Proceedings of

COGAIN 2008

‘Communication, Environment and Mobility Control by Gaze’

Edited by Howell Istance, Olga Štěpánková and Richard Bates

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

ISBN 978-80-01-04151-2

Welcome to COGAIN 2008!

This is the fourth international conference on communication by gaze interaction organised by the European Framework 6 Network of Excellence, COGAIN. The first conference in Copenhagen, held in 2005, was an opportunity for partners in the Network to take stock of current research in Europe. The second conference was an open event, held in Turin, and had 'gazing into the future' as its theme. Its purpose was to chart a research agenda for gaze-based communication over the coming 10 years. The third conference in Leicester attracted approximately 100 delegates from 16 countries. Its theme was gaze-based creativity and interacting with games and on-line communities, the first conference ever to have this as its main focus.

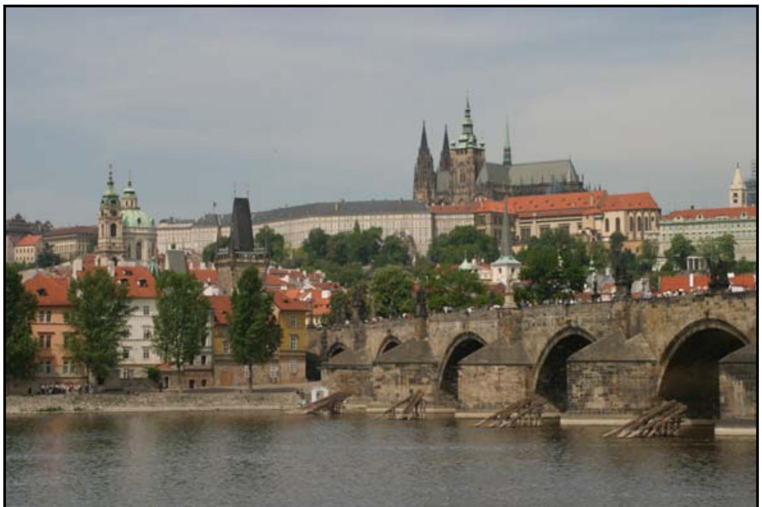
The theme of this year's conference is communication, environmental control and mobility control, particularly for people with motor impairments. We are expanding the areas in which gaze communication can be used effectively to include control of mobility and of the user's environment. This will require that gaze tracking systems can be used on mobile platforms and in a range of different lighting conditions. A person with motor neuron disease and an experienced user of gaze communication devices said recently "One of the future challenges should be to make a computer so that you can drive the wheelchair (safely!) using only an eye tracker". Enabling a suitable level of safety is critical if this challenge is to be met and we need to understand fully the safety issues involved, and investigate thoroughly solutions to these issues.

Gaze control has the potential to make really significant improvements to the quality of life of people with severe motor impairments. There is a need for faster and more versatile interaction techniques as well. It is important that the needs of users, both able-bodied and with disabilities are studied and that system solutions are evaluated against these. The conference reflects the importance of these aspects in its three paper sessions.

Now at the end of its fourth year, the COGAIN conferences have become the major international events that focus particularly on gaze communication for people with disabilities. It has become clear that there is a need for a permanent communication platform where all parties interested in gaze communication can meet, discuss and collaborate.
This is the aim of COGAIN association (http://www.cogain.org/association), which will be established during the second day of the conference (3rd September 2008) in Prague.

So welcome to Prague and enjoy the conference!

Howell Istance and Olga Štěpánková
COGAIN 2008 Conference Co-chairs



A highlight of any visit to Prague is the Charles Bridge. The stone Gothic bridge connects the Old Town and Malá Strana in Prague and was commissioned by Czech King and Holy Roman Emperor Charles IV in 1357. Charles Bridge is on the top of every Prague visitor's must-see list with thirty 17th century Baroque statues lining the bridge, and a beautiful view of the city. It is also popular with artists, musicians and souvenir vendors.

COGAIN 2008 Programme

8:30-9:00	**Registration**
09:00-10:30	**Keynote: The Human-Technical Challenge of Developing Gaze-Controlled Devices** Dr. Anthony Hornof, *University of Oregon, USA*
10:30-11:00	Refreshments
11:00-12:30	**Session 1: Overcoming Technical Challenges in Mobile and Other Systems** **Off-the-Shelf Mobile Gaze Interaction** J. San Agustin and J. P. Hansen, *IT University of Copenhagen, Denmark* **Fast and Easy Calibration for a Head-Mounted Eye Tracker** C. Cudel, S Bernet, and M Basset, *University of Haute Alsace, France* **Magic Environment** L. Figueiredo, T. Nunes, F. Caetano, and A. Gomes, *ESTG/IPG, Portugal* **AI Support for a Gaze-Controlled Wheelchair** P. Novák, T. Krajník, L. Přeučil, M. Fejtová, and O. Štěpánková, *Czech Technical University, Czech Republic* **A Comparison of Pupil Centre Estimation Algorithms** D. Droege, C Schmidt, and D. Paulus, *University of Koblenz-Landau, Germany*
12:30-14:00	Lunch
14:00-15:30	**Session 2: Broadening Gaze-Based Interaction Techniques** **User Performance of Gaze-Based Interaction with On-line Virtual Communities** H. Istance, *De Montfort University, UK,* A. Hyrskykari, *University of Tampere, Finland,* S. Vickers, *De Montfort University, UK* and N. Ali, *University of Tampere, Finland*

	Multimodal Gaze Interaction in 3D Virtual Environments E. Castellina and F. Corno, *Politecnico di Torino, Italy* **How Can Tiny Buttons Be Hit Using Gaze Only?** H. Skovsgaard, J. P. Hansen, *IT University of Copenhagen, Denmark* J. Mateo, *Wright State University, Ohio, US* **Gesturing with Gaze** H. Heikkilä, *University of Tampere, Finland* **NeoVisus: Gaze Driven Interface Components** M. Tall, *Sweden*
15:30-16:00	Refreshments
16:00-17:30	**Session 3: Focusing on the User: Evaluating Needs and Solutions** **Evaluations of Interactive Guideboard with Gaze-Communicative Stuffed-Toy Robot** T. Yonezawa, H. Yamazoe, A. Utsumi, and S. Abe, *ATR Intelligent Robotics and Communications Laboratories, Japan* **Gaze-Contingent Passwords at the ATM** P. Dunphy, A. Fitch, and P. Oliver, *Newcastle University, UK* **Scrollable Keyboards for Eye Typing** O Špakov and P. Majaranta, *University of Tampere, Finland* **The Use of Eye-Gaze Data in the Evaluation of Assistive Technology Software for Older People** S. Judge, *Barnsley District Hospital Foundation, UK* and S. Blackburn, *Sheffield University, UK* **A Case Study Describing Development of an Eye Gaze Setup for a Patient with 'Locked-in Syndrome' to Facilitate Communication, Environmental Control and Computer Access** Z. Robertson and M. Friday, *Barnsley General Hospital, UK*
17:30	Close

COGAIN 2008 Papers Index

Session 1: Overcoming Technical Challenges in Mobile and other Systems

Off-the-Shelf Mobile Gaze Interaction

Javier San Agustin
IT University of Copenhagen
Rued Langgaards Vej 7, 2300-DK
javier@itu.dk

John Paulin Hansen
IT University of Copenhagen
Rued Langgaards Vej 7, 2300-DK
paulin@itu.dk

Keywords
Gaze interaction, head-mounted display, head-mounted eye tracker, off-the-shelf

Introduction

In this paper we present a prototype of a mobile gaze interaction system based on a commercial head-mounted display (HMD) and an inexpensive webcam for tracking the user's eye movements. The components are off-the-shelf and our solution does not require any hardware modifications. The total cost of the hardware components (not including a laptop PC) is less than 200€.

HMDs are becoming increasingly popular as a means to obtain information on-the-spot in applications such as medicine, entertainment, augmented reality, maintenance or telerobotics (Liu et al. 1993, Tanriverdi and Jacob 2000, Broll et al. 2006). Displaying the information right in front of the user's eye(s) holds interesting potentials. For instance, a technician repairing a defective wire in a building can benefit from looking at maps and diagrams of the electrical installation on a head-mounted display, offering him the possibility of accessing important information without moving. During an operation, a doctor might need to look at different images and information of the patient being operated, and having them at-a-glance on an HMD can be more efficient than turning towards a desktop computer.

Even relatively high-resolution HMDs (640x480 and higher) are comfortable to wear, weighing around 100 to 200 grams. A growing number of companies is producing HMDs at a relatively low price (200 to 400 US$) for mini-PCs, mobile phones or mp4 video players. Most of the systems are non-immersive, allowing the user to maintain a view of the physical environment.

The new mobile displays create a demand for an efficient technique to interact with the information displayed in the HMD. When the hands are needed for other tasks, hand-controlled devices such as keyboard or mouse become awkward. Gaze interaction with the HMD can potentially provide a hands-free pointing technique (Bleach et al., 1998).

People using augmented and alternative communication tools may benefit from an HMD with gaze control. Daily activities, like driving a wheelchair, would not be interrupted when communicating. People without control of their hands could communicate on-the-move and in bed without requiring external assistance to reposition the equipment.

There are several challenges in the development of gaze interactive HMD systems. First and foremost present head-mounted eye trackers are expensive. Secondly, adding a gaze tracker to a display may increase the weight and make the complete system uncomfortable to wear. Thirdly, the user may feel “odd” wearing bulky gear in front of the eyes. In addition, tracking the eyes in mobile conditions may be more complicated than in a well-controlled environment as when sitting in front of a desktop computer. Light conditions will change as the user walks around different scenarios, introducing the need for robust eye tracking techniques (Hansen and Pece 2005). Movements of the head may also cause the camera to slip. However, if the display and the tracker are mounted on the same frame, issues related to head movements are eliminated from the gaze tracking process.

The purpose of the research project reported in this paper is to investigate the possibilities of building a low-cost mobile HMD system that allows for gaze control with a standard PC. The availability of such a system would make it possible for researchers on a limited budget to explore solutions to the challenges listed above. There might also be some users who would like to test a mobile system in real-life, even with the present shortcomings.

Gaze interaction on HMDs

Interaction on the HMD can be performed by means of gaze tracking. Although remote eye trackers are less intrusive than head-mounted, they do not represent a viable solution for interacting with mobile displays. Furthermore, recent advances in miniaturization of cameras, batteries and light sources have reduced the weight and intrusivenes of head-mounted eye trackers.

A number of such systems have been described in the literature. Babcock and Pelz (2004) presented a system to be used in off-line situations. It includes a camera that records the scene in front of the user. After recording a sequence, gaze information can be obtained and combined with scene information. Li et al. (2005) introduced a similar system that works in real-time. Although they use off-the-shelf components, their approach involves ingenious hardware modifications that require an advanced knowledge on electronics, which may prevent potential users from building the system. Smith et al. (2005) presented the ViewPointer, a head-mounted eye tracker that enhances context information when the user looks at pre-tagged physical objects by detecting whether the user looks directly at the object. This approach does not estimate gaze coordinates and thus is not suitable for interaction with a display.

Our prototype makes use of off-the-shelf components that do not require hardware modifications.

Hardware

Our system consists of a Sandberg Nightvision camera (Figure 1), which provides a resolution of 640x480 at 15 Hz or 320x240 at 30 Hz. It costs around €15 and weighs 100 grams. It has 6 built-in infrared LEDs. Infrared light improves the illumination conditions of the image and ease the detection of the eye features. We take advantage of the built-in infrared light to create a dark-pupil effect.

A commercial binocular head-mounted display (Vuzix DV920, Figure 1) is connected to a standard laptop PC. It provides a resolution of 1024x768 pixels and weighs 100 grams. The binocular HMD prevents ambient light from reaching the user's eye, eliminating most of the undesired reflections on the sclera and iris. However, the user can still see parts of the surrounding environment by looking above or below the 2.5 cm thick display frame. The current price is about €185.

Figure 1. Nightvision web camera (left). Binocular head-mounted display (right)

The camera has not been fixed to the HMD. Instead, it is mounted on a lightweight strap helmet that can be adjusted to the user's head. The camera may thus be conveniently positioned close to the eye to obtain a good image of the pupil. Figure 2 shows a user wearing the HMD and the head-mounted camera looking into the eye from below the HMD.

Figure 2. A user wearing the HMD and the head-mounted eye tracker

Algorithm for tracking the eye

The eye tracking algorithm uses the dark-pupil technique, and is based on fitting an ellipse to the contour of the pupil. A point on the contour is considered to have maximum gradient along any line extending from the initial guess of the center of the pupil. A set of 80 points on the contour are located by calculating the maximum gradient. The size of the pupil in the previous image is taken into account to calculate the length of the lines along which the gradient is calculated. This avoids taking points far away from the pupil as belonging to its contour.

Once the points on the pupil contour are located, an ellipse is fitted to these points. Since the number of points on the contour is usually high and very few points are located far from the pupil, we use a 2-step approach to estimate the ellipse. First, an initial ellipse is fitted to all the points. The shape of this ellipse might be deformed due to the presence of outliers. A second ellipse is then fitted using only the points that lie close to the first ellipse. Most outliers are eliminated by this technique. This approach avoids using iterative methods such as RANSAC, which are inefficient and require higher processing time.

Calibration

Gaze is estimated from the center of the pupil. A calibration process is required to map the pupil position to the HMD screen. A set of points is shown in the display and the user has to look at them in sequential order. A second order polynomial regression is then applied to estimate gaze (Morimoto et al., 1999). Calibration takes around 30 seconds.

Results

The accuracy of the system has been evaluated by conducting an experiment with three subjects. Each subject calibrated the eye tracker by looking at 16 targets displayed on the HMD. Upon completion of calibration, the user was instructed to look again at the 16 targets. Gaze location was estimated in real time during the test phase. No smoothing was applied to the estimated gaze coordinates.

Accuracy was evaluated under two different conditions. In the first one, the subject stood still. This situation is equivalent to using a remote eye tracker while maintaining the head still. In the second test the subject was instructed to walk along a corridor while looking at the targets presented on the screen. Figure 3 shows the accuracy in degrees for each of the users in both conditions.

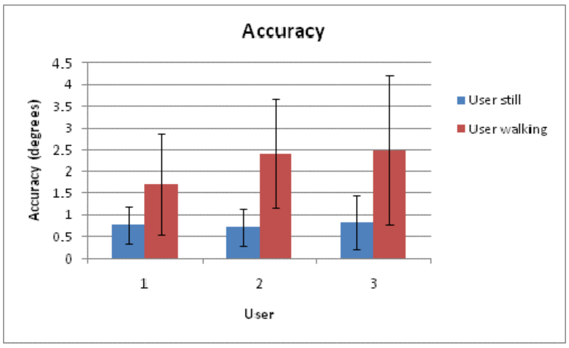


Figure 3. Accuracy for each user when standing still and when walking

When standing still, the average accuracy obtained is 0.766º ± 0.49. Since the camera is located close to the eye, a good quality image of the pupil is obtained, and thus the estimated center is very precise. On the contrary, in a mobile scenario where the user is walking the accuracy drops to an average of 2.20º ± 1.38. Camera and HMD are not fixed to each other, and therefore there are relative movements between both components. This introduces errors in the estimated gaze position. In addition, the system might slip as the user walks. Integrating the camera and the HMD into one element would improve the accuracy in a mobile scenario. Figure 4 shows the estimated gaze positions for one of the users when the user is standing still and when the user is walking.

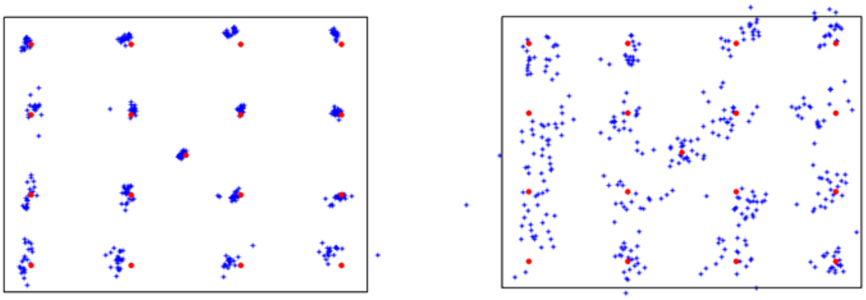


Figure 4. Estimated gaze positions for one user: standing still (left) and walking (right)

Discussion

We have developed and built a prototype of a head-mounted eye tracker that allows the user to interact with a commercially available head-mounted display. The whole system costs 200€ and weighs a total of 200 grams. The preliminary tests show an average accuracy under 1º when standing still and around 2.5º when walking. Since the camera and the HMD are not fixed to each other, the accuracy is affected by relative movements between both elements when the user walks. However, even while walking the accuracy is high enough to interact with noise-tolerant interfaces that have been specifically designed for gaze typing (e.g. Hansen et al. 2001, Hansen et al. 2008). While standing it is possible to interact with a normal windows environment through the use of standard gaze-clicking techniques (dwell, zooming or two-step magnification).

The prototype can undergo a number of improvements. Integrating the camera completely with the HMD is the most obvious next step. However, a complete integration of HMD and eye tracker will require some hardware modifications or special manufacturing. We are considering alternative solutions for a flexible mounting of the camera binding it close to the display but without covering the user's face. The design of a cool-looking face mount is probably the biggest challenge remaining. We expect that manufacturing companies of e.g. bike helmets, sunglasses, earphones or visors will have the competence to solve this problem.

Gaze can be used for pointing, but it lacks the ability to perform selections, i.e. clicking. Facial muscle activity through an EMG switch can provide a reliable solution to perform activations in combination with gaze pointing (Mateo et al., 2008). Voice recognition could also be used to activate certain predefined actions.

Acknowledgments

This work was supported by the European Network of Excellence COGAIN, Communication by Gaze Interaction, funded under the FP6/IST programme of the European Commission.

References

Babcock, J. and Pelz, J. (2004) Building a lightweight eyetracking headgear. *Proceedings of the 2004 symposium on Eye tracking research & applications*.

Bleach, G., Cohen, C.J., Braun, J. and Moody, G. (1998) Eye tracker system for use with head mounted displays. *IEEE International Conference on Systems, Man, and Cybernetics*.

Broll, W., Ohlenburg, J., Lindt, I., Herbst, I., Braun, A-K. (2006) Meeting technology challenges of pervasive augmented reality games. *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*.

Hansen, D.W. and Pece, E.C. (2005) Eye tracking in the wild. *Computer Vision and Image Understanding*.

Hansen, D.W., Skovsgaard, H.H.T., Hansen, J.P., Møllenbach, E. (2008) Noise tolerant selection by gaze-controlled pan and zoom in 3D. *Proceedings of the 2008 symposium on Eye tracking research & applications*.

Johansen, A. and Hansen, J.P. (2006) Augmentative and alternative communication: the future of text on the move. *Universal Access in the Information Society*, Volume 5, Number 2.

Li, D., Winfield, D. and Parkhurst, D. (2005) Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Liu, A., Tharp, G., French, L., Lai, S. and Stark, L. (1993) Some of what one needs to know about using head-mounted displays to improve teleoperator performance. *IEEE Transactions on Robotics and Automation*.

Morimoto, C.H., Koons, D., Amit, A., Flickner, M., Zhai, S. (1999) Keeping an eye for HCI. *Proceedings of the XII Brazilian Symposium on Computer Graphics and Image Processing*.

Mateo, J.C., San Agustin, J. and Hansen, J.P. (2008) Gaze beats mouse: hands-free selection by combining gaze and EMG. *Proceedings of the SIGCHI conference on Human factors in computing systems*.

Smith, J.D., Vertegaal, R. and Sohn, C. (2005) ViewPointer: Lightweight calibration-free eye tracking for ubiquitous handsfree deixis. *Proceedings of the 18th annual ACM symposium on User interface software and technology*.

Tanriverdi, V. and Jacob, R.J.K. (2000) Interacting with eye movements in virtual environments. *Proceedings of the SIGCHI conference on Human factors in computing systems*.

Fast and Easy Calibration for a Head-Mounted Eye Tracker

Cudel Christophe, Bernet Sacha, Basset Michel
Laboratory MIPS-EA2332, University of Haute Alsace
61, rue Albert Camus 68093 Mulhouse, France
{christophe.cudel, sacha.bernet, michel.basset}@uha.fr

Keywords
Eye tracker, easy calibration, infrared led, ransac, parallax

Introduction

Eye tracking techniques are used in many types of applications (Duchowski, 2007). We are focusing our work on wearable (or head mounted) eye trackers which can be used in the situation where head movements are free. Applications of this work can be found in automotive researches in our laboratory for road drivers behaviour (Basset et al., 2005).

Wearable eye trackers consist in a pair of glasses, helmet or headband equipped with 2 cameras : one for the pupil and the other for the scene. An infrared led placed close to the eye camera is used to ensure correct pupil illumination and allows for obtaining a corneal reflection if needed. A mathematical expression links the pupil position to the gaze direction or position. Many studies are already published (Villanueva et al., 2006, Li et al., 2005) about the determination of the Point Of Regard (POR) in the scene camera. The mapping between both cameras coordinates can be formulated by using a homography or a polynomial expression of 1^{st}, 2^{nd} or 3^{rd} order. Following the retained expression, the coefficients are determined with a set of calibration points issued from a grid of points located in a same plane.

This paper proposes an alternative scheme for the calibration procedure by using only a single infrared led as the pattern to be looked at. This can be done in the conditions detailed in the next section. Firstly, we demonstrate that this procedure offers a precision comparable to other methods, and secondly that during the calibration stage the eye tracker user can move his head and only a single led is mandatory instead of a grid of points.

Calibration modelisation

Figure 1 shows a schematic view from above of wearable eye tracker. To perform the calibration, eye tracker users have to look at points one after the other. These points must be located in a same plane and usually form a grid. To avoid parallax phenomena during the calibration, the head should stay in the same position. For each point, pupil position is automatically segmented, but the gaze direction in scene camera must be validated. If this validation is done manually (with a mouse click for example) an operator is often needed to help the eye tracker user. The calibration is performed when enough correspondences between eye and scene cameras are established in order to compute the mapping with a method of minimization.

We propose to reverse this scheme using only a single calibration point. In this case, a point set is collected by moving the head approximatively in a same plane during the calibration. This method is advantageous because we put a simple infrared led as calibration point. By using an IR filter in front of the micro-lens of scene camera, this infrared led is easily detected. This procedure is profitable because pupil centre and calibration points are both automatically detected with basic image processing algorithms. This offers an easy calibration for eye trackers.

Assuming that the head (or eye) remains in the same plane during the calibration, we demonstrate now that these two calibrations methods are identical. Here, we use homographies to formulate the relations between the planes of the model, but polynomial expressions could also be suitable. Because calibration is used to compute the mapping from eye to scene cameras, we use the eye camera as the geometric reference for the following equations. P is a calibration point P (in meters), vectors p_e and p_s are respectively the projection (in pixels) of P on eye and scene cameras.

Assuming that the observed scene is plane, the relation between P and its projections p_e and p_s respectively on Eye and Scene cameras are respectively:

$$P = H_1 . p_e \text{ and } p_s = H_2 . P$$

with the situation of Figure 1, H_1 and H_2 don't change during the calibration step and the relation between p_s and p_e can be written as:

$$p_s = H_2 . H_1 . p_e = H . p_e \text{ .}$$

If we consider the scheme of Figure 2 where the head moved, the projections of P become:

$$P = H_1' . p_e' \text{ and } p_s' = H_2' . P$$

Because as eye and scene cameras are rigidly attached, H is unchanged and:

$$p_s' = H_2' . H_1' . p_e' = H . p_e'$$

Thus p_s' and p_e' can be used to compute H.

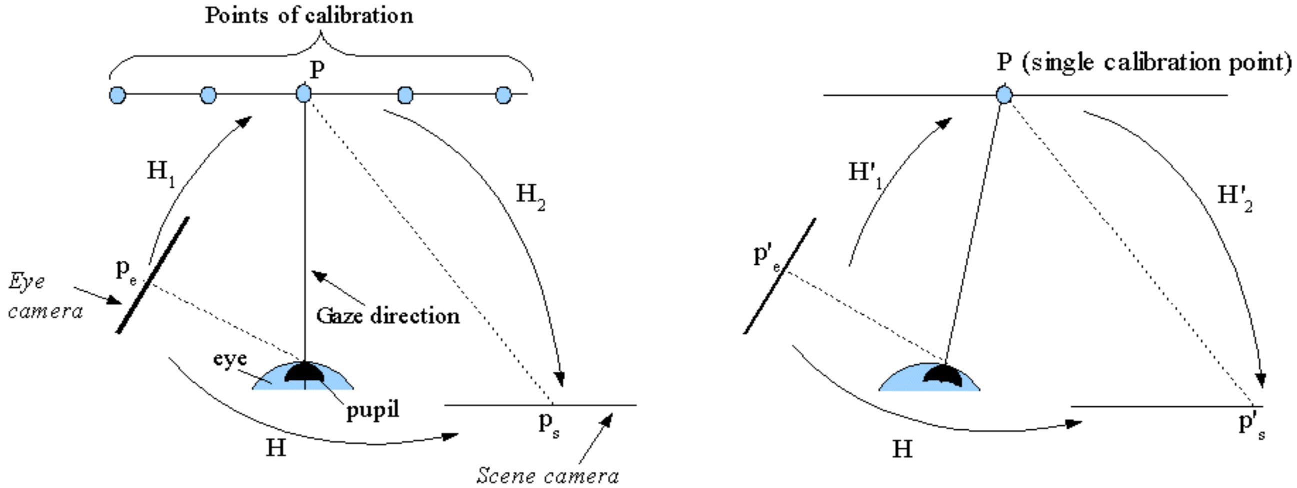


Figure 1. Top sight of head mounted eye tracker. P projection on scene camera and pupil projection on eye camera when gaze direction is oriented on P.

Figure 2. Example of a head movement on the left. H doesn't change in the eye camera reference.

This method is correct only if the head movements are confined in a plane. If this hypothesis is not respected, points coordinates will be affected by the parallax. On Figure 3, we present the error induced by parallax in the condition where the gaze is focused on a point P and where the eye tracker user moves following the axis of the gaze direction. In this case p_e stays at the same location but p_s changes of position in the image of the scene camera. With a calibration made at 1 meter (distance between P and user's eye) and with an eye tracker user moving from -0.5 to 10 meters, the parallax (dp) is characterized by variations of p_s position from 40 to 25 pixels (Figure 3). For head user staying in a depth of +/- 5cm during the calibration process, p_s coordinates are affected by an error below 2 pixels. This computation is obtained with intrinsic parameters of our scene camera and by considering that the scene camera and user eye are distant of 5 cm (Figure 4).

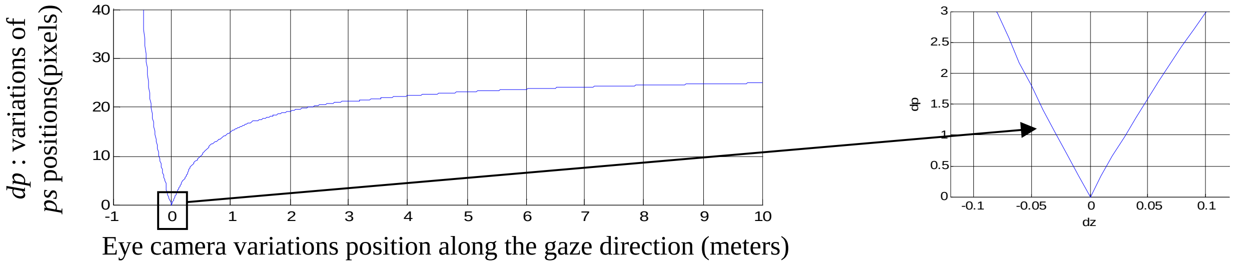


Figure 3. p_s variations when eye camera moves. Right figure is zoom around the calibration distance.

Calibration results

Compared to the classical calibration based on the use of a grid, our method is easier. In practice, the user starts the calibration and just needs to look at the infrared led from different positions. The acquisition of eye and scene images is continuous and takes only few seconds. Real times images processing algorithms are used to automatically segment both the pupil position in the eye image and the infrared led position in the scene image. This method allows to use more than 9 points to compute the mapping between eye and scene cameras with Levenberg Marquart algorithm. Because acquisition is continuous, mapping computation can be affected by false correspondences (outliers). We use a RANSAC algorithm (Fischler, 1981) to remove these outliers.

We propose to compare several kind of relations found in the literature (Li, 2005), which are used for eye tracking systems. Table 1 compares the precision of our method with the one obtained using 9 point grid. Points of calibrations were located 1 meter in front of the eye tracker. Here, the presented results are obtained with a mapping based on pupil center methods. The corneal reflection or glint methods have not yet been tested, but results should be closed. Table 1 here used the same points for the calibration and the measurement.

	Results with a grid of 9 points		Results with a single calibration point	
Expression between eye and scene camera	Distance mean Error (pixels)	Angle mean Error (degrees)	Distance mean Error (pixels)	Angle mean Error (degrees)
Homography	9.1	0.6°	9.8	0.6°
Polynomials expressions tested : $x_s = a_0 + \sum_{n=1}^{N}(a_{xn}.x^n + a_{yn}.y^n)$ $y_s = b_0 + \sum_{n=1}^{N}(b_{xn}.x^n + b_{yn}.y^n)$				
N=1 (first order)	15.6	1°	6.2	0.4°
N=2 (second order)	6.8	0.4°	3.9	0.3°
N=3 (third order)	3.9	0.2°	2.7	0.2°

Table 1. comparison of our method with the one obtained with 9 points calibration. The working distance is of 1 meter.

We give the mean error between the calculate and real gaze position on scene camera. Results are also presented as mean angle gaze direction error. The obtained results are comparable, which can be easily explained by considering that the parallax error is in practice close to the error made by an eye tracker user when the gaze position is manually validated during the calibration.

Figure 4 represents the latest version of our eye tracker. Software is running in real time and has been developed in C++ using CVB and OpenCV libraries.

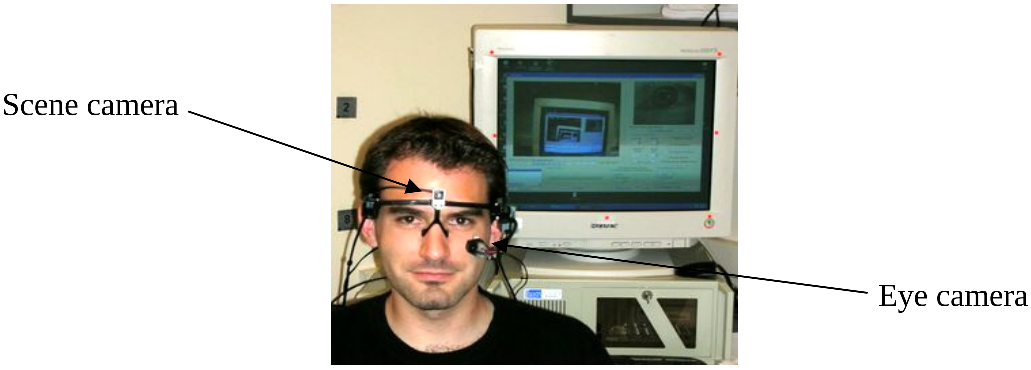


Figure 4. Illustration of our eye tracking system

Conclusion

This paper has presented an easy eye tracker calibration procedure, which takes only few seconds by using only a simple infrared led and with no external assistance. By using simple image processing methods, each calibration point is automatically detected, and the eye tracker user doesn't need to manually validate it. We showed that the results are comparable to a procedure with a grid of 9 points. We showed that the parallax is an important, factor which can blur the calibration but also the results. This procedure is well adapted for our applications, where we want to analyse the driver's behaviour for various situations of driving. The major advantages are that the driver's head doesn't need to be fixed and a single led is used instead of a grid of points for the calibration.

References

Duchowski, T. (2007) Eye Tracking Methodology theory and practice, 2nd edition, Springer editor

Basset, M., Cudel, C., Georges, V., Gissinger, G.L (2005) Visual characterization of road driver's behaviour. *WISP 2005, International symposium on intelligent signal processing*, Faro, Portugal, 1-3 sept. 2005, 6 pages.

Villanueva, A., Cadeza, R., Porta, S. (2006) Eye Tracking: Pupil orientation geometrical modelling, *Image and Vision Computing*, 24, elsevier, pp.663-679.

Li, D., Winfield, D., Parkhurst, D. J. (2005) Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 3, San Diego, USA, pp. 79.

Fischler, M. and Bolles, R. (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, vol. 24, *Communications of the ACM*, pp. 381-395.

Magic Environment

Luis Figueiredo
ESTG/IPG
Guarda
Portugal
Luis.figueiredo@ipg.pt

Tiago Nunes
ESTG/IPG
Guarda
Portugal
tiagomkey@gmail.com

Filipe Caetano
ESTG/IPG
Guarda
Portugal
fkaetano@gmail.com

Ana Gomes
Agrupamento Escolas Infante
D. Henrique, Viseu
Portugal
Ana.isabel.gomes@netvisao.pt

Keywords
Environment eye control

Introduction

In recent years, we have witnessed a great development in eye gaze systems that allow handicapped people to interact with the computer. Shi et Galley (2007) propose a remote eye tracking system with 3 cameras for environmental control, which proves the success of this new technology for handicapped people.

They have presented some works that allow the control of wheelchairs through the movements of the eyes. These systems usually use electromyography signals captured by electrodes placed around the user's eyes. After the processing of these signals, the systems can generate control signals for the wheelchair. Law et al (1999) and Barea et al (2002) are examples of these systems.

The following natural step is to join an eye gaze system like the one described by Figueiredo et Gomes (2007), with a conventional wheelchair to verify whether the control of an wheelchair only with the eye gaze is viable, in the current state of the technology.

In this paper two applications will be presented that allow environment control and an electric wheelchair control only with eye gaze.

Applications description

Environment Control

With the developed application of environment control we intend to provide the user with a simple and configurable tool according to his/her needs, involving low cost hardware, that enables the control of any infrared device or any electric device connected to a radio frequency receiver. We developed two different circuits with the blocks diagram presented in Figure1.

The emitter circuit communicates with the PC application by USB through a PIC microcontroller and allows recording infrared signals, the emission of infrared signals and the emission of radio frequency signals for the receiver module. The receiver module receives and processes the signals and allows the control of electric devices through the PIC microcontroller firmware. As for software, we developed an application that can create a limitless set of communication pictures and in each communication picture it can also create a high set of buttons. A function can be associated to each button in order to control an infrared device, an electric device, or both. The only thing that an eye gaze user will have to do is to select the communication picture button whose function he/she intends to activate.

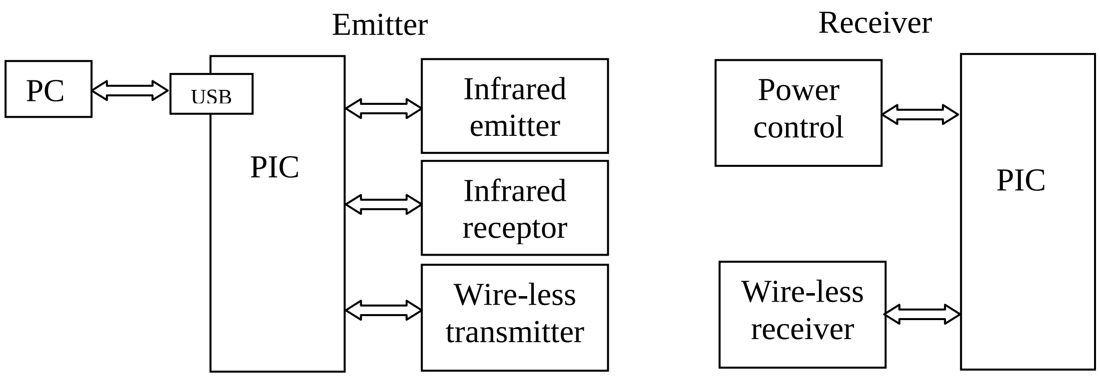


Figure 1: Block diagrams for the Emitter and Receiver

Wheelchair control

Since we used a low cost wheelchair for the tests, the first problem was the absence of a digital interface that allowed the connection between the PC and the wheelchair. This problem was solved with the development of an auxiliary circuit whose blocks diagram we present in **Figure 2**.

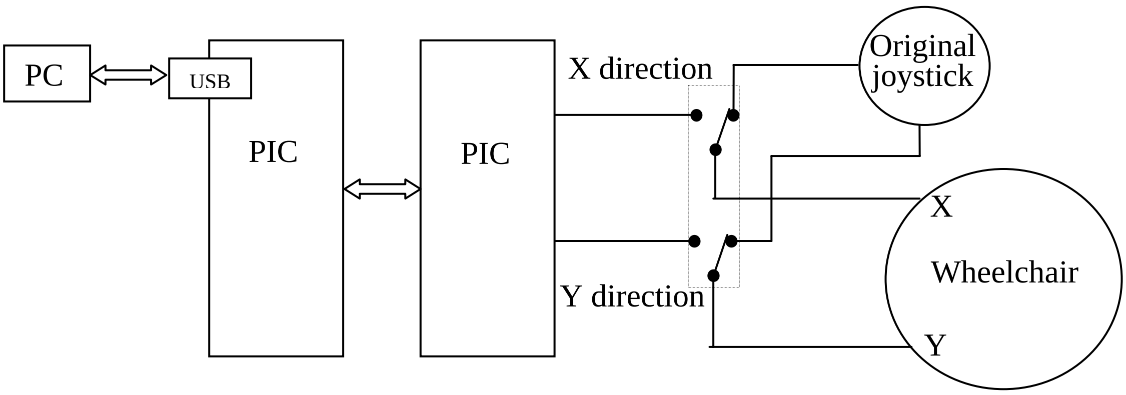


Figure 2: Software/hardware interface

For the wheelchair control, we try two approaches. First, the conceptual and simplest ones consist of placing a menu in the computer with 8 buttons indicating the different possible directions for the wheelchair movements. The user would have to activate each one of these buttons to follow in the desired direction. This solution has two problems. First, it is very complicated to control the wheelchair speed. Second, the user must constantly switch eye gaze from the laptop to the physical environment and vice-versa to be able to always lead the wheelchair in a safe position. This way of managing eye control was abandoned due to the deficient results obtained.

In the second approach, we remove the laptop from the user's sight and leaving only the digital cam to capture the user's face, which enables the eye gaze estimation. The user eye gaze directed to the cam will correspond to the stop position, i.e. the wheelchair is without movement when the user looks at the cam. The user can control not only the wheelchair direction but also its speed. To increase its speed the user must look up in relation to the cam. To diminish the speed he/she must look at a zone next to the cam. Looking at a zone below the cam the wheelchair runs backwards, also with the controlled speed with the eye gaze. For direction control (left and right), we obtained exceptional results, since we can control the wheelchair with high stability and precision. In fact, the system auto-adjusts the wheelchair movement by a natural feedback control system. If the user fixes his/her eyes in one target direction, and if the wheelchair doesn't take this direction, then the displacement of the user eyes will tend to increase in the inverse direction, which immediately implies the correction of its route. As it is not easy to translate the true control that we obtained with this system into words, graphs or tables, we will play a video that shows the first experience made with this system: http://www.youtube.com/watch?v=igIG-hMh3jU

Some simple security mechanisms were introduced:

- Disconnect the wheelchair whenever the PC does not send a control message to the wheelchair for a time period longer than 100ms.
- Activate the wheelchair when the user looks sequentially to the cam, to the right, to the cam, to the left and again to the cam. Only after these eye movements can the user control the wheelchair.
- Deactivate the wheelchair when the user looks to the cam (stop the wheelchair) for a fixed time. This allows the user to look wherever he/she wants to without moving the wheelchair when it is deactivated.
- Use a wireless communication device that disconnects the wheelchair in an emergency.

Future work

Since this basic work is to show, in practical terms, the viability of this project, we have already identified some future tasks to improve its functionality. We point out the following:

- Develop an Eye Gaze system using only a web cam instead of a high definition cam, considering that we have an excess resolution in the eye gaze determination for wheelchair control.
- Develop an Eye Gaze system that is more tolerant to the surrounding light, and allows its use outdoors.
- Provide the chair with a set of sensors that increase its security.
- Find alternatives to the computer to implement the Eye Gaze algorithms, which would increase the wheelchair autonomy. The use of embedded systems is fundamental for the future of this work.
- Test the wheelchair with real users.

Conclusion

The development of simple hardware devices that communicate with the PC can substantially increase the interaction between an Eye Gaze user and the environment. Being nothing new, this is still simple, economic, functional and easy to install.

It's easy and economic to construct the hardware that establishes a digital connection between the PC and a conventional wheelchair that does not have any digital interface.

It is possible to control the speed and direction of a wheelchair with great precision using an Eye Gaze system.

The wheelchair direction control uses a natural feedback system that automatically adjusts the wheelchair direction towards the point where the user is looking at.

The vibrations caused by the normal wheelchair movement do not interfere in the eye gaze detection.

Natural light interferes significantly in the Eye Gaze detection, which limits this system to just indoor use.

With these experiences and future work, the expression "what you see is what you get" may be transformed into "where you see is where you get".

References

Barea, R. Boquete, L. Mazo, M. López, E. (1999) Guidance of a wheelchair using electrooculography. *Proceeding of the 3rd IMACS International Multiconference on Circuits*, Systems, Communications and Computers (CSCC'99). July 1999.

Figueiredo, L. Gomes, A. (2007) Magic Eye Control. *COGAIN 2007*, Leicester, UK.

Law, C.K.H.; Leung, M.Y.Y.; Xu, Y.; Tso, S.K. (2002) A cap as interface for wheelchair control. *Intelligent Robots and System, 2002.* IEEE/RSJ International Conference, pp. 1439 - 1444 vol.2.

Shi, A. Gale, A. (2007) Environmental Control by Remote Eye Tracking. *COGAIN 2007*, Leicester, UK.

AI Support for a Gaze Controlled Wheelchair

Petr Novák, Tomáš Krajník, Libor Přeučil
Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University
Karlovo nam. 13, 12135, Prague, Czech republic
{novakpe, tkrajnik,preucil}@labe.felk.cvut.cz

Marcela Fejtová, Olga Štěpánková
Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University
Karlovo nam. 13, 12135, Prague, Czech republic
{fejtova, step}@labe.felk.cvut.cz

Other co-authors: Eduard Bakštejn, Zdeňka Lukešová, Tibor Strašribka, Jan Šourek, Pavel Štastný
(bakste1, lukesz1, strast1, sourej1, stastp3 @fel.cvut.cz)

Keywords
Smart wheelchair, hands-free control, environment sensing, robotics, safety

Introduction

Currently, one can distinguish two basic modes used for control of a wheelchair:

- In the direct mode, the wheelchair is driven in the same way as a car. The user indicates direction of the wheelchair movement using a steering wheel or a joystick and similar approach is applied for changing the speed.
- In the indirect mode, the user communicates with the wheelchair using a control panel reviewing the available predefined commands, e.g. "go 0,5 m forward", "turn 30 degrees right", … The control panel consists either of a set of different buttons suited for the particular user (HW solution) or it is represented by a GUI of a computer screen (SW solution).

The I4Control® system is a wearable system for gaze-computer interaction that is able to simulate the function of a joystick or to select from a grid-like structure using an appropriate GUI (Fejtová et al., 2006). Consequently, it can serve as a single input device for control of a wheelchair in both upper mentioned modes. This is certainly true from purely technical point of view. But this is not enough because safety of the resulting system has to be ensured! That is why special attention has to be given to the questions concerning reliability of the acquired signal and various ways it can be influenced or obscured. The fundamental danger related to gaze-based control comes from physiological reactions to certain stimuli that we have "build-in" to protect our eyes and even ourselves: we close eyes when a strong light flashes, we look into the direction of a loud sound, etc. It is a question how these immediate reactions can be distinguished from the intentional control signals the user wants to mediate to the controlled system. Moreover, the quality of the gaze control is significantly influenced by any change of light conditions for which the human eye has to adopt itself. Even if we use the best algorithms to evaluate point of gaze we can loose the control of the system for the time interval the human eye needs for adaptation to changed conditions. And this is not acceptable.

To make up for limitations of gaze control we have conducted several experiments with a wheelchair complemented by several AI features that have been developed in the field of intelligent mobile robotics. The resulting system is described in the section on an environment sensing system. Its functionalities seem to be useful not only for gaze-based wheelchair control but in a more general context. That is why we reconsider the indirect mode of wheelchair control and suggest its further refinement in the next section. In the conclusions there are mentioned some ideas for our future work towards construction of a smart wheelchair.

Environment Sensing System

To ensure safety of the wheelchair user and to support autonomy, the wheelchair has been equipped with a sensory system consisting of sonar and laser rangefinders, color camera and a notebook that conducts all necessary sensor data processing. The forward-looking color camera acquires images at 15 frames per second. The laser rangefinder is aimed to the front and provides a planar scan with 230° field of view and range of 4 m. Sonars are located at the back of the chair and are used to detect obstacles during backward movement. The wheelchair has been also equipped with a prototype odometric system previously developed for another project.

The safety is enforced by limiting the maximal speed of the wheelchair whenever nearby obstacles are detected by any of the aforementioned sensors. When moving forwards, the rangefinder scan is searched for objects closer than 1 m. If such objects are detected, the maximal speed is decreased and when such distance is 0.2 m, the wheelchair forward movement is turned off. Similar rules are introduced for backward movement and for sonar sensors. We plan to implement algorithm similar to insect-like navigation, where obstacle detection is based on optical flow computed from image sequence acquired by the camera mounted to the wheelchair.

The sensors are not used only for obstacle detection – their input is essential for construction of autonomous modes of navigation. So far, we have tested two algorithms based on data from color camera and one laser rangefinder based algorithm.

- First vision based algorithm (Kosnar et al., 2008) recognizes pathways in front of the wheelchair. The user first specifies, which parts of current image represent obstacles and what color has the path. The algorithm indicates, what trajectory will be followed. After the user confirms the trajectory, the wheelchair starts to move. While moving, estimated future trajectory is shown enabling the user to redefine obstacle and path colors on demand. Moreover, this algorithm can be used to create a graph like map of the environment. With this map, the driver can just specify required destination.
- Second vision based algorithm (Krajnik and Preucil, 2008) detects significant objects in the image, measures their positions and creates a simple description of the path the wheelchair follows. The description of the recorded path can then be stored in a corresponding database and later used to ensure autonomous traversal of the path by the wheelchair.
- Third algorithm incorporates laser rangefinder measurements into a two-dimensional map of indoor environment. After a reliable map is created, the path between any two reachable points on the map can be planned through dedicated AI algorithms. The wheelchair can safely follow the designed path provided upper mentioned obstacle detection is applied.

The refinements of indirect mode for wheelchair control

Let us consider the indirect mode ensured by a computer GUI. In this case the input is not limited to direct physical contact based on touch but it can be mediated by number of alternative interfaces including e.g. those applying gaze, voice or blow (Felzer and Nordmann, 2007).

The simplest approach offers the user to compose his/her journey from many elementary steps. GUI interface offers several buttons with corresponding labels for example: 1m forward, 2m forward, left 10degree, right 20degree. User selects appropriate button and the wheelchair performs requested action. In a more sophisticated setting the user can first design a sequence of elementary steps and finally give a command to perform them in one run. Here, the user must be able to interrupt movement of the wheelchair in any moment. This can be achieved either by using an independent input channel dedicated to this purpose or by specifying a special combination of the main control signals. This combination has to be such that it is highly improbable that the user executes it without specific intention. This type of wheelchair control is rather demanding and the movement takes a lot of time.

- The following options rely on incorporation of various AI features (Mandel et al., 2005) based on self-orientation and localization of wheelchair in world as well as on some methods of artificial intelligence (image detection, creating of map, smart localization, …). Those features we are currently applying have been briefly described in the former section. The first option the image from camera of the control system is displayed on user's screen. Control system detects (recognizes) some routes (footpath, road, …) and the user can select one of the offered possibilities. When selection is confirmed, the wheelchair starts to follow the requested path. Movement of wheelchair stops automatically whenever the control system detects some obstacle it cannot cope with itself.
- Further improvement is represented by the second option which incorporates learning. Suppose, the wheelchair has built-in a map of the environment it moves in and it offers a list of pre-created or learned paths. As soon as the wheelchair can identify its location, it is enough if the user selects his/her target position (for example: kitchen, bathroom, bedroom, …) and wheelchair can plan its journey to requested position itself by composing it from the parts listed among its ready-made paths.

The obstacle detection subsystem ensures that users' reactions to surprising stimuli do not negatively affect function and safety of the resulting system because wheelchair movement is automatically halted whenever the control system detects any serious problem (for example big obstacle).

Universal GUI for wheelchair control

Of course, the control system of wheelchair does not have to be restricted to a single option just described. User can make choice from the appropriate options according his/her actual location. In the home environment, it is possible to rely on pre-created paths and select target position, only. In structured outdoor environment (parks, pathways) it seems useful to use simple path recognition methods and in unstructured or otherwise complicated environment it is possible to use direct control of movement.

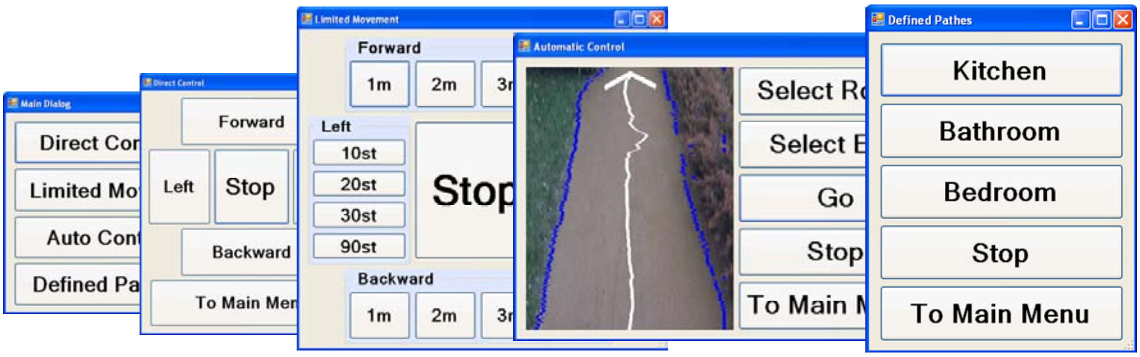


Figure 1. GUI interface of control system.

Moreover, the user can wish to switch among several input devices (buttons, eye movement recognition, …). To support freedom of choice while ensuring safety, the control system is divided into two parts. The first part includes mainly interface for input device and GUI interface (dialog with buttons) to select appropriate options / actions. This part also controls high level commands such as: go to target position. Second part takes care of autonomous movement of the wheelchair and ensures safety during the journey. This part performs movement commands (go, turn left, stop) and it checks permanently that the movement in intended direction is safe.

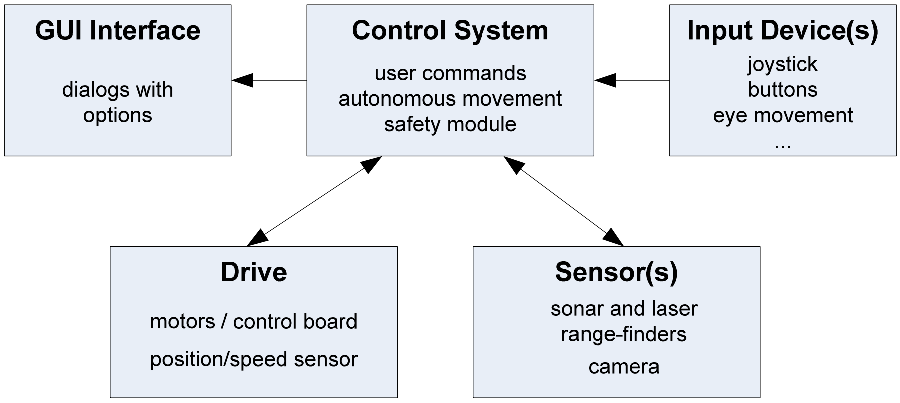


Figure 2. Block diagram of wheelchair system.

Conclusions

There are a number of AI algorithms that can improve wheelchair user's comfort and safety (Mandel et al., 2005). When considering them one has to take into account their time and memory requirements so that they fit the needs of the requested tasks and can be conducted by the HW available on the wheelchair (notebook in our case). As a next step, we are planning to implement a simple tracking program, which will simplify creation of the pre-defined paths: the user will be able to specify an object and the wheelchair will follow it, track its path and remember it. This approach will be used to support the learning based option mentioned as a refinement of the indirect mode of wheelchair control.

Acknowledgements

The presented research and development has been partially supported by the EU grant IST-2003-511598 COGAIN (Communication by Gaze Interaction) and by the Czech MSMT grant C06005 SYROTEK.

References

Fejtová M., Novák P., Fejt J., Štěpánková O. (2006): When can eyes make up for hands? *Proceedings of COGAIN 2006 'Gazing into the Future'*, pp.46-49.

Felzer, T., and Nordmann R. (2007) Consolidating Computer Operation and Wheelchair Control. *ASSETS 2007 - Proceedings of the Ninth International ACM SIGACCESS Conference on Computers and Accessibility, Tempe, AZ, USA*, ACM Press, pp. 239-240.

Felzer, T., and Nordmann R. (2007) Alternative Wheelchair Control. *RAT 2007 Proceedings of the International IEEE-BAIS Symposium on Research on Assistive Technologies, Dayton OH, USA*, IEEE Computer Society, pp. 67-74.

Krajník T., Přeučil, L. (2008) A Simple Navigation System with Convergence Property. *In Proceedings of European Robotics Symposium 2008*, Springer-Verlag, pp.282-292.

Košnar K., Krajník, T. Přeučil, L. (2008) Visual Topological Mapping. In Proceedings of European Robotics Symposium 2008, Springer-Verlag, pp.333-342.

Mandel, Ch., Huebner, K., Vierhuff T. (2005) Towards an Autonomous Wheelchair: Cognitive Aspects in Service Robotics. *Proc. of Towards Autonomous Robotic Systems (TAROS 2005)*, pp. 165–172.

A Comparison of Pupil Centre Estimation Algorithms

Detlev Droege
University of Koblenz-Landau
Universitätsstr. 1
56070 Koblenz
droege@uni-koblenz.de
++49 261 287 2769

Carola Schmidt
University of Koblenz-Landau
Universitätsstr. 1
56070 Koblenz

Dietrich Paulus
University of Koblenz-Landau
Universitätsstr. 1
56070 Koblenz
paulus@uni-koblenz.de
++49 261 287 2788

Keywords
gaze tracking, pupil centre detection

Introduction

The development of a low cost eye and gaze tracking system in Koblenz has been described previously (Droege et al., 2007). Given the low resolution of the relevant image parts the system suffers from insufficient accuracy with respect to the detection of pupil and/or iris center points and reflection (glint) centres.

To overcome this situation, several documented approaches have been implemented and evaluated to investigate their suitability for low resolution input data.

Algorithms

For pupil center detection, the following algorithms have been selected for investigation:

- the *Starburst* algorithm described in (Li et al., 2005)
- the *circle approximation* algorithm described in (Daunys & Ramanauskas, 2004)
- the algorithm of Pérez et al. (Pérez et al., 2003)
- the algorithm of Poursaberi et al. (Poursaberi & Araabi, 2005)
- the algorithm of Ohno et al. (Ohno et al., 2002)

All these algorithms are suitable for a dark pupil method based system like ours. They all perform an edge detection, possibly after some preprocessing steps, followed by different strategies to determine the center of an ellipsis or circle which fits to the detected edges around the dark pupil.

As an alternative to pupil center detection the estimation of the iris center can be used. While these methods are usually used for iris recognition systems, where the eye is held wide open and the whole iris is recognizable, gaze tracking systems have to deal with iris regions partially occluded by eye lids. But as these algorithms also detect the inner border of the iris, that is the pupil, they often can be adapted for gaze tracking purposes as well. The following methods have been examined:

- the algorithm of Li and Parkhurst (Li & Parkhurst, 2006)
- the algorithm of Zhu and Yang (Zhu & Yang, 2002)
- the algorithm Peng et al. (Peng et al., 2005)

- the algorithm of Daugman (Daugman, 2002)

Some of these algorithms explicitly require to remove any glints from the image before searching the pupil or iris center. While others are able to deal with distortions like glints, they usually perform better without such obstacles. Thus a preparative step to remove any glints is of advantage and can be used with most algorithms, provided no other distortions are introduced. These algorithms were considered:

- the algorithm of Li et al. (Li et al., 2005)
- the algorithm Mulligan (Mulligan, 1997)

Comparison

Given the goal to improve the performance of our existing system, the algorithms had to compare to the current implementation as described in (Geier, 2007).

To work with identical input data, several sequences of images have been recorded using our system. These were then replayed for every algorithm and the resulting coordinates have been recorded. Since the gaze direction estimation heavily relies on these coordinates, their accuracy is of major concern. To abstract from the subsequent calculations, which depend on the geometry of the setup (camera distance, screen size ...) only the positions in pixel coordinates have been compared. As the gaze estimation finaly is based on the direction and length of the line between the glint and the pupil center, this measure has been choosen as the value to compare.

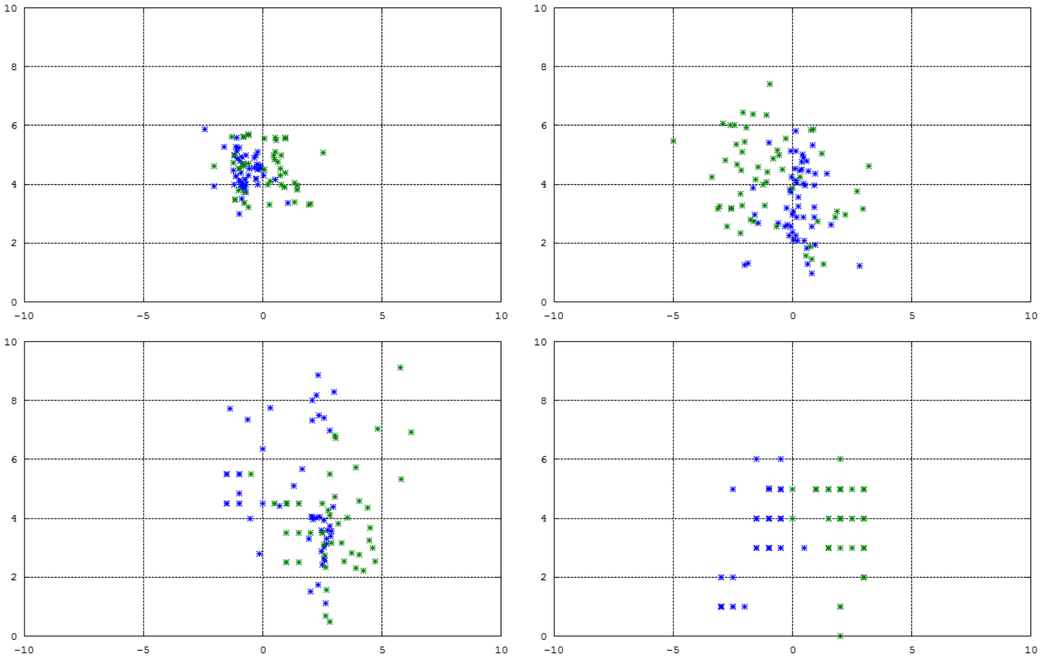


Figure 1: Results for the original algorithm, Starburst (top row), coordinate averaging and Daugmans algorithm

Figure 1 shows the results for an image sequence where a single point on the screen was focused. Any variation is due to the saccades occurring during fixation. Blue marks denote the left eye, green marks the right eye. All three tested algorithms show a much higher variation in the calculated positions opposed to the original. While Starburst and Daugmans algorithm suffer from the low resolution of the pupil image, coordinate averaging performs bad due to the presence of the glint, which heavily disturbs its results. Similar results can be observed with the other input sequences.

The other algorithms perform very similar to the original algorithm as can be seen in Figure 2, where the results of an image sequence where the eyes follow a slowly moving spot are shown.

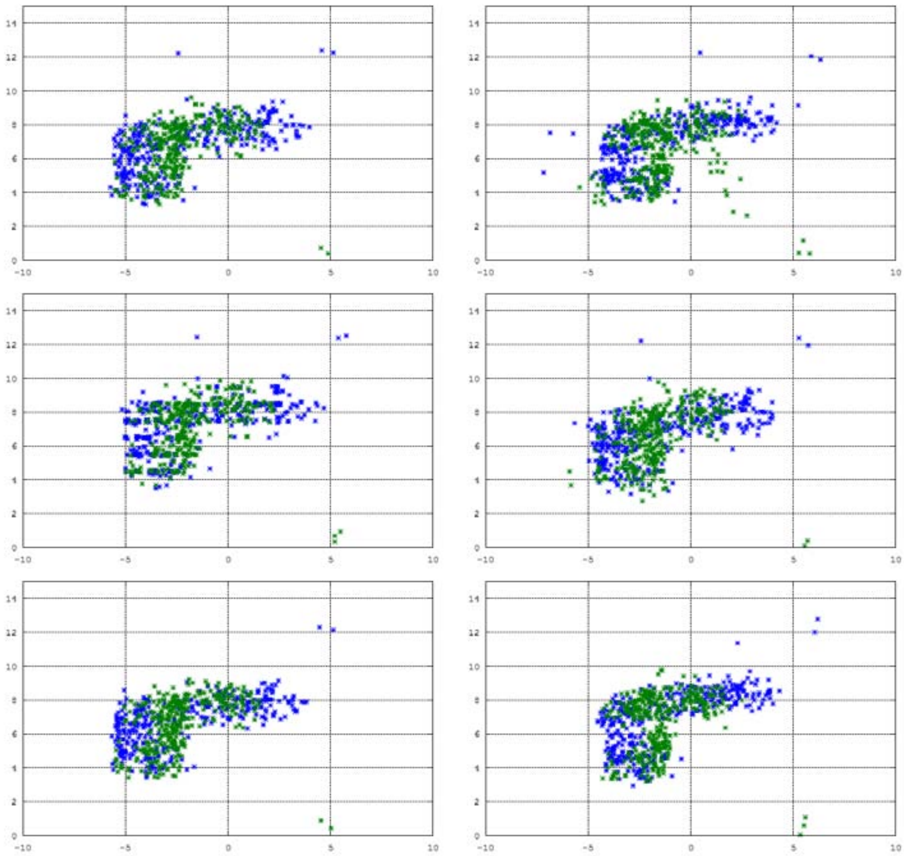


Figure 2: Results for the algorithms of Geier, Zhu et al. (top row), Poursaberi et al. and Perez et al. (middle row) Ohno et al. and Daunys et al. (circle approximation)

A quantitative evaluation is difficult, as no ground truth data exists to determine the detection error of the algorithms. A coarse estimation has been done by using an image sequence where the user followed a horizontally moving point. Allthough also saccades contribute to the real pupil positions, the mean deviation

algorithm	left eye	right eye
(Geier, 2007)	1.4508	1.1996
(Zhu & Yang, 2002)	1.8148	1.7803
(Poursaberi & Araabi, 2005)	1.3013	1.0860
(Pérez et al., 2003)	1.3842	1.2090
(Ohno et al., 2002)	1.2832	1.2341
(Daunys & Ramanauskas, 2004) Circle Approximation	1.2898	1.3303

Table 1: Mean vertical deviation for a sequence with a horizontally moving point

from the mean y-value has been calculated. The values in Table 1 give some hints on the accuracy of the algorithms, but the results are too close and vary too much to nominate a clear winner.

Conclusion

The aim of our comparison was to find an alternative to our pupil center detection, which in our previous system occasionally gave inaccurate estimates and thus was subject to improvement. While some of the algorithms published do perform slightly better in some situations, there is no clear "winner" for all situations. All algorithms suffer from the poor resolution of our chosen input device, some of them apparently are unable to deal with low resolution input at all. The differences are often caused by the presence of the IR-light reflection within the pupil.

Thus, besides hoping for higher resolution COTS cameras with high (IR-) light sensitivity, further attempts have to be made to tune the existing approaches to work better with low resolution input.

References

Daugman, J. (2002) How iris recognition works. *ICIP 2002, pp.* I:33-36.

Daunys, G. and Ramanauskas, N. (2004) The accuracy of eye tracking using image processing. *NordiCHI '04: Proceedings of the third nordic conference on human-computer interaction,* New York, NY, USA: ACM, pp. 377–380.

Droege, D., Geier, T. and Paulus, D. (2007) Improved low cost gaze tracker. H. Istance and R. Bates (eds.), COGAIN *2007, pp.* 37-40. `http://www.cogain.org/cogain2007/COGAIN2007Proceedings.pdf`

Geier, T. (2007) Gaze-Tracking zur Interaktion unter Verwendung von Low-Cost-Equipment. , Universität Koblenz-Landau, Campus Koblenz, Fachbereich 4: Informatik, Institut für Computervisualistik, Koblenz. (master thesis)

Li, D. and Parkhurst, D. J. (2006) Open-source software for real-time visible-spectrum eye tracking. H. Istance and L. Farinetti (eds.), COGAIN *2006 – gazing into the future*, pp. 18-20. (short paper) `http://www.cogain.org/cogain2006/COGAIN2006_Proceedings.pdf`

Li, D., Winfield, D. and Parkhurst, D. J. (2005) Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *IEEE Computer society conference on computer vision and pattern recognition* (CVPR '05) Washington, DC, USA: IEEE Computer Society, pp. 79–86.

Mulligan, J. B. (1997) Image processing for improved eye-tracking accuracy. *Behavior Research Methods, Instruments, and Computers,* 29(1), pp. 54-65.

Ohno, T., Mukawa, N. and Yoshikawa, A. (2002) Freegaze: A gaze tracking system for everyday gaze interaction. *Proceedings of the symposium on ETRA 2002: Eye tracking research and applications symposium, pp.* 125-132.

Peng, K., Chen, L., Ruan, S. and Kukharev, G. (2005) A robust algorithm for eye detection on gray intensity face without spectacles. *Journal of computer Science and Technology*, 5(3), pp. 127–132.

Poursaberi, A. and Araabi, B. 2005. A novel iris recognition system using morphological edge detector and wavelet phase features. *ICGST International Journal on Graphics, Vision and Image Processing, 05.*

Pérez, A., Córdoba, M., Garcia, A., Méndez, R., Munoz, M. and Pedraza, J. (2003) A precise eye-gaze detection and tracking system. *WSCG posters proceedings*, february 3-7, 2003.

Zhu, J. Yang, J. (2002) Subpixel eye gaze tracking. *FGR '02: Proceedings of the fifth IEEE international conference on automatic face and gesture recognition,* Washington, DC, USA: IEEE Computer Society, pp. 131.

Session 2: Broadening Gaze-Based Interaction Techniques

User Performance of Gaze-Based Interaction with On-line Virtual Communities

Howell Istance[1] **Aulikki Hyrskykari[2]** **Stephen Vickers[1]** **Nazmie Ali[2]**

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

[2]TAUCHI,
Department of Computer Sciences
FIN-33014 University of Tampere
FINLAND
ah@cs.uta.fi, nazmieali@gmail.com

Keywords
Gaze control, games, evaluation, virtual communities

Introduction

We present the results of an investigation into gaze-based interaction techniques with on-line virtual communities. The purpose of this study was to gain a better understanding of user performance with a gaze interaction technique developed for interacting with 3D graphical on-line communities and games. The study involved 12 participants each of whom carried out 2 equivalent sets of 3 tasks in a world created in Second Life. One set was carried out using a keystroke and mouse emulator driven by gaze, and the other set was carried out with the normal keyboard and mouse.. The study demonstrates that subjects were easily able to perform a set of tasks with eye gaze with only a minimal amount of training. It has also identified the causes of user errors and the amount of performance improvement that could be expected if the causes of these errors can be designed out.

Gaze driven mouse and keyboard emulation

The idea and the implementation of “Snap Clutch”, the gaze interaction technique used, is described more thoroughly elsewhere (Istance et al., 2008). In short, the principle is to use a gaze dwell to emulate a variety of input events. Four different emulator modes are available at any one time to the user, and a simple off-screen glance is used to change to one of these modes. A ‘mode’ in this context means a way of mapping between gaze data as inputs, and mouse or keyboard events as outputs. So, for example, a ‘pie menu’ mode can be constructed where the first gaze dwell generates a right button click at that position on screen. The second dwell generates a left mouse button click, then the mode is automatically de-activated, and so subsequent dwells generate no events.

The four modes can be chosen according to the actual application being controlled. In this study we used the following four modes:

- Glance up mode : Unconstrained looking around
    - no action on dwell (control off),
    - rotate left when looking inside the left hand edge of the screen
    - rotate right when looking inside the right hand edge of the screen
- Glance right mode: Right button click
    - a gaze dwell causes a right button click
- Glance left mode: Left button click
    - a gaze dwell causes a left button click

- Glance down mode: Locomotion
    - no action on dwell
    - constant streaming of ‘W’ keystroke events when the user looks in the main part of the screen
    - streaming of ‘A’ and ‘D’ keystroke events when the user looks into small square regions in the left and right hand sides of the screen
    - streaming of ‘S’ keystroke events when the user looks inside a thin strip inside the bottom edge of the screen causing the avatar to walk backwards

An approach to user performance investigations

The initial pilot study (Istance et al, 2008) showed that using Second Life with our gaze-based technique resulted in task completion times that were distinctively longer than when using conventional interaction techniques. In order to achieve parity of gaze interaction with normal keyboard and mouse, it is important to be able to identify usability issues with gaze control in terms of what influences the speed of interaction (time of task completion) and the types of errors made.

Partitioning task time into ‘productive’ time and ‘error’ time has long been a feature of usability engineering (Gilb, 1984). The time spent in a specific error condition represents the potential saving in task completion time if the cause of that specific error can be designed out so the user no longer makes that error. The relative savings in task completion times by addressing each of the types of errors identified represents a kind of cost-benefit analysis of redesigning different features of the user interface.

The Experiment

Subjects and apparatus. The study involved 12 participants. Ten of them were students and two were university lecturers who were experienced users of gaze interaction. Ages varied from 20 to 56, the average being 29. All subjects were able-bodied. The trials were carried out using a Tobii T60 screen integrated eye tracker, and the window contents during all of the trials were recorded using screen capture software.

Tasks. Two sets of three tasks were devised to be carried out within a purpose built world within Second Life. The world represented the computer science building at the university.

- The *locomotion task* required the subject to walk from the main entrance, up the main stairs (Figure 1), go into a room where there were display panels about individual modules and then report the module code from a particular panel. The difference between the two sets of tasks was the actual panel the participant was asked to report the code from.
- The *object manipulation* task required the subject to change a slide or request a web page from the main lecture theatre. In one task set, the participant changed the slide. This involved a right click on a panel button object to display a pie menu and then a left click to select the ‘Touch’ option. In the other set, the equivalent task was to request a web page to be displayed. This involved a left click on a panel

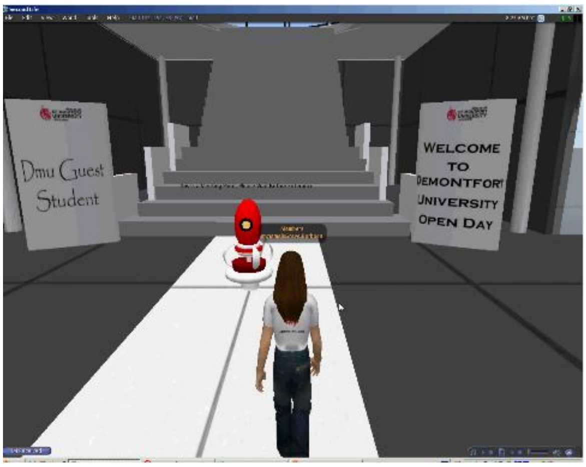


Figure 1. A locomotion task – searching for the target from the upstairs.

button object near the stage, and another left click to cancel to the request.

- The *application control* task required the user to change the appearance of their avatar. In one set, the task was to remove the moustache and in the other it was to raise the height of the eyebrows. The participant right-clicked the avatar to display a pie menu with an option to edit ‘appearance’. This caused a dialogue box to appear. The user had to select a group of features to edit from the vertical panel of buttons, then scroll down a list of features to make the required feature visible. A horizontal slider was used to change the selected feature.

Procedure. The subjects were split into two groups. One half did one task set with the keyboard and mouse, followed by the other task set using gaze control. The other 6 subjects started with gaze control followed by keyboard and mouse. None of the 10 student participants had used Second Life or gaze control previously.

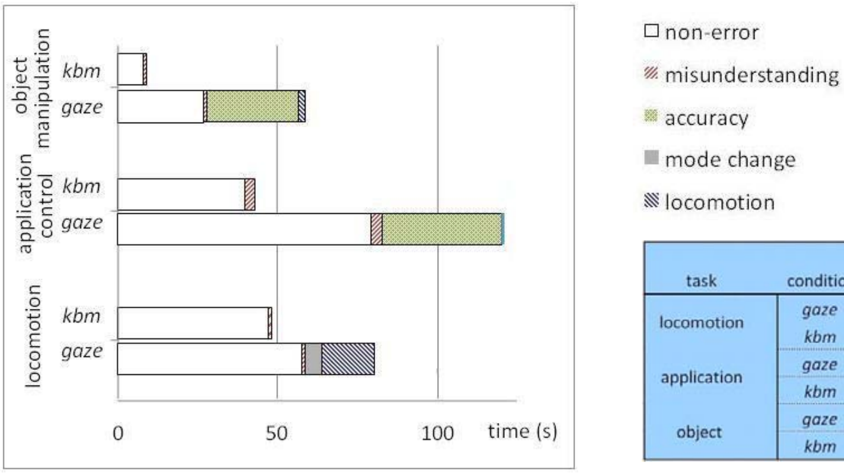
Each subject was given a 15 minute introduction to Second Life in the form of simple training exercises using a keyboard and mouse. This was followed by a 15 minute introduction to using gaze control. This also contained a series of simple training exercises.

Each task set began with the avatar standing by the main entrance to the building and the tasks were completed in the same order for all subjects, locomotion, object manipulation and application control. The task was first explained and then the subjects were asked to complete it. They were reminded of the tasks as needed during completion. After both task sets, subjects were given a brief questionnaire to complete. They were advised that they could withdraw from the trial at any time and that there were no penalty for doing so. No reward for participation in the trial was offered and the whole session took between 45-55 minutes to complete.

Analysis and Results

We identified four different categories of errors the participants made when performing the tasks. Videos of the trials were annotated using an open source video annotation application called Elan (ELAN, 2008). At the outset the data from one subject was marked up by two people and the consistency of the outcomes was checked. A number of minor adjustments to the original definitions of the error types were made as a result, otherwise there was a high degree of agreement between the analyses. The four main error catergories were the following:

1. Locomotion error – being of one of following
    - unintentional motion backwards (the gaze first moves through the ‘move backwards’ zone of the screen after glancing down to change into locomotion mode)
    - unintentional rotation - left or right (the person means to glance off screen to change modes when in ‘no control’ mode, but rotates instead)
    - turn overshoot (person deliberately turns while in ‘locomotion’ mode but turns too far and has to correct this.
    - walk overshoot (person tries to stop, but the change to ‘no control’ mode takes too long the person walks too far, and has to reverse)
2. Mode change error – an unintentional change of the mode; a subject tries to rotate while in locomotion mode – left or right – and changes mode by mistake by looking too far off screen
3. Accuracy error – a subject tries to click on a target but misses because of inaccurate pointing. If this resulted in the wrong selection being made this error included recovery from this.
4. Misunderstanding error - a subject misunderstands / mishears / forgets what to do – e.g. a subject goes in the wrong direction and later corrects the direction



task	condition	error-free time (s)	error count	error time (s)	err %
locomotion	*gaze*	57,97	4,55	21,40	22,68
	kbm	47,13	0,18	1,32	2,54
application	*gaze*	79,25	3,09	41,75	29,91
	kbm	40,15	0,45	2,73	5,95
object	*gaze*	18,99	2,45	21,89	50,05
	kbm	7,84	0,18	1,05	6,10

Figure 2. Average task completion times partitioned into error times (in four types of errors) and non-error times

For each subject and for each task, the time spent in each error condition was summed and this was deducted from the total task time, leaving the non-error time for each trial.

The outcomes of the trials for each of the three tasks are shown in Figure 2. Each task represented one example from the three main different categories of tasks performed in virtual environment (Hand, 1997; Bowman, 1999). Data from the locomotion task is at the bottom of the graph, the application control task data is in the middle and the object manipulation task data is shown at the top. The total lengths of the bars show the average total task completion times in seconds including errors.

There were significant problems with calibrating the eyetracker for one subject. She was able to complete all 3 tasks in the gaze condition but there were far greater accuracy errors than for any other subject (the error time was more than 3 standard deviations from the mean of all other subjects' error times for the application control and object manipulation tasks). Consequently all data from this subject was removed from the analysis.

The results show that all subjects were able to complete the three tasks using eye gaze. The non-error part at the bottom of the bars enables comparison of task times if the cause of the errors can be removed by design changes. The gaze:keyboard-mouse ratio of non-error times for the locomotion task is 1:1.2. The corresponding ratio for the application control task is approximately 1:2, and for the object manipulation task is approximately 1:2.5.

The error-free times in the gaze condition are encouraging, particularly for the locomotion task. With only a short training session, subjects would be able to complete the locomotion task using gaze nearly as quickly as with key commands if the cause of the locomotion errors could be removed. The reasons behind the locomotion errors are in part due to the speed of movement of the avatar in response to key commands generated by the emulator. This causes overshoot or undershoot of movement which then have to be corrected. This is largely due to the processing pipeline on a single computer (eyetracker – emulator – Second Life browser, and additionally, in the experiment condition, the video capture software). There may also be optimisations to the emulator software that could improve performance here. Another source of locomotion errors is the location of the backwards motion zone at the bottom of the screen. This meant that the gaze position first had to travel through this zone after changing into locomotion mode and the latency in the system caused an unwanted backwards movement as a result. These can be addressed by modifications to the behaviour of the locomotion mode and examining in detail the causes for response latency.

The biggest cause of errors in the application control and the object manipulation tasks was the difficulty of hitting the small control objects in the dialog boxes to change appearance. This was exacerbated by some latency in generating click events probably due to the processing pipeline. The best solution here to reduce these errors is probably to include some kind of zoom facility as is common with 2D gaze driven interfaces.

Subjects were asked in the post trial questionnaire to identify the most difficult aspects of the gaze control conditions. The majority of subjects said the application control task was the most difficult due to the lack of accuracy of gaze pointing. A smaller number said the slowness in generating click events, particularly in the object manipulation task, caused problems sustaining cursor position.

Conclusion

In summary, the study has been successful in revealing the extent and causes of performance differences between the gaze and keyboard-mouse conditions. It has enabled specific design changes to be identified to address these differences and has given an indication of the performance improvements that are likely to result from these changes. Importantly however, it has demonstrated the feasibility and potential for gaze based interaction with 3D virtual communities.

References

Istance, H., Bates, R., Hyrskykari, A., & Vickers, S. (2008). Snap Clutch, a moded approach to solving the Midas Touch problem. In *Proc. of the ACM Symposium on Eye Tracking Research & Applications (ETRA 2008),* ACM, New York, NY, 221-227, DOI= http://doi.acm.org/10.1145/1344471.1344523

Bowman, D., Johnson, D., and Hodges, L. (1999) Testbed evaluation of virtual environment interaction techniques. In *Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST '99).* ACM, New York, NY, 26-33. DOI= http://doi.acm.org/10.1145/323663.323667

Gilb, T. (1984) The "impact analysis table" applied to human factors design. In *Proc. Interact '84, First IFIP Conference on Human-Computer Interaction*, London, September 4-7, Vol. 2, pp. 97-101.

Hand, C. 1997. A Survey of 3D Interaction Techniques. *Computer Graphics Forum 16,* 5, 269-281.

ELAN (2008) A tool for the creation of annotations on video and audio resources. Freely available at http://www.lat-mpi.eu/tools/elan/

Multimodal Gaze Interaction in 3D Virtual Environments

Emiliano Castellina
Politecnico di Torino
Corso Duca degli Abruzzi,24
10129 Torino, ITALY
emiliano.castellina@polito.it

Fulvio Corno
Politecnico di Torino
Corso Duca degli Abruzzi,24
10129 Torino, ITALY
fulvio.corno@polito.it

Keywords

Gaze tracking, Multimodal Interaction, Experimentation, Games

Introduction

Nowadays the computer game industry is developing more and more innovative interaction and control methods for user inputs. Nevertheless Gaze tracking, that is a fast natural and intuitive input channel, is not exploited in any commercial computer game, yet.
In recent years several research groups started to study gaze tracking devices applied to computer games. In (Isokoschi & Martin, 2006) and (Isokoski et al., 2007) we find a comparison of different input methods, also including gaze tracking, for a first person shooter game. The study in (Dorr et al., 2007) shows that gaze tracking beats mouse control as input modality during a tournament of the classical Breakout game.
In (Istance et al., 2008) several uses modes to enable mouse emulation with gaze have been designed and tested avoiding the well-known *Midas Touch* problem. The methods proposed in that paper have been trialled in Second Life, an internet based 3D virtual world where users can interact with each other through avatars.
This paper present 6 different control methods for navigation and interaction in 3D games and reports a usability study on those techniques. Differently from previous works, the present research does not restrict attention to a particular technique or a particular application/game but it extends the evaluation to three different games, that require various skills and input schemes.

Control and Interaction Techniques

The control scheme for navigation and interaction in 3D Virtual Environments should allow the users to control their avatars, in particular the direction they are looking and the direction towards where they are moving. According to the application typology the game controller should allow more complex actions such as running, jumping, shooting and interacting with the environment objects.
Most Virtual 3D applications provide a control scheme based on a combination of keyboard and mouse inputs. Typically the gaze direction, called *Free look* or *Camera view*, is controlled by moving the mouse around, while the movement direction is controlled by keyboard.
In this context adding a further new input channel, additional and not alternative, such as gaze control, can revolutionize the interaction methods and user experiences. Our research has designed, developed and tested six different control techniques that involve gaze tracking and traditional input devices for navigation in 3D virtual environments.
Table 1 shows the control and interaction methods developed and tested by our research group.

Technique	Movements	Free Look	Actions
Multimodal interaction			
Gaze and keyboard (GTK)	K	G	G
Gaze and keyboard button (GKB)	K	G	K
Independent Gaze and Movements (IGM)	K	G	G
Gaze interaction			
Direct Gaze Control (DGC)	G	G	G
Virtual Keyboard (VK)	G	G	G
Gaze to Target (GT)	G	G	G
K = Keyboard	G = Gaze		

Table 1. Control and Interaction Techniques

Multimodal interaction

Gaze tracking and keyboard (GTK)

The user controls the *Free Look* by gaze interaction: when the gaze is directed to the 4 screen edges the camera view rotates toward the same direction with proportional speed to the gazed zone nearness to the borders. When the gaze direction comes back to the screen center the rotation stops. The gaze control allows also to interact with the objects in the environment. The starting and the termination of the rotation of the camera view and the actions activation are set by dwell time.
The keyboard is used to handle the movements of the user's avatar by arrow keys.

Gaze and keyboard button (GKB)

This control technique manages user's avatar movements and free look rotation with the same scheme of the previous method. The interaction with the environment is handled by a keyboard key (for example space) that replaces dwell time selection of the objects.

Independent Gaze and Movement (IGM)

Typically in 3D environment navigation scheme, free look and avatar movements are strictly bound, so the center of the camera shows the moving direction. This control scheme, instead, completely separates the control of movements from the control of camera. This behavior allows to simulate a person that walks in a direction and turns (right or left) his/her head. The direction of movements, controlled by keyboard, is indicated on the screen by an arrow, while the rotation of the camera is defined by gaze tracking.

Gaze interaction

Direct Gaze Control (DGC)

This method allows to control either the navigation or interaction in 3D environments by using only the gaze tracking inputs. The free look is managed with the same technique described above, whereas the forward movement is handled by selecting through dwell time the central zone of the screen (highlighted with a viewfinder). In this scheme the direction of navigation is strictly bound to the direction of the camera. In order to interact with the environment objects, contextual menus are displayed after a dwell time.

Virtual Keys (VK)

This scheme displays four semitransparent buttons in the middle of screen edges. The left and right buttons control the rotation of the camera while the upper and bottom button allow to navigate forward and backward. Each button is activated by dwell time selection.

Gaze to Target (GT)

This modality binds the user's avatar movements to predefined paths. The environment is enriched by anchor objects that define the locations that can be reached by the user. When the user selects by dwell time an anchor object then the avatar autonomously walks towards the selected point. After the anchor selection a confirmation menu is displayed n order to reduce *Midas Touch* error. The user can navigate the environment from anchor to anchor while the camera view is free and is controlled similarly to the GTK method.

Experimentation

The experimentation aimed to test the accuracy, speed and usability of the designed control and interaction techniques. It was divided in two phases. The first phase, involving 6 users, had the purpose to select the more promising techniques. The second phase extended the test of the selected methods to 15 users.
Three simple 3D games have been developed in order to test the control techniques. Those games use ETU driver (Bates & Spakov, 2006) to interact with Erica (*Erica System*, 2005), the eye tracker used for the experimentation. The first game shows a 3D home environment where the user should execute two kinds of task. In the first task (Figure 1(a)), the user has to navigate in the home and select a particular picture among the four pictures located in the environment. In the second task (Figure 1(b)), the user has to take the requested food from the fridge. The other two games aim at testing the pointing precision and speed, so the user is required to shoot target men in a shooting range (Figure 1(c)), and to shoot enemies, avoiding good guys, in a war path (Figure 1(d)).

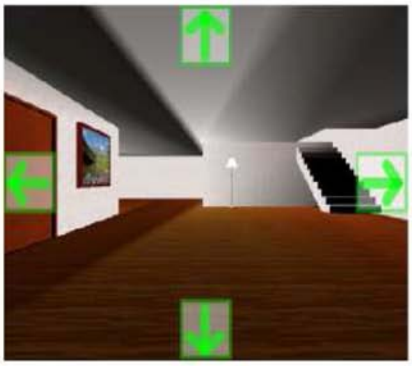
The users played each game for 6 minutes, divided in session of 30 seconds, while measuring their speed and precision.
The first round of preliminary tests highlighted that the more promising techniques were GKB, DGC and VK. During the second part of the experimentation the users tested the selected methods with the first game. The VK method has not been tested with the last three games because they did not allow free avatar movements but only free camera positioning.
Table 2 reports the precision percentage and the average elapsed time in the execution of the 2 tasks of the first game. The most precise technique was VK in both tasks, while the method that allowed the least elapsed time was VK in the first task and DGC in the second task. Table 3 shows a comparison of average elapsed time and precision in game 2 and 3 among DGC, GKB and the mouse. The mouse control allowed better precision in both games while the average elapsed time was equal for DGC and Mouse in games 2 and DGC had the least elapsed time in game 3.

Method	Find Picture		Take food	
	Precision (%)	Avg Time (s)	Precision(%)	Avg Time (s)
DGC	89	7.1	93	**8.2**
GKB	79	7.8	84	8.8
VK	**95**	**6.5**	**97**	8.6

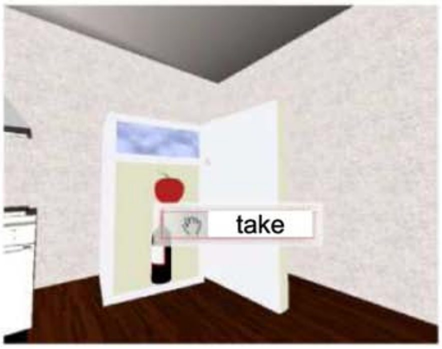
Table 2. Game 1 Test: Precision and Time

Method	Game 2		Game 3	
	Precision (%)	Avg Time (s)	Precision(%)	Avg Time (s)
DGC	68	**0.94**	45	**0.47**
GKB	49	1.04	52	0.67
Mouse	**90**	**0.94**	**68**	0.51

Table 3. Games 2 and 3: Precision and Time



(a) Game 1/Task 1



(b) Game 1/Task 2

(c) Game 2

(d) Game 3

Figure 1. ScreenShots of the games

At the end of the test the user filled an evaluation questionnaire in order to assess the usability of the proposed gaze-based control techniques and also to gather personal opinions and suggestions from the users. The analysis of the questionnaires shows that VK method has been perceived as the most accurate and fastest control type. User perceptions were clearly different to the real objective data reported in table 2 and 3 probabily because the amaze and the immersion provided by Gaze control give the players a more complete and better game experience that overcame the perfomance leaks.

Conclusions

The experimentation, proposed in this parer, showed a huge user interest about gaze based control applied to virtual 3D environment navigation and to game controlling. This kind of research aims at spreading the study and development of games and applications, based on gaze tracking device and addressed to common people. Spreading gaze tracking could have a relevant impact on the reduction of device costs. The decreasing of cost could also benefit the devices with a more noble purpose, i.e. the eye trackers used as Assistive Technologies for disabled people (Hansen et al., 2005).

References

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

Dorr, M., Bhme, M., Martinetz, T., & Barth, E. (2007). Gaze beats mouse: a case study. *The 3rd Conference on Communication by Gaze Interaction COGAIN 2007: Gaze-based Creativity and*

Interacting with Games and On-line Communities, 18–21.

Erica system. (2005). http://www.eyeresponse.com/.

Hansen, J. P., Hansen, A. S., Dan Witzner amd Johansen, & John, E. (2005). Mainstreaming gaze interaction towards a mass market for the benefit of all. *3rd international conference on universal access in human-computer interaction.*

Isokoschi, P., & Martin, B. (2006). Eye tracker input in first person shooter games. *The 2nd Conference on Communication by Gaze Interaction COGAIN 2006: Gazing into the Future*, 79–81.

Isokoski, P., Hyrskykari, A., Kotkaluoto, S., & Martin, B. (2007). Gamepad and eye tracker input in fps games: Data for the first 50 minutes. *The 3nd Conference on Communication by Gaze Interaction COGAIN 2007: Gaze-based Creativity and Interacting with Games and On-line Communities*, 11–15.

Istance, H., Bates, R., Hyrskykari, A., & Vickers, S. (2008). Snap clutch, a moded approach to solving the midas touch problem. In *Etra '08: Proceedings of the 2008 symposium on eye tracking research applications* (pp. 221–228). New York, NY, USA: ACM.

How Can Tiny Buttons Be Hit Using Gaze Only?

Henrik H.T. Skovsgaard
John Paulin Hansen
Department of Innovative Communication
IT University of Copenhagen
Rued Langgaards Vej 7, DK-2300, Denmark
{hhje; paulin}@itu.dk

Julio C. Mateo
Department of Psychology
Wright State University
3640 Colonel Glenn Highway
Dayton, OH 45435, USA
mateo.2@wright.edu

Keywords
Gaze interaction, assistive technology, cursor control.

Introduction

The limited accuracy of gaze trackers requires alternative methods to the point-and-click selection used in graphical user interfaces. Only a fraction of the interactive elements in Windows are actually critically small for current gaze tracking systems (i.e. less than 12x12 pixels), but they become serious obstacles for the workflow when using a "blunt" input. Increasing the size of icons or decrease the resolution of the screen may help in some cases, but the smallest elements on a computer interface may still create problems because of noise or off-sets on the gaze tracker. A two-step magnification method (Lankford, 2000) is provided by several commercial gaze communication systems to compensate for the inaccuracy. First, the user looks at the region of a screen in which the target is located. After a certain dwell time an enlarged version of the region pop up in a new window. The user can now make the selection in this enlargement with another dwell time selection. The method works well for most people, but the two-step process is likely to be slower than a single-step, direct selection.

Figure 3 The top-left Windows icons are taken from text typing application (16x16 pixels) and the top-right icons are common desktop icons (32x32 pixels). The three red targets (bottom) were used in the present experiment (size 6x6, 9x9 and 12x12 pixels).

Zoom–selection is a new method examined in this paper. When engaged, it presents a zoom-window around the user's point of regard in which a smooth animation shows the content of the window gradually increasing in size, as if approaching the user. The zoom-function allows for runtime course corrections during the selection process, proportional to the current level of magnification. At the end of the predetermined zooming time, the target in the centre of the zoom-window then becomes selected. Zoom-selection has been successfully used for gaze typing (Hansen et al. 2008). This paper examines zoom selection used for target selection in a windowed environment by comparing it to the two-step magnification method and – more briefly – to the simple dwell-time method. Ashmore et al. 2005 examined a gaze-contingent fisheye perspective for eye pointing and selection of magnified targets. The fisheye perspective (a so-called distortion interface) is hidden during visual search, but appears as soon as the user fixates a target. This technique provides an overview during search and the enlargement of targets during selection. However, we prefer to use zoom translations instead of a fisheye distortion since it keeps a better legibility in motion.

Experiment

In our experiment, the movement time (*Mt*) starts with the onset of the target and it ends when the user presses the space bar to trigger the selection process. With this approach, we also include reaction time in *Mt* to simplify the analysis.

All gaze-only interactive systems must discriminate between when users are navigating and when they are fixating. Their methods are unfortunately not standardized and the times it takes them to discriminate a fixation vary from tracker to tracker (Kumar et al. 2008, Salvucci and Goldberg, 2000). In order to avoid the uncertainties associated with the (unknown) device-dependent software fixation-detection techniques, we decided to initiate the selection process only when the user pressed a space bar. Consequently, selection time (*St*) is measured as the time it takes from the subject presses the space bar to the final selection process has been executed.

We performed three experiments to test three different gaze-only selection methods: simple dwell, two-step magnification and zoom-selection. Experiments 1 and 2 were conducted with a single target appearing in a circular fashion according to the ISO 9241-9 standard (ISO/DIS 9241-9, 2000), cf. figure 2a. In experiment 3 we presented several thousands tiny Windows icons simultaneously with just one blue target among them, cf. figure 2b. In all experiments the target would only appear when the user was fixating at a marked centre on the screen.

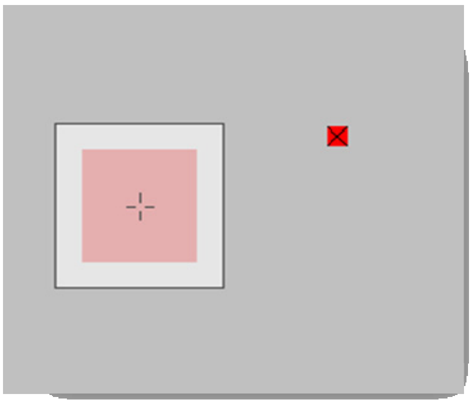


Figure 4a Screenshot from the two-step magnification selection. The red square indicates the target.

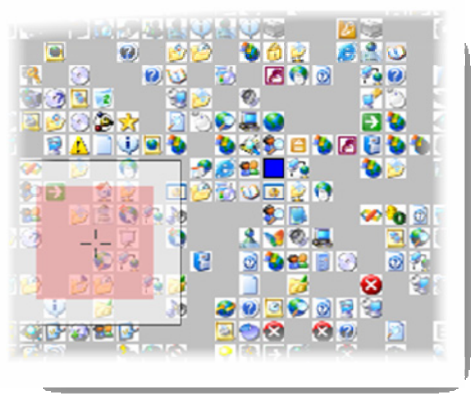


Figure 2b Screenshot of 2000 randomly spread icons used in experiment 3. The blue icon in the centre indicates the target.

Six participants (3 male, 3 female, mean age = 30 years) were recruited from the local university campus. The input device was a Tobii-1750 gaze tracker. The application ran on an IBM 1.86 GHz Intel Dual Core machine. The resolution was 1280x1024 pixels. The gaze tracker sampled at 50 Hz with a claimed accuracy of 0.5 degree on the visual angle. Roughly, this corresponds to 20 pixels in our test configuration.

The primary independent variable was **interaction technique** with three levels:

- Dwell: Provides a visual feedback on remaining dwell time by a pointer gradually decreasing its size from 68x68 pixels but without enlargements of the target.
- Two-step Magnification: A window (size 200x200 pixels) pops-up with a magnification (x5) of the gaze area at the point of regard when the user presses the space bar. In this window, the user can perform the second and final selection by another simple selection.
- Zoom selection: Pressing the spacebar triggers a graduate increase of objects within a window (size 300x300 pixels) until a final x10 zoom-level is reached.

We included additional variables to test a range of design options. Space constraints allow us only to report on the following variables in this paper:

- **Target size**: 6x6 pixels, 9x9 pixels and 12x12 pixels
- **Selection time:** In experiment 2 we used 1000 ms, 1500 ms, 2000 ms for the zoom selection interaction and 500 ms, 750 ms and 1000 ms for two-step magnification. In experiment 3 we used 1000 ms for the zoom selection and 500 ms for the two-step dwell magnification.

In order to compare the methods fairly, we decided to set the dwell times to half the zoom-time since the two-step magnification would take double as much time as the single-step zoom selection. Run-time corrections of pointer positioning were possible for both the dwell selection and the zoom selection.

To minimize asymmetric learning effects, the interaction method, target size, and selection time were counterbalanced using a balanced Latin Square. Furthermore, for both experiments, the mouse pointer was hidden to reduce visual distraction and prevent chasing. Audio feedback from the application informed the users about the outcome of their activation, playing a pleasant sound if the activation was a success and a warning sound if it went bad.

Experiment 1 consisted of 10 activations with the standard single-dwell activation technique using the ISO-format, with each of the 3 target sizes, (the largest 12x12 pixels shown first and the smallest 6x6 pixels shown last). The dwell time was set to 1000 ms. Experiment 2 was conducted immediately after experiment 1. Interaction method (two-step dwell magnification and zoom selection), target size (12x12, 9x9, and 6x6 pixels), and selection time (1000, 1500, and 2000 ms) were manipulated. The order of the conditions was counterbalanced. Ten activations were made with each of the combinations, giving 180 data per subject.

Finally, in experiment 3, conducted several days after experiment 1 and 2, we compared the two-step dwell magnification and the zoom selection technique. In this experiment, we used a windows-like layout with 2000 icons shown at once. Again, the sizes were 6x6, 9x9, and 12x12 pixels, but just tested with one selection time namely 1000 ms for the zoom selection and 500 ms for the two-step dwell magnification. A target would appear as the only blue icon among all the other 2000 small icons (c.f. figure 2b).

Results

Outliers (*Outlier* $> \mu + \sigma \cdot 3$) were first removed. This excluded 5 data of 180 in Experiment 1, 19 data out of 1080 in Experiment 2 and 16 out of 1080 in Experiment 3. We then performed ANOVA and Bonferroni post-hoc tests. Table 1 summarizes the results from all the experiments.

		6 x 6 pixels	9 x 9 pixels	12 x 12 pixels	Movement time (ms)	Selection time (ms)	Total time (ms)
Dwell (n=180)	**exp 1**	μ = 0.07, σ = 0.25	μ = 0.18, σ = 0.40	μ = 0.27, σ = 0.45	μ = 3374, σ = 3865	μ = 1000, σ = 0	μ = 4785, σ = 3878
Zoom (n=1080)	**exp 2**	μ = 0.30, σ = 0.46	μ = 0.44, σ = 0.50	μ = 0.57, σ = 0.50	μ = 1544, σ = 1205	μ = 1511, σ = 411	μ = 3055, σ = 1263
	exp 3	μ = 0.45, σ = 0.50	μ = 0.50, σ = 0.50	μ = 0.50, σ = 0.46	μ = 2019, σ = 1159	μ = 1000, σ = 0	μ = 3058, σ = 1195
Two-Step Dwell (n=1080)	**exp 2**	μ = 0.81, σ = 0.40	μ = 0.88, σ = 0.32	μ = 0.93, σ = 0.24	μ = 1422, σ = 966	μ = 2924, σ = 936	μ = 4346, σ = 1558
	exp 3	μ = 0.85, σ = 0.36	μ = 0.89, σ = 0.31	μ = 0.88, σ = 0.33	μ = 1678, σ = 798	μ = 2177, σ = 800	μ = 3865, σ = 1169

Table 1 Means (μ) and standard deviations (σ) of data.

Experiment 1: The grand mean of hit rates for a common dwell selection were 0.17, σ = 0.38 and the grand mean movement time was 4006 ms, σ = 4435. The mean hit rate for 12x12 pixels were 0.27, σ = 0.45, mean hit rate for 9x9 pixels were 0.18, σ = 0.40 and mean hit rate for 6x6 pixels were 0.07, σ = 0.25. The ANOVA showed a main effect of the target size on hit rate: $F(2, 179) = 4.269$, $p < 0.015$. This indicates that that the dwell selection method is rather imprecise, but it performs better when the target size increases.

Experiment 2: There was a main effect of interaction method on hit rate: $F(1, 1061) = 86.441$, $p < 0.0001$. The two-step dwell magnification was more accurate ($\mu = 0.88$, $\sigma = 0.33$) than the zoom selection ($\mu = 0.44$, $\sigma = 0.50$). There was a main effect of target size on hit rate: $F(2, 1061) = 19.155$, $p < 0.0001$. The post-hoc test revealed a main effect for all target sizes with p-values ranging from $p < 0.001$ to $p < 0.01$. The success rate for hitting a target gets significantly higher for each of the tested levels when target size increases.

The selection time is the time used for the activating the target after the subject triggers the space bar. Recall that the zoom selection was set to be twice as long as the dwell time for a fair comparison between the zoom-selection process and the two-step dwell magnification. The ANOVA showed a main effect on the selection method on selection time $F(1, 1061) = 1072.433$, $p < 0.0001$. Specifically, target selection took significantly longer with the two-step dwell selection ($\mu = 2924$, $\sigma = 934$) compared to the zoom selection ($\mu = 1504$ ms, $\sigma = 411$); in fact 94% longer even though the two methods had been balanced.

Immediately after experiment 1 and 2, the users evaluated the different interaction methods. They were asked which of the selection methods they perceived to be the fastest. A majority of the users (5 out of 6) found the two-step dwell magnification to be the fastest even though it was in fact 94 % slower. Not surprisingly, all of the users regarded the two-step dwell magnification to be most precise and all of them found the two-step dwell magnification to be easiest to use. Finally, the two-step magnification was also considered the most fatiguing by 4 out of 6 users; most likely related to the fact that this method requires multiple selections.

Experiment 3: In this highly clustered layout we found a main effect of selection method on hit rate: $F(1, 1073) = 215.1$, $p < 0.0001$, supporting the finding from experiment 2 that the two-step dwell magnification performs significantly better ($\mu = 0.87$, $\sigma = 0.33$) than the zoom selection ($\mu = 0.49$, $\sigma = 0.5$). Target size did not show a main effect. There was a main effect of selection method on selection time: $F(1, 1073) = 1086.141$, $p < 0.0001$. As in Experiment 2, the fixed zoom method was significantly faster ($\mu = 1000$ ms, $\sigma = 0$) than the two-step dwell magnification ($\mu = 2117$ ms, $\sigma = 800$).

Discussion and conclusion

The hit-rate of the zoom-selection were 0.44 and 0.49 in experiment 2 and 3, outperforming the conventional dwell-time selection technique that had a hit-rate of 0.17, while the two-step magnification method showed the highest hit-rate (0.87 and 0.88). The selection times of the zoom method were significantly faster than the selection time of the two-step method. Subjective evaluations from users indicated that the two-step method may be the most fatiguing, since it requires the double amount of clicks.

We have made a rough calculation of how long time it will take to perform 100 successful clicks on basis of the observed mean values of hit-rates and activation times. By dwell selection, the user would have to make 721 selections lasting approx. 5 seconds each, ≈ 60 minutes in total. With two-step dwell selections the user would have to make 124 selections lasting 5.8 seconds each, ≈ 12 minutes in total. Finally, zoom-selection would require 269 selections of 3 seconds duration, ≈ 13 minutes in total. Off course, it will most likely cause severe frustration when the user has to click repeatedly before hitting a target – and the user may even risk hitting a wrong nearby target. Therefore, this time calculation only indicates that the zoom selection may be equal to two-step magnification in terms of efficiency but not in terms of perceived usability.

Searching for lower bounds for how small targets the three methods would be able to handle, we also measured the distances from the final hit-points to the target centre. In average, the simple dwell activations were 16.7 pixels away from the real centre ($\sigma = 22.7$), the zoom-selections were 12 pixels off

(σ = 18.3) and the two-step magnification were 4.25 pixels off (σ = 8.8). Assuming that targets should be at least twice the size of the average offset to reliably hit the target, the results from the different selection methods indicates that a minimum target diameter would have to be around 34 pixels for dwell, around 24 pixels for zoom and 9 pixels for two-step dwell magnification if they are to consistently provide successful activations.

In conclusion, the findings suggest that both two-step magnification and zoom selection be included in gaze interactive systems. Because zoom selection are the fastest method and because it will handle the majority of Windows icons well (namely all of those larger than approximately 32x32 pixels), we suggest zoom selection to be the default method and the two-step dwell to be a second option that the user can engage when needed – i.e. when targets are really small or when the tracker becomes inaccurate.

Acknowledgements

This work was supported by the European Network of Excellence COGAIN, Communication by Gaze Interaction, funded under the FP6/IST programme of the European Commission.

References

Ashmore, M., Duchowski, A. T., and Shoemaker, G. 2005. Efficient eye pointing with a fisheye lens. In Proceedings of Graphics interface 2005 (Victoria, British Columbia, May 09 - 11, 2005). ACM International Conference Proceeding Series, vol. 112. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, 203-210.

Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices. International Standard, International Organization for Standardization (2000) ISO. ISO/DIS 9241-9

Hansen, D. W., Skovsgaard, H. H., Hansen, J. P., and Møllenbach, E. 2008. Noise tolerant selection by gaze-controlled pan and zoom in 3D. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (Savannah, Georgia, March 26 - 28, 2008). ETRA '08. ACM, New York, NY, 205-212. DOI= http://doi.acm.org/10.1145/1344471.1344521

Kumar, M., Klingner, J., Puranik, R., Winograd, T., and Paepcke, A. 2008. Improving the accuracy of gaze input for interaction. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (Savannah, Georgia, March 26 - 28, 2008). ETRA '08. ACM, New York, NY, 65-68. DOI= http://doi.acm.org/10.1145/1344471.1344488

Lankford, C. 2000. Effective eye-gaze input into Windows. In Proceedings of the 2000 Symposium on Eye Tracking Research & Applications (Palm Beach Gardens, Florida, United States, November 06 - 08, 2000). ETRA '00. ACM, New York, NY, 23-27. DOI= http://doi.acm.org/10.1145/355017.355021

Salvucci, D. D. and Goldberg, J. H. 2000. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications* (Palm Beach Gardens, Florida, United States, November 06 - 08, 2000). ETRA '00. ACM, New York, NY, 71-78. DOI= http://doi.acm.org/10.1145/355017.355028

Gesturing with Gaze

Henna Heikkilä
TAUCHI
Department of Computer Sciences
FIN-33014 University of Tampere, Finland
henna.heikkila@cs.uta.fi

Keywords
Gaze-based interaction, drawing application, gaze gestures

Introduction

This paper describes the first experiment of a PhD thesis project studying the use of gaze and gaze gestures in drawing applications. Drawing programs is an ideal platform for gaze gestures, since they can save screen space when fewer buttons or menus are needed in the application. Gestures also provide an easy access to a selection of often-used tools, for example copy, paste or undo. In some cases, gestures can even be performed faster than a selection from a menu. Some gaze-controlled drawing applications have already been implemented. However, none of these uses gaze gestures. EaglePaint (Gips and Olivieri, 1996) is one of the first drawing applications for gaze. In EaglePaint, a randomly colored line appears wherever the user looks. Hornof et al. (2004) have designed a drawing application called EyeDraw. In EyeDraw, the user draws pictures by using lines, squares, circles, and some predetermined shapes.

Gestures are mostly used in mouse and pen-based interfaces, but recently gaze researchers have become interested in them as well. Drewes and Schmidt (2007) and Istance et al. (2008) used gaze gestures to control an application. Drewes and Schmidt designed a set of gaze gestures that were based on eight directions (cardinal and half-cardinal points). See two example gestures in Figure 1. They implemented the gestures in a media player and used the gestures to play, pause and stop a track, to change a track or a media channel and to adjust the volume. They noted that the participants were able to do the asked gaze gestures and they used the whole screen to perform the gestures, which was reported to be easier than to perform them in small scale. Istance et al. added gaze-based interaction to Second Life and used gaze gestures to switch between different types of gaze and mouse controls.

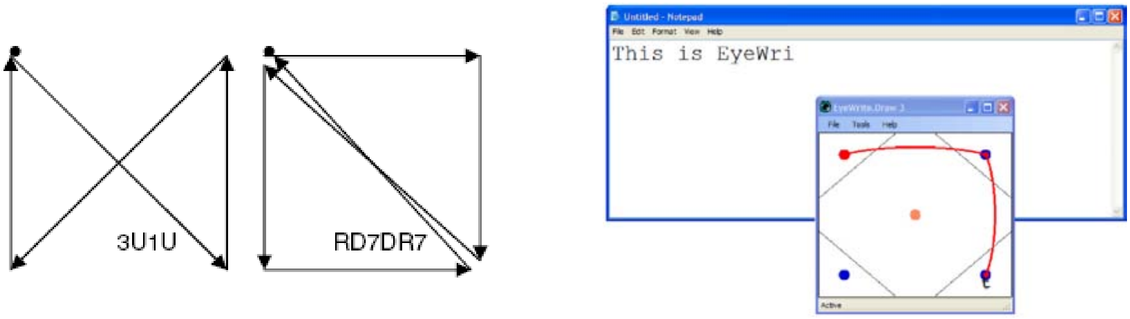


Figure 1. Example gestures Drewes and Schmidt's (2007) study (on left) and a screen caption from Wobbrock et al.'s (2008) EyeWrite (on right).

In addition to controlling an application, gaze gestures have also been used for writing with gaze. Wobbrock et al. (2008) implemented an application called EyeWrite (see Figure 1) that is based on their EdgeWrite method (for pen-based interaction). The shapes of the gestures in EyeWrite resemble roman letters so that they would be easier to learn and to remember. The gaze gestures are issued on a special writing window that shows guides to help the gesture drawing process. In their longitudinal study, Wobbrock et al. found that though the participants were slower when writing with gaze gestures than with dwell-based on-screen keyboard, they felt that the gaze gestures were an easier and faster way to write and caused less fatigue than the on-screen keyboard.

Another gaze writing style that resembles gaze gestures should be mentioned here. Isokoski (2000) designed a gaze writing prototype that is based on off-screen targets and a text input method called Minimal Device Independent Text Input Method (MDITIM). In MDITIM, the user writes using four principal cardinal directions (North, East, South, and West) and, in this prototype, these directions are marked on the monitor. The user writes by looking at the off-screen targets in a certain order. For example, when writing the letter y, the user looks at the off-screen targets in following order: South, East, South, and West. The letters create gesture-like gaze paths that the user can memorize.

Before designing gestures for a drawing application in detail, it is useful to know the usability of gestures in general. Usability here refers to users' ability to produce gestures and the system's ability to recognize them. As a first step, we carried out a controlled experiment.

Experiment

The goal of the experiment is to find out what kind of gaze gestures are easy and natural for the user. In this experiment, different shapes (such as lines, triangles, rectangles and circles) were presented to the participant and they tried to imitate them with their eyes.

Setup

For the test, 16 participants, 7 males and 9 females, were recruited. Their ages varied from 19 to 33 years. Four of the participants wore eyeglasses. Tobii 1750 eye tracker (with screen resolution of 1280x1024 pixels) and ClearView 2.6.3 were used during this experiment to track the participant’s eyes and to record the gaze data. The calibration was done with 16 points. With calibration, a test took about 30 minutes.

During the test, participants were asked to accomplish a set of tasks as fast and accurately as they could. Three of the tasks were done on an empty drawing area and another three tasks were done on a drawing area that had a model image in it. The tasks were:

- Draw with your gaze the letter L.
- Draw with your gaze a triangle.
- Draw with your gaze a line that goes from left to right and return to the starting point.
- Draw the form of the green rectangle (by looking at each of its corners one by one) with your gaze.
- Draw the form of the green circle (by following the circumference of the circle) with your gaze.
- Look once at each end of the green line.

All tasks were done with both small (250x250 pixels) and large (1180x920 pixels) drawing area. All tasks were repeated five times. In total, each participant did 60 tasks. The task order was balanced between participants.

Preliminary results

The analysis of the data is still ongoing and only preliminary results and some notes from the experiment can be presented at this point. The aim is to finish the analysis by September 2008.

In most cases, the shapes drawn during the tests resemble the asked shapes, at least when a human examines them. However, it is much more difficult for an application to recognise the shapes when the letter L resembles more the letter V. Participants commented that it was difficult to draw the shapes on almost blank surface. Especially, when the starting and the ending point of the shape were the same (e.g. triangle), it was difficult to remember where the drawing of that shape started, since there were no cues on a blank surface. Some participants used the borders of the drawing area as a help when drawing triangles and horizontal lines. This strongly suggests that the user needs some guides when doing gaze gestures. EyeWrite already uses guides to help the user to hit the target areas of the drawing window (Wobbrock et al., 2008). Drewes and Schmidt (2007) also offered helping lines in their test application, but the participants did not use them. However, the participants did use the whole display area for their gaze gestures and probably used the sides and corners of the application window or the display as a help.

After the test, most participants reported that the circle was the most difficult shape to draw. They said they had to concentrate more when trying to follow the curving form of the circle (this means more fixations on the gaze path, see Figure 2). When the task completion times are compared, the task concerning the circle shape done on the large drawing area stands out since it took about one and a half times as much time to complete the task than the second slowest task (rectangle shape done on the large drawing area). However, when comparing circle and rectangle shape tasks done on small drawing area, the task completion times are about the same in both tasks. These results are similar to the results from Tchalenko's (2001) studies. He discovered that curves are very difficult to draw because the user has to concentrate to achieve a better control of his eyes.

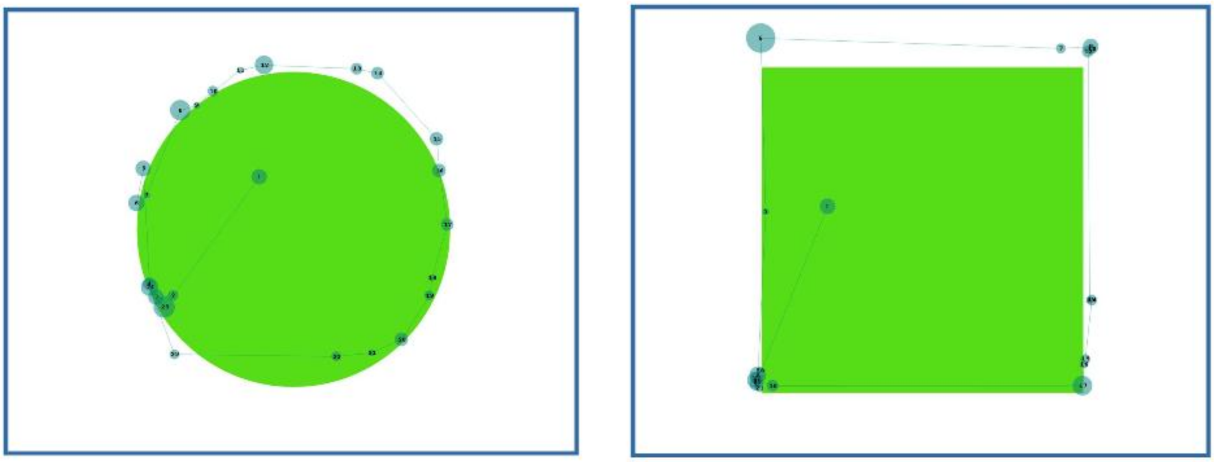


Figure 2. One participant's gaze path, when trying to follow the circumference of the circle (on left) and when trying to hit the corners of the square. Both show the calibration inaccuracy problem.

Next steps

The analysis of the gathered data continues. First, the accuracy between the gaze path and the model will be calculated (for tasks done with a model). Though the calibration was done with 16 points and its result was excellent, the drawn gaze paths have some systematic inaccuracy, especially on the upper part of the display (see Figure 2). This shows how vulnerable gaze pointing is to calibration errors. Next, the gesture duration and the distance that the gaze travels when performing a gesture will be taken under examination (for tasks done on an empty drawing area). This step relates to a challenge that the gaze gestures face: how to determine when the gesture starts and when it ends? Afterwards, an algorithm that recognises the gaze gestures will be created.

Conclusion

The experiment described aims to shed light on which kind of gestures would be easy to do with gaze and which kind of algorithm is needed to recognise the performed gaze gestures. Results will help when designing a set of gaze gestures that are suitable for a drawing application and for its tools and functions. When the set of gaze gestures is ready, they will be tested with users and compared against other available methods.

References

Drewes, H. and Schmidt, A. (2007) Interacting with the Computer using Gaze Gestures. In *Proceedings of INTERACT 2007*. Lecture Notes in Computer Science 4663, Springer, pp. 475–488.

Gips, J. and Olivieri, P. (1996) EagleEyes: An Eye Control System for Persons with Disabilities. *The Eleventh International Conference on Technology and Persons with Disabilities.* Available in http://www.cs.bc.edu/~eagleeye/papers/paper1/paper1.html.

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.

Isokoski, P. (2000) Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications* (ETRA '00). ACM Press, New York, NY, pp. 15–21.

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* (ETRA '08). ACM Press, New York, NY, pp. 221–228.

Tchalenko, J. (2001) Free-eye drawing. Point: Art and Design Research Journal, 11, pp. 36–41.

Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. (2008) Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (ETRA '08). ACM Press, New York, pp. 11–18.

NeoVisus: Gaze Driven Interface Components

Martin Tall
Varmdogatan 72, 257 33 Rydeback, Sweden
m@martintall.com
+46730611677

Keywords
Gaze Interaction, Midas touch, Target areas, Saccade selection

Introduction

The goal for this work has been to venture into novel interaction methods and implement these in reusable Graphical User Interface (GUI) components. My intention has been to create an interaction style that relies more on the specific properties of the human visual system, in which movement comes at a more constant and lower cost compared to moving a physical modality. Due to the proximity to natural human behavior this type of interaction should be very easy to learn. There is no new physical modality that the user has to map his or her intentions onto. Gaze interaction offers room for novel interaction techniques where objects appear or change when the user looks at them, without necessarily leading to a command execution. These reusable and configurable GUI components developed offer rapid development of future gaze driven applications.

Gaze Interaction Interface Components

The use of gaze data for interaction with computers is fundamentally different from more traditional computer interaction since there is no input modality (such as the mouse) to be acted upon. This requires specific interaction methods. Due to the physiology properties of the eyes a fixation covers an area of the screen that is larger than a traditional mouse pointer. Eye trackers will never be able to discriminate a gaze position for some of the smaller User Interface (U.I) components used in current interfaces. Hence, most of the existing applications for mainstream operating systems such as Microsoft Windows to be ill suited for gaze interaction. Additionally, the gaze data provided by the eye tracker is noisy and jittery. This factor has to be accounted for when designing gaze driven interfaces.

The commonly used dwell times creates a interaction style that is stressful to use since everywhere the user looks a command seems to be activated. This issue, known as the "*midas touch problem*" (Jacobs et al., 1993), enforces a constant roaming of the eyes. For example, the variance in text length displayed on buttons leads to involuntary activation on items that contain longer and thus more time consuming text strings. By further developing the use of target areas (Ohno, 1998) and displaying these dynamically hopefully the midas touch problem can be alleviated. This result in components that will display options only when the user is looking at them, providing a direct interaction style based on the contextual position of the users gaze. To handle the noisy and jittery gaze data I intend to use target areas that are larger than the buttons and icons used, this enables the gaze to remain on the target.

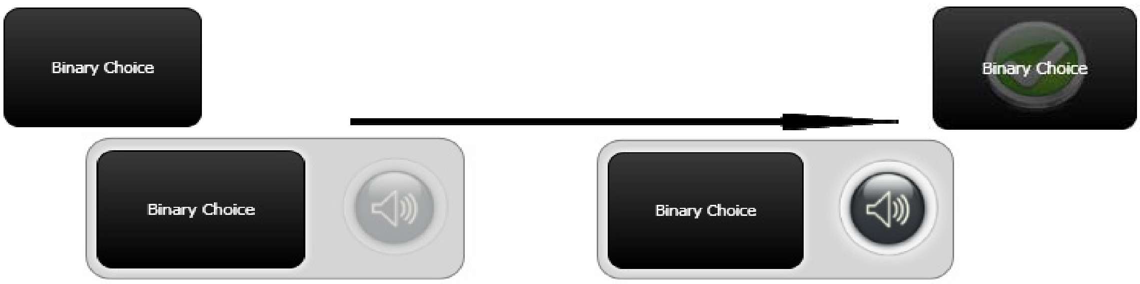
Implementation

When working with gaze as the only input, the midas touch problem as described earlier becomes a major issue. The behavior of the components has been shaped to reduce this as much as possible. In this case that includes a dynamically expanding target area which is activated by a fixation and creates a layer on top of the other components when activated and “rolled out”. Erroneous activations are reduced since the selection icons are not displayed on the interface in its original state, additionally a fixation on a button or menu does not cause a command to be issued (since a second saccade is required for performing the activation). When looking away from the component the activation icons are dynamically hidden from the interface which could reduce the error rate.

The Binary Choice Selection Button

This component resembles the traditional On/Off button where an option can either be selected or deselected, hence the name binary choice of either one or zero (true or false). The component was developed since the placement of text on dwell time activated icons causes involuntary activation (midas touch). The variances in length of the text of various buttons make the dwell time activation highly unstable. In other words, a button containing three words will more often be accidentally activated compared to a button with on one word.

1. Initial state *4. Selected state*



2. On fixation, opaque icon appears (speaker) *3. Fixation on the icon (opacity removed, glowing border)*

Figure 1. The Binary Choice component. Upon gaze entering the component a opaque layer expands to the right, reveling the saccade icon (2, shown as a speaker). A growing white border indicates the activation process (3). The changed/selected state is then indicated by the background of the component (4). The speed of the rollout and the activation threshold is configurable.

Fig. 2. Binary Choice, the target area (right box) is larger than the actual saccade/selection icon. It raises the tolerance for jitter by reducing the effect of noise from eye tracker

The Radial Saccade Pie Menu

The idea behind the component is to make use of dynamic allocation of the display area as well as providing a novel interaction method for activation. Upon fixating the button a set of icons are displayed at the top, left, right and bottom of the ellipse. An activation can then be performed by making a short saccade any of the selection icons. Since the second stage icons are displayed within the *parafoveal* field of view and always positioned at the same location (top, bottom, left and right) the user can effortlessly make a saccade to the desired icon. This reduces the chances of accidentally activating a command compared to one step dwell activation.

As the user becomes more aquatinted with the interface the activation times can be reduced, proving a fast and adaptable activation.

Fig. 3 The Radial Saccade Pie Menu. Upon gaze entering the component a opaque ellipse expands from underneath the button. Four icons appears on the ellipse. A fixation starts the activation process which is indicated by a glowing border. Both the expansion time and activation time can be configured. The number of icons used is optional between 1-4.

Prototypes

Each prototype uses the custom developed component and aims at evaluating their performance in tasks that are real world oriented, such as playing music or viewing pictures. The first prototype built is a gaze based version of the classic Memory card game. The goal of the game is to memorize the location of cards to find matching pairs. The purpose of the second prototype is to build a gaze based photo gallery. When the user fixates one of the photos the size of the canvas area expands providing a zoom effect. In this mode an additional menu bar is rolled out at the bottom of the panel. This menu houses a dwell icon that, on activation, brings the photo into full viewing mode. By enlarging a photo which the user is actively looking at the screen real-estate can be used in a more effective and dynamic way. The third prototype is the music player, it utilizes all of the components to create a music library which can be navigated by gaze alone. By navigating through the library a play-list featuring one or more songs from multiple artists/albums can be constructed.

Fig. 4. The third prototype, the music player is using all of the interface components.

Evaluation

The evaluation was divided into a sequence of tasks that were especially developed for the purpose. The first two steps of the evaluation concerns the performance of the individual components. The configuration of the components in terms of both interaction speed (feedback) and activation threshold (dwell) was configured in three modes. The three configurations had animation times of slow (500 ms.), medium (300 ms.) and fast (10 ms.) which means virtually no delay and causes the selection area to appear as soon as the gaze entered the component. In the same way the selection time (dwell) for each choice was configured with the same variables, hence the naming of the configurations are *long 500+500, medium 300+300* and *short 10+10*.

The second part of the evaluation regards the prototype applications which the participants were free to explore. The *user satisfaction* was measured by handing out two forms at the end requesting subjective opinions on the interface concerning the navigation, design, feedback, ease of use and stability. The Q.U.I.S (Chin et. al., 1987) and the IBM Computer Usability Satisfaction Questionnaires (Lewis, 1995) were used.

Results

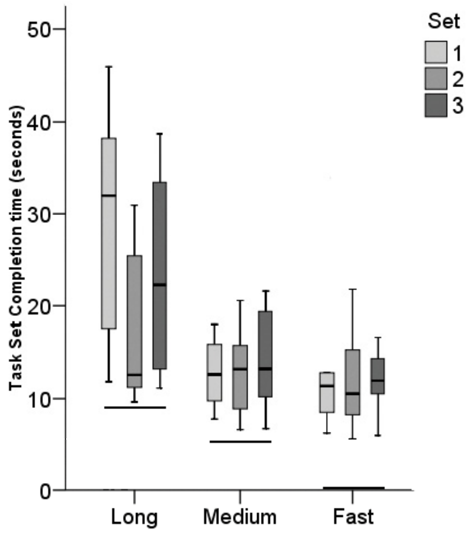


Fig. 5. Task set completion time

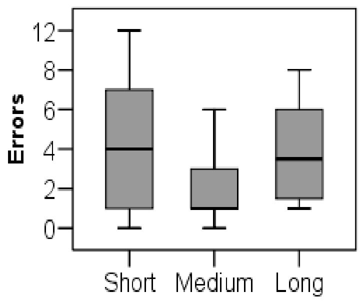


Fig 6. Binary Choice. Error rate

Binary Choice Component

The short temporal configuration (10+10) had a mean completion time per task set (selecting nine components) of 12 seconds with a standard deviation of 6 seconds compared to medium configuration (300+300) which have a mean time of 16 seconds with a standard deviation of 12 seconds. Finally the long activation time (500+500) produced a mean task completion time at 18 seconds with the standard deviation of 13 seconds .

Per indivdual component the short configuration had a mean activation time of 1 second, while the medium provided a mean 1.2 seconds. The long configuration displayed activation times well above the 500 ms (animation) + 500 ms dwell time required to perform a selection, when displaying a mean individual activation time on one and a half second.

Error rates are defined as the number of selections that exceed the nine needed to complete each task set. The highest error rate was found to be for the short configuration which also had the highest variance. The average mean was *short* 4.03 (SD=3.7), *medium* 1.71 (SD=1.6) and *long* 3.9 (SD=2.6). The bars in figure 6 show the mean average error rate over all sets in the three configurations.

Fig 5. Task completion times across the different configurations. The horizontal line indicate the theoretical time needed to accomplish the task.

Fig 6. Error rate for the different configurations. The short bar represents errors for the 10+10 millisecond configuration, medium equals 300+300 ms. and long 500+500 ms.

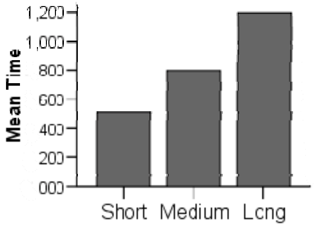


Fig. 7. Individual selection time.

Radial Saccade Pie Menu

The measures of time from gaze entering the component until a selection has been performed. Looking at the different configurations the short configuration (10+10) had a of mean 0.51 seconds with a standard deviation of 0.32 second. The medium configuration (300+300) delivers mean of 0.8 second (SD = 0.24) While the long (500+500) configuration of the component produced a mean of 1.2 seconds (SD = 0.3).

Response to the prototype applications

The majority of the participants found the interface to be stimulating and fun to use. All participants who were successfully calibrated and completed the two first steps in the evaluation were able to use the prototype application with none or very few instructions. The interface was perceived as clear, well structured and a majority was satisfied with how easy it was to use the system. The most prominent source of dislike for the interface came from offsets in the calibration which consistently led to higher error rates, longer task completion times and lower ratings in the questionnaires. The accuracy of the gaze position is essential for a positive experience. Using gaze interaction with a constant offset is cumbersome, this factor is represented by the high variance in frustration levels. These indicators correlates with the physical load participants experienced and further with the overall satisfaction of the interface. An offset creates a situation where no activations occur even if the participants reported starting at the components.

Future work

As the core technology of eye tracking more accessible a rich set of interface components is one important area in making gaze interaction more widespread. Future versions of the NeoVisus component library is likely to concern range selection, markers, text entry, communication and media functions, etc. The wide range for computer usage today requires flexible building blocks for rapid application development.

References

Chin, J. P., Diehl, V. A, Norman, K. (Sept. 1987) Development of an instrument measuring user satisfaction of the human-computer interface, Proc. ACM CHI '88 (Washington, DC) 213-218. CS-TR-1926, CAR-TR-328

Jacob, R.J.K. “The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look at is What You Get,” ACM Transactions on Information System, pp. 152-169, 1991

Lewis, J. R. (1995) IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. International Journal of Human-Computer Interaction, 7:1, 57-78.

Ohno, T. (1998). “Features of Eye Gaze Interface for Selections Tasks”. Proceedings of The Third Asia Pacific Computer Human Interaction – APCHI ’98. IEEE Computer Society. 1– 6

Session 3: Focusing on the User: Evaluating Needs and Solutions

Evaluations of Interactive Guideboard with Gaze-Communicative Stuffed-Toy Robot

Tomoko Yonezawa, Hirotake Yamazoe, Akira Utsumi, and Shinji Abe
ATR Intelligent Robotics and Communication Laboratories
2-2-2 Hikari-dai Seika, Kyoto 619-0288, Japan {yone,yamazoe,utsumi,sabe}@atr.jp

Keywords

Stuffed-toy Robot Guide, Gaze-communicative Guideboard, Wide-area Gaze-tracking

Introduction

In this paper, we introduce and evaluate an interactive guideboard with a communicative stuffed-toy guide-robot that behaves in correspondence to the user's gazing direction. The proposed system adopts our remote gaze-tracking method, which estimates the user's gaze angles based on image processing. The main purpose of this research is to provide intuitive and natural guidance interaction through an information board system that functions as if it were with a human guide who cares not only about the user's active attitude but also about the user's gaze. The system is applicable to both normal and *disabled/unmotivated* people who do not/cannot make an utterance but need some detailed information from the guideboard. It is also able to provide detailed information on any particular region of the guideboard corresponding to the subconscious interest of the user determined by her/his gazing direction. Our proposed interactive guideboard system combines the effective features of voice guidance following *the user's gaze*, based on our remote gaze-tracking method, and the impact of *anthropomorphism* using various non-verbal expressions (Duffy, 2003, etc.), including gazing behaviors (Fukayama et al., 2002, etc.), while adopting a scheme of human-human communication (Kendon, 1967).

We built a tentative interaction model for a gaze-communicative guide robot that performs in the following three steps: a) *drawing the user's interest or attention*, b) *attracting the user with appropriate communications*, and c) *guiding the user to the detailed information corresponding to the direction of the user's conscious/subconscious gaze*. Based on this model's three stepwise interaction modes, we implemented the interactive guideboard with a gaze-communicative stuffed-toy robot. In this paper, we show the experimental results on the effectiveness of i) *the presence of the anthropomorphic guide robot* in the system, ii) *the attentive behavior* (turns of the robot's head) to the guideboard, and iii) *the gaze-corresponding guidance to the appropriate information based on both the direction of the robot's head toward the content and the information given by of the vocal guidance.*

Related Research

Many research efforts on gaze applications have considered the user's gaze as a conscious control interface for disabled people (Shi et al., 2007, etc.). In contrast to these works, we apply the user's gaze to detect both subconscious and conscious desires for further information; moreover, our approach does not require the user to wear any fixture.

On the other hand, there has been much research on communication robots that utilize gaze. Several studies have examined the social recognition/interactions between humans and robots related to gaze

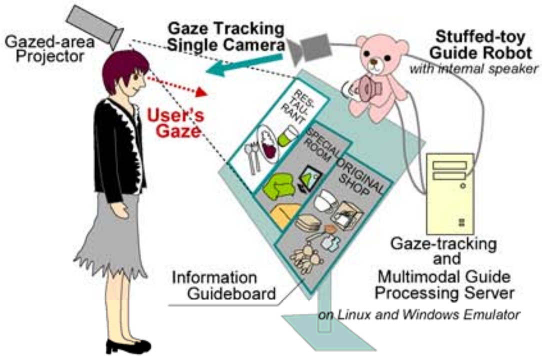


Figure 1. System Structure

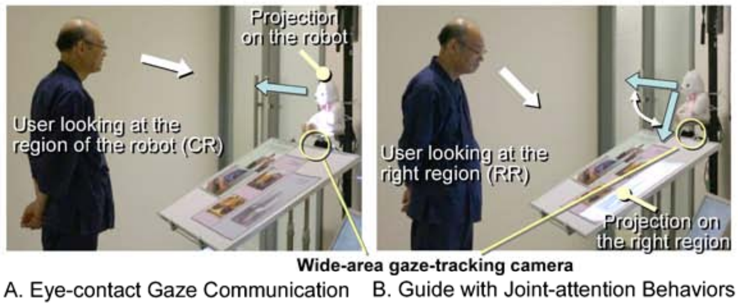


Figure 2. Operation View of Two-mode Guide System

(Breazeal et al., 2005; Sidner et al., 2005, etc.). These works primarily dealt with the facial and gazing behaviors of robots in imitative interactions based on a particular model. However, the appropriate application of joint attention, which is an important attentive component of gaze behavior, has not been sufficiently discussed.

Many research works have shown the importance of anthropomorphic agents in natural/intuitive guidance and assistance to people for various purposes, such as personal guide system (Sumi et al., 1998, etc.) Meanwhile, Katagiri et al. showed how the different behaviors of an agent differently affect the user's knowledge and performance (Katagiri et al., 2001). In this paper, we aim to adopt a kind of persuasive power of the robot's gazing behavior corresponding to the user's gaze and thus attracting the user's gaze toward an intuitive guide system based on our primary model of gaze communication (Yonezawa et al., 2007).

Structure of Gaze-communicative Guide System

Gaze-communicative Guide Robot

To provide natural and intuitive guidance through stepwise interactions, our gaze-communicative guideboard system is implemented with a robot agent in three different reactive modes corresponding to the user's gaze. First, the robot tells the user that it can give guidance on the details of the guideboard, corresponding to the user's gaze, when her/his face is detected. Next, when the user looks at the robot, the robot first reacts to the eye contact as initiation of communication, and then it tells the user the guidance information. Finally, when the user looks at some particular region of the guideboard, the robot provides the hierarchically composed information corresponding to the duration of gazing at the region. The robot also behaves as if it is looking at both the board and the user to express an anthropomorphic effort and intention to give guidance.

The guide system consists of a guideboard, our gaze-tracking system, an illuminating projector, and a stuffed-toy guide robot featuring vocal, assistive and gazing behaviors. The guideboard is divided into three regions, and its assumed scenario is guidance in a hotel lobby. This hotel guideboard has the following parts: the left region (LR) shows information and pictures of the restaurant, the center region (CR) describes specially equipped rooms, and the right region (RR) displays information on the hotel's in-house shop (Figure 1). Figure 2 shows examples of the system operations. The stuffed-toy robot reacts to the user's gaze toward it as the initiation of eye-contact communication (Figure 2-A) and provides the detailed information on each region corresponding to the user's gazing direction (Figure 2-B).

We conducted a demonstration experiment in the lobby of a hotel and found that the users operated the system with their gaze irrespective of age or gender, as we designed. Some of the users, whose gaze were not the actual target, observed the interaction between a particular user and the region described by the robot's vocal information (Figure 3-A and 3-B). On the other hand, children enjoyed the interaction with the robot's behaviors by freely using their gazes (Figure 3-C). These observations showed the gradual transition of interactions from gaze communication to the gaze-corresponding guidance.

Figure 3. Demonstration experiment in lobby of hotel

Wide-area and Calibration-free Gaze-tracking Method

The guide robot needs to perceive the user's gaze, which moves in a *wide-area* space such as a room. However, the current gaze-tracking systems still need fixtures, calibrations, and limitations on the distance to the system (Newman et al., 2000; Ohno et al., 2000, etc.). In our method, we employ a single-camera-based gaze-tracking method using a high-resolution camera (Yamazoe et al., 2008) to detect the user's attention or direction of interest in a *wide area*. This is possible because the method can estimate the user's gaze direction from low-resolution face images (320×240 pixels) with eye-region images (30×20 pixels). Figure 4 shows an overview and the results of the wide-area and calibration-free gaze-tracking method. This method, by applying facial-feature tracking and 3D eyeball-model estimations, offers the advantages of not needing any attachment devices to users and not requiring the user to wear attached devices or to perform preliminary actions such as looking at reference points for calibration. Figure 5 shows examples of gaze-tracking angles relative to the board. The averaged accuracy of the angles for each region is shown in Table 1. Accordingly, this system's robot and illuminated board could react to the user's gaze at sufficient response speed and accuracy for natural gaze communication.

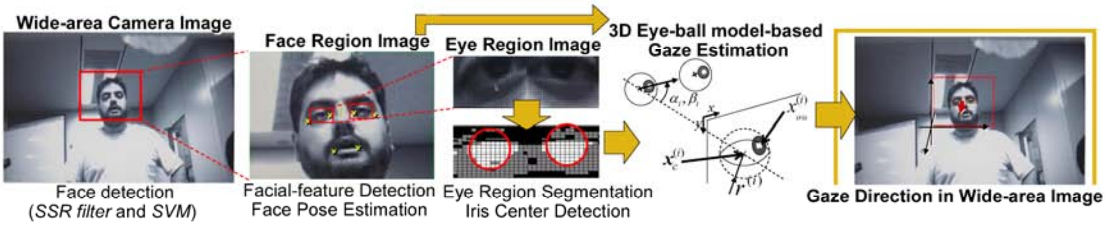


Figure 4. Wide-area Gaze-tracking by Single Camera

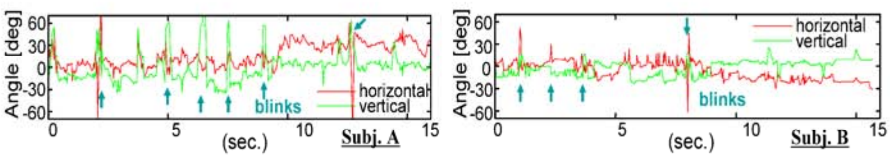


Figure 5. Examples of estimated gaze angles

Table 1. Gaze-tracking accuracies

(region)	*LR*	*CR*	*RR*	robot
horizontal	3.81°	3.51°	2.43°	1.40°
vertical	6.89°	5.44°	5.79°	9.11°

Evaluations of Gaze-communicative Robot on Guideboard

Based on our assumed stepwise interaction model, we examined the effectiveness of the gaze-communicative robot and the gaze-corresponding guidance in the following three types of subjective evaluations performed by 25 people aged from twenties to thirties (13 females and 12 males). These evaluations were made to verify the effectiveness of i) *the robot's presence* in the user's passive guidance (EX1), ii) *the gazing behaviors of the robot* (EX2), and iii) *the gaze-corresponding guidance* (EX3) using both voice and motions of joint attention.

To simplify the contents of the guidance, the guideboard shows only two figures: a triangular pole in LR and a cube in RR, and the voice guidance simply speaks the name of the object on the guideboard. The subjects evaluate each stimulus using a five-point rating scale to evaluate the relevance (5: very relevant, 4: somewhat relevant, 3: even, 2: somewhat irrelevant, 1: irrelevant) of the following statements: `(a) certainty factor`, `(b) attractive factor`, `(c) naturalness`, and `(d) before-after change of interest`. The short sequences of the guidance were shown to the subjects as the experimental stimuli in randomized order for counterbalance.

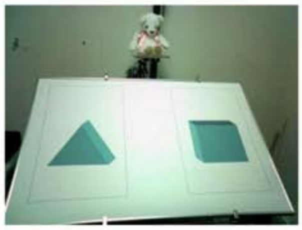


Figure 6. Simple board in evaluations

Conditions: We prepared different conditions to compare them and to verify the effectiveness of each issue as above i) to iii) in every experiment. In `EX1`, we prepared the condition α_1 as the automatic vocal guidance without the robot and the condition β_1 as the automatic vocal guidance with showing the robot. For `EX2`, we prepared the condition α_2 as a motionless robot with vocal guidance and the condition β_2 as the robot's vocal guidance with its gazing behavior corresponding to the guiding content. In `EX3`, we prepared four conditions to verify two different issues: gaze-corresponding vocal guidance and gaze-corresponding direction of the robot' gaze. So we prepared the condition {`r+,v+`} in that the robot gazes at the figure which the subject looks at, and the vocal guidance explains the figure; for example, the robot talks "This triangle cookie is cocoa flavored." The condition {`r+,v−`} was prepared as the robot gazes at the same figure with the subject, but the content of vocal guidance is the opposite figure. In the condition {`r−,v+`}, the robot gazes at the opposite figure but the vocal guidance corresponds to the subject's gazing figure. In the condition {`r−,v−`}, both the robot's gazing direction and the content of vocal guidance are corresponding to the opposite figure of the subject's gaze.

Instructions: The subjects stood in front of the guideboard. The distance from the guideboard to the subjects was about four feet. In `EX1` and `EX2`, the subjects were instructed only to observe the guideboard system. In `EX3`, the subjects were instructed to select and look at one figure and observe the guideboard system. The subjects were also instructed to do same things for the other figure.

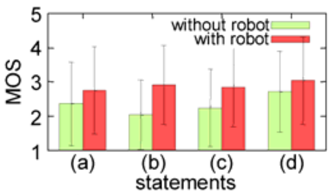


Figure 7. MOS for the robot's presence

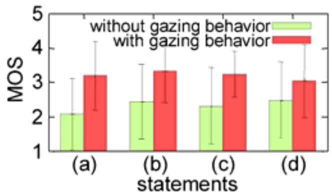


Figure 8. MOS for gazing behaviors

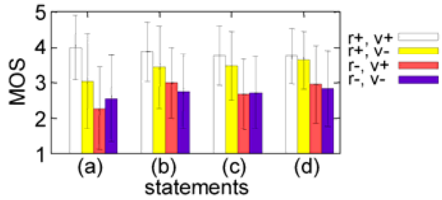


Figure 9. MOS for gaze-corresponding guide

Table 2. ANOVA of MOS for Robot's Presence

	(a)	(b)	(c)	(d)
$F_{(1,24)}$	9.6	27.9	12.0	5.37
p	<u><.01</u>	<u><.01</u>	<u><.01</u>	<u>.029</u>

Table 3. ANOVA of MOS for Robot's Gazing Behaviors

	(a)	(b)	(c)	(d)
$F_{(1,24)}$	23.1	27.9	23.3	7.19
p	<u><.01</u>	<u><.01</u>	<u><.01</u>	<u>.013</u>

Table 4. Two-factor ANOVA of Gaze-corresponding Guide

dir.		(a)	(b)	(c)	(d)
r+ vs r- (robot)	$F_{(1,24)}$	11.6	14.3	23.8	14.2
	p	<u><.01</u>	<u><.01</u>	<u><.01</u>	<u><.01</u>
v+ vs v- (voice)	$F_{(1,24)}$	3.68	4.11	0.61	1.13
	p	0.067⋆	0.054⋆	0.442	0.298
intrr.	$F_{(1,24)}$	9.17	0.342	1.30	0
	p	<u><.01</u>	0.563	0.266	–

dir.: direction, *intr.*: interaction

Results: The MOSs (means opinion scores) comparing different stimuli are shown in Figure 7–9 as the results of `EX1`–`EX3`, and their ANOVA results ($df. = 24, \alpha = .05$) are marked with underlines for each significance, with ⋆ for significant tendencies (Table 2–4). `EX1` is a comparison of 1) the guideboard without the robot and 2) the guideboard using the robot (Figure 7). The results of ANOVA for EX1 (Table 2) show every significant difference. The results of EX2, comparing 1) the guidance without the robot's motion and 2) the guidance with the robot's gazing behaviors at the corresponding region of the guideboard, also show every significant difference in each statement (Figure 8, Table 3).

The results of EX3 comparing correspondence (+) or non-correspondence (−) of the guiding behaviors with joint attention(r) and voice guide (v) are shown in Figure 9. Here, r+ means the robot gazes in the same direction as the user's gaze, and r− means the robot gazes in the opposite direction to the user's gaze. In addition, v+ means the appropriate contents of the vocal guide for each figure, such as "This is a tent" for *LR* and "This is a building" for *RR*, and v− indicates the inappropriate contents of the voice guide. Two-factor ANOVAs show that the robot's gazing direction corresponding to the user's gazing position obviously increases the quality of the guidance, compared with the only somewhat significant tendencies of the vocal guide's contents (Table 4).

Discussion

As shown in the previous section, our results verified the effectiveness of i) the robot's presence, ii) the robot's gazing behaviors toward the guided region, and iii) guidance corresponding to the user's gaze. These results indicate the beneficial effect of gaze-communicative guidance in enhancing the attractiveness and reliability of a guide system, based on positive impressions such as affection and naturalness.

Consequently, the anthropomorphic presence, the anthropomorphic behaviors, and gaze-corresponding interaction of the gaze-communicative guide make a positive impression on the users of this system. These results suggest the possibility of removing the psychological or physical burdens, that are actually experienced by disabled or hesitant people in asking for further detailed information beyond that on the guideboard.

Conclusion

This paper introduced and evaluated an implementation of an interactive guideboard system using a communicative stuffed-toy guide robot that behaves in correspondence to the user's gazing direction, which is estimated by our remote gaze-tracking method. The primary experiment on the system demonstrated an intuitive and enjoyable system for guidance interaction, performing at sufficient speed and accuracy. Furthermore, we verified both the effectiveness of *the anthropomorphic guide agent* and *the gaze-corresponding guidance* through analyses of subjective evaluations. As future works, we are considering not only vocal guidance and the robot's gazing behaviors with simple illumination but also other modalities such as the robot's sign language and the interactive information on the board to expand the target users toward deaf people.

Acknowledgements

This research is supported by National Institute of Information and Communications Technology. We would like to thank Dr. Norihiro Hagita and Dr. Kenichiro Shimokura for this opportunity, and other ATR members for cooperating our experiments.

References

Breazeal, C., Buchsbaum, D., Gray, J., Gatenby, D., & Blumberg, B. (2005). Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots. *Autonomous Robots*, *11*, 31–62. (Issues 1–2)

Duffy, B. (2003). Anthropomorphism and the social robot. *Robotics and Autonomous Systems*, *42*(3), 177–190.

Fukayama, A., Ohno, T., Mukawa, N., Sawaki, M., & Hagita, N. (2002). Messages embedded in gaze of interface agents – impression management with agent's gaze–. *Proc. ACM SIGCHI*, *1*, 41–49.

Katagiri, Y., Takahashi, T., & Takeuchi, Y. (2001). Social persuasion in human-agent interaction. *IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 64–69.

Kendon, A. (1967). Some functions of gaze-direction in social interaction. *Acta Psychologica*, *26*, 22–63.

Newman, R., Matsumoto, Y., Rougeaux, S., & Zelinsky, A. (2000). Real-time stereo tracking for head pose and gaze estimation. *Proc. FG2000*, 122–128.

Ohno, T., Mukawa, N., & Kawato, S. (2000). Just blink your eyes: A head-free gaze tracking system. *Proc. CHI2003*, 950–951.

Shi, F., & Gale, A. (2007). Environmental control by remote eye tracking. *COGAIN 2007*, 49–52.

Sidner, C., Lee, C., Kidd, C. D., Lesh, N., & Rich, C. (2005). Explorations in engagement for humans and robots. *Artificial Intelligence*, *166*, 140–164.

Sumi, Y., Etani, T., Fels, S., Simonet, N., Kobayashi, K., & Mase, K. (1998). C-map: Building a context-aware mobile assistant for exhibition tours. *Community Computing and Support Systems, LNCS 1519*, 137–154.

Yamazoe, H., Utsumi, A., Yonezawa, T., & Abe, S. (2008). Remote gaze direction estimation with a single camera based on facial-feature tracking without special calibration actions. *Eye-tracking Researches and Applications*, 245–250.

Yonezawa, T., Yamazoe, H., Utsumi, A., & Abe, S. (2007). Gaze-communicative behavior of stuffed-toy robot with joint attention and eye contact based on ambient gaze-tracking. *Proc. of ICMI07*, 140–145.

Gaze-Contingent Passwords at the ATM

Paul Dunphy, Andrew Fitch, Patrick Olivier
Culture Lab
School of Computing Science
Newcastle University
{p.m.dunphy,p.l.olivier}@ncl.ac.uk

Keywords

Security, Graphical passwords, Passfaces, Shoulder surfing

Introduction

Knowledge-based authentication (e.g. passwords) has long been associated with a vulnerability to *shoulder surfing*; being stolen by attackers overlooking the interaction. In order to combat such threats, steps can be taken to either alter the form of the challenge made to the user, or make use of an interaction technique that is resistant to information leakage. We consider the latter, and empirically evaluate the usability of gaze-contingent interaction as a solution to shoulder surfing in an ATM scenario. We combine this technique with *Passfaces* graphical passwords; potentially more memorable than PINs and well suited to accept gaze-based input. To create a naturalistic setting for our study we utilise the *immersive video* technique originally deployed in the design of pervasive computing systems (Singh et al., 2006). We demonstrate the efficacy of the approach, a usable graphical password entry system that is impossible to attack by direct observation.

Gaze-Contingent Graphical Passwords

The use of an eye tracker to input alphanumeric passwords was presented in a system called *EyePassword* (Kumar et al., 2007), and the application to graphical password schemes has also been considered (Hoanca & Mock, 2006), however no implementations or evaluations of such a setup have been reported to date. Graphical passwords potentially offer improved memorability over passwords and PINs due to image recognition/recall being an easier memory task for humans over recall of words and numbers. Today, research continues into exploring the vast design space of graphical authentication. *Passfaces* is a simple image selection-based graphical scheme that also takes advantage of innate human ability to recognise previously seen faces. The authentication secret assigned to each user is a sequence of n (usually 5) face images which they are required to identify in a sequence of 3×3 grids of faces, each of which comprises an assigned face and eight decoy faces. A number of studies have reported Passfaces to be usable and memorable (Brostoff & Sasse, 2000) (Valentine, 1998a), even over a long period of time (Valentine, 1998b).

We developed software to loosely simulate ATM behaviour using a Tobii x50 eye tracker for input. Our system had 5 states: (1) *Idle*: the system displays various offers from banks on a loop in the fashion of a typical ATM, waiting for a customer; (2) *Calibration*: the user is led through a calibration process; (3) *Playtime*: the user is familiarised with the eye tracker by playing a short game involving selecting fruit on-screen; (4) *Enrolment*: the user is assigned the five face images to comprise their face password, and must spend time forming a memory association; (5) *Login*: the user must recognise and select the faces assigned at enrolment. The user is presented with a sequence of five 3×3 grids, containing eight decoy faces and one assigned face, upon selecting a face in a grid the next grid is displayed. A face selection was assumed after a 0.5 second dwell. For a successful login all 5 faces must be recognised and feedback to this effect was given only after the final face selection. A number of these states are illustrated in figure 3.

As users would be standing whilst interacting with the eye tracker (positioned at a fixed height), it was important to give feedback on the visibility of their eyes to the eye tracker, so they could re-create a good stance and debug any lack of responsiveness themselves. We included the Tobii *trackstatus* indicators of eye visibility; and dynamically changed the screen background colour depending on head distance from the eye tracker.

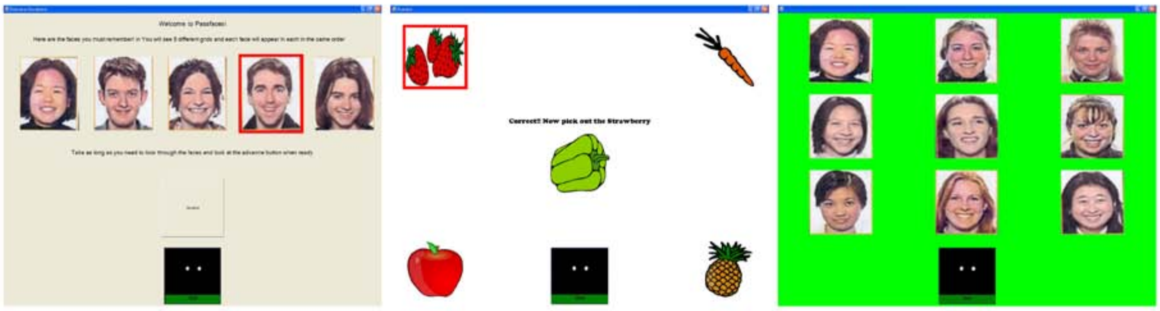


Figure 1. Display designs for the different interaction stages: (1) enrolment; (2) playtime (experimenting with gaze-contingent interaction); and (3) *Passfaces* login challenge.

Study Environment

A key design concern of the study was the recreation of the sights, sounds and experiences that are typical of ATM machine usage in busy public settings. The configuration of our ATM study had 4 such elements: (1) *RFID* credit cards; (2) spatial and physical configuration of the ATM; (3) use of an array of large video screens in which to immerse the user at our ATM; and (4) the display of ambient video and sound of an actual ATM to simulate the sensory experience of real ATM usage. The simulated ATM was crafted to be of a familiar physical configuration, and identified users by their unique RFID credit cards assigned to them at the outset of the study, the unique identifier encoded on the card enabled the system to determine if the user required a calibration or could proceed directly to login. The recognition that ATM usage in a public setting can be very different to the desktop led us to re-create many of the common distractions. To capture ambient visuals and sound we recorded video footage at an actual ATM. We positioned our ATM in the centre of three screens and displayed ambient footage on a loop on the left and right screens with a still image of a wall on the centre screen (figure 2).

Figure 2. Configuration for the ATM study.

Procedure and results

Twenty student participants (16 male and 4 female) took part in the study; 11 of these were majors in Computing Science or Information Systems (*average age* = 21, *STD* = 0.8). Participants played the role of bank customers (users of the ATM) and would start the procedure by placing their RFID credit card on the RFID reader underneath the screen. After undergoing a satisfactory calibration (where a moderator could assist to determine quality) and passing the playtime and enrolment phases, the participants attempted to login with no number of incorrect attempts causing a lockout. Where a reminder was required, this option was included and access to it logged by our system. Participants were asked to login 3 times per session, with each login approximately 15 minutes apart. After a successful login the system thanked the participant for using the ATM and instructed them to remove their card. Participants returned 3 days later to the second session to perform further logins, but this time the calibration, playtime and enrolment phases were not required as details were stored in the database.

In the enrolment phase participants spent some time forming a memory association with the faces that comprised their graphical password. On average users spent 24.9 seconds on this screen but there was a wide variation ($STD = 12.88$) with one participant taking as long as 64 seconds. Figure 3 describes the number of attempts participants required to login at both the first and second sessions. In the first login of the first session performance was worst, as participants came to terms with their face password (a new experience for all). The first login saw participants just as likely to need one attempt as more than three attempts. In fact, in the first login alone, participants made 62 attempts; as many as in the following two logins combined. Performance improved for logins 2 and 3 as users increasingly logged in at the first attempt.

Three days after the first session participants returned and we were keen to observe any retention of faces in memory, and skill at using the eye tracker. At this session there was a marked improvement in performance over the first session with only 8 instances out of 60 (20 participants and each was asked to login 3 times in the session) where more than one attempt was needed, 127 attempts in total were required at the first session and just 73 at the second session ($t = 3.42; p < 0.01$). This superior performance was also reflected in memorability of the faces as reminders of the face passwords were requested significantly less in session 2. The system recorded 24 at session 1 and only 2 at session 2 ($t = 3.99; p < 0.01$)

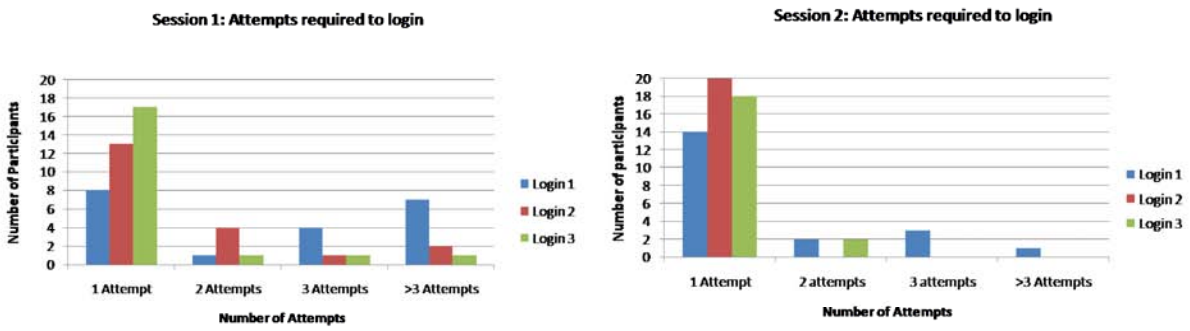


Figure 3. Attempts required for a successful login

At the end of the study participants were interviewed in small groups and asked for their perceptions and opinions on the proposed system and encouraged to relate it to their daily lives. Some of the taller participants criticised the height of the eye tracker as they had to adopt a slightly crouched position (a vulnerable position at an ATM). This problem has been experienced before in trials of iris biometrics at the ATM interface (Coventry et al., 2003). The calibration process frustrated some participants for whom it did not work first time, exacerbated by the sense that they were doing something wrong. One participant commented that the system felt more hygienic as no physical contact with a keypad on the system was required. Another found the experience of eye tracking to be confusing, trying to touch buttons on-screen as though interacting with a touchscreen. Also the ambient sights and sounds did not seem to cause undue distraction for the participants, perhaps indicating a *cocktail party effect* where focused attention on the task at hand drowns out the extra stimuli.

Concluding remarks

We have reported initial investigations into the potential of eye tracking to be applied as a usable solution to shoulder surfing in an authentication context. Despite initial uncertainty about an unfamiliar technology, user performance in the study was good and we witnessed a significant improvement in skill with the eye tracker technique across two short sessions. Also the eye tracking input seems to eliminate the threat of shoulder surfing, something we are interested to validate in future work. One concern regards our inclusion of the *trackstatus* indicator displaying visibility of the eyes to the camera within the eye tracker, but it seems unlikely this can leak accurate information of gaze to an adversary.

There are obstacles to commercial adoption aside from cost alone. The average time to login was 20 seconds, with the fastest time recorded being 8 seconds. Compared to a typical PIN entry of just a few seconds there is still a significant deficit. Eye trackers themselves have well known limitations in terms of *failure to enrol* errors as some users can consistently achieve a less than perfect calibration. Future plans include exploring the design space of authentication techniques suited to gaze-contingent interaction and further increasing the ecological validity of our ATM environment for future lab-based studies.

References

Brostoff, S., & Sasse, A. (2000). Are Passfaces more usable than passwords? A field trial investigation. In *Hci 2000: Proceedings of people and computers xiv - usability or else* (p. 405-424). Springer.

Coventry, L., Angeli, A. D., & Johnson, G. (2003). Usability and biometric verification at the atm interface. In *Chi '03: Proceedings of the sigchi conference on human factors in computing systems* (pp. 153–160). New York, NY, USA: ACM.

Hoanca, B., & Mock, K. (2006). Secure graphical password system for high traffic public areas. In *Etra '06: Proceedings of the 2006 symposium on eye tracking research & applications* (pp. 35–35). New York, NY, USA: ACM.

Kumar, M., Garfinkel, T., Boneh, D., & Winograd, T. (2007). Reducing shoulder-surfing by using gaze-based password entry. In *Soups '07: Proceedings of the 3rd symposium on usable privacy and security* (pp. 13–19). New York, NY, USA: ACM.

Singh, P., Ha, H. N., Kuang, Z., Olivier, P., Kray, C., Blythe, P., et al. (2006). Immersive video as a rapid prototyping and evaluation tool for mobile and ambient applications. In *Mobilehci '06: Proceedings of the 8th conference on human-computer interaction with mobile devices and services* (pp. 264–264). New York, NY, USA: ACM.

Valentine, T. (1998a). *An Evaluation of the Passface Personal Authentication System .* (Technical Report. London: Goldmsiths College University of London.)

Valentine, T. (1998b). *Memory for Passfaces after a long delay.* (Technical Report. London: Goldmsiths College University of London.)

Scrollable Keyboards for Eye Typing

Oleg Špakov and Päivi Majaranta
Department of Computer Sciences / TAUCHI
FIN-33014 University of Tampere, Finland
{oleg.spakov; paivi.majaranta}@cs.uta.fi

Keywords
Eye typing, text entry, eye tracking, gaze input

Introduction

Text entry is one of the main interaction tasks in gaze-controlled interfaces. The primary method of eye typing consists of selection of keys from an on-screen virtual keyboard (for a review of gaze-based text entry methods, see Majaranta and Räihä 2007). The user types by pointing at each character by gaze and dwelling on it for a certain amount of time, using *dwell time* as an activation command. Typically, only one keystroke per character (KSPC) is needed since most letters can be directly pointed at and selected. Having all characters visible at the same time requires space. The keys on the virtual keyboard must be big enough because of the accuracy limitations of eye tracking devices. This is true especially with "low-cost" systems that are based on off-the-shelf video or web cameras and have limited spatial resolution. Obviously, if the keyboard occupies most of the screen estate, it significantly limits the space available for other applications.

Several attempts have been made to solve the problem of coping with the inaccuracy of the measured point of gaze and still preserving maximum screen space. Decreasing the number of keys can be used to save screen space (Miniotas et al. 2003). However, bigger keys are more often needed to enable the use of an eye tracker with low spatial resolution (Hansen et al. 2001), or to enable an end-user with eye tremor or involuntary movements to point at items on screen comfortably enough (Donegan et al. 2006). Thus, in some cases, having fewer keys is a requirement for any tracking at all and would therefore not save screen space. Isokoski (2000) used off-screen targets in order to preserve maximum screen space. To type a character, the user fixates at the off-screen targets in a certain sequence. The resulting *gaze gesture* is mapped to a character or command. Some recent gaze gesture systems use parts of the screen itself as active areas for the gesture recognition (Drewes and Schmidt 2007; Porta and Turina 2008) or show a small special area where the entering of the gaze gestures happens (Wobbrock et al. 2008). All these systems save screen space but learning the gesture based alphabet takes time. They also typically require several (typically 2-4) strokes per character. In experiments, users have achieved the average speed of 5-8 words per minute (Porta and Turina 2008; Wobbrock et al. 2008).

Miniotas et al. (2003) developed Symbol Creator. A character is created by combining two (or more) symbols. Hence, two keystrokes produce one character (with few exceptions). The symbol parts and their combinations resemble hand written characters or their parts (similarly as 'o' and 'l' put together forms 'd'), which helps in learning the symbols. The Symbol Creator has eight keys in a one-row virtual keyboard. Showing only one row of keys leaves most of the screen estate free for other purposes.

Our goal was to develop a keyboard that saves screen space but will still be immediately usable and not require any special learning. Our idea is to use a keyboard layout that is already familiar to the user (such as QWERTY) and to save screen space by only showing part of the keyboard. In the following sections, we first describe the design of the reduced keyboards, which we call *scrollable keyboards*. We will then report results from an experiment where the keyboard was tested.

Scrollable Keyboards

For the "full" keyboard, we used a common keyboard layout, QWERTY, shown in Figure 1 on the left. For the experiment, we decided to leave out special characters and punctuation (other than the comma and dot keys). Two space keys were placed in the end of the second and the third row.

Figure 1. Full (3-row) keyboard, 2-row and 1-row scrollable keyboards.

The 2-row keyboard (Figure 1, in the middle) has only two rows of keys visible at any time. To reach the third row, the user needs to select one of the special scroll keys on the left. The 1-row keyboard (Figure 1, on the right) only shows one row. The scroll keys, "up" and "down", are located on the sides of the keyboard. In both, the scrolling is cyclic; an invisible row can be reached using either one of the scroll buttons. The scrolling produces animated feedback which takes 150 ms. Obviously the KSPC measure is more than one for the scrollable keyboard, since at least one extra keystroke (scroll key) is required to reach a hidden row.

The visible distance between rows was extended because the drifting of the measured gaze position is higher in vertical direction than in horizontal direction with the tracker we used (see the method section below). Even though the visible buttons are circles, the gaze reactive area for each button is a rectangle (approx. 1.5*3.0 degrees if the distance between the user and the monitor is 45 cm). The buttons were selected using dwell time of 500 ms, constant throughout the experiment. Animated feedback indicated the progression of the dwell time, and the key became "pressed" (shown as pressed "down" for 150 ms) when selected.

Method and Procedure

8 volunteers (aged 23-47 years, 5 male, 3 female) took part in the test. They were students or staff, and all had participated in other related eye typing experiments earlier. Experienced participants were used to minimize the learning period. All were fluent in English and familiar with the QWERTY layout.

The experiment was conducted in the usability laboratory at the University of Tampere. A head-mounted EyeLink eye tracking system was used to measure participants' eye movements. The iComponent software which has a plug-in for EyeLink was used to implement the experimental keyboard and to save data. The setup consisted of operator and subject monitors, adjustable chairs and tables. The chair was set so that the participant's eyes were at approximately 45 cm from the 17-inch monitor.

For the experiment, 30 easy to memorize phrases were chosen from a set of 500 phrases by MacKenzie and Soukoreff (2003). Punctuation was removed and the phrases were case-insensitive. Participants were instructed to eye type the phrases as fast and accurately as possible. They were instructed to ignore mistakes and to carry on with a phrase when a mistake was made (our keyboards did not have a backspace key).

Each session started with a short training period. To provide a basic level of familiarity with the experimental software, participants were given one practice phrase (about 25 characters) prior to data collection. The experiment had 3 conditions: 3-row (full), 2-row, and 1-row keyboard. There were 8 sessions for each testing condition (1 session per day). Each session included 6 phrases (average length of 26.3 characters) for each condition, shown one at a time. Thus, the number of entered characters was approximately 8*8*3*6*26.3 ≈ 30300 (1152 phrases). A session lasted approximately 10-15 minutes.

Results

The typing rate was measured in words per minute (wpm). In the last session, the average typing speed was 15.06 wpm for the full keyboard, 11.12 for the 2-row keyboard, and 7.29 wpm for the 1-row keyboard. The average error rates varied between 1-5%, with large variance between participants during the whole experiment. In the last session, the average error rates were below 2% for all conditions (see Figure 2).

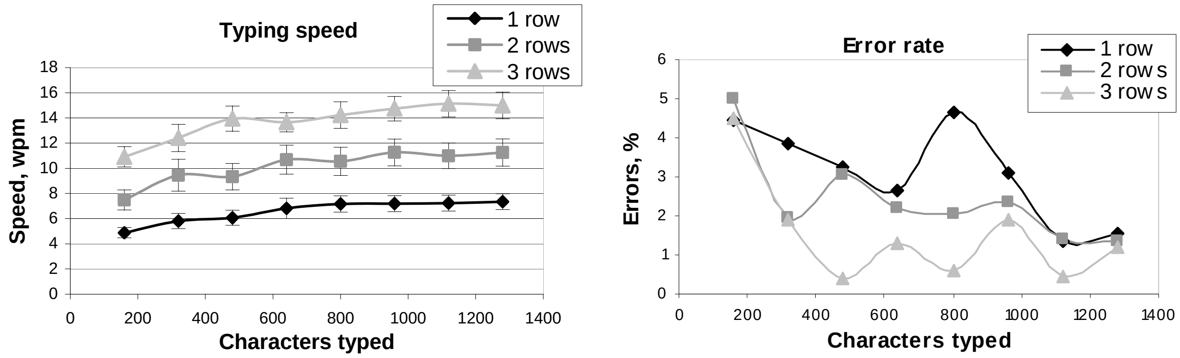


Figure 2. Typing speed (left) and error rate (right).

The selection time for the scroll buttons, letter keys and space was measured. Especially, monitoring the usage of the scroll button is interesting, because it shows how the participants learned to use the scrollable keyboards with only partially visible layout. Figure 3 below shows the selection times for the 1-row (on the left) and 2-row (on the right) keyboards. The average selection times of the scroll buttons were 1107 and 1268 milliseconds for the 1-row and 2-row keyboard, respectively. If the constant dwell time of 500 ms is removed from the full selection time, the search time for each button is approximately 500 ms.

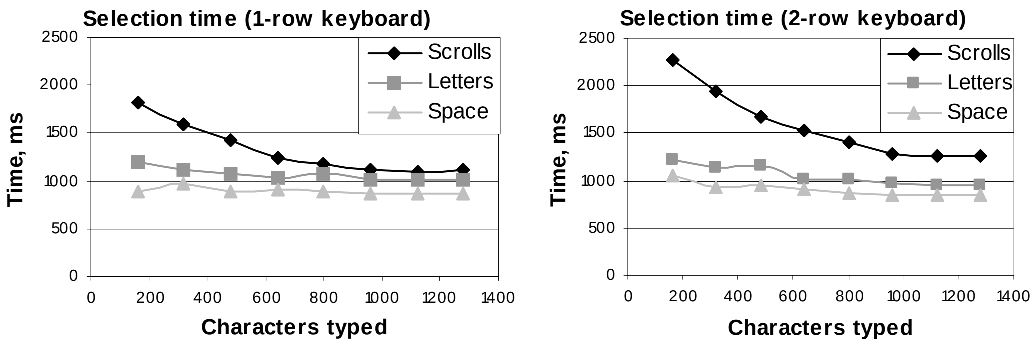


Figure 3. Selection time for the 1-row (left) and 2-row (right) scrollable keyboards.

Analysis of the scroll button usage shows that it slightly decreased in time and the average percentage of the scroll button clicks among all clicks were 39% (1.64 KSPC) and 16.5% (1.2 KSPC) for the 1-row and 2-row keyboards, respectively. Participants used different strategies with the scrolling keyboards. Half of them memorized the location of letter and rows so that they could choose the shortest route to the invisible row. For example, after ‘e’ (located on the top row) the user can reach ‘n’ (on the bottom row) by one scroll up instead of two scrolls down in the 1-row keyboard. Thus, the number of scroll usage was minimized. Some participants never scrolled the layout from top line up (to the bottom) or vice versa, because they did not want to lose orientation in scrolling. In this case, more scrolling was required but the participants still did not spend time in searching for the target letter. Finally, one participant did not memorize the distribution of letters across rows but always visually scanned any row to find the desired letter, and used only one direction of scrolling (up). This strategy resulted in the slowest typing speed. The difference between the fastest and slowest participant was approximately 3 wpm within each condition.

Conclusion

We have shown that scrollable keyboards, which reduce the space taken by the full (3-row) keyboard by 1/3 or 2/3, can be efficiently used to enter text by gaze. The typing speed reduced only by 26.0% for the 2-row and 51.6% for the 1-row keyboard. Furthermore, the increase in the rate of keystrokes was quite reasonable, from 1 KSPC to 1.64 KSPC and 1.2 KSPC with the 1-row and 2-row keyboard, respectively. The results are encouraging compared to e.g. gesture based interfaces that always require several strokes per character (albeit the saccades needed to make such eye strokes can be very fast).

The typing speed and KSPC can be further improved using an optimized layout organized according to letter-to-letter probabilities. However, the optimized layout requires longer learning time. (Results of our experiment with the optimized layout will be reported elsewhere later.)

The scrolling keyboards may be especially useful in casual typing situations, for example, filling in web forms where the overview of the full web page is important. Scrolling could also be useful in accessing the key rows that are not needed as often as letters, such as number, punctuation and function keys. Finally, the user should be able to easily adjust the number of visible rows to support the optimal layout in each situation.

Acknowledgements

We would like to thank all volunteers who participated in the study and Tatjana Evreinova for comments.

References

Donegan, M., Oosthuizen, L., Bates, R., Istance, H., Holmqvist, E., Lundälv, M., Buchholz, M., and Signorile, I. (2006) *D3.3 Report of user trials and usability studies*. Communication by Gaze Interaction (COGAIN), IST-2003-511598: Deliverable 3.3. Available online at http://www.cogain.org/results/reports/COGAIN-D3.3.pdf

Drewes, H. and Schmidt, A. (2007) Interacting with the Computer Using Gaze Gestures. In *Proceedings of INTERACT 2007, LNCS 4663*, Springer, 475-488.

Hansen J.P., Hansen D.W., and Johansen A.S. (2001) Bringing Gaze-based Interaction Back to Basics. In C. Stephanidis (ed.) *Universal Access In HCI*, Lawrence Erlbaum Associates, 325-328.

Isokoski, P. (2000) Text Input Methods for Eye Trackers Using Off-Screen targets. In *Proceedings of the Eye Tracking Research & Applications Symposium, ETRA 2000*, ACM Press, 15-21.

MacKenzie, I.S. and Soukoreff, R.W (2003) Phrase sets for evaluating text entry techniques. In *Proceedings of CHI 2003: Extended Abstracts on Human Factors in Computing Systems*, ACM Press, 754-755.

Majaranta, P. and 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* (pp. 175-187), San Francisco: Morgan Kaufmann.

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

Porta, M. and Turina, M. (2008) Eye-S: a full-screen input modality for pure eye-based communication. In *Proceedings of ETRA '08*, ACM Press, 27-34.

Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. (2008) Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of ETRA '08*, ACM Press, 11-18.

The Use of Eye-Gaze Data in the Evaluation of Assistive Technology Software for Older People

Simon Judge
Barnsley Assistive Technology Team
Block 14, Barnsley District Hospital Foundation Trust, Gawber Road. S75 2EP
Simon.judge@nhs.net
01226 432159

Steven Blackburn
Sheffield University, ACT Programme
ICOSS, Sheffield, S1 4DP
s.blackburn@sheffield.ac.uk
0114 222 6253

Keywords
Usability Study; Assistive Technology Software; Older People; Simplified Computer Access;

Abstract

This paper reports on recent work undertaking usability study of a software-based assistive technology. The software was developed to support increased opportunities and interactions for people in residential nursing homes and extra-care housing. The objective of the project was to allow older people and those with early onset of dementia to have access to some of the functionality of modern computers. The software could also have applications in other markets, such as schools and for older people living at home. The intention is to provide opportunities for active participation and facilitate more access to hobbies, interests, past-times and to develop and maintain social networks. The complex interface of modern computers otherwise often excludes people from access to digital media including video and internet telephony, games and activities, information and resources on the internet and other facilities that may be useful to them if presented in a different way.

The study presented is being carried out in 3 residential homes with 20 participants. Eye-gaze recording was a key element of the usability testing. The study methodology was designed to provide feedback towards the design of the software and to better understand the use of computers by this target group.

This paper presents the results of the first stage of the usability study, in particular the paper concentrates on the use of the eye-gaze data. The design of the sessions allowed participants to explore the system independently and then to complete some pre-defined tasks. The users' interaction with the computer was recorded through video, audio, screen and eye-gaze recording as well as a data-log of the physical and eye interaction. The process of acquiring eye-gaze data with this fairly non-typical cohort is examined and the value of this data in contributing to the design of this software is explored.

Introduction

The Advanced Care Technologies (ACT) Programme is a European funded venture to investigate the most effective applications of Assistive Technology and Telecare (ATT) to raise the quality of life for South Yorkshire's ageing population. A major theme of the ACT Programme is evaluation of the effectiveness and user acceptability of ATT products. It seeks to do this through collaborative partnerships between the university, industry and the health and social care systems.

The usability study described in this paper was conducted within the SIM WIN project. The SIM WIN project is an evaluation of an intuitive computer-based system developed for residents of residential care homes. The software was designed to provide enhanced accessibility to activities and interests, and improve social interaction and networking to family and friends through videoconferencing facilities. The motivations behind the development of SIM WIN was to provide a means of mental stimulation, as an alternative (or complementary) to the traditional social activities existing in UK residential care homes. The University of Sheffield leads the SIM WIN project, in collaboration with Barnsley Hospital and a non-for-profit residential care home provider based in Sheffield, UK.

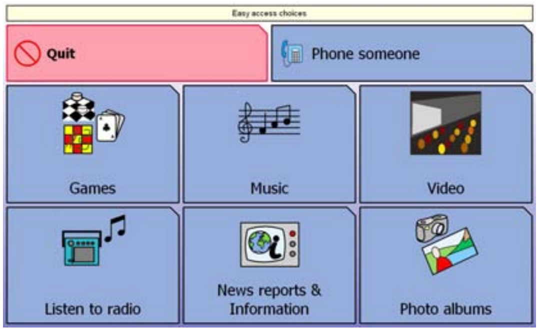


Fig 1. SIM WIN system (screenshot)

Eye-tracking or eye-gaze is a method of measure where people are looking while they perform particular tasks. Almost sixty years ago, Fitts, Jones and Milton pioneered the use of eye tracking (or eye-gaze) to assess the usability systems for airline pilots [Fitts, P. M., Jones, R. E., & Milton, J. L. (1950). Eye movements of aircraft pilots during instrument- landing approaches. Aeronautical Engineering Review 9(2), 24–29]. Since then, eye-gaze techniques has been used to assess a wide range of human-computer interactions., including military technology, menu-based software systems and website design. Literature concerning the implementation of eye-gaze techniques to assess the usability of software by older people is limited with only a few published studies, for example, Obrist et al (2007) used eye-gaze methods to investigate the usability an interactive TV application between a group of older people (50+) with a group of younger adults [Marianna Obrist, Regina Bernhaupt, Elke Beck and Manfred Tscheligi, Focusing on Elderly: An iTV Usability Evaluation Study with Eye-Tracking, Lecture Notes in Computer Science, Volume 4471: 66-75; 2007]

The aim of the study described in this paper was to evaluate the usability of the SIMWIN system, in addition the study aimed to investigate the role of eye gaze techniques as a tool for assessing the usability of computer-based systems developed for older people.

Study Design

The study is being carried out in 3 residential homes with 20 participants. Participants were a mix of day-centre attendees and residential home residents, all participants were not currently computer users and were aged 63 and above. The oldest participant was 96 years old. Participants had a range of associated medical conditions, including some functional and cognitive limitations. Recruitment was carried out by casual researchers through information sessions and demonstrations, inclusion criteria were diagnosis of dementia and any physical incapacity which would hinder the basic operation of the touchscreen (e.g. blindness).

Each participant undertook an initial usability session of approximately 20 minutes.The first half of the session allowed the participant to explore the system and during the second half the participant was asked to complete 6 tasks. The session was recorded using screen capture with eye-gaze data overlaid, video and audio capture and data-logging of mouse and gaze interactions.

Fig 2. Examples of recorded data

This session was followed by 3 months of regular training, personalisation and support sessions in the residential homes in a group environment. At the time of writing, this training period is ongoing and the first session of the usability testing has been studied and initially analysed. A follow-up stage of the study is planned to allow comparison of participant's use pre and post the training and support. The study methodology was designed in order to provide feedback towards the design of the software and to better understand the use of computers by this target group.

Analysis

A number of analysis stages are planned:

- Qualitative analysis of the recorded data (screen, eye, video and audio)
- Hand-coding of the recordings to analyse (tasks completed, task success rate, number of cells selected in each task)
- Quantitative analysis of the data-logged data (mouse and gaze interactions)

At the time of writing, the qualitative analysis has been completed and is briefly described below.

Preliminary Results

The qualitative analysis was conducted to identify and code themes within the data related to the usability of the system. Below is a brief summary of the main relevant themes.

Design Compromises and Features

Design compromises predominately emerged around the nature of the interface and making this less confusing for participants. A number of sub-themes emerged relating to 'confusability':

- The label used on the buttons: participants sometimes struggled to select 'correct' buttons in tasks, despite repeated prompting and having looked at the button. This may be related to the language and/or icon of the buttons not being relevant or understood by the participant.
- Not understanding the function of the button: some participants seemingly pressed buttons without fully understanding what the effect of pressing it would be.
- Not 'seeing' buttons: buttons on the system were sometimes represented in slightly different ways and in different positions. The eye gaze data showed that some participants, when trying to achieve a function, found it difficult to discriminate between different buttons in order to select the function. Several instances of 'not seeing' buttons – where participants' scan-paths showed that they had looked at a number of other buttons, but not the target one – where observed, particularly where target buttons were a different shape to the rest of the buttons and/or in the corners of the screen.

A few other sub-themes related directly to design considerations:

- Missing buttons: one specific part of the system, video playing, did not have a 'stop button' – a quarter of participants specifically noticed this design anomaly. In other comparable screens with a 'stop button' participants were easily able to identify and use the button, confirmed through observation of the eye-data.
- Content: in some parts of the system, participants can clearly been seen to understand the difference between the interface buttons and the content and also to demonstrate an expected cause and effect by looking to the content area after pressing a button. Analysis of the data also highlighted some areas of the system where participants did not seem to find it clear which areas of the screen were active buttons and which were displaying content.

Understanding and Cognitive Load

A number of themes emerged around participants understanding of the operation of the system, these themes can be broadly grouped into issues to do with navigation, the intuitiveness of interaction:

- Navigation:

Participants showed varying levels of understanding of the navigation: For example, there was evidence of confusion between the use of the 'Do something else' and 'Quit' buttons. However, some participants also showed good understanding of the concept of the 'Do something Else' button – frequently using it intentionally to choose another type of activity after having scanned and rejected the other options. Participants also showed varying abilities to understand the concept of the navigation between the levels in the system – most participants managed to show understanding of moving between the top level and second level to choose a specific activity. Eye gaze data showed that many participants actively scanned the available options on each level and then subsequently actively chose their preferred choice.

- Competence/intuitiveness:

Participants displayed varying levels of competence and intuitive understanding of the human-computer interaction. Many participants, during the first period of use of the system, showed an intuitive understanding of the touch-screen and how to use it, some other participants needed some instruction on the touch-screen, however they then learnt its operation. Some participants were also able to explore the system independently without prompting, including some of the more complex tasks in some cases, for example navigating through multiple levels to select preferred music tracks. For some participants, memory of the system sometimes appeared to affect their competence at using the system.

Discussion

Initial analysis of the data from the usability studies of this SIMWIN system software has shown it to provide a useful source of information for the design and study of this Assistive Technology software for older people. The recording of eye-gaze data has been demonstrated to be successful with this cohort which might have otherwise been considered challenging. Although it would have been possible to run the study without the eye data, the analysis has shown that combined with the other data streams (screen, audio, video) it provides a very rich source of data. A number of the themes that developed from the qualitative analysis of the data were reinforced through observation of the eye traces – for example, noting the eye track path across choices before a selection helped confirm that users were intentionally choosing options. Another example of the usefulness of the eye-data is shown in one of the themes where participants appear to find it difficult to see one of the buttons – without the eye data, the reason for their difficulty in selecting this button would be difficult to induce.

The use of an eye-gaze system did have, however, some practical difficulties: for example, it was difficult or impossible to calibrate the system for some older people who wore quite thick spectacles. Also, the use of wheelchairs by some participants proved problematic when it came to locating the eye-gaze screen in front of them in the ‘real environment’ of a residential home.

Other forms of analysis of the data will take place to try to establish further information from the data; hand coding will help establish task success rates and times taken to complete tasks; and the data-log from the software will be analysed to see if this provides a useful information source. In parallel with this aspect of the project, a further study is being carried out to establish the cohort’s opinions on the concept and use of this system.

Possible future research topics have already been identified – these include investigating specific aspects of interface use by this cohort – for example the relationship between content and buttons and optimal layout and positioning of these elements. The project is also likely to generate future research work regarding the use and accessibility of computers for older people.

A Case Study Describing Development of an Eye Gaze Setup for a Patient with 'Locked-In Syndrome' to Facilitate Communication, Environmental Control and Computer Access

Zoë Robertson
Barnsley Assistive Technology Team
Medical Physics, Block 14, Barnsley General Hospital, Gawber Road, Barnsley, S75 2EP
zoe.robertson@nhs.net

Marcus Friday
Barnsley Assistive Technology Team
Medical Physics, Block 14, Barnsley General Hospital, Gawber Road, Barnsley, S75 2EP
marcus.friday@nhs.net

Keywords
Eye gaze, MyTobii, environmental control, email, 'locked-in syndrome'

Introduction

For people with severe physical disabilities eye gaze technology offers a way of accessing direct communication and in addition a way of controlling their immediate environment and enabling access to functions such as email and the internet. These additional functions have the capacity to enhance a person's independence and also their social inclusion hence improving quality of life.

This paper presents a case study of a person who has very limited physical movement including limited and involuntary eye movement and looks at the process of assessing for, and setting up his eye gaze system. It also highlights the challenges faced during the development process.

In order to maintain confidentiality the patient has been referred to as Mr X.

Process

Background

Mr X had a brainstem stroke in January 2005 and was diagnosed at that stage as having 'locked in syndrome'. The Barnsley Assistive Technology Team became involved towards the end of 2005 when other health professionals, such as speech and language therapists and occupational therapists, were trying to find methods that Mr X could use to access equipment such as a communication aid.

Mr X has a minimal amount of jaw and eyebrow movement and communicates 'yes' and 'no' by looking down and up respectively. His eye movement is limited to vertical movement with little or no lateral movement. Mr X and his wife have a very effective communication method in which his wife provides an auditory scan of the alphabet. Although Mr X and his wife communicate very efficiently using this method, other people such as family members and carers, rarely use it as effectively.

Following initial assessment the Barnsley AT Team investigated several different access methods to attempt to identify a reliable and comfortable way for Mr X to obtain a switch action. Early on in the assessment process the possibility of trying eye gaze was considered but soon dismissed as at that time a system was not available which could accommodate the user having vertical eye movement only. The

COGAIN report from 2005 highlights some of the physical difficulties that can make using eye gaze difficult for some people (Donegan et al, 2005).

During the period that Mr X was trying various switch alternatives Barnsley AT Team had built links with the COGAIN project, part of which is to consider using eye gaze for people who may have more challenging requirements and, due to developments in eye gaze systems, it was suggested to Mr X that eye gaze could be tried.

Assessing for eye gaze access

The initial aim was to set up a system for Mr X to give access to the alphabet, numbers, some basic editing and punctuation tools and a rest page. Mr X was assessed using the MyTobii system. The initial challenge was positioning the MyTobii appropriately for Mr X. The first assessments were done with the MyTobii attached to the kitchen surface with Mr X in his chair in the doorway. To solve a height adjustable table was purchased. This provided better positioning of the MyTobii and also enabled Mr X to try the system whilst in his chair and whilst in bed.

The second major challenge was that, due to Mr X having only vertical eye movement and a degree of nystagmus, (involuntary eye movement), it was difficult for Mr X to achieve a good calibration. In the early assessments a calibration performed by another person was used. Although not ideal, this did enable Mr X to be able to operate the system and indicated that with better calibration this could be a successful access method.

Initial assessment sessions focussed on identifying ways of maximising Mr X's success with eye gaze. To facilitate assessment a grid which plays musical notes was used. This musical grid provided a rewarding and relatively stress free way to practise as there was, for example, no pressure to spell out words. Having identified the potential of eye gaze for Mr X a long process of refinement and development began.

Initial grid development

The initial trials had suggested that Mr X would only be able to access a single column of cells and that these would be best positioned in the centre of the display with the workspace, (area where typed letters appear), to the left. In addition previous assessment had suggested Mr X would probably manage five rows. Using these grid constraints an initial communication grid was set up for Mr X based on the scanning system which he uses with his wife.

Despite still having difficulty achieving a good calibration Mr X was able to use this initial grid set up to type a sentence. In addition, having tried this initial grid Mr X requested that each row be made a different colour, with the same colour scheme used throughout the grid set to aid his ability to distinguish between cells more easily.

Due to advances in technology at this stage the second of the initial challenges, the calibration was solved as the MyTobii software had advanced to enable one eye to be used for control. This enabled a good calibration to be achieved and so improved the precision of his selections.

The colours were added to the grid set and at the following assessment Mr X was asked questions about the set up, which he was able to answer using the MyTobii. Mr X was asked what he liked about the system and he responded that it was great because most people won't use the method he uses with his wife. He was also asked what he didn't like and his comments were about things he wanted changing within the grid set (i.e. workspace to be positioned at the bottom with rows filling the width of the screen, some comments on position of some of the letters and when he wanted it to speak). Finally, he was asked about other things he would like to be able to do, and he was keen to have some basic environmental control, document production and email.

Development of grid set

At this stage the grid set went through major changes and having solved the earlier challenges further challenges arose. The first of these was that there were concerns regarding having the workspace at the bottom of the screen as this reduced the height of the cells and it was possible that this could cause selection problems for Mr X. The second challenge was and is accessing some of the additional functions such as document production and email whilst only having a very limited number of cells available and a small workspace. The third challenge was identifying ways of enhancing the speed at which Mr X can communicate.

The first stage of this process was that the workspace was moved, the rows were extended to be the full width of the screen and the labels on the cells were repeated across the cell (this was requested by Mr X again to enhance the distinction between the cells). In addition a page of environmental control functions was added. As mentioned above the second challenge at this stage was that Mr X can only access five cells per page and one of those cells is always taken up with a way of getting to a different page. This means considering the navigational process and number of selections to get to certain functions is essential. A grid was added to give Mr X access to the additional functions required (e.g. Environmental Control). Adding basic environmental control was relatively easy as the functions Mr X required were single functions (e.g. lights) and so only took up one space on a grid. When these changes had been made a further assessment was performed to test the changes. Mr X proved that he was able to manage with the six rows and reported that he preferred the set up.

The grid underwent further development to enable Mr X to switch between communication, document production, environmental control and email. The document and email production utilise the same alphabet and editing grids however the workspace is changed accordingly when entering these modes. Additional grids have also been introduced to give the specific commands required for these functions for example a contacts list. This is where the limit of five cells has been a major challenge as to give all the required functions for efficient and independent email use takes up multiple five cell grids and trying to set these up to limit cognitive load and number of selections has been complex.

Work has also been carried out regarding the third challenge of increasing the communicative speed. For each letter Mr X types he makes three selections and so giving Mr X methods to avoid typing every word in full would enhance communicative speed. Three options were considered to approach this challenge. The first of these was word prediction. This posed additional issues as word prediction requires a display of the possible words and due to the five cell limit this resulted in either Mr X being taken to a separate prediction page or having a single cell on the front page. These options were demonstrated to Mr X however due to his visual impairment he found it difficult to see the predicted words and got frustrated and agitated. The second option tried was abbreviation expansion, for example defining that if ‘hh’ is typed it could be expanded to ‘hello, how are you?’ However with abbreviation expansion within the Grid 2 software a cell is needed to display the possible expansion and then that cell is selected to choose the expansion. Again Mr X did not like this due to it requiring a cell on the home page and it being difficult to see. The third option explored was auto replace. This is similar to abbreviation expansion however when a unique letter combination is entered it is automatically expanded to a phrase, (e.g. typing ‘hh’ results in ‘hello, how are you?’). Mr X felt that this was the best option at this stage.

Conclusion

The process has highlighted important challenges and possible solutions when using eye gaze for a person with severe physical disabilities who wants to access a range of functions. This is not finished as Mr X identifies further requirements as he uses the system more. He currently would like access to music and has had a photograph album added. He would also like to access the web and initial work has looked at this however the constraints of workspace size and number of cells does present particular issues when considering web access.

Further work is required to look at expanding the functions available using eye gaze when a person is limited in the number of targets they can manage due to physical limitations.

Acknowledgements

The authors would like to thank the patient, his family, therapists and carers. We would also like to thank Mick Donegan for his help and support.

References

Donegan M, Oosthuizen L, Bates R, Daunys G, Hansen JP, Joos M, Majaranta P & Signorile I (2005) D3.1 User Requirements Report with Observations of Difficulties Users Are Experiencing. Communication by Gaze Interaction (COGAIN), IST-2003-511598: Available from: http://www.cogain.org/results/reports/COGAIN-D3.1.pdf (accessed 14 May 2008).

COGAIN 2008 Keynote by Dr. Anthony Hornof

The Human-Technical Challenge of Developing Gaze-Controlled Devices

Abstract

Since the summer of 2003, students, collaborators, and I have been working together to develop eye-controlled interfaces. We have met with some success, such as with the development of EyeDraw and EyeMusic. EyeDraw is software that is specifically designed to enable children with motor impairments to draw pictures using their eye movements. EyeDraw has been extensively tested and validated, and is now distributed with a commercial eye tracker. EyeMusic is a system developed for computer musicians (without motor impairments) that enables a performer to control a new media art performance with just his or her eye movements. EyeMusic compositions have been performed at major computer music conferences.

Working on these projects, my students, collaborators, and I have encountered many challenges, both technical and human-centered, which are probably consistent with the difficult challenges faced by the COGAIN community in general. Some of these challenges include: Understanding and decomposing a human task to the point that it can be dictated by a series of eye movements, developing eye-controlled software within the constraints of existing frameworks for programming graphical user interfaces, connecting software across platforms, working with children and adults with severe motor impairments as software testers and collaborators, getting comfortable and integrated with a unique physical and social environment, providing roles for caregivers and siblings in the software, and building teams that span incredibly disparate disciplines and practices.

My current research efforts have for the moment put eye tracking software development on hold, and instead focus on spending time with children with severe motor impairments and their caregivers. The goal is to figure out how to move the eye tracking software development process out of the isolated lab so that it can better mesh with actual usage and practice. Along the way to designing new gaze-controlled technology, developers can perhaps benefit by learning and using other "lower tech" methods for communicating with a person with impairments. It is my hope and expectation that by facing these challenges head-on that COGAIN and like-minded researchers can better solve the incredibly difficult problem of delivering complex, thoughtful, and easy-to-use communication by gaze interaction.

Biography

Dr. Anthony J. Hornof is an Associate Professor in the Department of Computer and Information Science at the University of Oregon. He joined the faculty in 1999 and was promoted with tenure in 2005. Dr. Hornof earned his Ph.D. in 1999 and his Master's degree in 1996, both from the University of Michigan, and both in Computer Science and Engineering. He received a B.A. in Computer Science from Columbia University in 1988. After college, he remained in New York City for five years (1988-1993) where he worked as an information technology specialist for Deloitte and Touche, and also part-time as a deejay at nightclubs such as Save the Robots and M.K. He also pursued mixed-media painting during these years, and his work was featured in group shows in New York City. In 1993, he redirected his creative and intellectual energies towards a career in academia, where he now integrates his interests in computing, human factors, and creative expression. Dr. Hornof is published in the leading human-computer interaction conferences and journals, and has been awarded over $1.75 million in single-investigator research grants, including multiple awards from the National Science Foundation and the Office of Naval Research.

Dr. Anthony J. Hornof
Department of Computer and Information Science
University of Oregon
Eugene, Oregon 97403-1202
USA
email: hornof@cs.uoregon.edu