they have a simple function in our system: they are expected to send commands into the control system, only. The considered input devices range from the simplest ones (like finger switches) over more complicated ones like devices for detection of arm or head movement up to those most complex and expensive ones (e.g. precise eye trackers indicating the place the user observes), for examples see e.g. [Chapman and McCartney], [Webb] or the web page of Ability

Technology (http://www.abilitycorp.com.au/). Our goal is to suggest a unified protocol for all these types of input devices so that they can become mutually interchangeable. Thanks to this protocol the input devices can be simply connected into the control system using number of ways (USB, LAN, BlueTooth, ZigBee, …).

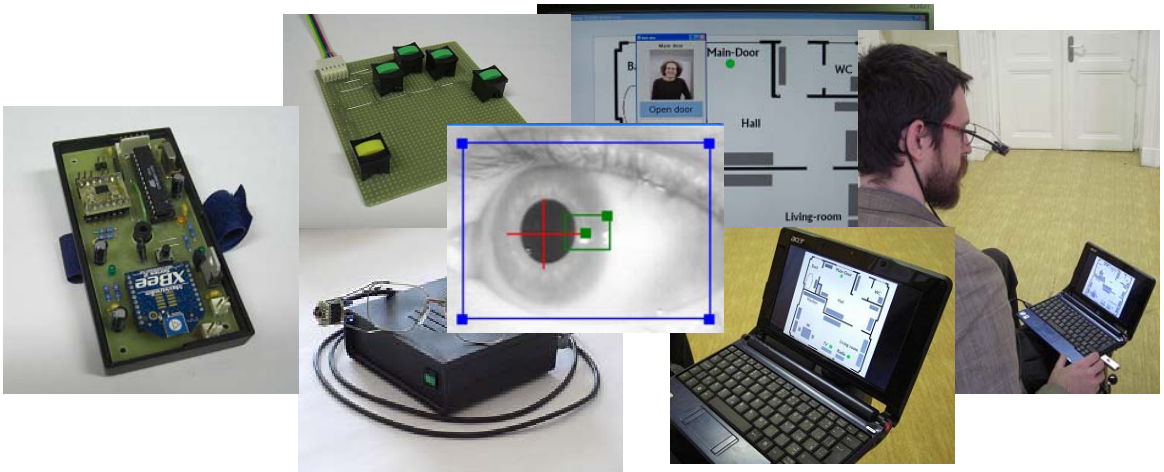


Figure 1. Input devices; arm movement, finger switches, low-cost eye-tracker, computer screen, input device on wheelchair.

Within our created system, we distinguish three categories of input devices First category (e.g. precise eye-tracker) provides input in absolute coordinates mainly in direction X and Y and allows to confirm the selected action. User can easily and quickly select requested icon or action on the computer screen. Second category (e.g. head or arm movement) provides only directions like left, right, up, down and confirm elected action. For the user these input devices are not as fast as the first type but they are sufficient for certain tasks. The last category (e.g. simple buttons) provides nothing but just a few switches (in some cases no more than one). The user must use only limited possibility to select requested icon or action and this can take quite a long time. In our protocol, each input device must send information to the control system that will identify its type (its possibility) so that the control system in the case of necessity can autonomously complement some signals necessary for control of some output devices (from direct selecting icon to only scanning icon after icon).

The Central Unit

We expect that common low-cost personal computer (PC) or home multimedia centre can take the role of the central unit as our control SW is neither huge nor resource demanding. Input and controlled devices are connected into the PC using many types of connections:

- USB, RS232 – near-by devices like IR control for TV set and radio, …
- BlueTooth, ZigBee – medium-distance devices like actuators for windows, …
- LAN, WI-FI – remote devices like those close to the home entrance (camera, door lock, …).

Central unit or another screen can show the plan of a room or flat with many icons that represent particular devices. These icons inform not only about actual state of devices but invoking the particular icon can start a dialog for control of the underlying device. Selecting or activating particular icon is based on used input device and number of input commands.

From the view of the central unit and input or output device there is no difference concerning their interconnection. Each device has its own driver within the control SW that is composed from three main levels. The lowest level, named low-level driver, ensures real communication for example throw USB or

LAN. Creating this low-level driver there is not difference between using for example USB or LAN. This part is almost the same for all devices. Next level, named high-level driver, provides some degree of autonomous behaviour. This part sends all requirements from the user to the real device, updates GUI interface according to the actual state from the device and it can even tell other devices' high-level drivers to perform some consequential or necessary actions. Such feature represents certain degree of autonomous behaviour or cleverness. The last level is a dialog for controlling particular device. Template for this dialog provides the control system. Each (namely output) device has only its own configuration for the control dialog. Unifying of control dialogs is a significant advantage. All control dialogs have the same type of buttons and the same structure of layout that is transparent to the user. It is also possible to rearrange graphical appearance of control dialogs by our cushy editor to be more suitable for particular user.

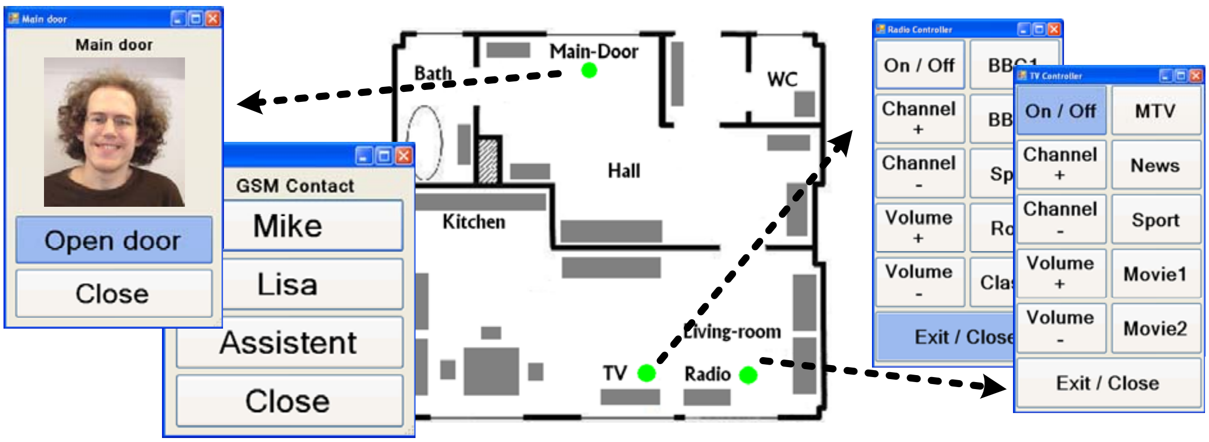


Figure 3. Main screen with examples of control dialogs.

Output devices / actuators

The control system behaves as if there is just one input device but many output devices. In contrast to input device the output devices are much simpler and cheaper [Chapman and McCartney], [Webb]. Up to now we have created output devices for controlling TV set or radio (replacing common IR remote control), electrical plugs, measure of temperature and so on. Some special kinds of devices are universal sensor / actuator to provide for example input for doorbell or control door lock. Thanks to the unified protocol the output devices can be connected into the control system using USB, LAN, BlueTooth, …. There are some more special devices like GSM modem for SMS or call on pre-defined numbers in the case of emergency.

Figure 3. Ouput devices (USB remote IR TV control, RS232 GSM modem, LAN entrance door system).

Future work

The described system was partially created and tested but much of the work remains to be done. Now, it is possible to control quite complex appliances like TV set or radio with many functions. We want to extend our system to be able to control not only devices that are ever more complicated but also the simple devices that people use in every day life, e.g. lift / elevator (call in, send to). For such reason we use low-power wireless technology named ZigBee to create wireless connection between the target device, for example lift / elevator and wheelchair. The unified protocol can be used independently of the communication channel. Next important thing is to create some level of self-localization within room, flat or house for the mobile user. Already mentioned ZigBee technology itself can detect its position / localization with accuracy of less then 1m among other devices using this technology. Based on such a feature, it is possible to show for example on user's display panel on wheelchair only those devices that are in appropriate range. This behaviour simplifies the intended implementation and it seems to be more suitable for severely challenged user. The user will not have to go through many menus to select the requested device on the screen but the control system itself will provide offers only for device(s) in appropriate range.

Conclusion

Described system was designed to be consisting of common and low-cost parts. Further extension with devices that are more sophisticated or connection into already existing smart home control system is also possible. The described system is not limited for use with gaze controlled input devices. On the contrary, it was tested with several alternative input devices with limited number of input commands. Our intention was to provide well-transparent display that provides not only information about the actual status of environment but also the means for easy control of this environment. We have given special attention to communication among devices to enable control of the user's surroundings not only from static position like chair or bed but also from the dynamically changing position than can be reached e.g. by a wheelchair. The presented research and development has been partially supported by the EU grant IST-2003-511598 COGAIN (Communication by Gaze Interaction) and the Czech grant No. MSM6840770012 "Transdisciplinary Research in Biomedical Engineering II".

References

Ability Technology (*www.abilitycorp.com.au*): Environmental Control System for People with Spinal Injuries. (September 1999), Technical report

Aiello, M., Dustdar, S.: Are our homes ready for services? A domotic infrastructure based on the Web service stack, Pervasive and A domotic infrastructure based on the Web service stack, Pervasive and Mobile Computing, v.4 n.4, p.506-525, August, 2008

Chapman, K., McCartney, K. (September 2001) Smart home for disabled people. Cutting Edge Conference, issue number, University of Portsmouth UK, Oxford.

Novák, P., Krajník, T., Přeučil, L., Fejtová, M., and Štěpánková, O. (2008) AI support for Gaze Controlled Wheelchair. Proceedings of COGAIN 2008,- Communication, Environment and Mobility Control by Gaze http://www.cogain.org/cogain2008/COGAIN2008-Proceedings.pdf , pp. 19-22.

Špakov, O., ETU driver - Eye-Tracking Universal Driver http://www.cs.uta.fi/~oleg/etud.html

Webb, S., D.(April 15, 2005) Environment control and computer interface for people with severe physical disabilities. *Technical report*, University of Technology Curtin. http://www.computing.edu.au/~webbsd/

A Gaze-Contingent, Acuity-Adjusted Mouse Cursor

Michael Dorr
University of Lübeck
Ratzeburger Allee 160,
D-23538 Lübeck,
Germany
dorr@inb.uni-luebeck.de

Christoph Rasche
Justus-Liebig-Universität
Giessen
Otto-Behaghel-Str. 10F,
D-35394 Giessen,
Germany
rasche15@gmail.com

Erhardt Barth
University of Lübeck
Ratzeburger Allee 160,
D-23538 Lübeck,
Germany
barth@inb.uni-luebeck.de

Keywords

gaze-based input, gaze-contingent rendering, human-computer interaction

Introduction

Gaze-based computer interfaces so far have often focused on using gaze as the sole input channel, for instance for eye-typing or for moving the mouse cursor. While such display control can be essential for disabled persons, it is generally rather awkward to use because gaze cannot be placed precisely on objects; furthermore, it cannot be used quickly enough for sequential input because of the *Midas touch* problem that both orienting and acting are transmitted through the same modality (Jacob, 1993). One solution to this problem can be not to rely on gaze exclusively, but to only use it as an assistive measure, for example to place the cursor at the gaze position only if mouse activity is registered (Zhai, Morimoto, & Ihde, 1999; Drewes & Schmidt, 2006). In the following, we present another such gaze-assistive approach; we address the problem that the position of a small mouse pointer is often forgotten during browsing, so that the pointer has to be relocated by a visual search before it can be used again. A common shortcut is that users wiggle the cursor briefly, thereby making it more conspicuous. To achieve a similar effect without the need of a manual intervention by the user, we render the mouse cursor in a gaze-contingent fashion and increase its size as a function of the distance from the centre of gaze (cf. Fig. 1). Because this size modification roughly follows the cortical magnification factor, the pointer has optimal visibility even in the user's visual periphery while keeping a normal-sized appearance when looked at directly. In some sense, this is the opposite of many previous gaze-contingent systems that made use of the reduced sensitivity of the visual periphery to leave out details, e.g. for compression or reduced rendering complexity (Perry & Geisler, 2002; Böhme, Dorr, Martinetz, & Barth, 2006).

Implementation

Many eye-tracking studies are tested on platform modifications not readily accessible to other users. We therefore decided to use open-source software – despite its much smaller share in the mass market – which lends itself naturally to niche and research domains because it can be flexibly modified and distributed to other researchers and users, and opted for the Linux operating system. We were then faced with the choice whether to write our own prototype software application, to adapt an existing application, or to make the gaze-contigent cursor be used system-wide. The first two options are appealing because they certainly require the least effort to implement; as a drawback, they are of fairly little use outside the scope of this research project. Unfortunately, the last option would involve writing a complete proxy server for the X window system (Schneider, 1997), a somewhat daunting task. We therefore decided to settle on a middle ground and adapt the *Qt* cross-platform application and user interface framework (Trolltech, 2009). The advantage here is that with only minor modifications to a few well-encapsulated C++ classes in the *Qt* library, any application that is based on *Qt* (including the well-known KDE desktop system) can make use of the gaze-contingent cursor; by linking either against the modified or the original library, it is also possible to switch behaviour on a per-application level.

One of the central application of today's desktop computers is the web browser. Unfortunately, both the KDE default browser *konqueror* and the *WebKit*-based browser that comes shipped as a demo with the *Qt* library implement their own cursor handling on top of *Qt*. Therefore, we modified the latter application as well.

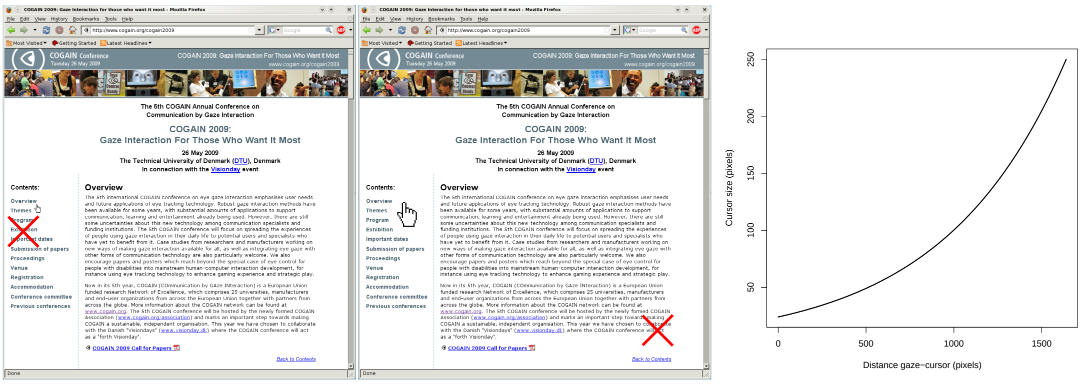


Figure 1. Schematic illustration of the gaze-contingent mouse cursor. Left: The subject is scanning the menu column to the left (gaze indicated by the red cross); the mouse cursor is small because it is still visible (para-)foveally. Middle: When the subject has read the text body and the distance gaze to cursor is large, the mouse cursor is enlarged to ensure visibility even in the visual periphery. Right: Cursor size as a function of distance between gaze and cursor.

Experimental validation

Four subjects, department members and undergraduate students, participated in a small usability study; all were naive regarding its purpose. The goal was to engage the subject in a task that required frequent scrolling within the windows and switching between windows in order to exploit the use of the gaze-contingent cursor as much as possible, given the limited duration of a typical evaluation (about 10 minutes). Subjects were asked to perform some typical web browsing activities such as looking up flight information, comparing prices for consumer electronics, etc. The web browser was placed on the right half of the screen and results of the tasks had to be entered into an editor window that covered the left half of the screen. During the experiment, eye movements were registered using an SMI iView X RED remote eye tracker running at 50 Hz. The size of the mouse cursor varied from 24 (size_{min}) to 250 (size_{max}) pixels according to the formula

$$\text{size}_{\text{cursor}} = \text{size}_{\text{min}} \cdot \left(\frac{\text{size}_{\text{max}}}{\text{size}_{\text{min}}}\right)^{d/d_{\text{max}}},$$

with d the current distance between gaze and cursor and d_{max} the maximum possible distance on the screen, i.e. the length of the screen diagonal (at a resolution of 1280 by 1024 pixels).

At the end of the test, subjects were asked to evaluate the use of the gaze-contingent cursor. They were asked to rate the gaze-contingent cursor on a scale from -5 (highly annoying) over 0 (neutral) to +5 (highly useful).

Results

None of the subjects rated the gaze-contingent cursor as annoying. In fact, one of the subjects only became aware of the nature of the cursor when she read the instructions to evaluate the system after the experiment session (this subject therefore did not assign a numerical rating). However, this may be explained by the fact that the tasks placed a high cognitive load on this subject, who had only very recently acquired the German language (all the web sites to be visited were in German). The remaining three subjects rated the cursor positively (+1, +1, and +4, respectively); two of them commented they "did not have to look for the cursor anymore". Nevertheless, two subjects also reported that the cursor sometimes seemed to move by itself (presumably when jitter either in the eye or the gaze measurement induced a slight "pumping" effect, i.e. size variations), indicating the need for a more elaborate gaze data filtering.

Conclusion

We have presented a gaze-contingent mouse cursor whose size changes as a function of visual eccentricity to remain visible to the user even in the visual periphery. A small usability study showed that users were not distracted by the variable size, but rather rated the gaze-contingent cursor to be a positive addition to the user interface. A more objective performance evaluation will have to be carried out in the future to test whether this subjective preference also corresponds to a higher input efficiency. The presented idea could also be extended to editor cursors.

In order to implement the gaze-contingent mouse cursor, we have modified the publicly available *Qt* library; this opens up the possibility to use our cursor from a vast variety of different applications without the need to recompile them. The source code of our modifications is available upon request.

Acknowledgements

Our research has received funding from the European Commission within the project GazeCom (contract no. IST-C-033816) 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. We acknowledge Henry Schütze's assistance with programming and data collection.

References

Böhme, M., Dorr, M., Martinetz, T., & Barth, E. (2006). Gaze-contingent temporal filtering of video. In *Proceedings of Eye Tracking Research & Applications (ETRA)* (pp. 109–115).

Drewes, H., & Schmidt, A. (2006). Mauszeigerpositionierung mit dem Auge. In *Proceedings of Mensch & Computer 2006* (p. 185-94).

Jacob, R. J. K. (1993). Eye movement-based human-computer interaction techniques: Toward non-command interfaces. *Advances in Human-Computer Interaction*, *4*, 151-80.

Perry, J. S., & Geisler, W. S. (2002). Gaze-contingent real-time simulation of arbitrary visual fields. In B. E. Rogowitz & T. N. Pappas (Eds.), *Human Vision and Electronic Imaging: Proceedings of SPIE, San Jose, CA* (Vol. 4662, p. 57-69).

Schneider, J. (1997). *X11-big-cursor minihowto.* "http://i11www.ira.uka.de/~schneid/X11-big-cursor/master/".

Trolltech. (2009). "http:/www.qtsoftware.com".

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit* (p. 246-53).

Optimizing the interoperability between a VOG and a EMG system

Mikel Ariz[1], Javier Navallas[1], Arantxa Villanueva[1], Javier San Agustin[2], Rafael Cabeza[1] and Martin Tall[2]
(1) Electrical and Electronic Engineering Department. Public University of Navarra.
Campus Arrosadia sn. 31006 Pamplona (SPAIN)
(2) Inc Group. IT-University of Copenhagen
Rued Langgaards Vej 7 2300.Copenhagen (DENMARK)

Keywords
Multimodal Interface, EMG, VOG, Midas Touch, gaze interaction, eye tracking

Introduction

Gaze tracking systems have long been used to support communication for people with severe motor disabilities. Although big efforts have been put into developing more robust gaze trackers, reliable selection by gaze remains an unsolved issue. Our eyes represent a natural substitute for the mouse, but selecting objects only by gaze is a more delicate task. Almost 20 years ago, Jacob (1991) described what is known as the Midas Touch problem: every object we look at would be selected.

In order to reduce or eliminate the Midas Touch problem, multimodal input can be considered. Gaze interaction can be complemented by alternative input techniques, such as brain computer interfaces or electromyographic switches. Many of these technologies have demonstrated their feasibility by themselves but little work has been focused on combining different technologies and studying its possibilities. Remarkable works can be found in (Mateo et al., 2008) (Popescu et al., 2006) (Surakka et al., 2004) (Hansen & Junker, 2006).

Most previous work on the combination of gaze pointing and EMG clicking has been directed towards showing a speed advantage of gaze-EMG over the mouse in an ideal desktop scenario. However, the use of an EMG system in a noisy environment, such as walking or using a wheelchair, might introduce a high number of undesired muscle activations, and therefore a high number of clicks that might deteriorate the interaction.

Our work focuses on increasing robustness against noise by building a tightly integrated system that combines gaze pointing and EMG clicking. This is achieved by requiring a fixation on the gaze tracker before issuing a click event when a muscle activation is detected. Both systems, i.e. the gaze tracker and the EMG switch, have been developed by our group and are presented independently in the following sections. A description of the multimodal interface by combining both technologies is provided next, followed by an explanation of the experimental setups and the results obtained. Finally, the conclusions are presented.

Video Oculographic System

The eye-tracking system implemented in our lab is based on a single camera and two infrared light sources. The image processing module detects the dark pupil center and the two glints positions. A user calibration using a 4x4 grid of points is carried out and accuracies under 0.5º are achieved after calibration in a moderate head movement scenario. The gaze-tracking system is shown in Figure 1.

Figure 1. Picture of the gaze-tracking system

Electromyographic System

Electromyography is an important topic not only from the point of view of the work presented in this paper, but also in research fields that study muscle behavior and activity modeling. Various methods have been explored to detect muscle activity during the last years (Staude et al., 2001). The techniques to detect muscle activity can be divided into two groups: surface EMG, in which one or more electrodes are placed on the user's skin, and needle EMG, which requires inserting a needle electrode into the muscle tissue. Surface EMG has been chosen for its low invasiveness and good performance. In our system, the surface electrodes are attached to the skin over the muscle area, the forehead is many times employed since the muscle signal is faster compared to other muscles, such as those of the hand. Our system uses disposable, self-adhesive pre-gelled electrodes. Three electrodes should be used, two of them attached to the skin around the muscle zone and a third one working as ground. In the system developed in our lab, the forehead is selected to place the activity detection electrodes. The third one, working as ground, is placed in the wrist. Figure 2 shows a user wearing the electrodes. The hardware of the system is completed by bio-signal amplifier.

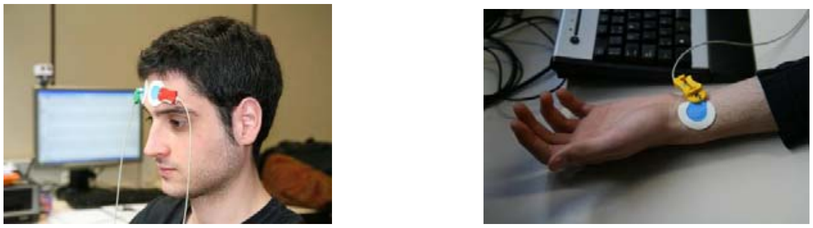


Figure 2. User wearing two electrodes on the forehead and one electrode on the wrist

In order to detect muscle activation, some techniques analyze the signal in frequency domain while others limit their analysis to temporal domain. We have focused our attention on temporal analysis since the computational complexity is reduced. Most detection algorithms consist of up to three basic processing stages [3], namely signal conditioning, detection unit, and post-processing. Different algorithms were tested in a previous analysis in order to select the most appropriate one: Hodges and Bui, Bonato et al., Lidierth, Abbink et al., AGLR Step and AGLR Ramp (Staude et al., 2001). In general, all systems establish a threshold value that should be exceeded by a function of the input samples to determine that activation has occurred. After a careful study, AGLR Step was selected as the algorithm with best balancing between false detection rate and simplicity.

Multimodal Interface VOG+EMG

Once both systems were implemented and tested, they were combined in a single interface. The eye tracking system was used to move the cursor in the screen while muscle activation was employed as click signal by sending a mouse click event to the operating system.

Two versions of the system were implemented. The first one was named as system with no communication (SWNC). In this system, EMG and VOG work independently. Each time a muscle activation is detected, a click is sent to the system, regardless of the VOG status. In the second modality, a communication protocol is established between both systems, i.e. system with communication (SWC). Thus, the EMG receives information from the VOG system about gaze behavior. More specifically, the VOG sends information to the EMG system about the fixation status of the eye. Our hypothesis is that voluntary activations will be produced only in maintained fixation situations. In other words, muscle activations in cases in which the eye is not fixating (making saccadic movements) can be rejected. Once an activation is detected, the EMG system will check eye fixation status before sending a click event to the OS. The EMG will send a click event to the OS if a fixation status is maintained during a specified amount of time. This time is defined as the activation time (see Figure 3). Activations in no fixation situations can happen due to involuntary user's actions (spasmodic activation) or due to system noise. In both cases, these undesirable activations can be modeled by introducing noise to the system. The experiments carried out to compare both systems, i.e. SWC and SWNC, are described in the next section.

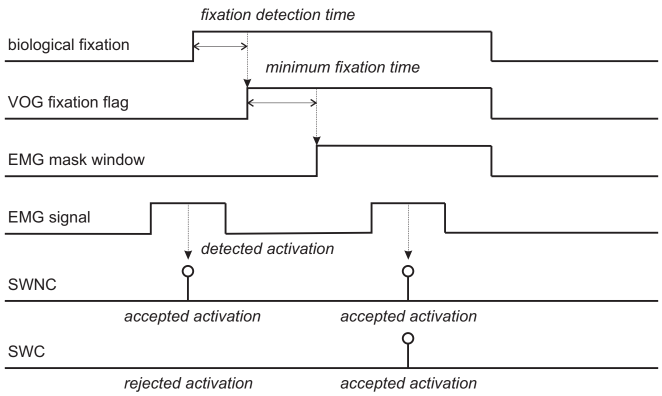


Figure 3. Sequence diagram comparing the SWC and SWNC systems. In the SWC system activations will be sent if a fixation is maintained during the activation time.

Method

Participants and Apparatus

A total of 10 volunteers, ranging from 24 to 41 years old, participated in the study. They were randomly divided into 2 groups, one of which tested the system with communication, while the other one tested the system without communication. 2 users had previous experience with gaze tracking and EMG activation; each of them was assigned to a different group. A 17" monitor with a resolution of 1280 x 1024 was used to present the target-acquisition task. The gaze tracker and EMG system presented above were used. Figure 4 shows the experimental setup with one of the participants during the experiment.

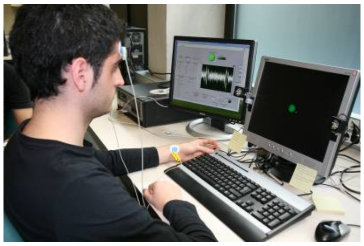


Figure 4. Experimental setup.

Design and Procedure

The experiment was conducted using a 2 x 3 x 2 between-subjects design. Factors were *communication* (no communication / communication), *noise level* (no noise, low noise or high noise), and *activation time* (200 ms or 400 ms).

The experiment consisted on a target-acquisition task as specified in the ISO 9241-Part 9 standard, following a similar procedure to Zhang and MacKenzie (2007). 16 targets were arranged in a circular layout, and participants had to point at each target using their eyes and select it by performing a frowning or tightening their jaw. The participants were instructed to select the targets as fast and as accurate as possible. The size of the targets was fixed to 150 pixels in diameter. Prior to starting the experiment, participants calibrated both systems and ran a warm-up trial to become acquainted with the multimodal interface. Each participant completed a block of 64 trials for each combination of noise level and activation time either with or without communication between gaze tracker and EMG. The order of the 6 blocks was counterbalanced to neutralize learning effects. In a normal situation, a user might alternate two types of tasks: a visual task such as browsing or reading, and a selection task to activate a menu item or a link. During the visual task no clicks should be performed, since they could lead to an undesired activation. In order to simulate the situation where the user is not fixating and does not want to issue an activation, we added a moving object between targets that appeared for a random time of between 2 and 4 seconds. Once the time elapsed, this object disappeared and a new target that the user had to select appeared. Participants were instructed not to perform any activation while the moving object was on the screen.

A noisy environment in which undesired muscle activations occur was simulated by introducing random activations to the system. A homogeneous Poisson process was chosen to model the noisy activations. In the low noise (LN) condition, the average time between noisy activations (the inverse of the intensity of the Poisson process) was 8 seconds, while in the high noise (HN) condition it was 4 seconds. In each trial we measured completion time (i.e. time required to select the target since the moment it appears), unsuccessful activations (i.e. targets selected when the cursor was outside the target), involuntary selections (i.e. targets selected by noise and not by an EMG activation), and clicks produced by noise.

Results

Results were analyzed using four 2 x 3 x 2 mixed-design ANOVAs, with *communication* being a between-subjects factor, and *noise level* and *activation time* being within-subjects factor. Completion time, error rate, involuntary selections and clicks by noise were analyzed as the dependent variables. All data were included.

The completion time was compared for the group using the SWC and different activation times. The statistical test (Wilcoxon Rank Test) concluded that activation times of 400 ms did not result in significantly larger completion times ($p=0.11$) comparing to 200 ms.

The type of communication between gaze tracker and EMG had a significant effect on completion time, $F(1, 8) = 33.145$, $p < 0.05$, but noise did not, $F(2, 16) = 1.009$, $p > 0.05$. Completion time was higher when communication between systems existed. In the case of error rate, communication had a significant effect, $F(1, 8) = 11.419$, $p < 0.05$, but noise level did not $F(2, 16) = 1.489$, $p > 0.05$. Error rate was lower in the case of communication. Figure 5 shows average completion times and error rates for each communication and noise level combination.

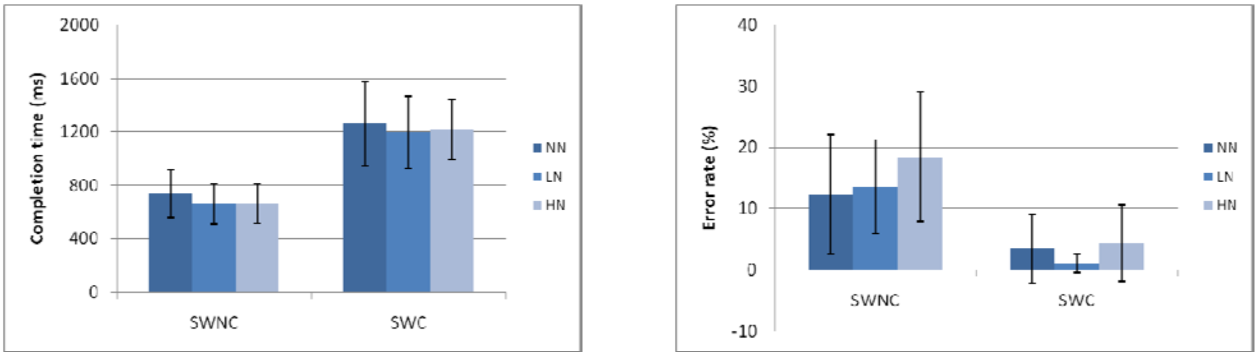


Figure 5. Mean completion time and error rate for each communication and noise level combination. Error bars show the standard deviation.

Involuntary selections was affected by both communication ($F(1, 8) = 25.36$, $p < 0.05$) and noise level ($F(2, 16) = 55.26$, $p < 0.05$). The number of involuntary selections was lower when there was communication between gaze tracker and EMG, and for lower noise levels. Similarly, the number of click events was lower when there was communication ($F(1, 8) = 412.12$, $p < 0.05$), and for lower noise levels ($F(2, 16) = 233.51$, $p < 0.05$). Figure 6 shows average involuntary selections and clicks due to noise per target for each communication and noise level combination.

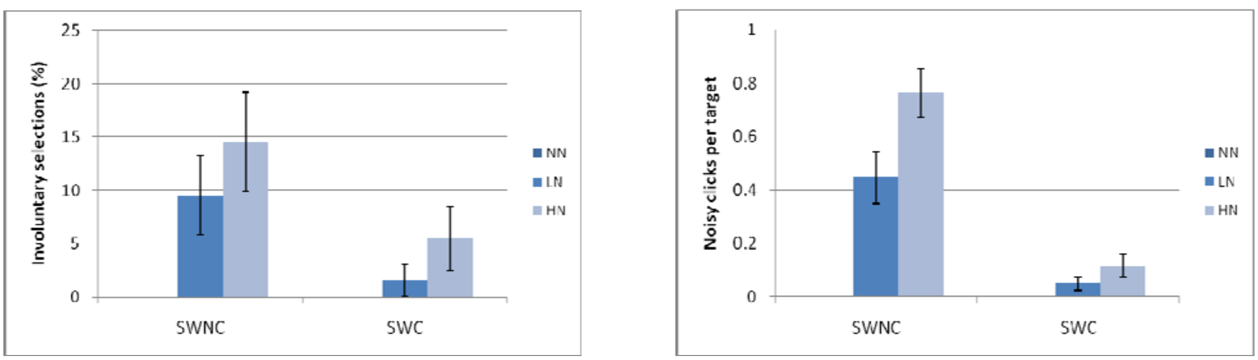


Figure 6. Mean involuntary selections and noisy clicks per target for each communication and noise level combination. Error bars show the standard deviation.

Conclusions

A multimodal user interface that combines gaze pointing and EMG clicking has been designed and implemented. Two versions of the interface have been evaluated. The first one considers VOG and EMG as independent systems while the second one permits the EMG to know the fixation status of the user according to the information provided by the VOG system. In this manner, muscle activations that occur when there is no fixation are rejected. The requirement that for every muscle activation there must be a fixation in order to issue a click event increases the time required to select the targets, since in some cases the first muscle activation performed by the user will not occur during a fixation, and therefore the user will need to activate the muscle again. However, the benefit of the tight VOG-EMG integration comes in the presence of noise.

The error rate decreases from 14.8 to 2.9% when comunication is introduced. There is also a reduction in the number of clicks due to noise per target. Since a fixation is required, most noisy activations that occur when the user is performing a visual task such as reading or browsing will not become a click. The results obtained in this study suggest that a close integration between gaze tracker and EMG switch might be beneficial in noisy environments where false activations are likely to occur. This can be the case when the user is driving a wheelchair or suffers from non-controlled movements.

References

Abbink, J., Van Der Bilt, A., & Van Der Glas, H. (1998). Detection of onset and termination of muscle activity in surface electromyograms. *J. Oral Rehab.*, 25, 365–369.

Bonato, P., D'Alessio, T., & Knaflitz, M. (1998). A statistical method for the measurement of muscle activation intervals from surface myoelectric signal during gait. *IEEE Transactions on biomedical engineering*, 45 (3), 287–299.

Hansen, J. P., & Junker, A. M. (2006). Gaze pointing and facial EMG clicking. In H. Istance & L. Farinetti (Eds.), *Cogain 2006. Gazing into the future*.

Hodges, P., & Bui, B. (1996). A comparison of computer-based methods for the determination of onset of muscle contraction using electromyography. *Electroencephalography and Clinical Neurophysiology/ Electromyography and Motor Control*, 101 (6), 511–519.

Jacob, R.J.K. (1991). The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get, *ACM Transactions on Information Systems*, 9152-169.

Lidierth, M. (1986). A computer based method for automated measurement of the periods of muscular activity from an EMG and its application to locomotor EMGs. *Electroencephalogr Clin Neurophysiol.*, 64(4), 378–380.

Mateo, J., San Agustin, J., & Hansen, J. P. (2008). Gaze beats mouse: hands-free selection by combining gaze and emg. *In CHI '08 extended abstracts on human factors in computing systems* (pp. 3039–3044). New York, NY, USA: ACM.

Popescu, F., Badower, Y., Fazli, S., Dornhege, G., & Mueller, K.-R. (2006). EEG-based control of reaching to visual targets. *In Dynamical principles for neuroscience and intelligent biomimetic devices -abstracts of the EPFL-LATSIS symposium* (pp. 123–124).

Staude, G., Flachenecker, C., Daumer, M., & Wolf, W. (2001). Onset detection in surface electromyographic signals: a systematic comparison of methods. *EURASIP J. Appl. Signal Process.*, 2001 (1), 67–81.

Surakka, V., Illi, M., & Isokoski, P. (2004). Gazing and frowning as a new human-computer interaction technique. *TAP*, 1 (1), 40-56.

Zhang, X., & MacKenzie, I. S. (2007). Evaluating eye tracking with ISO 9241 – Part 9. Proceedings of HCI International 2007, China, 779-788.

Gameplay experience in a gaze interaction game

Lennart Nacke, Sophie Stellmach, Dennis Sasse, Craig A. Lindley
Game and Media Arts Laboratory, Department of Interaction and System Design, School of Engineering, Blekinge Institute of Technology
PO Box 214, SE-374 24 Karlshamn, Sweden
Lennart.Nacke@bth.se, sstellmach@gmail.com, Dennis.Sasse@gmail.com, Craig.Lindley@bth.se

Keywords
Game experience, flow, immersion, gaming with gaze, human-computer interaction

Introduction

Eye tracking has been well researched in the field of human-computer interaction (Duchowski, 2007; Jacob, 1990). In addition, interacting with gaze as an input modality for digital games has recently become a popular area of study (Isokoski et al., 2007; Jönsson, 2005; Kenny et al., 2005; Smith and Graham, 2006; Špakov, 2005).

Jönsson (2005) compared eye and mouse control as input for two three-dimensional (*3D*) computer games and found that gaze control was more accurate, game experience was perceived as subjectively more enjoyable and committing. Smith and Graham (2006) studied eye-based input for several game types, with principally *3D* navigation. Their results show that participants felt more immersed when using the eye tracker as a gaming input device. Kenny et al. (2005) developed a first-person shooter (*FPS*) game that logs eye tracking data, video data and game internal data, which were correlated with each other. They found that players fixate the center of the screen for a majority of the time. Isokoski et al. (2007; 2006) describe the advantage of gaze pointing in *FPS* games as aligning the camera (view frustum of the player) to the target becomes obsolete, when aiming is decoupled from view. Their results indicate that gaze input for *FPS* games can compete with "killing efficiency" of gamepad input, but leads to more ammunition used due to problems of targeting accuracy. Thus, the game developed for this study was focusing solely on navigational challenges using gaze as input to a *3D FPS*, allowing us to assess *gameplay* experience with subjective questionnaires for this novel input modality.

In contrast to previous studies, which mainly compared mouse and gaze interaction in terms of efficiency and accuracy (Agustin et al., 2007; Dorr et al., 2007; Isokoski et al., 2007), the purpose of the study reported here is to broadly investigate gameplay experience in a *3D* gaze interaction game by testing the reliability of a range of subjective experiential questionnaires. As there are currently no established measures for assessing notions of, for example, *flow* (Csikszentmihalyi, 1990) in games, this investigation may be the foundation for a more thorough comparison of input modalities using subjective experience assessment questionnaires. While gameplay experience itself is currently not well understood, this is especially true for gaze interaction games. Thus, Gowases et al. (2008) claimed that it is very important to evaluate immersion and user experience in eye tracking games. Their study shows that players felt more immersed when using a gaze-based input method in comparison to using a mouse. We conducted a field experiment to investigate gaze steering in a *3D* game[1]. We assess gameplay experience in detail for this gaze interaction game using self-report game experience (IJsselsteijn et al., n.d.), flow (Csikszentmihalyi, 1990) and presence (Vorderer et al.,

[1] A video of the field experiment game stimulus and setup can be watched online at http://www.youtube.com/watch?v=6PZpsWzjnvE

2004) questionnaires. The novelty of this paper stems from it using statistical correlations of the previously mentioned experience questionnaires to explore key points of game experience in a gaze interaction game.

Method

Stimulus. The stimulus for this experiment was a game-mod level developed using the *Half-Life 2* Source SDK platform (Valve Corporation, 2004). The game level was designed around the concept of navigation in a *3D* virtual environment, thus, the goal was to navigate successfully (i.e. without falling off) on a catwalk to a door indicating the end of the level (see Figure 1). The keyboard keys (*W-A-S-D*) were used to control locomotion of the player, while gaze input was used to control the first-person camera view[2] (usually controlled with the mouse in *FPS*). Navigational challenges were the labyrinthine structure of the catwalk and several oil drums placed in the way of the player. Players highscore was evaluated according to playing time and times they fell off the catwalk. We chose to use this game, since we wanted to investigate the gameplay experience related to navigational challenge of using gaze as a steering method in an *FPS* game.

Figure 1. The 3D gaze interaction stimulus, a *Half-Life 2* level.

Participants. Data were recorded from *30* random people, *2* female and *28* male, at *Dreamhack Winter 2007*[3], aged between *14* and *32* (*Mean (M)* = 18.67, *Standard Deviation (SD)* = 4.26). Participants were gamers that were interviewed at a booth at *Dreamhack* and invited to participate in the experiment. *76.7%* started playing games when they were younger than *12* years old, *23.3%* did this when they were between *12* and *20* years old. *53.3%* considered themselves to be skilled players. *86.7%* preferred keyboard and mouse as their gaming input device, while the rest opted for joypad and joystick. Four people wore glasses and four people wore contact lenses (all of them near-sighted), one person was a bit cross-eyed with the left eye pointing a bit more to the left. Finally, *56.7%* indicated that they play computer games every day.

Experimental design and apparatus. The experiment was designed to assess gaze steering experience in a game level. Our questionnaires only focused on evaluating gaze-input game experience. However, we did allow participants to additionally play the game levels with a mouse[4]. Each participant was allowed to get used to playing with gaze in another test level first and then played the stimulus level until they had reached the door at the end of the level. Components of game experience were measured for the gaze condition using

---

[2] The mapping of gaze input can be described like this: If a player looks to the left side of the screen, the camera (or view) will also turn to the left, if he looks to the right side, the view will turn right, etc. - player controls her view in the virtual world by looking at different areas of the monitor.

[3] Dreamhack is the world's largest computer game festival, for more information see http://www.dreamhack.se

[4] Again, the primary focus of this study was not a comparison between gaze and mouse input, but an assessment of game experience for gaze interaction. The playing times with the mouse were much shorter (~1 minute), thus not resulting in measurable experience. We cross-checked the difference between gaze and mouse input with a few questions: *83.4%* indicated that they had to concentrate more for navigating correctly with gaze interaction. *63.3%* preferred gaze input to mouse input as interaction method. Only, *43.3%* felt limited by the gaze interaction in their navigation through the level.

a game experience questionnaire (*GEQ*) (IJsselsteijn et al., n.d.), which measures the experiential dimensions of *immersion*, *tension*, *competence*, *flow*, *negative affect*, *positive affect*, and *challenge*. Each component consists of *6* items – each indicating a feeling statement – to which agreement is measured on a five-point scale ranging from *0* (not agreeing with the statement) to *4* (completely agreeing), thus resulting in *42* statements. Component scores are computed as the average value of its items. The questionnaire is based on focus group research (Poels et al., 2007) and subsequent survey investigations among frequent players as part of the *EU*-funded *FUGA* project (Contract: *FP6-NEST-28765*). In addition to that the Flow State Scale (*FSS*) from Jackson and Marsh (1996) and items from the *MEC* Spatial Presence Questionnaire (Vorderer et al., 2004) were used to check for possible correlations with the *GEQ*. The gaze game was played using a *Tobii T120* eye tracker.

Procedure. The experiment was conducted as part of a booth from Blekinge Institute of Technology that was set up at *Dreamhack Winter* in Jönköping, Sweden from November 29 until December 2, 2007. Participants were selected from visitors of the event and invited to participate in the study to enter a lottery for winning a radio-controlled mini-helicopter at the end of the event. After playing, participants were asked to fill out the questionnaires, then thanked for their participation and escorted off the booth.

Results

The results of the game experience questionnaire showed a very positive game experience. Most notably positive affect (M = 3.31, SD = 0.45), immersion (M = 3.11, SD = 0.57) and flow (M = 2.84, SD = 0.74) scored high for the gaze input game. Challenge (M = 2.63, SD = 0.52) and competence (M = 2.37, SD = 0.80) scored high compared to prior studies (Grimshaw et al., 2008; Nacke and Lindley, 2008). As expected, negative affect (M = 0.37, SD = 0.41) and tension (M = 1.01, SD = 0.52) dimensions scored low, indicating a very pleasant game experience. Figure 2 shows average gameplay experience scores of the *GEQ* for the gaze interaction game.

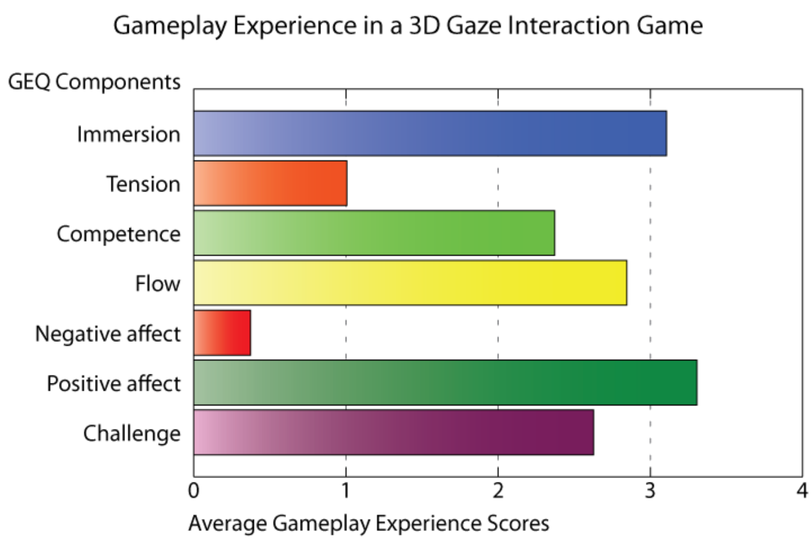


Figure 2. Average gameplay experience component scores recorded from playing the gaze interaction *Half-Life 2* mod.

The *FSS* (Jackson and Marsh, 1996) rates *36* items in a *1* (strongly disagree) to *5* (strongly agree) response format to assess *flow* experience (Csikszentmihalyi, 1990). It consists of several subscales (see Figure 3). The average flow value – calculated as a mean of all items – was high (M = 3.59, SD = 0.55) in comparison to values reported by Jackson and Marsh (1996). The highest flow component was *autotelic experience* (M = 4.43, SD = 0.77). Figure 3 shows the average scores of all the subscales in comparison.

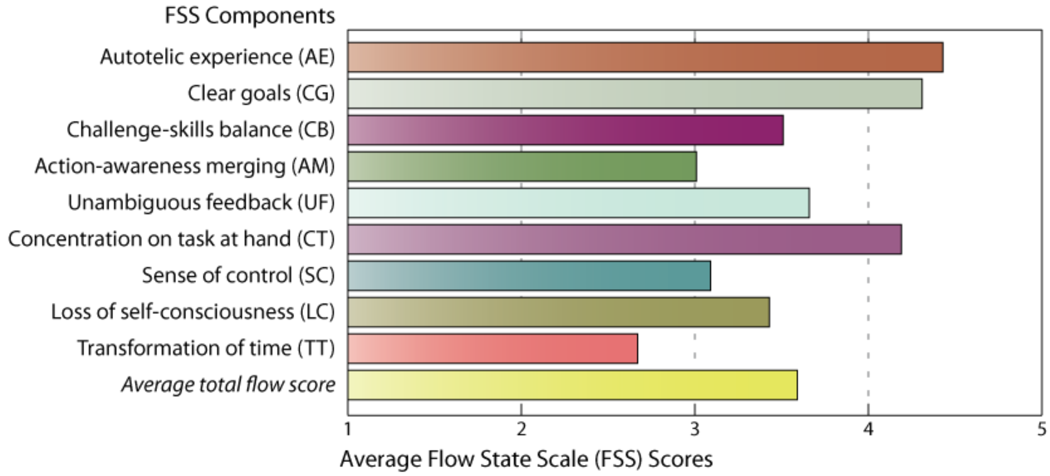


Figure 3. Average *FSS* scores recorded from playing the gaze interaction *Half-Life 2* mod.

Finally, we assessed mean scores for the *MEC* Spatial Presence Questionnaire (Vorderer et al., 2004). It can be noted that spatial presence possible actions (M = 3.68, SD = 0.81) ratings were significantly higher than spatial presence self-location (M = 3.21, SD = 0.02) ratings, $t(29) = -3.22$, $p < .01$. This could be attributed to the gaze interaction experience being more immersive than the content in the game.

When correlating the questionnaire items between *GEQ* and *FSS*, we found highly significant positive correlations (using Pearson's r[5]) between *challenge-skills balance* and *competence* ($r = .57^{*}$[6]), *clear goals* and *competence* ($r = .49^{*}$), *sense of control* and *competence* ($r = .61^{*}$), *autotelic experience* and *competence* ($r = .55^{*}$) and the overall *FSS flow* value and *GEQ competence* ($r = .58^{*}$). The *FSS flow* value and the *GEQ flow* value did not significantly correlate, in fact they were significantly different, $t(29) = -4.41$, $p < .001$. Interestingly, we also found very significant positive correlations between the items *unambiguous feedback* and *positive affect* ($r = .48^{*}$), *transformation of time* and *spatial presence self location* ($r = .50^{*}$), and between *autotelic experience* and *immersion* ($r = .51^{*}$). Also of interest was a significant *negative* relationship between *sense of control* and *challenge* ($r = -.37$, $p < .05$), indicating that less interaction control results in higher challenges. These relationships indicate the complexity of game experience for *3D* gaze interaction games.

Conclusion

Due to the experimental setting, the novelty of navigating in a game at a computer festival could be high. However, our findings indicate that gaze interaction games[7] provide a positive game experience, where the challenge of controlling the game by gaze (and keyboard) results in positive affection and feelings of flow and immersion, which can be reliably assessed using the questionnaires presented in this study. In line with the results of Gowases et al. (2008), *immersion* and *spatial presence possible actions* ratings for the gaze interaction game focusing on navigational challenge, were high and a relationship between *autotelic experience* and *immersion* was found, which demands further investigation. Having assessed questionnaires as usable tools to investigate game experience with a gaze navigation game, we plan to conduct more studies

[5] Pearson's correlation coefficient (r) is a standard measure for testing the relationship strength between two variables.

[6] * denotes $p < .01$ (which stands for highly significant correlations in this case)

[7] In this context, gaze interaction games refers to *3D* gaze interaction *FPS* similar to the mod that we have created, where the player exerts effort to control her view in the virtual world by looking at different areas of the monitor in the real world. This way of controlling camera view in an *FPS* game seems to have exciting potential in terms of gameplay experience and will serve us as an entry point to future experiments.

in a controlled laboratory setting that will allow us to study and log (see Nacke et al., 2008) different game and input types more thoroughly.

Acknowledgements

This research was supported by "*FUGA* - The Fun of Gaming: Measuring the Human Experience of Media Enjoyment", funded by the European Commission (Contract: *FP6-NEST-28765*). We thank all participants involved in this study. Also, we would like to thank Charlotte Sennersten for her help in organizing the Dreamhack experiments as well as Jessica Lundqvist, Anders Vennström and their colleagues from *Tobii Technology AB* for supplying and supporting us with the eye trackers.

References

Agustin, J. S., Schantz, J., and Hansen, J. P. (2007). Eye Trackers: Are They Game? In: *Proceedings of the 2nd Conference on Communication by Gaze Interaction*, Leicester, UK, Communication by Gaze Interaction (COGAIN), pp. 20-24.

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. New York: HarperPerennial.

Dorr, M., Böhme, M., Martinetz, T., and Barth, E. (2007). Gaze beats mouse: a case study. In: *Proceedings of the 2nd Conference on Communication by Gaze Interaction*, Leicester, UK, Communication by Gaze Interaction (COGAIN), pp. 16-19.

Duchowski, A. T. (2007). *Eye tracking methodology: Theory and practice* (Second ed.). New York: Springer.

Gowases, T., Bednarik, R., and Tukainen, M. (2008). Gaze vs. Mouse in Games: The Effects on User Experience. In: *Proceedings of the International Conference on Advanced Learning Technologies, Open Contents & Standards (ICCE)*, Taipei, Taiwan, pp. 773-777.

Grimshaw, M., Lindley, C. A., and Nacke, L. (2008). *Sound and Immersion in the First-Person Shooter: Mixed Measurement of the Player's Sonic Experience*. Paper presented at the Audio Mostly 2008 - A conference on interaction with sound.

IJsselsteijn, W. A., Poels, K., and de Kort, Y. A. W. (n.d.). The Game Experience Questionnaire: Development of a self-report measure to assess player experiences of digital games.

Isokoski, P., Hyrskykari, A., Kotkaluoto, S., and Martin, B. (2007). Gamepad and eye tracker input in first person shooter games: Data for the first 50 minutes. In: *Proceedings of the 2nd Conference on Communication by Gaze Interaction*, Leicester, UK, Communication by Gaze Interaction (COGAIN), pp. 11-15.

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 2006: Gazing into the Future*, Turin, Italy, pp. 78-81.

Jackson, S. A., and Marsh, H. W. (1996). Development and validation of a scale to measure optimal experience: The Flow State Scale. *Journal of Sport and Exercise Psychology, 18*, 1, pp. 17-35.

Jacob, R. J. K. (1990). What you look at is what you get: eye movement-based interaction techniques. In: *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, Seattle, Washington, United States, ACM, pp. 11-18.

Jönsson, E. (2005). *If Looks Could Kill – An Evaluation of Eye Tracking in Computer Games.* Unpublished Master's Thesis, Royal Institute of Technology (KTH), Stockholm, Sweden.

Kenny, A., Koesling, H., Delenay, D., McLoone, S., and Ward, T. (2005). A preliminary investigation into eye gaze data in a first person shooter game. In: *Proceedings of the 19th European Conference on Modelling and Simulation*, Riga, Latvia, pp.

Nacke, L., Lindley, C., and Stellmach, S. (2008). Log who's playing: psychophysiological game analysis made easy through event logging. In: *Proceedings of Fun and Games, Second International Conference*, Eindhoven, The Netherlands, Springer, pp. 150-157.

Nacke, L., and Lindley, C. A. (2008). Flow and Immersion in First-Person Shooters: Measuring the player's gameplay experience. In: *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, Toronto, Canada, ACM, pp. 81-88.

Poels, K., Kort, Y. d., and Ijsselsteijn, W. (2007). "It is always a lot of fun!": exploring dimensions of digital game experience using focus group methodology. In: *Proceedings of the 2007 conference on Future Play*, Toronto, Canada, ACM, pp. 83-89.

Smith, J. D., and Graham, T. C. N. (2006). Use of eye movements for video game control. In: *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology (ACE)*, Hollywood, California, ACM Press, pp.

Špakov, O. (2005). *EyeChess: the tutoring game with visual attentive interface*. Paper presented at the Alternative Access: Feelings & Games.

Vorderer, P., Wirth, W., Gouveia, F. R., Biocca, F., Saari, T., Jäncke, F., et al. (2004). *MEC Spatial Presence Questionnaire (MECSPQ): Short Documentation and Instructions for Application* (No. IST-2001-37661).

Selecting commands in 3D game environments by gaze gestures

Stephen Vickers[1] **Howell Istance**[12] **Aulikki Hyrskykari**[2]

[1]Human-Computer Interaction Research Group,
De Montfort University, The Gateway,
Leicester, LE1 9BH, UNITED KINGDOM
svickers@dmu.ac.uk, hoi@dmu.ac.uk

[2]Human-Computer Interaction Unit (TAUCHI),
Department of Computer Sciences
FIN-33014 University of Tampere, FINLAND
ah@cs.uta.fi

Keywords
gaze control, gaze gestures, games, MMORPG

Introduction

People with motor impairments can benefit greatly from being able to take part in Massively Multiplayer Online Role-Playing Games (MMORPGs), such as World of Warcraft. These are 3D graphical worlds which have many similarities with virtual environments but may be inhabited with thousands of players simultaneously. We are investigating how to use eye gaze as a high bandwidth input modality for the range of tasks necessary to play MMORPGs. Our emphasis in this work is facilitating all interaction by means of gaze, rather using gaze as a complementary modality to mouse, keyboard and gamepad. There are several broad categories of tasks associated with playing MMORPGs, namely locomotion and camera control, fighting with other characters, manipulating objects and equipment, and communicating with other players. The work we have done to date has shown that gaze can be almost as effective as keyboard and mouse in controlling locomotion. Selecting icons representing objects or commands on the game client interface using gaze is subject to the same problems of inaccuracy as are common with gaze controlled 2D desktop applications. A significant additional problem, however, when using gaze is that the player's visual attention is distracted from the action taking place in the game. This is when gaze position is being used for making dwell-click selections on icons located in shelves or tool-bars at the edge of the client window. Some way of activating commands that minimises the distraction is highly desirable. Gaze gestures offer a potentially powerful solution because:

- they can be activated 'close in' to the central avatar position
- they should be faster than dwell as no need to wait for the dwell period to expire, although this will depend on how complicated the gesture is
- accuracy problems should be removed as the player no longer needs to look at and hit small targets to activate commands

Background and Related Work

Gaze gestures have already been used for text entry. Bee and Andre (2007) implemented a gaze gesture scheme based on Quikwriting, which was devised by Perlin (1998) for entering text with a stylus on a PDA. In this scheme, there are 8 active zones around a central rest position. To select a letter, the stylus is moved into the 'major' zone containing the letter, then it moves to the 'minor' zone corresponding to the position of the letter within the major zone. The zones in Quikwriting are labelled as shown in Figure 1. The path shown the figure selects the letter 'l'. Bee and Andre identified visual feedback as a significant design issue when the path is described by gaze position instead of a stylus. This refers to the user wanting to check the positions of the letters in the zones, and wanting to check the content of the string already entered. If the user glances back at the layout of letters after they have started to enter a gesture, this will probably cause an unwanted gesture. Similarly, if the user glances at the string they have already entered mid-gesture, this will also cause the gaze path to cross an active zone, and again result in an unwanted gesture. They proposed design solutions to each of these problems in their implementation. They compared performance on a text entry task using gaze gestures with using a conventional gaze-operated visual keyboard. They reported text entry rates of 5 wpm with gestures compared with 7.8 wpm for the visual keyboard.

Wobbrock et. al. (2008) described a scheme called Eyewrite, which they had adapted for use with gaze from a scheme they had earlier developed for use with a stylus and small display (called Edgewrite). Here there were only 4 active zones arranged around the central rest position, but in this scheme individual letters were represented by patterns which bore some resemblance to the shape of the letters (Figure 2). They reported an extensive longitudinal study where performance was studied over 14 sessions and they reported text entry rates of 5 wpm for gesture and 7 wpm for visual keyboard use, which were similar to those found by Bee and Andre.

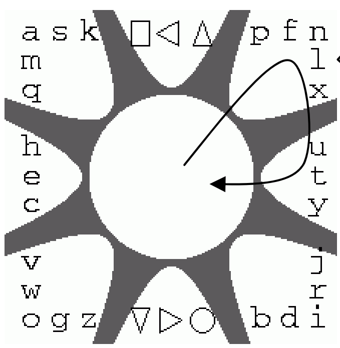


Figure 1. Quikwriting (stylus gesture produces letter 'l')

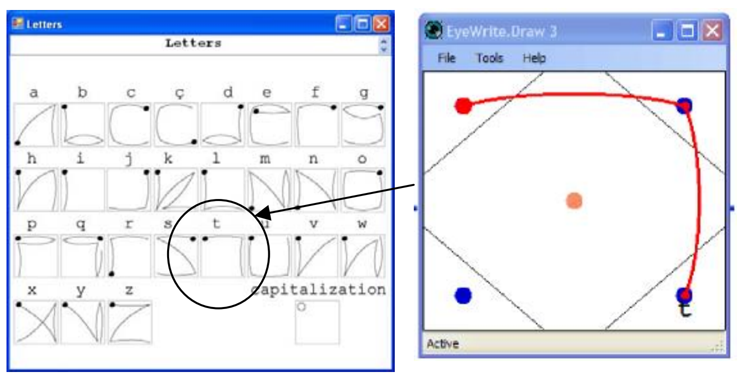


Figure 2. Eyewrite (gaze gesture produces letter 't')

Other approaches to using gaze gestures have sought to define deliberate patterns of scan path data without any notion of active zones, and then to establish how reliably these patterns can be extracted or identified in sequences of normal activity, such as reading (Drewes and Schmidt, 2007, Heikkila 2008).

Design Solution

The general approach we have adopted to gaze control of MMORPG's is keyboard and mouse emulation. World of Warcraft allows keyboard and mouse events to be assigned to a wide range of interface action. These events can be generated by a separate application that recognises gaze gestures. The notion of gesture 'zones' as embodied in Quikwriting and Eyewrite is quite acceptable for our purposes, as one of the main reasons for using gestures is to allow the player to keep their visual attention on the central region of the screen. We can arrange the active zones around the avatar as long as the visual disruption they cause is minimised. The active zones are visible, but nearly transparent. We have adopted a reduced version of Quikwriting as the basis for the initial design solution. There are only 4 active zones (one on each corner, akin to Eyewrite) allowing a total of 12 recognisable gestures. We carried out a study of gaze positions during normal play, which showed most of the time the player is looking within a fairly small central region of the window. However, the size of the overlay area containing the active zones is configurable to accommodate individual preferences for camera angles and fields of view. The key and mouse events allocated to each of the gestures are also configurable; the configuration dialog is shown in Figure 3.

The appearance of the game client window with the active areas visible is shown in Figure 4. There is no labelling on the zones themselves, but a small reminder overlay showing which commands are currently associated with the 12 gestures can be made visible and located over any part of the client window (visible on the right in Figure 4). The figure shows the three moves in a gesture that selects the upper element associated with the bottom left zone (zone 4A, command ‘spell A’, cf. the circles in Figure 3), in a similar way to the Quikwriting scheme.

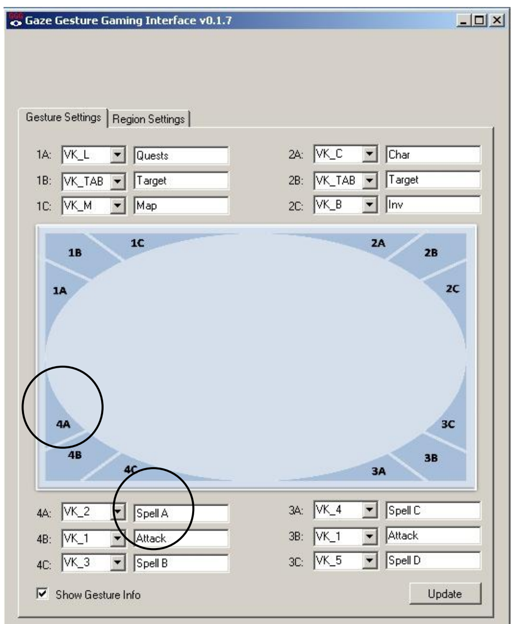


Figure 3. Gesture configuration dialog, showing the command generated in Figure 4

Figure 4. Performing a gesture in World of Warcraft, to select Spell A in this case

In our previous studies, participants have had to select a target monster or creature (a ‘mob’) by dwell-clicking on the target. This was difficult if the mob was moving at the time. Therefore we mapped a gesture that would automatically select the mob that was nearest to the avatar regardless to whether it was moving or not (zones 1B and 2B – ‘Target’ in Figure 3). The user could choose to cycle through several different mobs close to them by repeating the gesture. We have earlier developed a system for rapid mode switching using glances to support game playing (Istance et.al. 2008). In this context, a mode is a single means of interpreting gaze events, such as: left dwell click, right dwell click, locomotion control, turn gaze control off and so on. The use of gaze gestures for command selection is integrated with the mode selection software.

Initial user experience

The gaze gesture interface is fully implemented and operational, but we have not performed the actual user tests for it yet. However, we have been testing the interface more informally, and the user experiences of one informal pilot study are presented below.

Only eight of the twelve possible gestures were assigned to fighting actions. The remaining gestures were assigned to general interface actions such as "Open Quest Log", "View Map" and so on. Two of the most common actions were duplicated with two gestures that were of the shortest pattern: target a mob was assigned to 1B and 2B (Figure 3) and basic attack was assigned to 4B and 3B. Mirroring the means of selection was natural for the user in two ways. Firstly, the mobs would always first appear in the distance, thus being in the top 1/3 of the screen and the gesture used to select them uses the upper quadrant zones. Secondly, as the mob could appear anywhere from left to right in the area of the screen, the user could perform the gesture in the left or right zones that were closest to the mob, and the presumed object of attention.

Initially, the pressure of performing a gesture in a time-constrained situation, such as fighting with a creature was found to be difficult. The user would panic and fail to make the required gesture even after several attempts causing frustration. Especially during the first few minutes of play, the user would try to make the gesture as quickly as possible but would actually generate an unrecognisable eye movement. Additionally, getting used to having the zones present on the screen and not being distracted by them took some time. However, after more practice the user became relaxed and reported thinking less about performing the gestures and more on playing the game. Consequently, the gesture was made more calmly and not with quick erratic eye movements.

The gestures that were used to target the mobs and use the basic attack were easily recalled. However, as the other attacks were performed less frequently, remembering the necessary gesture was difficult. The user had to refer to the gesture reminder overlay. After the initial use sessions, the user stated that the use of gestures for initiating the fighting commands was comparable with dwell-clicking. However, this was not necessarily due to the speed of performing the gesture. The length of time to perform a gesture was not accurately recorded in this initial user evaluation but a review of the screen captured video shows that it took between 1 and 1.6 seconds to perform a successful gesture. The most significant advantage reported was less frustration through not having to accurately position a cursor over a small target and then dwell for a period of time.

Conclusion and Future Work

The next phase in the work will be to study in detail the use of gestures for tasks with considerable time constraints, primarily fighting tasks. We intend to compare performance using the gesture scheme described in this paper with dwell-clicking a set of 12 semi-transparent command buttons arranged around the avatar. While gesture-based interaction appears to offer considerable benefits over dwell-clicking, the extent to which these can be realised needs further investigation. If we can establish that these benefits exist, then rapid gesture-based interaction represents a significant departure from the conventional dwell-clicking paradigm, with great advantages for gaze-controlled game playing and gaze controlled applications in general.

References

Bee, N., and Andre, E. (2008) Writing with Your Eye: A Dwell Time Free Writing System Adapted to the Nature of Human Eye Gaze, *Lecture Notes in Computer Science, Volume 5078/2008 Springer Berlin / Heidelberg*, 111-122

Drewes, H., and Schmidt, A. (2007) Interacting with the Computer Using Gaze Gestures. *Proc. of INTERACT 2007*, 475-488

Heikkilä, H. Gesturing with gaze, *Proc. of the 4th Conference on Communication by Gaze Interaction (COGAIN 2008)*, 65-69.

Istance, H., Bates, R., Hyrskykari, A., and Vickers, S. 2008. Snap clutch, a moded approach to solving the Midas touch problem. *Proc. of 2008 Symposium on Eye Tracking Research & Applications (Savannah, Georgia, March 26 - 28, 2008). ETRA '08. ACM,*

Perlin, K. (1998) Quikwriting: continuous stylus-based text entry. *Proc. UIST '98,* 215-216, a demo available at http://mrl.nyu.edu/~perlin/demos/Quikwrite2.html

Wobbrock, J.O., Rubinstein, J., Sawyer, M., and Duchowski, A.T (2008) Longitudinal evaluation of discrete consecutive gaze gestures for text entry. *Proc. 2008 Symposium on Eye Tracking Research & Applications (Savannah, Georgia, March 26 - 28, 2008). ETRA '08. ACM*, 11-18.

GazeTrain: A case study of an action oriented gaze-controlled game.

Lasse Farnung Laursen
DTU-Informatics
Technical University of Denmark
DK-2800 Kgs. Lyngby, Denmark
lfla@imm.dtu.dk

Bjarne K. Ersbøll
DTU-Informatics
Technical University of Denmark
DK-2800 Kgs. Lyngby, Denmark
be@imm.dtu.dk

Keywords
Gaze-Interaction, Game, Human Computer Interaction, Accessibility, Usability, Usability Heuristics

Introduction

Creating an engaging and worthwhile gaze-controlled game poses a significant number of challenges. The challenges are compounded by the fact that a number of popular game mechanics rely on quick reactions, and complex input. The straightforward solution to this issue is to avoid these game mechanics, and use the ones that do not require quick reactions. However, avoiding specific mechanics can lead to excluding certain types of game genres, thus reducing the variety of an already meager games selection. With the intention of providing further variety to physically challenged gamers, we present *GazeTrain*, inspired by a classic action oriented puzzle game *Pipe Dream* (Wikipedia, 04.03.2009) developed in 1989. We describe the process, necessary compromises and possible pitfalls of redesigning and implementing a game concept for use with gaze control. Preliminary tests and their results are also briefly discussed at the end of the article. The game and associated source is available for free download from The COGAIN Association's homepage: *http://www.cogain.org*, in the hopes of motivating further development within the field of leisure applications using gaze-control. In addition to the downloadable game, it will be possible for gamers to contribute feedback and donations towards further development.

In *Pipe Dream*, a source of inspiration for *GazeTrain*, the player is responsible for assembling a set of short pipes through which an unidentified liquid can flow. The short pipes are placed in a grid structure and cannot be removed once placed. Individual pipe pieces can however be replaced, at a point penalty, if the liquid has yet to flow through it. Constructing a sufficiently long line of short pipes for the liquid to flow through, allows the player to advance to the next level. The player is provided with only one pipe-piece to place and a short overview of upcoming pieces. The game ends when the liquid catches up to the end of the current pipe, and it is not sufficiently long. A freeware implementation of *Pipe Dream* can be downloaded from the following location: http://members.chello.at/theodor.lauppert/games/pipe1.htm.

Developed over a two month period, *GazeTrain* (see Figure 1) charges the player with placing traintracks in front of a moving train to guide it and solve simple tasks. Although the challenges in the game vary, depending on which mode the game is played in, the ultimate goal remains the same. Earn as much money as possible by traversing new tracks and delivering cargo at various cities. The game modes consist of: Practice Mode, Resource Mode, and Timed Mode. Practice Mode allows novice players a chance to familiarize themselves with the interface without being distracted or pressured by other game elements. Resource Mode charges the player with delivering a set number of resources as fast as possible. Timed Mode challenges the player to deliver as much cargo as possible, given a set time limit. This time limit is extended as the player delivers cargo.

Game Design Choices

Pipe Dream, which *GazeTrain* is loosely based upon, is comprised of multiple game mechanics. A number of these lend themselves towards being gaze-controlled, where as others do not. During development a few core concepts from *Pipe Dream* were adopted directly into the design of *GazeTrain*, a number of them were modified before being adopted, and some were dropped altogether. Instead of detailing every core concept and the necessary modifications to make them gaze-control compatible, we describe what we consider to be the most critical design choices, which are applicable to several core concepts, below:

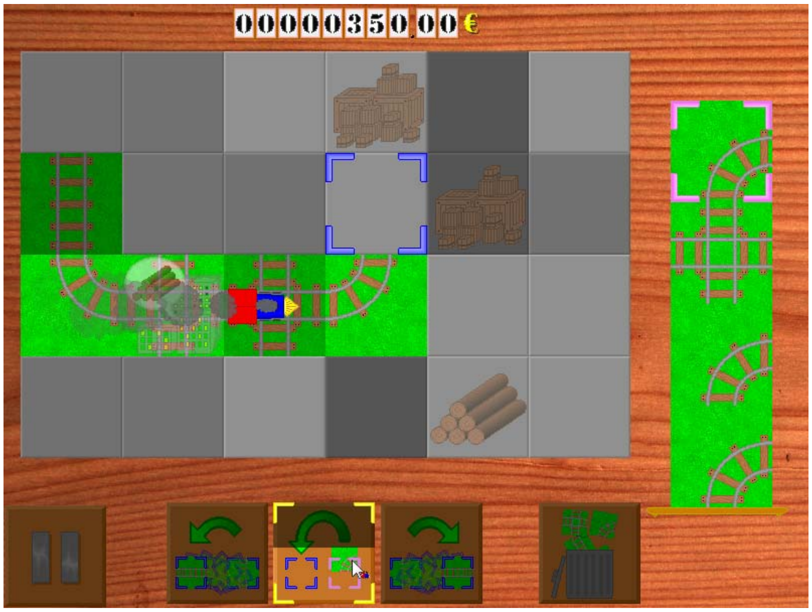


Figure 1. In-Game Screenshot of GazeTrain.

- **Customizability** – Physically challenged gamers will, to a varying degree, be unable to control their interaction with a given application. Interaction can be hampered by involuntary muscle spasm or inferior eye tracking. If a game allows for the customization of various game and/or interface elements, the user may be able to make the game compensate for the involuntary actions. The more customizable a game is, the higher the likelihood that a motivated, but physically challenged, gamer can tailor it uniquely to her needs. However, it is important to note that the developer must not shift the burden of properly tweaking various game elements, over to the user. For example, the developer should strive to properly balance the games difficulty, and then make it possible to modify, for the benefit of the user. Note that this customizability should preferably extend beyond game elements and into meta-game territory, such as the Graphical User Interface.
- **Soft penalties** – It goes without saying that any game should strive to deliver a positive experience to its users. According to flow theory (Csikszentmihaliy, 1990), people are happiest when they are in flow. Specifically, when they are just able to overcome the challenges presented to them. If the challenges are too overwhelming, they will become frustrated, but if the challenges are too easy, they will become bored. According to Csikszentmihaliy, entering flow requires that both the challenge presented and skills required to overcome it, are at their peak. Some people describe this state of flow as completely focused on the task at hand and being unaware of the world around them. But flow theory can be difficult to apply in an action oriented gaze-controlled game. Game mechanics requiring quick reactions, and involuntary player actions, are at odds, especially in this regard. If the

game cannot detect involuntary actions, then providing the player with a challenge they are just able to overcome, is very difficult. As previously stated, one solution is avoid these troublesome mechanics. But this might deprive the player of an enjoyable game mechanic. An alternate solution is to still use these mechanics, but alter the game to soften penalties for any mistakes. For example, if the player fails to react properly within a certain amount of time in *Pipe Dream*, the game ends. The corresponding penalty in *GazeTrain* is that the player either gains fewer points, or receives less time to play. A much softer penalty than an abrupt ending to the game. Using this softer penalty means that the challenge to the player remains largely intact, but the frustration of failure, either due to voluntary or involuntary action, is reduced.

- **Unorthodox interface control** – A core game mechanic in *Pipe Dream* is the ability to quickly react and place game pieces in the correct position. As previously stated, the penalty for being unable to react correctly in time has been reduced in *GazeTrain*, to better accommodate players using gaze-control. However, the standard dwell click technique used in a number of gaze-controlled games remains at odds with this fast-paced requirement. Standard protocol would involve an extended dwell time when an action of some importance is undertaken. This extended dwell time was replaced with an additional, but shorter, dwell click. Figure 2 shows the complete number of steps required to place a single tile in *GazeTrain*. First, a source tile is selected, then a destination is chosen, and finally the action is confirmed by selecting the "transfer tile" button. The selection of the source and destination would be required regardless of interaction technique. The relevant difference is that instead of letting the user dwell an extended period of time on the destination square, a movement of the mouse pointer (via the eyes) and shorter dwell time is required.

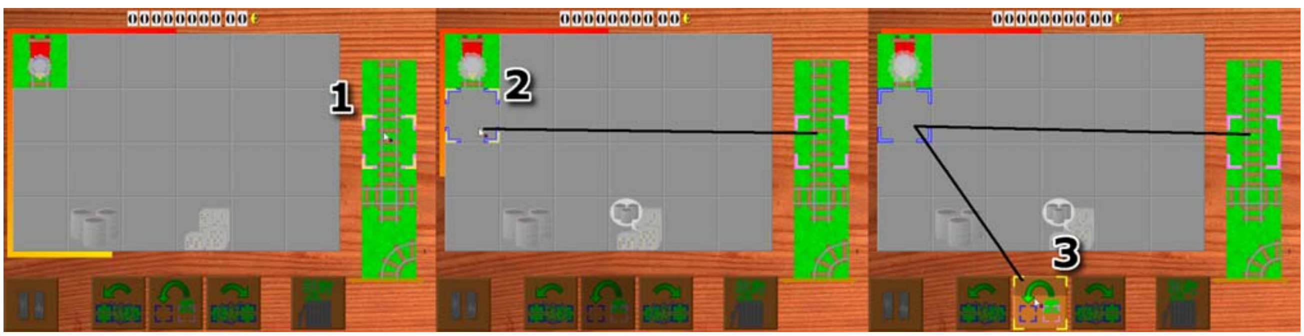


Figure 2. 3 step track tile placement process.

- **Active Elements** – Moving or animated elements in a gaze-controlled game can cause unintended effects as they have a tendency to attract the attention of the player. Once the players' attention has been attracted, the mouse pointer will follow. In an attempt to avoid unintended actions due to misdirected attention, all of the consequential actions require the user to look within an area containing no excessively active elements. Every action, which can be performed without looking in that specific area, is of no consequence or can easily be undone.

User Testing

Two volunteers (both male) participated in the preliminary evaluation. Neither with any previous eye tracker experience. The test platform was an ERICA Eye Tracker comprised of a Sahara Slate PC (Tablet PC) with an Intel Pentium M 1.3 Ghz processor and a 12.1 inch screen with a resolution of 1024 x 768.

Each participant played the practice mode of the game for approximately ten minutes. As mentioned earlier, the practice mode allows the player to familiarize themselves without any additional pressure. During the tests, each participant was asked to verbalize their thoughts. It should be noted that during each participants test the eye tracker had to be recalibrated to allow for proper interaction. After each test, a short interview was conducted with the participant to gain further insight into their experience. We have listed what we believe to be the most relevant observations below:

- **Dwell Time** – Initially, the dwell time was set to be 500 ms, based on the settings used in EyeDraw (Hornof et al., 2004). The first participant experienced a number of problems with this setting and the dwell time was increased to 1500 ms (closer to the setting described in EyeChess (Spakov and Miniotas, 2005). The second participant revealed during the interview that he felt the dwell time was somewhat too long. The difference in preferred dwell time is probably related to the players' ability and experience, as implied by Spakov and Miniotas. However, we also believe that a tolerable dwell time is related to the accuracy of the eye tracker. The more precise an eye tracker is the less trouble a user will have, which could lead to a preferred shorter dwell time. If the eye tracker is less precise the user may prefer a longer dwell as to ensure no mistaken actions are performed. All of these observations further underline the need for customizability to allow for the best user experience. Additionally, it may be beneficial to use different dwell times for different interface elements. One volunteer commented that the dwell time for the control panel elements should be reduced since these elements were much easier to differentiate than others. It is important to note that the accuracy of the eye tracker also affects the perception of how easily certain interface elements are selectable.
- **Interface Design** – When possible, the interface should consider the need for additional applications to be visible during its execution. For example, the ERICA eye tracker provides a video feed window which provides feedback on whether or not the users' eye is in view and being tracked properly. Ideally the interface should also be customizable to some degree, allowing the user to place interface elements in preferred locations or even change their sizes to allow for easier access.

Conclusion and perspectives

Although there exists a number of resources (Donegan, 2006) intended to aid the development of gaze-controlled software, there is still a need for further research and more comparative studies. The observations presented in this article are intended to provide further insight, and help similar projects during the decision making process to avoid potential pitfalls.

For future tests of *GazeTrain*, and probably most other gaze controlled applications, it would be an advantage to measure the accuracy of the used eye tracker during the tests. The accuracy has a significant effect on both the users experience and perceived interface usability. However, regardless of what concrete measurements yield, each physically challenged gamer will have their own personal set of preferences and perhaps disabilities which results in very specific requirements of the application in question. Without knowing exactly which limitations the user has, customizability is one of the best approaches to the problem.

Acknowledgements

The project has received funding from the Technical University of Denmark and the EU Network of Excellence COGAIN, for which we are thankful. We are also thankful for the loan of an Eye Tracker from the COGAIN pool during development. In addition we wish to thank Mick Donegan, ACE Centre for his insightful feedback and Henrik Jensen, IT-University for his valuable input.

References

Csikszentmihaliy, M. (1991) Flow: The Psychology of Optimal Experience. Harper Perennial.

Donegan, M. et al. (2006) D3.2 Report on features of the different systems and development needs. *Communication by Gaze Interaction (COGAIN)*, IST-2003-511598.

Hornof, A. J., Cavender, A., and Hoselton, R. (2004) EyeDraw: A system for drawing pictures with eye movements. *Proceedings of ASSETS 2004: The Sixth International ACM SIGACCESS Conference on Computers and Accessibility*. ACM Press, New York, NY, pp. 86-93.

Spakov, O., and Miniotas, D. (2005) EyeChess: A Tutorial for Endgames with Gaze-Controlled Pieces. *Cogain 2005*, pp. 16-18

Wikipedia (05.03.2009), Pipe Mania (video game).

Detecting Search and Rescue Targets in Moving Aerial Images using Eye-gaze

James Mardell, Mark Witkowski and Robert Spence
Department of Electrical and Electronic Engineering
Imperial College London,
Exhibition Road, London SW7 2BT, U.K.
{james.mardell, m.witkowski, r.spence}@imperial.ac.uk

Keywords
Gaze assisted target location, Wilderness Search and Rescue (WiSAR), Unmanned Aerial Vehicles (UAV).

Introduction

The search for a person missing in a wilderness area can be approached by flying a remotely piloted Unmanned Aerial Vehicle (UAV) to the most likely estimated location of the missing person and conducting a systematic search of the area based on a live video feed of the terrain below. A human operator observes this imagery in order to identify the missing person or associated artefacts (e.g. Goodrich, 2008). Our aim is to discover whether a characterisation of eye-gaze behaviour, coupled to its continuous monitoring, can enhance the likelihood of success in the search and rescue process.

To create a simulated moving terrain image suitable for the experiment, high-resolution (6 inch/pixel) selected parts of aerial photographs obtained from the Montana State Library Natural Resource Information System (http://nris.mt.gov/gis/) were converted into a video stream. The resulting video simulated the feed from a UAV travelling at four different apparent speeds: 20, 40, 80 and 120 mph (approx. 9, 18, 36 and 54 ms^{-1}) for an image swept width of 320 ft (97.5 m). A typical mini UAV for this task might travel at around 40 mph (e.g. McLain and Beard, 2004). The range of speeds was chosen to bracket the view from cognitively "too slow" to "too fast". We prepared four video sample types at a resolution of 640x640 pixels, two showing a forested area, with and without targets, and two showing scrubland, with and without targets. Each video sequence represents a search area of 320 ft by 3840 ft (97.5x1170 m) and lasted for 144, 72, 36 and 20 seconds at the respective speeds. In all cases we assume a downward facing camera[1] and ignore various flight artefacts such as vibration and banking effects.

We presented these sequences in three distinct modes. In the first "video stream" mode, features and targets appear at the top of the fixed video window and move continuously towards the bottom (as though the UAV were flying towards the top or "northwards" on the screen). In the second video mode, features appear at the right and travel to the left ("eastward" flight across the screen). In the third mode, the complete image sequence was segmented into a series of non-overlapping static images and presented sequentially in the style of a Rapid Serial Visual Presentation (RSVP) "slide-show" mode (as Cooper *et al.*, 2006). In this mode targets and features appear static on the screen for the duration of the presentation. Overall timings were maintained. We did this to see if moving video images were more effective or preferable to the equivalent static sequences. Figure 1 illustrates the scenario and presentation modes.

We found distinct gaze behaviour differences between the video mode and the "slide-show" mode. We report here on the effects of different speeds and identify the possibility of assisted target identification in the

[1] In practice a WiSAR UAV may have a steerable camera, and the pilot may require a different view of the terrain to maintain control.

domain of search and rescue. The remainder of the paper summarises the experimental method and presents some preliminary findings of note. From these we identify some conjectures that we plan to investigate in further detail.

Figure 1. The Flight and Presentation Mode Scenario, video (above), RSVP (below). See text for timings.

Experimental Procedure

Our approach was to record and characterize eye-gaze behaviour of a number of volunteer participants (mostly drawn from our student population), when asked to locate moderately distinctive targets (automobiles) within the video and image sequences. Each participant gave consent, and was introduced to the task from a script. A standard 9-point gaze calibration procedure was performed. Any visual conditions were noted, and all participants successfully completed both the calibration and experimental procedure.

In our preliminary trial, we tested 10 participants with four order randomised presentations of each of the sequence mode types (forest, scrub, target, no-target), each participant viewed a sequence at each of the four speeds and each saw at least one example of the "northward", "eastward" and RSVP presentations (one duplicated in each case)[2]. Participants were also requested to press the space bar when they identified a target; false negatives and positives were noted. Gaze movements were recorded throughout the presentations using an LC-Technologies (VA, USA, www.eyegaze.com) gaze recording system. The X and Y screen (approx. 15 pixel accuracy) coordinates of the gaze position were recorded every 16.667 ms. The participant sat with an eye distance of approximately 72 cm from a 15" LCD screen of 1024x768 pixels.

At the end of each presentation the participants were asked to rank (on a scale of 0 to 10) the task just performed on four separate issues: i) "How immersive was the video?" ii) "How nauseous did you find the video?" iii) "How hard/easy was it to locate the target?" iv) "What did you think of the speed of the video?" Participants were also permitted to offer comments both during and after the experiment, and these were noted.

Observations and Discussion

Figure 2A-D, show the gaze density plots for each of the four speeds for the "eastward" video motion. The plots are an accumulation across all participants, with invalid gaze points eliminated; the plot densities are normalised. The histogram above each plot (along the direction of motion) shows the distribution of individual gaze point X-values in each column of the video display. The histogram at the right of each plot (across the direction of motion) shows the distribution of gaze Y-values in each row. Means and standard deviations (indicated by blue and orange sidebars) are shown in each case.

There is a very marked asymmetric distribution along the direction of movement at each speed, the eye apparently naturally being positioned to make best use of the available visual space as features enter the presentation area. The movements across the motion are clearly centred, and show a distribution around this. It may be noted that the range of this (vertical) motion is more restricted at the highest speed (Figure 2D), and

[2] That is, no participant saw the same sequence mode or at the same speed twice.

we surmise that the high transit rate severely curtails the opportunity for across motion search. We had considered the possibility that "northward" travel results would be influenced by cultural issues such as reading direction. This appears not to be the case, as very similar equivalent results were obtained from the "northward" travel (not shown) instances to those presented here in Figure 2A-D. Figure 2E-F show equivalent gaze density plots at the 40 and 120 speeds for the serial RSVP mode. Unsurprisingly, the gaze point distributions are largely symmetric about the centre region in both the X and Y directions.

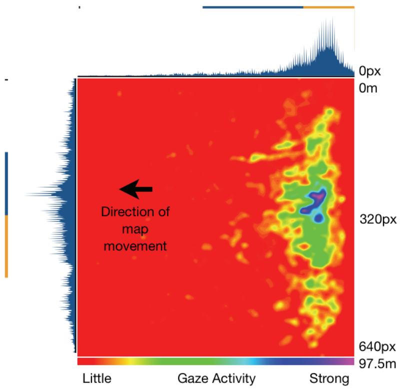


Figure 2A. Eastward, 20 mph

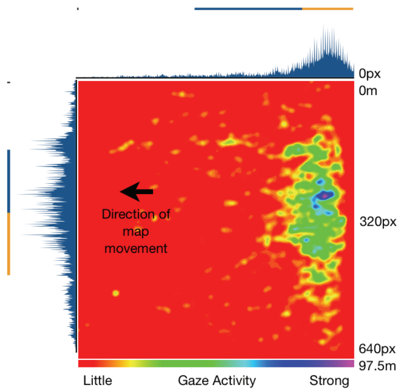


Figure 2B. Eastward, 40 mph

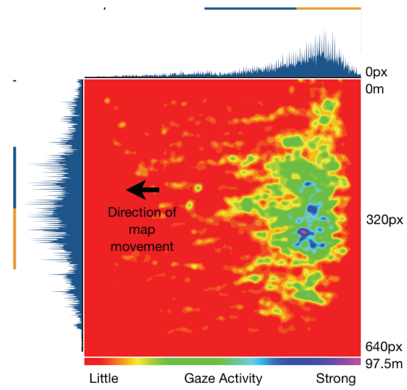


Figure 2C. Eastward, 80 mph

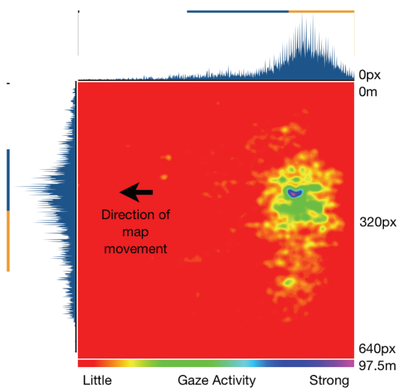


Figure 2D. Eastward, 120 mph

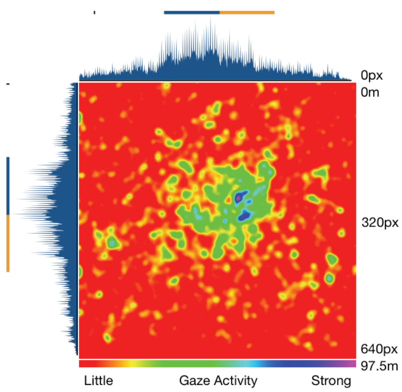


Figure 2E. RSVP, 20 mph

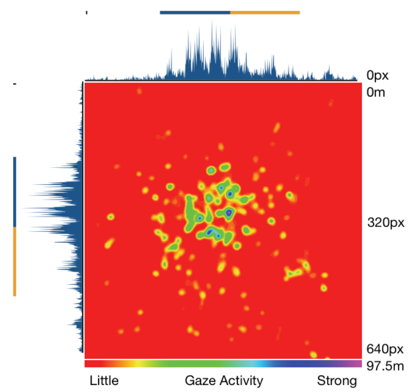


Figure 2F. RSVP, 120 mph

Initially we had conjectured that it might be possible to exploit this asymmetry of distribution in the video sequences to automatically detect or confirm when targets were observed, as there is also a clear tendency for observers to observe and track potential targets along the direction of motion. Due to the spread of gaze points it is not possible to distinguish targets from other features on the basis of position alone. However, Figure 3 indicates a clear tendency of participants to track targets for significantly longer than non-targets that are also naturally salient (typically trees, boulders, light soil areas, etc.).

Figure 3 (left) shows the complete gaze trace for one participant observing the "Eastward"/scrub/target/80 mph sequence. Figure 3 (right) shows several hand segmented tracking episodes extracted from the trace and clearly shows the track associated with the target to be substantially longer than any of the others. The target is shown circled in the single video frame used as background. Preliminary analysis suggests this extended tracking effect appears to be consistently present for each of the higher motion speeds (at 80 mph, 6 out of 7 possible targets; at 120 mph, 4 out of 4), but that the gaze pattern tends to revert to a normal saccadic/fixation search at lower motion speeds (20 mph, none detected; 40 mph, 1 of 3). The RSVP mode clearly precludes the exploitation of the tracking phenomena. As a rule of thumb, the detection traces were some three times longer than the others.

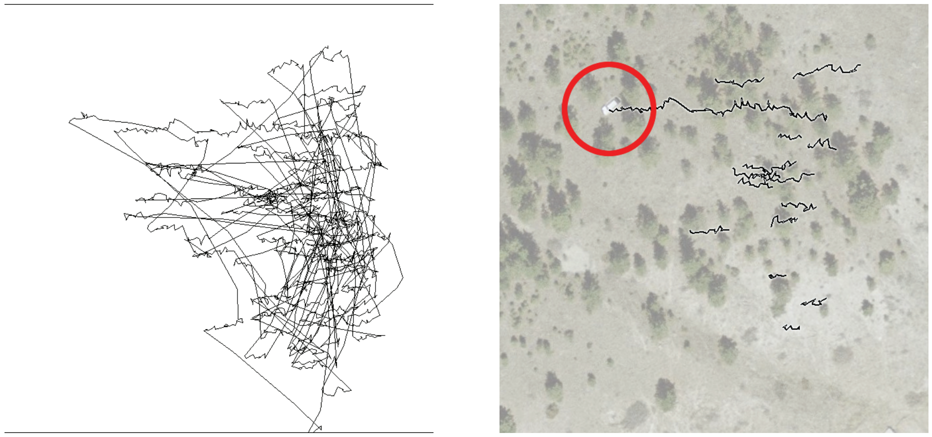


Figure 3. A gaze plot for "Eastward" travel (left); Hand segmented tracking episodes (right)

Participants reported little or no tendency to nausea when observing any of these sequences, though they are very short compared to a typical full search pattern. We note a reported tendency in this small sample for a preference to viewing the RSVP mode presentations over the video. In general participants reported the slow presentation rate (20) to be too slow, to the point of impatience, the fastest rate (120) leaving them concerned that they had missed targets. A larger sample and further analysis is required to confirm these observations.

Conclusions and Future Work

The observations from our pilot study suggest a number of proposals for the enhancement of the search and rescue (and related tasks) process. We consider the following possibilities to merit further investigation:

To monitor gaze data in video streams to determine if unreported targets nevertheless cause characteristic tracking or other gaze behaviours. Mello-Thoms *et al.* (2002) have, for instance, reported that gaze traces of trained radiologists inspecting mammograms for cancer clearly fixated on some clear disease indicators that were not subsequently reported. The exploitation of any such effects in a search and rescue task may help to avoid unfortunate false negatives, particularly during a long observation shift.

Slowing the moving image when the monitored eye-gaze tracks for an extended period to allow more careful inspection, or magnify the area around or otherwise highlight potential targets detected by gaze (we have simulated these effects, with encouraging results).

References

Cooper, K., de Bruijn, O., Spence, R. and Witkowski, M. (2006) A comparison of static and moving presentation modes for image collections, in: Celentano, A. (Ed.), *proc. Conf. on Advanced Visual Interfaces* (AVI 2006), ACM Press, pp. 381-388.

Goodrich, M.A., Morse, B.S., Gerhardt, D., Cooper, J.L., Quigley, M., Adams, J.A. and Humphrey, C. (2008) Supporting wilderness search and rescue using a camera-equipped mini UAV, *Journal of Field Robotics*, **25**-1, pp. 89-110.

McLain, T.W. and Beard, R.W. (2004) Unmanned air vehicle testbed for cooperative control experiments, *proc. 2004 American Control Conference*, IEEE, pp. 5327-5331.

Mello-Thoms C., Dunn S., Nodine C.F., Kundel H.L., and Weinstein S.P. (2002) The Perception of Breast Cancer: What Differentiates Missed from Reported Cancers in Mammography?, *Acad. Radiol.*, **9**-9, pp. 1004-1012.

Feasibility Study for the use of Eye-movements in Estimation of Answer Correctness

Minoru Nakayama
CRADLE
(The Center for R & D of Educational Technology)
Tokyo Institute of Technology
Ookayama, Meguro, Tokyo 152-8552
Japan
nakayama@cradle.titech.ac.jp

Yuko Hayashi
Human System Science
Tokyo Institute of Technology
Ookayama, Meguro, Tokyo 152-8552
Japan

Keywords
eye-movements, answer correctness, user's response estimation, discriminant analysis

Introduction

Many web sites often ask users to make responses such as "yes" or "no" to confirm the validity of the context of the explanation or the commercial message. These responses are required to use computer input devices, and are often of a troublesome nature because users have to understand the context and assess the significance. Eye-movements can be used to evaluate document relevance (Puolamaki, Salojarvi, Savia, Simola, & Kaski, 2005), and to estimate user's certainty (Underwood, 2005; Nakayama & Takahasi, 2006; Nakayama & Takahasi, 2008). The above results suggest that some features of eye-movements may indicate a viewer's interest or certainty for a simple word matching task. If the degree to which a statement was understood, or context reasoning were assessed using features of eye-movements, this kind of estimation can be applied to online tests, evaluating the impact of commercial messages, etc. The possibility of assessing a situation of contextual understanding and the dependence on the degree of the task difficulty should be evaluated in an experiment, however.
This paper addresses the feasibility of predicting user decision correctness of inferential tasks using eye-movements while the user makes "yes/no" responses based on his/her understanding some statements presented to him/her. The characteristics of features of eye-movements are also discussed.

Method

Experimental Task

The subjects were first asked to understand some definition statements which described locational relationships between two objects (Figure 1). Second, ten questions in statement form were given to determine the degree of understanding, which asked subjects to evaluate whether each question statement was "true" or "false" (Figure 2). The subject's responses were classified as correct or incorrect according to the statement because the question statements were generated using a rule of logic (Takita & Nakayama, 2004). Each statement was displayed one after another, and response to questions were made by clicking the left or right mouse button. No feedback was given for responses. Each presentation lasted 5 seconds for a definition statement, and 10 seconds for a question statement. When the subject responded to a question statement, the display moved to the next task. The difficulty of the definition statements was increased at tasks 3, 5 and 7. Five sets of data were created for each of three task levels, making 150 data responses for a total of 15 task sets. The subjects were 6 male university students ranging from 23 to 33 years of age.

Eye-movement measuring

During the experiment, subject's eye-movements were observed using a video-based eye tracker (nac:EMR-8NL). The task was displayed on a 20 inch LCD monitor positioned 60 cm from the subject. The subject rested his or her head on a chin rest and a small infra-red camera was positioned between the subject and the monitor, 40 cm from the subject. The tracker was calibrated at the beginning of the session, eye-movement was tracked on a 800 by 600 pixel screen at 60 Hz. The accuracy of the spatial resolution of this equipment is noted in the manufacturer's catalog as being a visual angle of 0.1 degrees. Eye-movement data was recorded on a PC as time course data, while subjects read and understood the text content of each statement. The tracking data was converted into visual angles according to the distance between the viewer and the display. Eye-movements were divided into saccades and gazes using a threshold of 40 degrees per second (Ebisawa & Sugiura, 1998).

Results

All responses which took longer than 4 seconds were classified as incorrect responses, because of the irregularity of taking such a long time to make a decision. The following features were summarized using eye-movement data from within the first 4 seconds. The accuracy of the responses across the number of statements is summarized in Figure 3. The accuracy decreases with the total number, and this suggests that the number of statements can be used to control the task difficulty, and that the task is the easiest for the first 3 statements and the hardest for the last 7 statements.

Difference in features of eye-movements

Some features of eye-movements were extracted by trials and errors, using metrics based on the saccade feature which was extracted. The mean of saccade features, saccade lengths in degrees and saccade differences in degrees per second are summarized in Figures 4 and 5. All mean values for correct responses are higher than are the ones for incorrect responses, except statement 3, where the values are almost equal. To determine the differences between correct and incorrect responses, two-way within-subject ANOVA was conducted for saccade length and saccade difference respectively. The factor of answer correctness is significant for both saccade length ($F(1,20)=27.8$, $p < 0.01$) and saccade difference ($F(1,20)=24.7$, $p < 0.01$). The task difficulty factor is not significant for both. This suggests that the transition of eye-movements for incorrect responses is smaller than is the one for correct responses, and this phenomenon is independent of the task difficulty.

The results of saccade frequency duration and saccade time are summarized in Figures 6 and 7 respectively. In these metrics, means of incorrect responses were higher than were means for correct responses. According to the results of two-way within-subject ANOVA, the answer correctness factor is significant for both saccade frequency ($F(1,20)=21.7$, $p < 0.01$) and saccade time ($F(1,20)=67.6$, $p < 0.01$), though the task difficulty factor is not significant.

These results indicate that when subject's responses are correct, the saccade length is long and the saccade time is short.

Estimation of Answer Correctness

According to the above results, the features of eye-movements are significantly different between correct and incorrect responses. Here, a linear discriminant analysis is introduced to examine the possibility of estimating answer correctness using features of eye-movements without behavioral metrics. Linear discriminant analysis was conducted for each level of task difficulty. The discriminant accuracy was calculated using a cross validation procedure. Discriminant functions were created for each level, using all features except the automatic selection of variables by the software program. In the results, discrimination performance for statement 3 is summarized in Table 1. In this table, "correct-correct" and "incorrect-incorrect" predictions are accurately estimated, and the accuracy is 68.7%. According to the binomial distribution (Shiba & Watabe, 1976), significant classification may be at over 65%, so this classification is above the level of chance. The functions and accuracy of the number of statements are summarized in Table 2. All features have been standardized to allow comparison of each variable's contribution. In a comparison of the coefficients of each feature, the value of the saccade time is relatively larger than it is for other features. This means that the saccade time is the largest component of the decision making estimation function. The order of the values of saccade length and saccade frequency change with task difficulty as the number of statements increases. More detailed analysis of eye-movement features and improvement of the prediction accuracy will be a subject of our further study.

Conclusion

The feasibility of predicting the correctness of user's decision using eye-movements was examined while users alternatively made "yes" and "no" responses to questions. Some features of saccadic eye-movements were extracted to indicate the decision processes. The results show there are significant differences between correct and incorrect responses, and that the level of difficulty of the task did not affect these deviations. To determine the performance of estimating user's decision making correctness, a discriminant analysis was conducted. The accuracy of the discrimination is significant, and the contributions of the features were evaluated for their performance. These results provide evidence that the features of eye-movements during reading statements can be used to estimate answer correctness.

References

Ebisawa, Y., & Sugiura, M. (1998). Influences of target and fixation point conditions on characteristics of visually guided voluntary saccade. *The Journal of the Institute of Image Information and Television Engineers*, *52*(11), 1730–1737.

Nakayama, M., & Takahasi, Y. (2006). An estimation of certainty for multiple choice responses using eye-movement. In *Electronic proceedings of the 2nd conference on communication by gaze interaction – cogain 2006: Gazing into the future* (pp. 67–72).

Nakayama, M., & Takahasi, Y. (2008). Estimation of certainty for responses to multiple-choice questionnaires using eye movements. *ACM TOMCCAP*, *5*(2). (Artile 14)

Puolamaki, Y., Salojarvi, J., Savia, E., Simola, J., & Kaski, S. (2005). Combining eye movements and collaborative filtering for proactive information retrieval. In A. Heikkil, A. Pietik, & O. Silven (Eds.), *Proceeding of acm-sigir 2005* (pp. 145–153). New York, USA: ACM Press.

Shiba, Y., & Watabe, H. (1976). *Toukeiteki houhou ii (stastistical method ii)*. Tokyo, Japan: Shinyosha.

Takita, W., & Nakayama, M. (2004). Influence of audio-visual presentation style for sentences on their understanding. *IEICE Technical Report*, *104*(132), 17–22.

Underwood, G. (2005). Eye fixations on pictures of natural scenes: Getting the gist and identifying the components. In G. Underwood (Ed.), *Cognitive processes in eye guidance.* Oxford, UK: Oxford University Press.

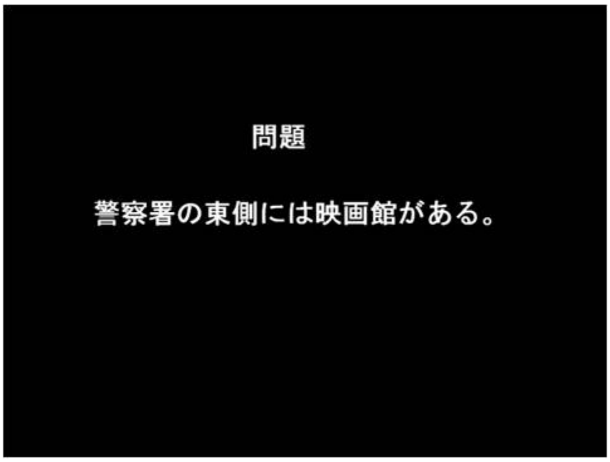


Figure 1. A sample of a definition statement: "A theater is located on the east side of a police station."

Figure 2. A sample of a question statement: "There is a post office on the south side of the theater."

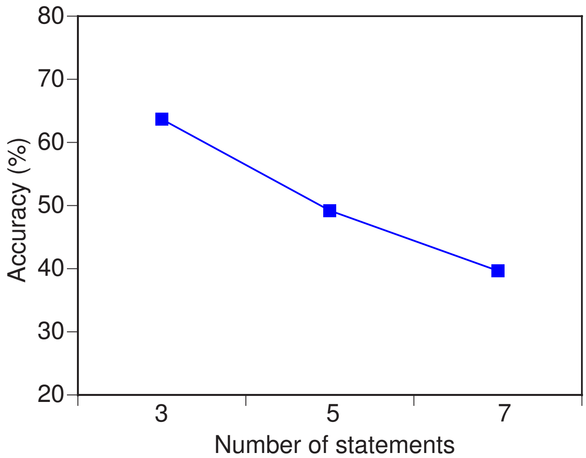


Figure 3. Response accuracy across the number of statements.

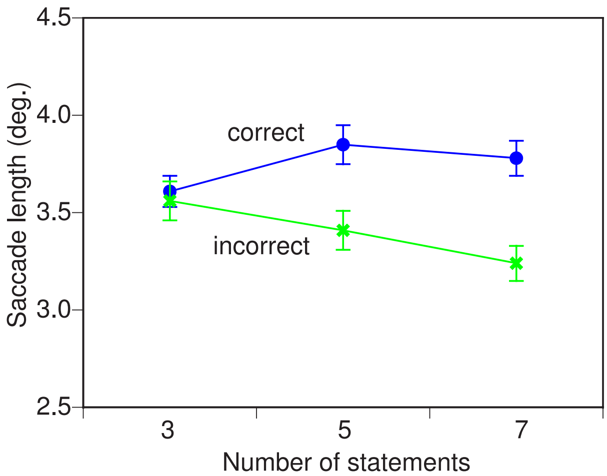


Figure 4. Saccade length

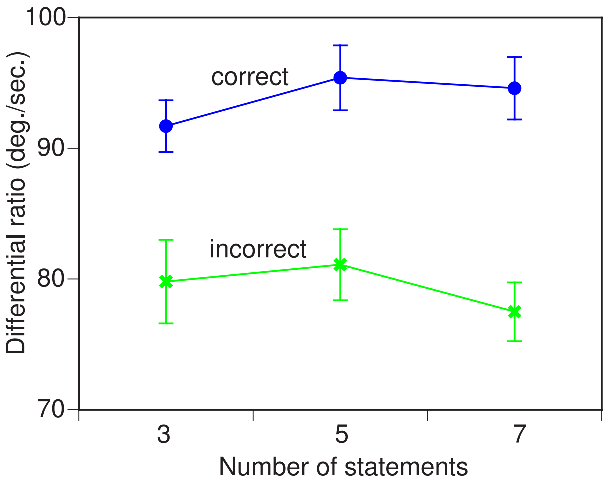


Figure 5. Difference in saccade length

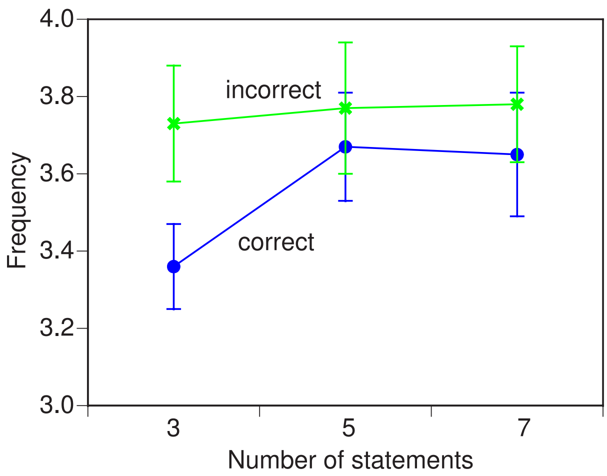


Figure 6. Saccade frequency

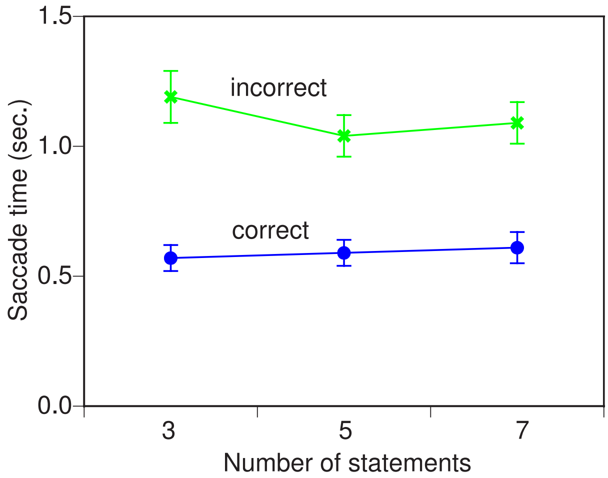


Figure 7. Saccade time

Table 1. Discriminant result(No. of statements=3).

Subject's response	Prediction Correct	Prediction Incorrect	Total
Correct	156	35	191
Incorrect	59	50	109
Total	225	75	300

Table 2. Discriminant Functions and Accuracies.

No. of statements	Discriminant function	Accuracy (%)
3	$0.75 * f_1 - 0.04 * f_2 + 0.28 * f_3 - 2.02 * f_4 + 1.56$	68.7
5	$0.77 * f_1 - 0.02 * f_2 + 0.52 * f_3 - 2.01 * f_4 - 1.15$	69.6
7	$0.52 * f_1 + 0.004 * f_2 + 0.55 * f_3 - 1.22 * f_4 - 3.20$	68.3

f_1: saccade length, f_2: saccade difference
f_3: saccade frequency, f_4: saccade time

Eye Tracker Connectivity: Alternatives to ETU-Driver

Gintautas Daunys
Siauliai University
Vilniaus str. 141, Siauliai, LT-76353, Lithuania
g.daunys@tf.su.lt
+370 41 595836

Vytautas Vysniauskas
Siauliai University
Vilniaus str. 141, Siauliai, LT-76353, Lithuania
vytautas.vysniauskas@tf.su.lt
+370 41 5958365

Keywords
Eye tracker, Eye tracker connectivity, Gaze tracking applications, Interaction by gaze, Interprocess communications

Introduction

It is evident that standardisation of eye tracker connectivity to applications is important for software designers, developers of software applications and end-users. The lack of common standard leads to situation, when eye tracking hardware and software from different manufacturers operate only on their own standard that is not compatible with other manufacturers' operating standards. Then software from one developer cannot operate with other eye tracker. This limits the availability of applications for end-user, which has eye tracker from specific brand. Detailed review of situation in 2005 can be found in deliverable of FP6 project COGAIN (Bates et al., 2005).

One approach to solve this situation is ETU-Driver (Bates and Spakov, 2006), which can be downloaded for free from http:// www.cs.uta.fi/~oleg/etud.html. It is based on Microsoft COM technology. ETU-Driver has modules, which are using API's supplied by manufactures, to obtain data from eye tracker and convert it to common format. In this manner different applications can be implemented with ETU-Driver and can receive data from whichever eye tracker that has an interface with that driver.

On one hand, the driver is an example of good practise in the area. On another hand, ETU-Driver doesn't seem to be permanent solution of the eye tracker connectivity issue. It is confirmed by the next reasons:

- Data from eye tracker reaches application via some steps that are wasting resources of computing.
- There are no recommendations how to build eye tracker software for simplest data transfer.
- It is evident that for embedinging new eye tracker a new module for that device must be build.
- COM technology is obsolete and programmer must input additional efforts to program.
- Maintenance of driver in one site (TAUCHI unit of Tampere University) also is an issue.

A solution can be find in a standardisation of a gaze data transfer. In the next section techological issues will be discussed. It will be indicated that problem could not be solved with Plug and Play technology, widely used for other hardware. Furthermore, the arhitecture of interaction will be discussed. Finally, details of implementation and results will be provided.

Technological Background

One opinion is to use Plug and Play (PnP) technology to connect eye tracker to a computer. The issue is that in other PnP input devices data is produced by hardware and driver software, which operates in kernel mode, facilitates data transfer to applications. When a new hardware device is connected to bus and it transfers via bus its identification code to operation system (OS) and by this code OS installs driver for the device from local disk or from the Internet. Each vendor uses the unique code for every device. Commonly PnP is used for widely using devices as data storage, network, display and other. PnP is used as technology to load the correct driver for particular hardware and function. However, it is impossible to use or difficult to implement PnP with different eye trackers. In the case of eye trackers data is produced by software. Widely used video based eye tracking systems need powerful computations. Data are produced by user mode software. So PnP currently is not solution for eye trackers.

The problem of gaze tracker connectivity to applications can be assigned to issue of interprocess communications (IPC). Eye tracking software and applications are running in the same computer as different processes. Every application runs in protected mode, it means that one process can not directly write or read to other application memory.

There are many methods for interprocess communication. A partial list of them is next: file, socket, pipe, named pipe, semaphore, shared memory, message passing, memory mapped file, mailbox. The method could be selected by specific requirements for data transfers. Requirements can be listed as:

- fast and errorless data transfers;
- minimal resources requirement;
- asynchronous transfers (it means that eye tracker and application can communicate by events, rather then sending request and waiting for answer);
- connectivity of software components written in different languages;
- connectivity of components operating in different computers on LAN. It is required, when eye tracker software runs on other computer then application. When this requirement conflicting with speed requirement it could be ignored for choosing method for one computer systems.

Also it is worthful to consider as a mean of interaction between applications hooking. Hooking allows to filter Windows messages or send new ones. Hooking creates background for mouse actions emulation. Mouse emulation is very useful tool for interaction with applications by gaze because all most Windows application responds to mouse events.

Solution

Simplest solution for connectivity is mouse events emulation in eye tracker software. Mouse emulation is already used by most of eye tracking manufactures (Bates et al., 2005). The issue is that mouse events could lead to lose of some gaze information obtained by eye tracker. So it is desirable that it will be supplied more data and emulation was implemented in other software component.

The suggestion leads to architecture shown in Figure 1. From first sight it seems the same as in case of ETU-Driver. All good practice obtained from ETU-Driver could be used in new architecture. The difference is that architecture is more simple and easy understandable. Here between eye tracker and application is intermediate component, which is called Interaction Broker (further broker). Its role is to establish communication between eye tracker and any application.

Operation is such as selected method of IPC eye tracker transfers data to the broker. The broker transfers data to application, as some method of IPC. But the broker is Windows application, which could be reconfigured.

The broker communication methods with applications can be different and perhaps not a single at the same time. The authors give preference to hooking of mouse events.

The requirements for broker component:

- Standardized interface with eye tracker;
- Standardized interface with applications;
- Standardized feature by which eye tracker or applications can obtain handle to broker window (data);
- Standardized way of negotiations of broker with other software components.
- Possibility to discover available eye tracker and to launch it.

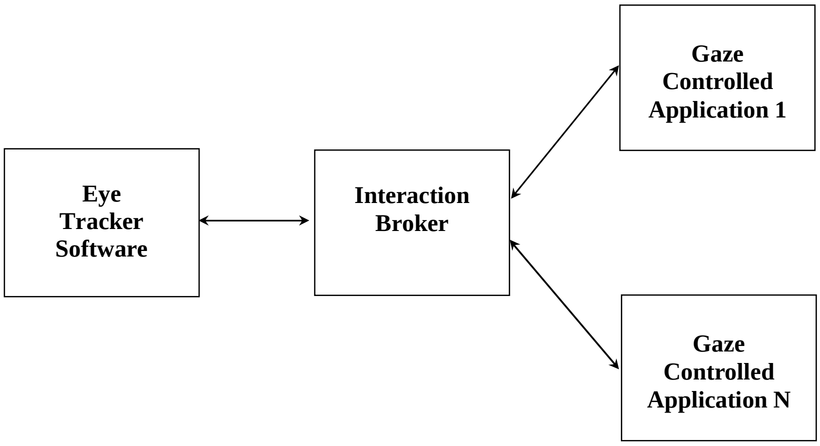


Figure 1. Simplified structure of connectivity architecture

Preliminary Results

To find good solution there is need to experimentally test some proposed architecture implementations. The first prototype was build as described below.

The broker was written in C# using Microsoft Visual C# 2008 Express edition with service pack 1 and its code is managed. The window of the broker has title “COGAIN Broker” and it helps to find them from eye tracker software or applications for negotiations. The Broker launches mouse messages hooking procedure packed to DLL (it is requirement from Microsoft MSDN documentation). It receives data from eye tracker by Windows message WM_COPYDATA. Then broker transfer the data to hooking procedure, which generates mouse events.

The eye tracker software is written in unmanaged C++ code using Microsoft Visual Studio 2003. It finds a handle of the broker window by its name and sends WM_COPYDATA messages (see the code in Figure 2).

The gaze controlled application also is written in C#. It has an implemented handler for WM_MOUSEMOVE and WM_COPYDATA messages. It saves obtained data and a time moment of a data delivery to a file for off-line analysis. The eye tracker software also sends data with a time stamp.

Experimentally there were compared the time moment from the time stamp and the time moment of the message delivery. This allowed to measure time delay of data transfer from the eye tracker to the application. Results show that there were no lost data packets. The mean time delay is approximately 7 milliseconds, standard deviation is about 3 milliseconds.

In future the plans are to try different methods of interprocess communication and measure its performance.

```
/* Declaration  of variables */
HWND  hwnd; //Handle of COGAIN Broker window
HWND  het;  //Handle of Eye tracker window
COPYDATASTRUCT cds;     // Data structure for sending with message
GAZEDATA gd;      // Structure for gaze data
/* Finding of the handles of running COGAIN Broker window and own window */
hwnd = ::FindWindow( NULL, "COGAIN Broker" );
het = (HWND)this->GetSafeHwnd(),

/* When gaze data are ready */
// Filling structure variable gd with gaze data

// Filling the COPYDATA structure
cds.dwData = 1;               // message identifier
cds.cbData = sizeof(gd);      // size of data to send
cds.lpData = &gd;             // address of gaze data structure
//Sending the message with gaze data
if( hwnd != NULL )
::SendMessage( hwnd, WM_COPYDATA, (WPARAM)het, (LPARAM) (LPVOID) &cds );
```

Figure 2. An example code how the eye tracker sends data to the broker

Conclusion

Connectivity of eye tracker can be implemented by the methods of interprocess communication or remoting. The flexible architecture of communication was developped. The architecture was implemented using C# and C++ programming languages, Windows messaging, and hooking of mouse events. The experimental prototype indicated that there was an acceptable time delay of the gaze data transfer.

References

Bates, R., Istance, H., Oosthuizen, L., and Majaranta, P. (2005) D2.1 Survey of De-Facto Standards in Eye Tracking. COGAIN, IST-2003-511598: Deliverable 2.1. Available at http://www.cogain.org/results/reports/COGAIN-D2.pdf

Bates, R. and Spakov, O. (2006) D2.3 Implementation of COGAIN Gaze Tracking Standards. COGAIN, IST-2003-511598: Deliverable 2.3. Available at http://www.cogain.org/results/reports/COGAIN-D2.3.pdf

TEDA, a SW tool supporting customization of eye tracking algorithms

Petr Novák, Pavel Moc
Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University
novakpe@labe.felk.cvut.cz, pavel.moc@amedis.cz

Olga Štěpánková, M. Uller, Lenka Nováková
Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University
{step, uller, novakova}@labe.felk.cvut.cz

Keywords
Eye position detection, algorithms for detection, evaluation, database, result visualisation

Introduction

Most gaze-based communication solutions share the same initial phase when being introduced to a new client: the tool has to be set up for the specific user. This process corresponds to such a modification of the parameters of the used eye-tracking algorithm that suits best the needs of the considered client, i.e. it minimizes number of recognition errors [Daw-Tung, L. and Chen-Ming, 2004]. This quality is influenced by a number of features that are related to technical characteristics of the processed image (brightness, application of IR, etc.) as well as to specific properties of the user's face (colour of iris, lashes and brow, maximal open position of eyelid, etc.). It is well known that sensitivity of various algorithms towards these features varies a lot. Finding the proper functional parameter setting for the specific client is sometimes really difficult and it needs lot of experience. Often, one has to try number of settings before an acceptable solution is found [Fejtová et al., Li et al., Rasmusson et al.]]. To support this experimental approach we have developed a SW tool **TEDA** (**T**esting of **E**ye **D**etection **A**lgorithms) that helps to ensure the necessary experiments offline in the quiet space of computer laboratory and come back to the customer with a reasonable solution. TEDA is a DB solution offering reliable means for comparison and evaluation of available or new algorithms and their setting based on real life pictures of the considered customer complemented by information about true *reference position* of the eye (or more precisely centre of its pupil).

TEDA is an empty DB system complemented by several useful services. First of all, it is intended for storing images or videos of the scene to be recognized by the used eye-tracking algorithm and taken with the considered customer as well as algorithms (and their parameter settings). Second, as soon as the database is filled in there are available following additional services supporting upper mentioned experimental approach:

- The user can conduct his/her *own experiments*, i.e. apply to the stored images his/her algorithm(s) and save the obtained results (the calculated position of pupil centre and the corresponding processing time of the used algorithm) for each picture.
- The system provides *graphical review* of the obtained results related to the difference between the reference point and the point identified by the used algorithm(s) provided the used images are annotated (there is identified the reference point, i.e. true position of the pupil in the considered picture). There is also provided a graphical output that helps to identify those images that are hard for the applied algorithm, because the algorithm fails to find the reference point properly (see radar picture at the Fig.4).
- If the images contained in the database are not annotated, the user (a human expert) can *manually annotate* them in an interactive mode when the user is expected to point by the cursor to the pupil centre

on each presented picture. This position is stored by the system as the referential position for the considered image.

Structure of the TEDA system

The TEDA tool is build using an open-source solution, namely **Postgres Plus** database server and MS NET Framework [Agarwal and Huddleston]. Its functionalities and implementation is described in detail in [Novak et al.] and it can be dowloaded upon request from NIT.FELK.CVUT.CZ/DRUPAL/SOFTWARE. TEDA is composed of five main parts (Fig. 1). The first part is responsible for loading input data, namely the images or video sequences with an eye (or its movement) in any of the frequently used formats (BMP. JPG, MPEG, AVI) and complemented by additional data (for example intensity of surrounding light) into the database (see Fig. 2). This part also serves as a simple editor where one can specify the reference values, e.g. the reference position of the eye or diameter of the pupil in the input image or each image in the video sequence. The second part picks up the input data from the database, provides them to the main core application and saves the results back into the database. The part (3) represents the core of the program and is responsible for all data transmissions necessary for the processing. It hands on data from the part (2) as input to the algorithms through the appropriate interface (4), takes results provided by the algorithm and transfers them to the graphic module (5). The part (4) represents an interface that converts data provided by the DB into the format or structure required by the processing algorithms and back. The last part (5) is a graphical unit that can show the stored output of algorithms in many graphical ways.

The current version stores in the database a limited set of features of the considered images, namely surrounding light (lux), IR light applied on eye, reference position of eye, detected position of eye, detected diameter of pupil and probability of output data from algorithm. Other features can be added easily.

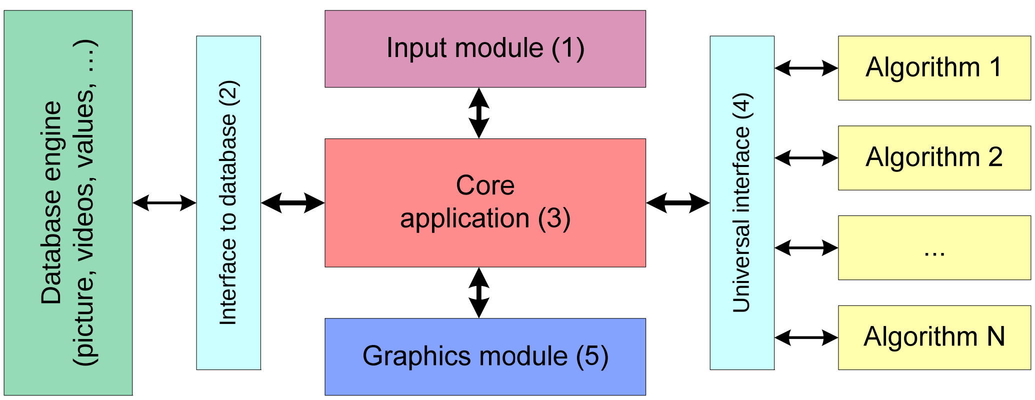


Figure 1. Overall structure of the designed TEDA system.

Each algorithm is represented by its program library (e.g. DLL) and it has to follow the universal interface specifying uniform data types and structures that allow sending the same image data to any type of image processing algorithm. Any already existing algorithm can be complemented by a so-called wrapper so that the upper mentioned requirements are ensured.

Using the system

First, it is necessary to load the input data into the database (images and video sequences) and complement them by the reference values of eye position (and diameter of pupil if needed). Following figure shows the dialog for loading input data and creating reference values as well as a window showing already loaded data.

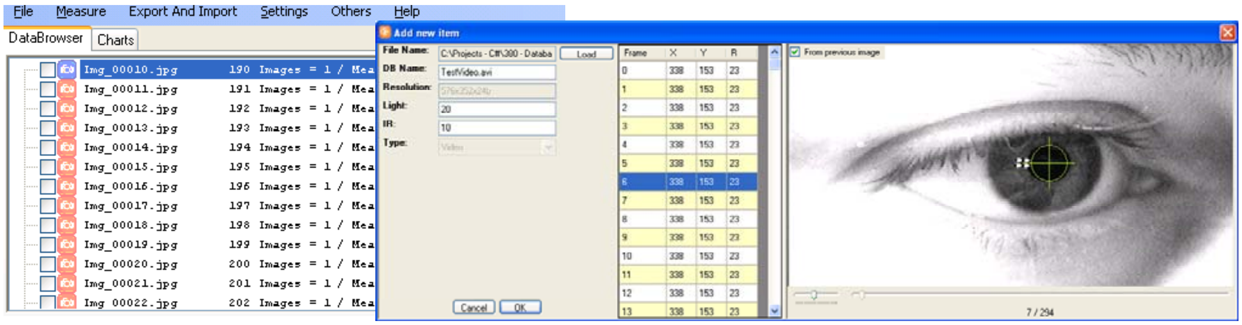


Figure 2. Screenshots: the input data browser, loading data into the database.

Next, it is possible to select what input data will be analyzed by which algorithm (from the current offer) and start processing. One can select images and video sequences at one time. Application can send picture after picture to algorithm independent of its source position (stand-alone image or video sequence). One can also set up rotation or flip of picture if this is requested by the used algorithm.

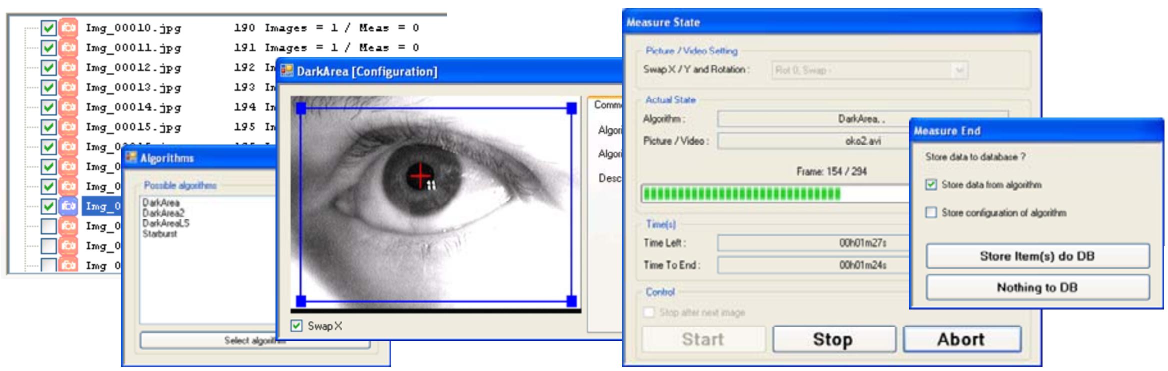


Figure 3. Screenshots showing the application process: data and algorithm selection, algorithm configuration, process information, storing the results.

Evaluation of data

All results provided by the algorithm can be stored into the database. Under each particular image or video sequence, there is list of applied algorithms (and their modifications) as well as the corresponding results. To review the results one can just select the pictures or video sequences of interest and the program will display graphs (Fig. 4) showing the difference between the reference eye position and the position detected by the algorithm (either as the straight distance or for each axis separately). Moreover, the tool offers also other useful types of graphs, e.g. the 'radar graph', where the image numbers are on the perimeter and the distance from the centre corresponds to the difference between the reference value and the detected position.

To ensure more sophisticated analysis of the generated results, it is possible to export all the selected data (input and output) into a simple text file so that they can be downloaded into any data analysis tool of choice (e.g. Excel or Matlab) for further analysis.

Figure 4 illustrates one of the natural ways for usage of our tool: the data of a single customer are analyzed by one image processing algorithm under several parameter settings. Numerous experiments can be easily ensured before the best choice is selected. It is worth to notice, that the conducted experiment shows that the lowest threshold does not have to lead to the best results (see the results corresponding to the frames 35-40 and 95-100 at the Fig.4).

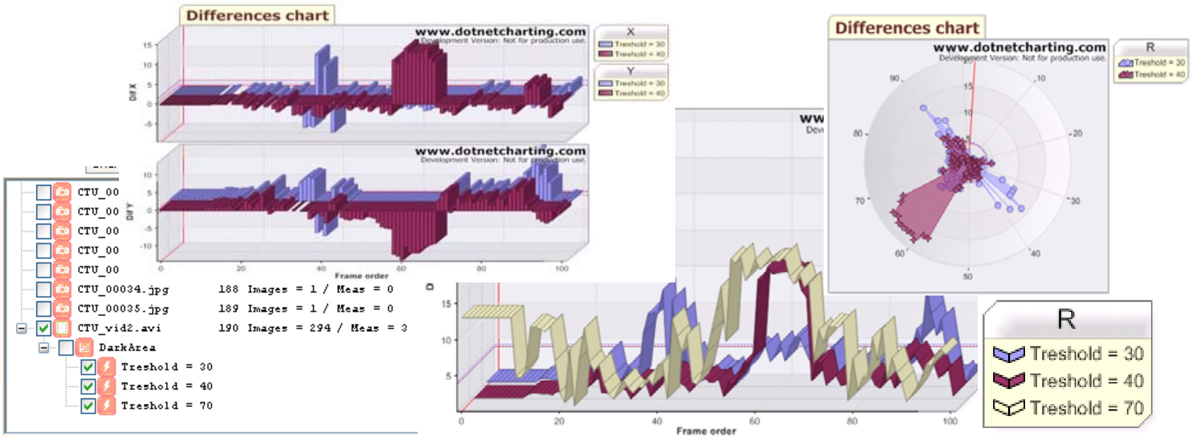


Figure 4. Graphical presentation of the results provided for the same images processed by 3 different algorithms.

Conclusions

TEDA tool offers a user friendly environment for conducting different experiments and it is available upon request from the first author. The interested user can *download an empty DB*, fill it with his/her own images and take advantage of all functionalities described in this contribution. The structure of the DB is not absolutely fixed - the current version offers a possibility to extend the set of basic features characterizing properties of the considered images. An important advantage of our solution is that it supports sharing some data (real life images) with other interested parties: the completed DB can be uploaded back to the dedicated server under its own name and thus contribute to creation of a dedicated data repository. Existence of such a data repository based on real life pictures complemented by information about true *reference position* of the eye (or more precisely centre of its pupil) will provide reliable means for comparison and evaluation of available or new image processing algorithms to be used in the context of gaze based interaction.

The presented research and development has been partially supported by the EU grant IST-2003-511598 COGAIN (Communication by Gaze Interaction) and the Czech grant No. MSM6840770012 "Transdisciplinary Research in Biomedical Engineering II".

References

Agarwal, V., V., and Huddleston, J., Beginning C# 2008 Databases From Novice to Professional, Apress2008

Daw-Tung, L. and Chen-Ming, Y.(2004) Real-time eye detection using face-circle fitting and dark-pixel filtering . *Multimedia and Expo, 2004.* ICME '04, Int. Conference on, Vol. 2, pp. 1167-1170 Vol.2.

Fejtová, M., Fejt, J., Novák P., Štepánková, O.(2006) Eye as an Actuator. In Computers Helping People with Special Needs, *Berlin: Springer*, vol. 4061, pp. 954-961.

Li, D., Winfield, D., Parkhurst, D. J. (2005). Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *Proceedings of the IEEE Vision for Human-Computer Interaction* Workshop at CVPR, 1-8.

Novák, P., Moc, P., Štěpánková, O., Guerrero, G., Cabeza, R., Villanueva, A.: D5.7 Open Unformation Source for Eye Tracting, to be dowloaded from www.cogain.org, 2009

Rasmusson, D., Chappell, R., and Trego M. (1999) QUICK GLANCE: EYE-TRACKING ACCESS TO THE WINDOWS95, http://www.dinf.ne.jp/doc/english/Us_Eu/conf/csun_99/session0153.html

Multimodal Gaze-Based Interaction

Sandra Trösterer
Team mEyeInt, Chair of Human-Machine Systems
Technische Universität Berlin
Franklinstr. 28/29, 10587 Berlin
str@mms.tu-berlin.de

Jeronimo Dzaack
Team mEyeInt, Chair of Human-Machine Systems
Technische Universität Berlin
Franklinstr. 28/29, 10587 Berlin
jdz@mms.tu-berlin.de

Keywords
Multimodal interaction, gaze tracking, tool support, evaluation, human-computer interaction

Introduction

Gaze-based interaction allows interacting with a computer system in a contact-free and straightforward manner. It is used for all-embracing interaction, i.e. for navigation and manipulation tasks. Nevertheless, this kind of interaction has its drawbacks. One of the main problems is that the gaze is always *online*, although "ideally, the interface should act on the user's eye input when he wants it to and let him just look around when that's what he wants" (Jacob, 1990, p. 13). To overcome this drawback several approaches exist, e.g. to use blinks or an appropriate dwell time in order to activate elements on the screen. However, these solutions have their limits, e.g. when using a specific dwell time, there is always the danger to activate elements accidentally or the need to stare at an element longer than normally necessary. Especially when the gaze shall be used not only to activate an element, but also to manipulate it (e.g. rotate the element), the situation becomes even trickier. We want to use gaze-based interaction in everyday life, for computer games or for desktop applications. We conclude that the gaze is a good means to navigate on the screen, but when it comes to manipulation tasks, gaze alone has its limits. The question is, how can we support gaze-based interaction? Several approaches exist that integrate different interaction modalities and software architectures (e.g. Kumar et al., 2007). However, most of these approaches are designed for specific research questions and applications and thus are limited to a small set of modalities. Our approach is to extend gaze-based interaction by an unlimited set of modalities such as gestures, speech, facial expressions or even brain activities to multimodal gaze-based interaction. This could help to overcome the drawbacks of a pure gaze-based interaction, but would still provide the advantages of contact-free interaction. To design and evaluate supported gaze-based interaction from a psychological and ergonomic point of view we developed a software framework. In this paper we present the conceptual aspects and a validation study of iCOMMIC (integrated Controller for Multimodal Interaction; Dzaack et al., 2009), our tool and method to combine, integrate and evaluate different input and output modalities in order to interact with a computer system in a multimodal way.

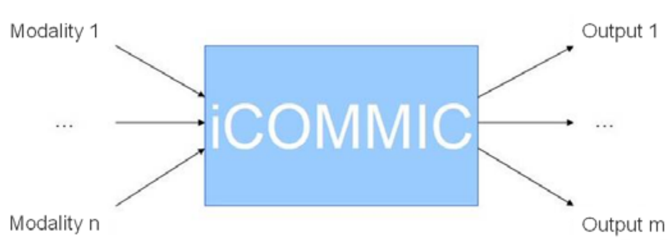


Figure 1: Simplified concept of iCOMMIC.

iCOMMIC – integrated COntroller for MultiModal InteraCtion

iCOMMIC is an integrative software framework for multimodal human-computer interaction. It enables (1) to integrate different input and output modalities, (2) to combine these modalities in any way and (3) to record the user interaction data in an appropriate way for psychological investigations. For example it allows to connect the output of an eyetracker to the cursor or an external light bulb to signal a specific behaviour as well as a gesture input to a languge generation system. Figure 1 shows a simplified concept of iCOMMIC.

To allow this multidimensional mapping, the main programm of iCOMMIC is designed in a layered architecture that handles the management, the mapping and the exchange of incoming and outgoing information. The data exchange between the main programm and the input and output devices is based on a specific protocol that is adjustable by the user. In this context we engineered a wrapper concept, that enables the encapsulation of data from multiple devices in a controlled data stream and allows in this way the same processing and handling of data from different devices. The data flow of iCOMMIC is shown in Figure 2. Currently iCOMMIC supports gaze, mouse and keyboard as input modalities and on-screen display and mouse as output devices.

The tool was developed in an iterative software developmement process at the Chair of Human-Machine Systems. During the design and engineering process requirements, which were deduced from several expert interviews, were taken into account.

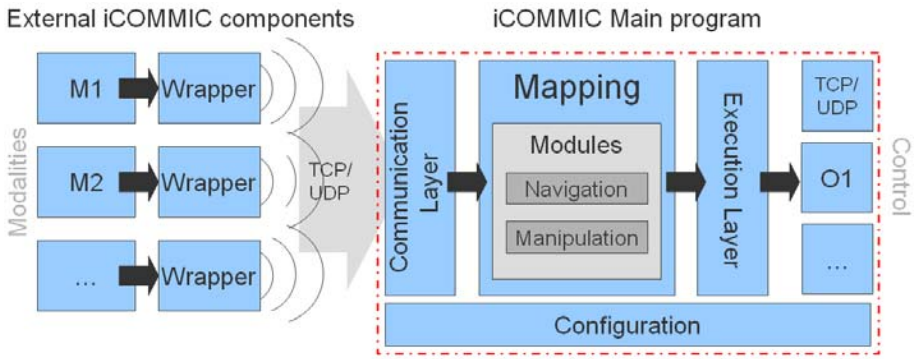


Figure 2. Data flow in iCOMMIC: connection of different modalities (M) (left side), mapping of modalities to the output devices (O) of the computer (middle), execution of control commands (right side).

1. Capturing and connection of different modalities

Every modality and its device provide individual information. Regarding the visual modality, an eye-tracker provides e.g. information about gaze direction. In order to process information from diffent devices in an equal way and to use it for the interaction, the information is captured by wrappers. These are components, that can be implemented individually and in a customized manner for each input device. The wrapper encapsulates the modality-specific chararacteristics and processes the raw data to generate a standardized data stream, that can be interpreted and processed by the main program. The advantage of the wrapper concept is, that different interaction techniques and filter functions can already be realized in the wrapper. For example for gaze data it would be possible to implement different filter functions that differentiate between intended blinks or normal eye lid activity or between saccades and fixations. Finally the wrapper is connected to the main program through the communication layer of iCOMMIC by network protocols (i.e. Transmission Control Protocol (TCP) or User Datagram Protocol (UDP)). In this layer the data stream of a wrapper is received and processed, as soon as the wrapper connects to the main system.

2. Mapping process

The mapping process is the core process of iCOMMIC. This process makes it possible to combine different input and output devices with each other. The mapping is implemented in two modules that enable the universal control of a computer: (1) a navigation and (2) a manipulation module. That means, that all navigation operations like scrolling or mouse movements are processed through the navigation module and all selection and manipulation operations are processed through the manipulation module. The main idea is, that it is possible to switch-on or switch-off the navigation and manipulation function of an input device in a user-defined way. For example a user can use the data gained from an eye-tracker to navigate on the screen, but use a mouse-click or a gesture to manipulate the element, at which a user looks at. In short, the mapping process works as follows: for every modality that successfully connects to the main program, an internal process is started, that receives and prepares the data stream for the mapping process. In a first step the data is controlled for syntactic correctness and the needed processing modules are identified and loaded, according to the configuration settings of the user. Then the methods of the navigation and the manipulation modules are accessed and based on the delivered parameters control commands for the connected output devices are executed. For example, if an eye blink is sent to the main programm this event can be transformed into a mouse click or a sound output as specified in the configuration by the user. Due to the modular set-up of the mapping process, an independent development and extension of the central mapping modules is possible.

For further data analysis, all data streams of the connected modalities are recorded and stored in separate standard formatted data files. The raw data is enhanced with informations about e.g. the name of the modality, timestamps and synchronisation information.

3. Execution layer

The execution layer makes sure, that the control commands, that were generated in the mapping process, are executed by the output devices. It tranfers the control commands of the mapping process to a command that is processible by the particular computer device. This layer is a central entity that enables the connection of different output devices. To support the multimodal interaction all commands and data are provided in a specific network protocol comparable to the wrapper protocol.

Verification Study

In order to show the applicability of iCOMMIC for psychological studies and for everyday use, and to show the correctness of the logged data, we conducted a verification study with 12 subjects. Three kinds of interaction were investigated using iCOMMIC: (1) dwell time based gaze-interaction (250 vs. 500 ms dwell time), (2) mouse interaction and a (3) combination of gaze-based and mouse interaction, i.e. the gaze was used to navigate and the mouseclick was used for manipulation. The experiment was realized as a randomized within-design, i.e. all subjects had to do all kinds of interactions. The subject was seated in front of a 19" monitor and had to perform a simple searching task (see Figure 3). The goal was to find the target *X* in one of the corners of the display (*O*s were used as distractors) and to confirm its position with the corresponding button on the screen. This had to be done within a time interval of 4000 ms. After a button-press or time-out, a fixation cross was presented for 1000 ms before the next trial began. For each kind of interaction, the subject performed 20 training trials, followed by 40 main trials. Gaze-data was recorded with the remote eye-tracker IntelliGaze from Alea Technologies. Apart from the data recorded by iCOMMIC, reaction times and errors were investigated. Generally the configuration of the different kinds of interaction could be easy set in iCOMMIC. All data was recorded in the specified way and was synchronized accurately. Currently we are analyzing the gained data regarding the efficiency of the different interactions. The results indicate that the mouse interaction worked best regarding mean reaction time (1117 ms) and error rate (1,04 % of all trials were incorrect due to time-out or false-response errors). The mean reaction time of the mouse-based interaction is in each case significantly lower than the reaction time for the 250 ms (1320 ms) and 500 ms

(1606 ms) dwell time based gaze-interaction and the combination of gaze and mouse (1471 ms, all *p*<.01). No significant differences in reaction time could be found between the combined and the dwell time based interactions, although naturally in the 250 ms condition the reaction times are significantly lower (*p*<.01) than in the 500 ms condition. Regarding the error rates of the gaze-assisted interaction forms, we found a 4,17 % error rate in the 250 ms condition, 3,33 % in the 500 ms condition and 1,46 % in the combination condition. We conclude, that the combination of gaze and mouse seems to be the most efficient interaction form apart from the mouse interaction. The reaction time was similar to the dwell time based conditions, but with the advantage that the error rate was much lower. We believe that the high efficiency of the mouse interaction is due to the fact, that it is a highly trained interaction form and that in our experimental setting the mouse cursor had not to be moved over long distances. Further data analysis, that will focus on latencies and fixation durations, will give us further hints.

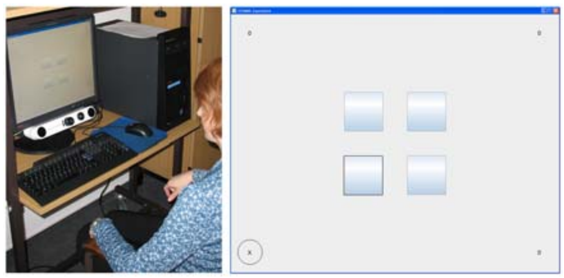


Figure 3. Experimental setting and searching task (target *X* and the corresponding button are marked).

Conclusion

iCOMMIC is a tool that supports the development and evaluation of multimodal gaze-based interaction. Due to its modular set-up it can be simply extended and enhanced by further input or output devices. Currently we are working on the realization of a gesture recognition tool and a wrapper, that connects this modality with iCOMMIC. The integration of speech and electroencephalography data (EEG-data) are further steps. Future research will focus on the investigation and comparison of different kinds of multimodal gaze-based interaction as a basis for the transfer of this kind of interaction to daily life applications. To facilitate psychological investigations, iCOMMIC is currently extended by a profile functionality, i.e. specific configuration settings can be stored as profile and simply loaded into the system on demand. We believe that our approach provides a base for the realization of new interaction concepts, that enable the user to interact with a computer in a more natural and efficient way, thereby using the advantages of the different modalities a human possesses.

References

Dzaack, J., Trösterer, S., Nicolai, T. and Rötting, M. (in prep. 2009) iCOMMIC: Multimodal Interaction in Computing Systems. *Proceedings of the 17th World Congress of the International Ergonomics Association*, Peking.

Jacob, R. J. K (1990) What you look at is what you get: Eye movement-based interaction techniques. *Human Factors in Computing Systems: CHI ’90 Conference Proceedings*. ACM Press, pp. 11-18.

Kumar, M., Paepcke, A. and Winograd, T. (2007) EyePoint: Practical Pointing and Selection Using Gaze and Keyboard. *Proceedings of ACM CHI 2007 Conference on Human Factors in Computing Systems*. ACM Press, pp. 421-430.

Trends and Techniques in Visual Gaze Analysis

Sophie Stellmach[1], Lennart Nacke[2], Raimund Dachselt[1], and Craig A. Lindley[2]

[1]User Interface & Software Engineering Group, Department of Simulation and Graphics, Faculty of Computer Science, Otto-von-Guericke University
PO Box 4120, D-39016 Magdeburg, Germany
sstellmach@gmail.com, dachselt@acm.org

[2]Game and Media Arts Laboratory, Department of Interaction and System Design, School of Engineering, Blekinge Institute of Technology
PO Box 214, SE-374 24 Karlshamn, Sweden
Lennart.Nacke@bth.se, Craig.Lindley@bth.se

Keywords
Gaze visualization, trends, virtual environments, qualitative and quantitative survey

Introduction

Visualizing gaze data is an effective way for the quick interpretation of eye tracking results. In the two general application areas of eye tracking, diagnostics and interaction (Duchowski, 2002), recently there has been put much effort into gaze interaction for three-dimensional (*3D*) virtual environments (*VEs*) (Castellina and Corno, 2008; Isokoski and Martin, 2006; Smith and Graham, 2006). However, since diagnostic studies benefit from visualizations of eye tracking data for understanding complex relationships between gaze behavior and stimuli, developing visualization techniques for *3D VEs* is a fundamental next step in eye tracking research.

A classification of gaze visualization techniques by Špakov (Špakov, 2008, pp. 37-49) emphasized the limited variety of suitable techniques for *3D* stimuli. The most widely used procedure for investigating gaze data for dynamic and *3D* stimuli is to analyze superimposed gaze plots over video recordings on a frame-by-frame basis. This quickly results in a monotonous and time-consuming process. The lack of suitable techniques for a more efficient gaze analysis of *3D VEs* results in the desire for enhanced procedures. Such techniques may provide quick insights into gaze behavior for evaluative studies of, for example, digital games, model designs, and product placement in virtual worlds. The purpose of the research presented here is to establish a foundation for improving gaze visualizations of eye tracking data. We conducted a survey with professionals and researchers working with different stimulus types to find out more about the importance of gaze visualizations and general requirements for improved eye tracking analysis. This research aims at gaining a formal understanding of gaze visualization techniques and applying this knowledge to the design and development of novel visualizations especially for *VEs*.

In this paper, preliminary findings from mixed-method (some quantitative, but a major emphasis on qualitative) questions will be presented and discussed in light of the proposed gaze visualization framework. It has to be noted that this was primarily a qualitative investigation, which explains the small and *not* randomly selected sample size, thus following the purposeful sampling strategy discussed by Creswell (2007, pp. 125-129).

Method

Participants. Ten eye tracking professionals and researchers aged between 28 and 57 years (*Mean (M)* = 37.7, *Standard Deviation (SD)* = 9.38) participated in this survey online. Of the total, 50% (*N* = 5) were female and 50% (*N* = 5) were male. Participants had been working with eye trackers between 2 and 15 years with an average of 7.2 years (*SD* = 4.49). Having to grade their knowledge concerning eye tracking on a scale from 1 (little knowledge) to 5 (much knowledge), the average knowledge of participants was high (*M* = 4.2, *SD* = 0.63).When asked how many studies they had worked on that incorporated eye tracking analysis, participants' experience ranged between 3 and 50 studies (*M* = 15.9, *SD* = 14.71).

Survey design, apparatus and procedure. The survey consisted of demographic, mixed-method (eight quantitative and four qualitative) questions. The quantitative questions were aimed at evaluating the importance of visualizations for eye tracking analysis ("*How important were visualizations for the analysis of your eye tracking studies?*", "*How would you assess the importance of sophisticated gaze visualization techniques for dynamic virtual environments?*"). These evaluations were done on a scale from 1, not important, to 5, very important. Other quantitative questions were aimed at uncovering the stimuli types used in these studies (static [two-dimensional (*2D), 3D*], dynamic [*2D*, *3D*], interactive [*2D*, *3D*]). The qualitative questions asked about personal experiences ("*What are your experiences in designing and analyzing eye tracking experiments employing dynamic interactive stimuli?*"), weaknesses of current gaze visualizations ("*Where do you see weaknesses in current gaze visualization techniques?*") and desirable features for gaze analysis ("*What features would you desire for a simple and intuitive gaze analysis?*"). The survey was implemented online using the tool LimeSurvey[1] (Version 1.70+). Thirty-two researchers were selected from searches on major eye tracking publication venues (*COGAIN*, *ETRA* and *ECEM*) and together with staff from *Tobii Technology AB*. Anonymous identifiers were assigned to each participant and they were then invited via email to participate in the online survey. No financial incentives were offered for participation.

Preliminary results

Quantitative results. Visualizations for eye tracking analysis as conducted by individual researchers were assessed as important (*M* = 3.7, *SD* = 0.95). Gaze visualizations for dynamic *VEs* were estimated to be a little more important (*M* = 3.9, *SD* = 0.88), although not significant, $t(9)=-0.56$, $p>.05$. Ninety percent of the participants had already used static *2D* stimuli in their experiments. Following this, 70% had drawn on interactive *2D* stimuli (user interfaces) and 40% had employed *2D* dynamic stimuli (videos). However, only 20% had already used *3D* stimuli of all kinds (static, dynamic and interactive), see also Figure 1.

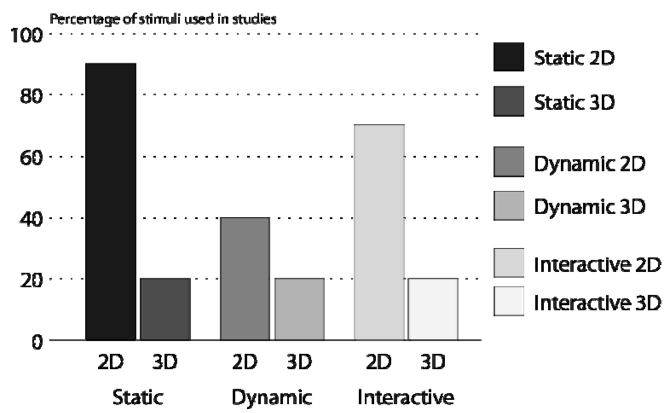


Figure 1. Percentages of stimuli (sorted by types) used by researchers in the study.

Qualitative results. Among the ten people interviewed, six people made a statement about their application areas (multiple selections were possible): four employed eye tracking for diagnosis and four for interactive applications. Two respondents had conducted studies using virtual *3D* stimuli.

[1] Open source survey application mainly developed by Carsten Schmitz and available for download at www.limesurvey.org

Although gaze visualizations are generally considered important (*M* = 3.7, *SD* = 0.95), researchers using eye tracking for interaction studies declared that they "*almost never did any visualizations*" (10 years work experience with eye tracking). For diagnostic purposes, however, visualizations provide "*the only way to quickly review and analyze what participants saw / did not see*" (8 years[2]). Thus, gaze visualizations can play an essential role in detecting gaze patterns. This is especially beneficial when trying to communicate findings to customers. Nevertheless, one respondent argued that visualizations "*do not prove anything*" (8 years).

In spite of the importance granted to gaze visualizations by most interviewees, several drawbacks were also specified. An important issue is the lack of effective dynamic visualization methods. In fact, eight people agreed with the following statement: "*So far, nobody has come across a simple and intuitive way to visualize gaze data for dynamic stimuli.*" (Ramloll et al., 2004; Špakov, 2008). One person disagreed, claiming that gaze replays are an intuitive way to visualize gaze positions. Another participant named several approaches for improving gaze visualizations for videos (Duchowski and McCormick, 1998; Nikolov et al., 2004; Sadasivan et al., 2005), but granted the difficulty of considering these techniques simple or intuitive.

A general problem for gaze visualizations is the danger of misinterpretation, which was mentioned by one interviewee. Reports on weaknesses of current visual gaze analyses mainly targeted two areas: features of available tools and of visualizations. Gaze analysis tools were criticized for the lack of multiple views and properly formatted data. In addition, further improvements of fixation identification algorithms and the conversion of processed visualizations into table data was demanded. It was also mentioned that gaze visualizations for *3D* contexts need to be improved and should provide aggregated comparisons (i.e. *3D* heat maps). Some participants warned that scan paths from multiple participants (especially in *3D*) may result in visual clutter as well as problems with temporal synchronization. Another respondent advised that "*simple gaze path animations may not help in seeing repeating patterns*" (8 years).

Concerning eye tracking studies within *VEs*, one interviewee requested the possibility to "*compare multiple scan paths in the VE while still allowing the person viewing this to control their own position/orientation in the VE*" (15 years). Another respondent (10 years) provided the following recommendations: "(*1) Easy way to specify the 'variables-of-interest' (not of AOIs*[3]*).(2) Multiple visualization and easy-to-follow links between them. (3) Ability to select and edit gaze data in a very flexible way (in tables, interactive, etc), having the variables-of-interest recalculating and multiple visualization repainting 'on-the-fly'*". Further requests include:

- Integration of signal processing methods like frequency analysis for automatic pattern detection
- Automatic relocation of *AOIs* for moving objects (videos)
- Visualization of transitions between different views
- Integration of external data sources
- Multiple views, overview-and-detail, backtracking
- Data format that allows use of statistical programs
- Annotation tool (allowing researchers to attach and share comments in the playback)
- Not being time consuming to set up and analyze (quick deployment)

Discussion & Future Work

This paper described current trends and needs within eye tracking research concerning improvements in gaze analysis techniques. The presented results clearly imply the importance of gaze visualizations for diagnostic use. Aggregated visualizations such as heat maps may be integrated for static *3D* scenes. A very interesting result is that a majority of eye tracking researchers have not used *3D* stimuli so far. Reasons for this may

[2] In the following X years denotes X years of work experience with eye tracking.

[3] Areas of interest

include a higher complexity to develop sophisticated *3D* scenarios and the use of tedious frame-by-frame evaluation of session videos. Beside the adaption of visualizations to stimuli, several features concerning usability of available analysis tools have been mentioned, including: overview, details-on-demand, backtracking, etc. (see also the *Information Seeking Mantra,* (Shneiderman, 1996)).

In conclusion, by identifying the trends and requirements for gaze visualization in *VEs*, we now have specifications for developing tools that can appropriately visualize various stimulus types in *VEs*. Our future work includes further investigation of visual analysis techniques for eye tracking as well as the prototyping of a tool that incorporates knowledge gained from this survey. The prototype tool will include the design and development of novel gaze visualizations for static *3D VE*s.

Acknowledgements

This research was supported by "*FUGA* - The Fun of Gaming: Measuring the Human Experience of Media Enjoyment", funded under the 6th Framework Programme of the European Commission (Contract: *FP6-NEST-28765*). We thankfully acknowledge the participation of the eye tracking professionals and researchers in this study. Also, we would like to thank Jessica Lundqvist from *Tobii Technology AB* for recommending some of her contacts for participation in this study.

References

Castellina, E., and Corno, F. (2008). Multimodal Gaze Interaction in 3D Virtual Environments. In: *Proceedings of the 4th Conference on Communication by Gaze Interaction - COGAIN 2008: Communication, Environment and Mobility Control by Gaze*, Prague, Czech Republic, COGAIN, pp. 33-37.

Creswell, J. W. (2007). *Qualitative Inquiry & Research Design* (2nd ed.). London, UK: Sage Publications.

Duchowski, A. T. (2002). A Breadth-First Survey of Eye-Tracking Applications. *Behavior Research Methods Instruments and Computers, 34*, 4, pp. 455-470.

Duchowski, A. T., and McCormick, B. (1998). Gaze-contingent video resolution degradation. *Human Vision and Electronic Imaging III, 3299*, 1, pp. 318-329.

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 2006: Gazing into the Future*, Turin, Italy, pp. 78-81.

Nikolov, S. G., Newman, T. D., Bull, D. R., Canagarajah, N. C., Jones, M. G., and Gilchrist, I. D. (2004). Gaze-contingent display using texture mapping and opengl: system and applications. In: *ETRA '04: Proceedings of the 2004 symposium on Eye tracking research & applications*, San Antonio, Texas, USA, ACM Press, pp. 11–18.

Ramloll, R., Trepagnier, C., Sebrechts, M., and Beedasy, J. (2004). Gaze data visualization tools: opportunities and challenges. In: *Eighth International Conference on Information Visualisation (IV'04)*, London, UK, IEEE Computer Society, pp. 173–180.

Sadasivan, S., Greenstein, J. S., K.Gramopadhye, A., and Duchowski, A. T. (2005). Use of eye movements as feedforward training for a 46synthetic aircraft inspection task. In: *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, Portland, Oregon, USA, ACM, pp. 141–149.

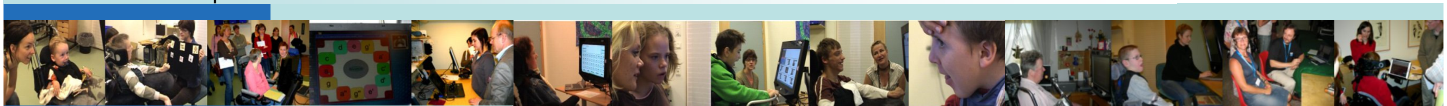
Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In: *Proceedings of IEEE Symposium on Visual Languages*, Washington, DC, USA, IEEE Computer Society, pp. 336-343.

Smith, J. D., and Graham, T. C. N. (2006). Use of eye movements for video game control. In: *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology (ACE)*, Hollywood, California, ACM Press, pp.

Špakov, O. (2008). *iComponent - Device-Independent Platform for Analyzing Eye Movement Data and Developing Eye-Based Applications.* Unpublished PhD Thesis, University of Tampere, Tampere, Finland.

Clinical experiences from assessment and introduction of eye gaze systems

For individuals with severe motor disorders, computerised activities might be a possibility to manage independently. Though, they often need special solutions to get computer access. For some users, an eye gaze system might be the only way of undertaking activities independently. The main aim of Dart's involvement in the COGAIN-project is user-trials for evaluation of different eye-gaze-systems, development of framework applications to assist in the initial assessment and implementation for individuals with severe motor dysfunction, some also with cognitive and visual problems. Trials with different users have given information on different issues and needs of improvement to make eye gaze systems an effective and efficient alternative for users with complex disabilities

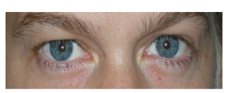


User trials have included 51 users in the ages 2 to 63 years old. These persons have varying diagnoses; Severe Cerebral Palsy, Multiple Scleroses, Amyotrophic Lateral Scleroses, Spinal Muscle Atrophy, Duchenne Muscular Dystrophy, Traumatic brain injury, Brain stem stroke, Metabolic conditions and Rhett's syndrome.

When making an assessment it's important to analyze the characteristics of the eye gaze system, and to find out what the user wants to be able to perform with the technical aid.

Start with a thorough assessment of the user's abilities;
- best working position
- function for mouse click
- visual problems or limited control of eye movements
- are there other things that might affect use of eye gaze - like problems opening the eyes, squinting, nystagmus etc?
- other aspects like cognition, literacy skills, hearing etc

Which of the strengths can we use to overcome the difficulties?



When using an eye gaze system some users may benefit from having the computer screen at bit higher than is generally advisable. This may be because the user opens his/her eyes more when looking up, allowing the eye gaze system to recognise where the user is looking more easily. Though, it is important to be attentive to the user's neck to avoid pain

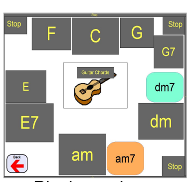
Adjustable screen gives more choices for comfortable positioning

Avoid reflections and direct sunlight

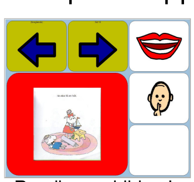
Consuming less energy might lead to increased performance and the possibility to work for a longer period.

It's important to give the user all opportunities available.
Find motivating activities and have fun!
Don't work in to sessions that are too long – the first trials with an eye gaze system might be tiring. Take breaks!

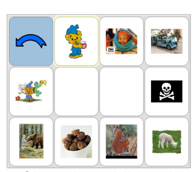
Example of applications for introduction of eye gaze system



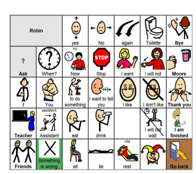
Playing guitar chords



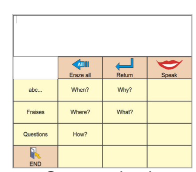
Reading a children's book



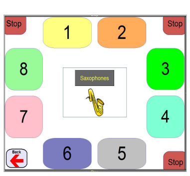
Choosing children's songs



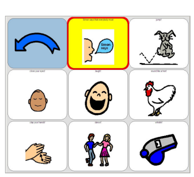
Communication with PCS-symbols



Communication with words/letters

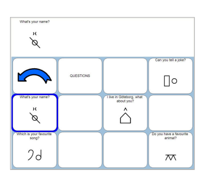


Playing saxophone loops

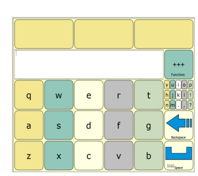


Playing "Simon says"

Choosing music



Asking questions with Bliss-symbols



On-screen-keyboard

REGION VÄSTRA GÖTALAND
SAHLGRENSKA UNIVERSITY HOSPITAL

Eva Holmqvist och Margret Buchholz
Council Certified Specialists in Occupational Therapy
For more information contact: eva.holmqvist@vgregion.se

DART Centre for AAC and AT

COGAIN 2009: Gaze Interaction For Those Who Want It Most

www.cogain.org/cogain2009

COGAIN Conference

Tuesday 26 May 2009

COGAIN
Communication by Gaze Interaction

Sponsors

DTU

ITMAN

DTU Informatics
Department of Informatics and Mathematical Modelling

ISBN: 978-87-643-0475-6