Head-Mounted Gaze Trackers for Controlling the Environment

Diako Mardanbegi

IT University of Copenhagen
Software and Systems Section

A Thesis Submitted for the Degree of
Doctor of Philosophy
July 2013

SUPERVISOR

Associate professor Dan Witzner Hansen
IT University of Copenhagen, Denmark

Abstract

This thesis aims to show that head-mounted gaze trackers can help to extend the domain of gaze-based applications into the everyday life. Gaze tracking has been well researched in the field of human-computer interaction [9], [16]. However, gaze interaction is still mostly limited to help and assist a small group of people with severe motor-skill disabilities. Gaze interaction is mostly done where a single user is interacting with a computer screen while sitting in front of a monitor. A high degree of flexibility can be obtained with head-mounted gaze trackers, where the gaze tracker is mounted on the head, thus allowing the gaze to be estimated when e.g. walking and driving. This thesis investigates the use of head-mounted gaze trackers for interaction with the environment (not necessarily a computer display) in natural everyday life situations. Gaze interaction in the mobile situations, however, poses several challenges such as:

- Using the gaze as a pointing mechanism in 3D where the geometrical relationship between the user head, body and the objects in the environment is unknown.

- Conventional gaze-based selection strategies are unsuitable for interaction with the real physical objects due to their limitations.

- Limitations of the regular head-mounted gaze trackers in accurately estimating the gaze point in 3D space.

This thesis deals with some of these challenges. The main contributions are:

1. Introducing a novel gaze-based interaction technique based on the gaze point and head gestures that are measured through the eye-movements. The major advantage of the method is that the user keeps the gaze on the interaction object while interacting with it.

2. Extending the gaze interactive applications into mobile situations for controlling the computer displays, stationary and non-stationary objects in the environment.

3. Defining and describing the parallax error in terms of head-mounted gaze tracker using the epipolar geometry, and presenting a method for real-time compensating for the parallax error.

Acknowledgments

I would like to thank my supervisor Dan Witzner Hansen who was both a supervisor and a good friend. Thanks for his patience, motivation, valuable information and guidance, and specially all the espressos that we had together.
I would also like to thank John Paulin Hansen for encouraging me to start a PhD in the Innovative Communication Group at ITU.
Thanks goes to my parents for all their support, and to my wife for her constant love and sharing the ups and downs.

Reader's Guide

This dissertation is a collection of 10 papers (Chapter 2-12). 9 papers have been peer-reviewed and published and one is about to be submitted. Each paper includes related works, explanations and conclusions. The first chapter of the thesis aims to provide a technical reader's guide. The first chapter provides summaries and motivation for each paper and tries to stress the interrelation between the papers and more importantly how they address the research questions proposed in Section 1.1.2. Some paragraphs come with lateral margins that are intended to help the reader see the main point of the paragraph. They also provide additional information and links to the other relevant sections.

Preface

List of Publications

Papers

The papers are listed in order of the submission time. Except for the first paper of the list below which is about to be submitted, all the other papers have been published.

Mardanbegi, D., and Hansen, D.W. "Real-Time Compensation for Parallax Error In Head-mounted Gaze Trackers". Journal of Machine Vision and Applications (to be submitted) — Chapter 9

Mardanbegi, D., and Hansen, D.W. "Gaze Activation Techniques". Proceedings of the ACM symposium on Eye Tracking South Africa ETSA '13, ACM Press, Cape Town, South Africa, 2013. — Chapter 5

Hales, J., Rozado, D., **Mardanbegi, D**. "Interacting with Objects in the Environment by Gaze and Hand Gestures" 3rd International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction (PETMEI2013) at 17th European Conference on Eye Movements (ECEM 2013), Lund, Sweden, 2013. — Chapter 3

Ling, R., **Mardanbegi, D.**, and Hansen, D.W. "Synergies between head-mounted displays and head-mounted eye tracking: The trajectory of development and its social consequences" Experts' Workshop on Living Inside Mobile Social, Boston University, Boston, USA, 2013. — Chapter 11

Mardanbegi, D., and Hansen, D.W. "Gaze-Based Controlling a Vehicle" CHI 2013 Workshop on "Gaze Interaction in the Post-WIMP World", Paris, France, 2013. — Chapter 4

Kurauchi, A.T., Morimoto, C.H., **Mardanbegi, D.**, and Hansen, D.W. "Towards Wearable Gaze Supported Augmented Cognition" CHI 2013 Workshop on "Gaze Interaction in the Post-WIMP World", Paris, France, 2013. — Chapter 12

Mardanbegi, D., and Hansen, D.W. "Eye-based head gestures for interaction in the car" In Proceedings of the 2012 ACM Conference on Au- — Chapter 7

tomotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12). ACM, Portsmouth, NH, USA, 2012.

Chapter 8 **Mardanbegi, D.**, and Hansen, D.W. "Parallax error in the monocular head-mounted eye trackers" In Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12). ACM, New York, NY, USA, 689-694.

Chapter 6 **Mardanbegi, D.**, Hansen, D.W., and Pederson, T. "Eye-based head gestures: Head gestures through eye movements". Proceedings of the ACM symposium on Eye tracking research & applications ETRA '12, ACM Press, California, USA, 2012. (Awarded as best full paper + best student paper)

Chapter 2 **Mardanbegi, D**., Hansen, D.W., "Mobile gaze-based screen interaction in 3D environments", Proceedings of the 1st Conference on Novel Gaze-Controlled Applications (NGCA2011), Blekinge Institute of Technology, Karlskrona, Sweden, 2011.

Chapter 10 Pederson, T., Hansen, D.W., and **Mardanbegi, D.**, "Investigations of the Role of Gaze in Mixed-Reality Personal Computing", Proceedings of the 2nd Workshop on Eye Gaze in Intelligent Human Machine Interaction (IUI'2011), Stanford University, Palo Alto, California, USA, 2011.

Abstracts

Mardanbegi, D., Hansen, D.W. "Real-time parallax error compensation in head-mounted eye trackers". Proceedings of the Scandinavian Workshop on Applied Eye Tracking SWAET 2012, Karolinska Institutet, Stockholm, 2012.

Book Chapters

Hansen, D.W., Villanueva, A., Mulvey, F., and **Mardanbegi, D.** Cogain Book, Section 6, Chapter 1, Introduction to Eye and Gaze Trackers

Hansen, D.W., Mulvey, F., and **Mardanbegi, D.** Cogain Book, Chapter 6: Discussion and Future Directions

Reports

Mardanbegi, D., & Hansen, D. W. (2010). Reflections of Head Mounted systems for Domotic Control. IT Technical report, IT University of Copenhagen.

Tools Developed in this Project

Haytham Gaze Tracker is an open source gaze tracker suited for head-mounted or remote setups. It provides real-time gaze estimation in the

user's field of view or in the computer display by analysing eye movements. Haytham offers gaze-based interaction with the real objects in the environment as well as with computer displays in fully mobile situations. The software is built by C# , using Emgu and AForge image processing libraries. More information about the Haytham Gaze Tracker can be accessed at http://eye.itu.dk.

Contents

Abstract v

Acknowledgments vii

Reader's Guide ix

Preface xi

1 Introduction 1
- 1.1 Motivations and Research Questions . . . 1
  - 1.1.1 Motivation: Mobile Gaze Interaction . . . 2
  - 1.1.2 Research Questions . . . 4
- 1.2 The Eye . . . 4
  - 1.2.1 Eyeball . . . 5
  - 1.2.2 Eye muscles and movements . . . 6
    - 1.2.2.1 Fixations . . . 7
    - 1.2.2.2 Saccades . . . 8
    - 1.2.2.3 Smooth pursuit movements . . . 8
    - 1.2.2.4 Vestibulo-ocular movements . . . 8
- 1.3 Gaze Tracking . . . 9
  - 1.3.1 Remote vs Head-Mounted Gaze Trackers . . . 10
- 1.4 Thesis Statement and Research Contributions . . . 12
  - 1.4.1 Gaze Pointing . . . 13
  - 1.4.2 Gaze-Based Activation . . . 15
    - 1.4.2.1 Gaze Activation Techniques for Interaction in 3D . . . 16
    - 1.4.2.2 The New Gaze-Based Activation Strategy . . . 18
    - 1.4.2.3 Applications . . . 19
  - 1.4.3 Parallax Error in HMGTs . . . 20
    - 1.4.3.1 Describing the parallax error . . . 21
    - 1.4.3.2 Real-time compensation for parallax error . . . 21
  - 1.4.4 HMGT, HMD & Wearable computers . . . 22

1.4.4.1 Fusion of HMGT and HMD 23

2 Mobile Gaze Based Screen Interaction In 3D Environments 25

3 Interacting with Objects in the Environment by Gaze and Hand Gestures 31

4 Gaze Based Controlling a Vehicle 41

5 Gaze Activation Techniques 49

6 Eye Based Head Gestures 55

7 Eye Based Head Gestures for Interaction In The Car 65

8 Parallax Error In The Monocular Head-Mounted Eye Trackers 71

9 Real-Time Compensation for Parallax Error 79

10 Investigations of the Role of Gaze in Mixed Reality Personal Computing 87

11 Synergies Between HMDs and HMGTs 93

12 Towards Wearable Gaze Supported Augmented Cognition 109

13 Discussion & Future Work 117
13.1 Gaze Pointing 117
13.2 Activation Strategy 118
13.2.1 Detecting the Fixed-Gaze 118
13.2.2 Separating the VOR from the Natural Eye Movements 120
13.2.3 Why VOR? 122
13.3 Parallax Error 122
13.4 HMGTs & HMDs 123

Bibliography 125

List of figures 129

List of tables 131

- Chapter 1 -

Introduction

This chapter provides the motivations for this work and describes the interrelation between the following chapters. Section 1.1 introduces the main motivation for this research as well as the research questions that are addressed in this thesis. The main research contributions of the thesis are addressed later in Section 1.4, but before that, the basic information that are necessary for better understanding the contents are presented in Sections 1.2 and 1.3. Section 1.2 briefly describes the anatomy of the eye which is needed for modelling the eye and it is mainly used in the Chapters 8 and 9. This is followed by a brief presentation of the basic physiology and movements of the eye (e.g., vestibulo-ocular reflex) that are needed in order to introduce eye-based head gestures (Chapter 6) that is one the main contributions of this work. Section 1.3 introduces the gaze tracking context and describes the difference between the head-mounted gaze trackers (which is the main focus of this thesis) and the remote gaze trackers.

§ 1.1 Motivations and Research Questions

Gaze-tracking devices are becoming smaller, more robust, and therefore, the use of gaze tracking can move into more natural settings (where the user can move around freely and interact with the objects in the environment). This thesis investigates how the domain of gaze-based interactive applications can be extended to our daily activities. The present work focuses on using head-mounted gaze trackers (HMGTs) for interacting with computers and in general with the objects in the environment (e.g., a TV or a light in the living room). This section presents the motivation for this work including a brief background of the topic. The research questions are formulated and introduced afterwards.

1.1.1 Motivation: Mobile Gaze Interaction

Gaze Interaction

Eye movements and the user's gaze point can be used in a wide variety of application domains. The variety of gaze tracking applications can broadly be divided into two categories [9] diagnostic applications where the eye tracker provides objective and quantitative evidence of the user's visual and attentional processes or neurological disorders (e.g., understanding how the consumer's visual attention is distributed over different forms of advertising which is important in the market research), and interactive applications where the gaze tracker is used as an input device of an interactive system (e.g., in human-computer interaction), and the system responds to the users gaze in some manner [9].

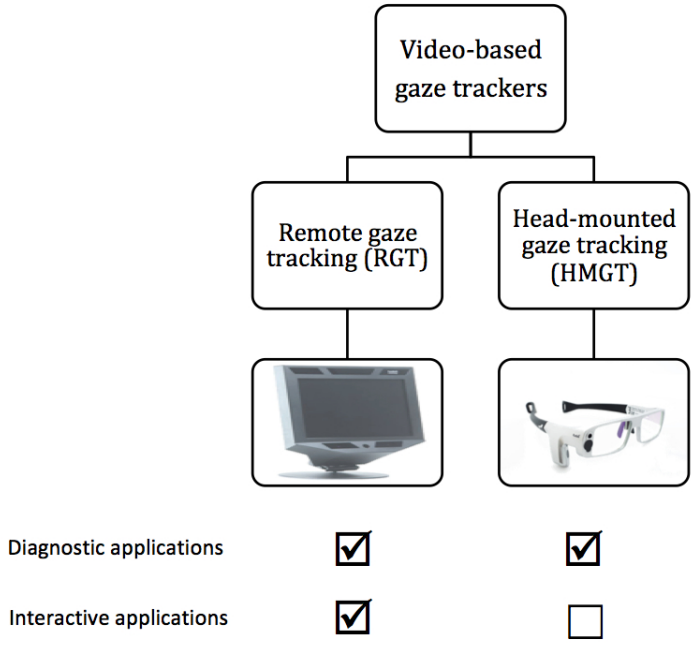


Figure 1.1: *The application domains of the head mounted gaze trackers have been limited to the diagnostic applications. Curtesy of Tobii Technologies*[1]

Mobile Gaze Interaction

Both remote gaze tracker (RGT) and HMGT have been used for diagnostic proposes such as in medical research (neurological diagnosis), psychology, and some specific applications like marketing and usability [9]. However, today, gaze interaction is mostly done with a single user sitting in front of a computer screen using a remote gaze tracker [23] (Figure 1.1). Gaze interaction has therefore been limited to help and assist disabled people by interaction with only one fixed display. Remote gaze trackers do not allow

[1] Tobii Glasses, http://www.tobii.com/

the user to be mobile and move freely, and interaction with multiple displays requires multiple RGT for each display.
With the increasing number of displays (TVs, computer monitors, mobile devices and projectors) used ubiquitously in our daily life, it is clear that gaze-based interaction holds potential for being more than a tool for interaction with a single display and aimed for limited user groups (e.g. disabled people). This thesis shows that HMGTs can be used for interaction with a computer display (similar to RGT) and even multiple displays in a fully mobile scenario in which the user can move around freely in an environment while using gaze for interaction. Mobile gaze-based interaction with displays can be generalized where several displays and users can interact simultaneously. Estimating the gaze point in the environment by HMGTs allows for extending the interaction space from the 2D space of interactive displays to the real 3D world for controlling the everyday objects such as a lamp, a fan, and etc. Figure 1.2 shows the general idea of gaze interaction in everyday life using a HMGT with compared to the limited scenario of gaze-based interaction by a RGT.

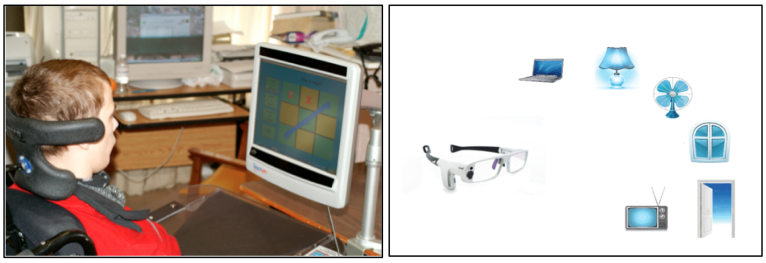


Figure 1.2: *(left) Gaze-based interaction with one computer display using RGT (right) mobile gaze-based interaction with objects in the environment using HMGT.*

With head-mounted gaze tracking technology getting better, smaller and lighter every year, it is likely that in the near future HMGT functionality will be compact enough to fit into wearable displays such as Google Glass and Vuzix smart glasses M100 [2]. From the point of view of HCI, we see that gaze as a pointing mechanism will in the short-term likely be add functionality to wearable computing devices. There is, however, a range of novel gaze-based applications waiting to be investigated in the long-term. This will include gaze-enhanced head-mounted computing devices and with improved principles for gaze-based interaction. Mobile gaze-based interaction with virtual and real objects would be highly useful in a wide variety of fields.

[2]http://www.vuzix.com/

1.1.2 Research Questions

Based on the above research motivation, the main research questions are formulated and categorized into four groups. All of these questions arise from the question of: **How the head-mounted gaze trackers can be used for interaction with the environment?**
These four groups of research questions have been addressed in the Section 1.4 and the following chapters.

Gaze Pointing
Section 1.4.1

The geometric relationship between the user, gaze tracker and the object in the environment is not fixed in the natural situations.

1. Given an estimated gaze point in the scene image, how can it be related to the objects in 3D?

2. How can gaze interaction with displays be extended to situations where the user is mobile and is able to interact with multiple displays?

3. How to use gaze pointing for interaction with real stationary or non-stationary objects in the environment?

Activation Strategy
Section 1.4.2

Besides pointing, activation commands needed for controlling an object can be obtained from the information provided by a gaze tracker.

1. Are traditional gaze-based activation techniques suitable for mobile gaze interaction in 3D?

2. Is there any other alternative to the traditional gaze activation techniques that allows gaze trackers to be used by people with and without disabilities for interaction with their environment?

Parallax Error
Section 1.4.3

A common problem with many HMGTs is that they introduce gaze estimation errors when the distance between the point of regard and the user is different from when the system is calibrated.

1. What are the main parameters that influence the parallax error in a HMGT?

2. How to deal with the parallax problem when using the HMGTs for interaction in 3D?

HMGT, HMD & Wearable computers
Section 1.4.4

Using gaze as an interaction modality for wearable computers.

1. How can the future wearable mixed reality systems benefit from gaze and mobile gaze tracking?

§ 1.2 The Eye

The main characteristics and the important aspects of the eye, the anatomy of the eye, and the basic eye movements that are relevant to gaze interaction and eye tracking, are briefly described in this section.

1.2.1 Eyeball

Figure 1.3 shows the main parts of the human eyeball. From the optical point of view, the pupil (black central circle), the iris (coloured part), and the scelra are the most interesting parts that can be observed from the outside. Pupil and iris are covered by a transparent layer called cornea that contributes to approximately 2/3 of the refracting power of the eye. The boundary between the cornea and the sclera is called limbus. The

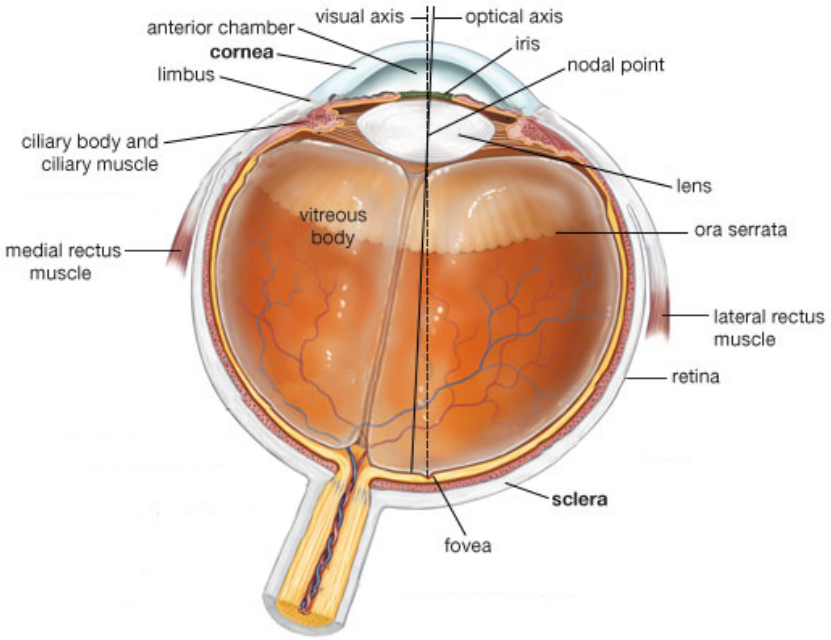


Figure 1.3: *Human eyeball. Adapted from [11]*

reflection of a light source from the anterior surface of the cornea is called *glint* (figure 1.4).

The lens has a variable shape. The change in the shape of the lens changes its refractive power and therefore, the eye can accommodate to an object a certain distance away. Behind the lens, the light passes through the vitreous humor and is received at the retina that contains photoreceptors. The retina consists of different regions. The central part is a small area called the fovea which is the region of acute vision and has a diameter about 1.8 mm (1° of visual angle). The geometrical symmetry axis of the eye-ball is called optical axis (Figure 1.3). The optical axis is actually the approximated symmetry axis of the eye because the eyeball is not completely symmetric. This approximated symmetry axis (optical axis) is defined as the line through the centres of curvature of the lens (or the cornea) and the eyeball. The fovea is not located centrally, around the optical axis. Therefore, the visual axis of the eye is defined as the line joining the fixation

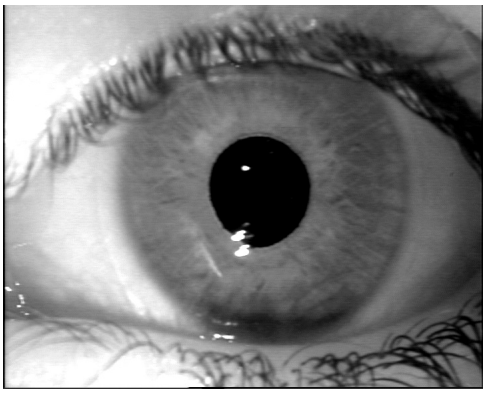


Figure 1.4: *Eye image and the reflection of the light on the cornea*

point and the fovea. The optical and visual axes intersect at the cornea center, with an angular offset that is subject dependent. There is another axis called the line of sight which is defined as the line joining the fixation point and the centre of the entrance pupil. The line of sight (LoS) is not fixed because the fluctuations in pupil size change the position of the pupil centre. However, LoS is important from the point of view of visual function, including refraction procedures, because it defines the centre of the beam of light entering the eye.

1.2.2 Eye muscles and movements

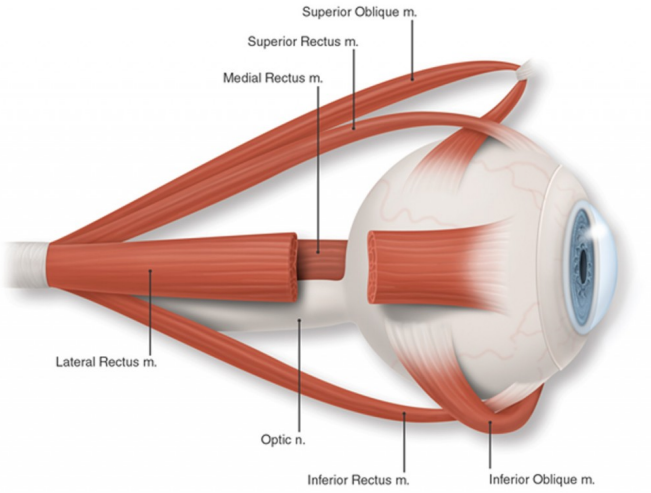


Figure 1.5: *Extraocular muscles. Adapted from [10]*

The eyeball rotates inside its socket around the centre. The center of the

eyeball lies about 13.5 mm behind the front apex of the cornea. Figure 1.5 shows the extraocular muscles which are six muscles that move the eyeball. The actions of these muscles and the movements that they create are listed in the table 1.1. There is another muscle called "levator palpebrae superioris" that unlike the recti and oblique muscles does not move the eyeball. Since its tendon inserts into the upper eyelid, it raises the upper eyelids, and opens the eyes.

Table 1.1: *Eye muscles*

Muscles	Action
Superior rectus	Moves eyeballs superiorly (elevation) and medially (adduction), and rotates them medially.
Inferior rectus	Moves eyeballs inferiorly (depression) and medially (adduction), and rotates them medially.
Lateral rectus	Moves eyeballs laterally (abduction).
Medial rectus	Moves eyeballs medially (adduction).
Superior oblique	Moves eyeballs inferiorly (depression) and laterally (abduction), and rotates them medially.
Inferior oblique	Moves eyeballs superiorly (elevation) and laterally (abduction), and rotates them laterally.
Levator palpebrae superioris	Elevates upper eyelids

The term eye movements is used to describe any rotation of the eyes relative to the head. Any eye movement requires the action of all the eye muscles (to varying degrees) [20]. All different types of eye movements made by these muscles can be observed and be detected from the outside. Video-based eye trackers measure these movements through the eye image (e.g., tracking the eye features, position of the pupil relative to the eye corners, or orientation of the iris).

Different types of eye movements are classified in different ways in the literature [20]. However, only the basic types of eye movements and their functions, and some basic definitions used to describe eye movements, that are relevant to this dissertation, are briefly introduced in the following sub-sections.

1.2.2.1 Fixations

Fixation is a temporal state that occurs when the eye is 'fixed' on an object of interest and is relatively still. We gather data from our surroundings

during a fixation. The range of the fixation duration is from 100 ms to over 500 ms and the average duration is 200–250 ms [34].

1.2.2.2 Saccades

Saccades are rapid, ballistic, and conjugate eye movements that abruptly change the point of foveal fixation. A saccade is a very specific type of eye movement. Indeed, not any movement of the eye is saccade. The small, jerk-like, and involuntary eye movements (*micro-saccades*) that occur during fixations, the eye movements that occur when following a small moving object (*smooth pursuits*), and the reflexive eye movements when moving the head while the gaze is fixed (*Vestibulo-ocular movements*) are different than saccades which will be described subsequently. The range of the amplitude of saccades varies from the small movements (e.g., made while reading), to the larger movements made while for example gazing around a room. Definition of saccadic speed and amplitude is very dependent on the equipment and protocol used to measure it [9]. However, one approximation of saccade duration in terms of degrees is the minimum duration of 20-30 ms for the first 5° (visual angle) of the movement and the duration increase of 2 ms per additional angle [7].

1.2.2.3 Smooth pursuit movements

Smooth pursuit movements are slow tracking movements of the eyes designed to keep a moving stimulus on the fovea [31]. When the target velocity is more than about $15°s^{-1}$, the smooth movements are supplemented by saccades, and above about $100°s^{-1}$ pursuit becomes entirely saccadic [20]. Smooth pursuit starts following the target after a latency of 100-150ms. When the speed of the target is faster than $2°s^{-1}$, a "catch-up" saccade occurs about 75ms after pursuit has begun. This correcting saccade brings the fovea back close to the target, and after that the pursuit continues with the speed of $v_{pursuit} = 0.9v_{target}$.

1.2.2.4 Vestibulo-ocular movements

When the point of regard is fixed and the head moves, vestibulo-ocular movements (VOR) stabilize the eyes relative to the gazed object, thus compensating for head movements and stabilizing the image on the retina. The anatomy of the vestibular system consists of two structures that are located in inner ear: Semicircular Canals that detect and respond to angular acceleration and deceleration of the head, and Otolith System that detects and responds to position of head with respect to linear acceleration and the pull of gravity. Vestibulo-ocular movements are in the opposite direction and with the same speed as the head movement. The delay between the head

and eye movements is about 10ms [31] that makes the vestibulo-ocular reflex one of the fastest reflexes in the human body [37]. The main reason that the VOR is so fast is that the signals from the semicircular canals in the inner-ear are sent as directly as possible to the eye muscles. This process involves only three neurons.

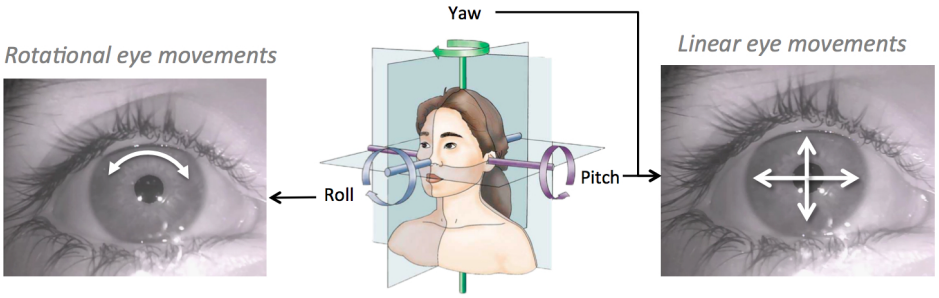


Figure 1.6: *Head rotations and the corresponding rotational and translational reflexive movements of the iris/pupil. The image in the center is adapted from [38].*

The VOR has both rotational (AVOR) and translational (TVOR) aspects [26]. Rotational (torsional) VOR used for compensating for the head roll, is mainly controlled by the two oblique muscles that rotate the eyeball around the visual axis. AVOR can be measured by measuring the iris torsion when it rotates around the LoS axis. TVOR which is mainly for compensating the head yaw and pitch, can be measured by measuring the movements of the pupil center in the eye image (Figure 1.6).

The "gain" of the VOR is defined as the eye velocity divided by head velocity during the head turn: $gain = \frac{v_{Eye}}{-v_{Head}}$. The gain of the horizontal and vertical VOR is usually close to 1.0, but the gain of the torsional VOR is generally low [8].

§ 1.3 Gaze Tracking

Eye tracking refers to monitoring the eye movements. A gaze tracker is a device that measures the eye movements and additionally estimates the user's gaze using the information obtained from the eyes. There are a number of techniques for measuring eye movements. The most common and widely used technique is video-based eye/gaze tracking that uses video cameras to record the image of the eye and extracts the information from the eye image. Depending on the gaze estimation technique employed, the output of the gaze trackers may be the Point-of-Regard (PoR) or gaze direction in 3D space, or it may be a point in a 2-dimensional image (e.g., the user's field of view (scene image) or a computer display) [13]. The core problem of gaze

estimation is finding the relation between the eye data and gaze. There are different approaches to find this relationship depending on the hardware employed and the knowledge available about geometrical relationships between the eye, head, and hardware components [13].

1.3.1 Remote vs Head-Mounted Gaze Trackers

Video-based gaze trackers can be categorized into two different types: Head mounted gaze tracker (HMGT) and Remote[3] gaze tracker (RGT).

Figure 1.7: *(left) RGT estimates the gaze point in a fixed two-dimensional space e.g., computer display. (right) In contrast, HMGT estimates the gaze in the user's FoV.*

In RGT, the system components (mainly the eye camera) are placed away (remote) from the user e.g. on a table. RGT systems usually only allow for estimating the point of regard (PoR) on a planar surface (fixation plane) e.g., a computer display. An attractive property of remote gaze tracking is that it is non-invasive, however it has a limited field of view. In order to compensate for small movements of the head [4], RGTs usually use one or more infrared light sources that are fixed relative to the fixation plane. The stationary light sources (mounted on a table or a screen) are used as reference points for gaze estimation. The gaze estimation space of a RGT which may be a 2D plane (e.g., a computer screen) or a fixed 3D volume around the light sources ([19], [14]), is defined as a space containing all fixation points that can be estimated by the system. The gaze estimation space of a RGT is limited to a space around the fixed light sources. The range of movements of the subject's head is limited when using a RGT because the head should be in the field of view of the camera in a way that the reflections of the light sources are on the surface of the cornea.

Unlike the RGT systems, HMGTs allow for a higher degree of mobility[5] and they have at least one camera for capturing the eye image and another

[3] Unlike the HMGT, the eye camera is not mounted on the user's head

[4] For example, the maximum range of head movements is about 37 × 17cm for Tobii TX300

[5] Head-mounted gaze trackers are also called mobile gaze trackers.

one for capturing the scene image. HMGT systems are commonly used for estimating the gaze point of the user in his field of view (Figure 1.7). Mobility is the most important advantage of the HMGT systems compared to RGT systems, which makes the HMGTs suitable for mobile usage like walking and driving. The main limitation of HMGT systems is that the camera have been mounted on the head. However, as with many other electronic devices, HMGT technology has become smaller and more agile using smaller and better cameras.

The HMGTs may or may not have the scene camera that records the image of the user's field of view. HMGTs that do not have the scene camera, they usually estimate the line of gaze relative to the user's head (or the center of the eyeball) (Figure 1.8) or estimate the PoR as a point in a plane in front of the user but it is measured relative to the head coordinates system. In these cases, in order to be able to compensate for the head movements, and to know at which point in space the user is looking, more information is needed about the position of the head in space. This information can be obtained from pos sensors connected to the user's head.

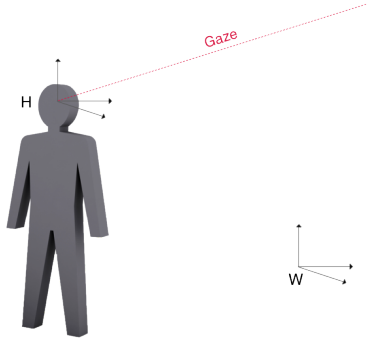


Figure 1.8: *HMGTs that don't have a scene camera, usually estimate the line of sight in the head coordinates system (H) which can move relative to the world coordinates system (W)*

HMGTs that have a scene camera, estimate the gaze point in the scene image. This requires a mapping function between the eye features and the scene image. This is obtained through a calibration procedure (Figure 1.7.left).

In general gaze trackers can provide an abundance of information about the subject and the environment (Figure 1.9). Different types of eye-related information can be obtained from the eye camera (with both RGT and HMGT) including: measurement of eye movements, estimation of the user's gaze, monitoring of the behaviour of eye muscles and frequency of blinking (e.g, as one of the indicators of the user's fatigue [35]), measurement of the

[6]http://www.tobii.com/

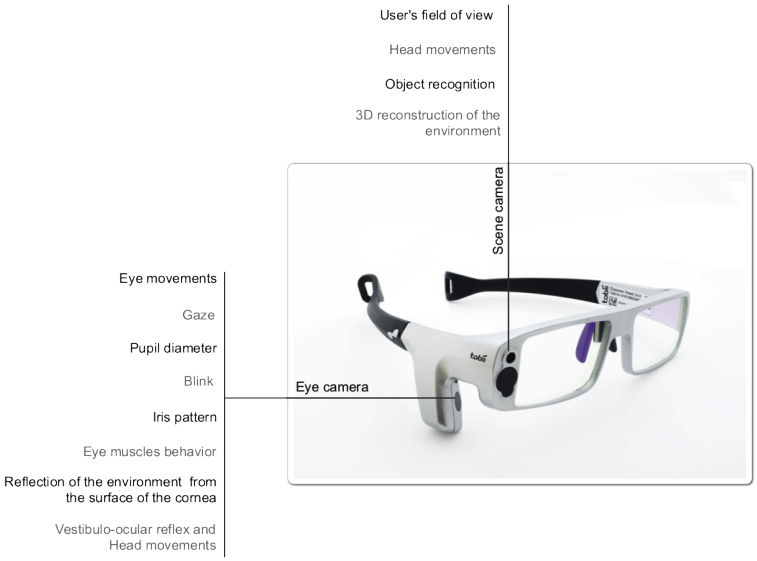


Figure 1.9: *Many different types of information can be obtained from the eye and the scene cameras of a HMGT. Curtesy of Tobii Technologies*[6]

pupil diameter (e.g., as an indicator of the cognitive load [30]), extracting the iris pattern (e.g., used as a biometric), projection of the the light in the user's environment from the surface of the cornea [24], and measurement of the vestibulo-ocular reflex that coordinates eye movements relative to head movements (e.g., as a way of measuring the head rotations (roll, tilt, and pan) through the eye movements). On the other hand, the scene camera of a HMGT is like a regular camera that records the user's field of view. It can be used for many different purposes such as: object recognition in the user's field of view, reconstructing the environment, and for monitoring the head movements of the user. In general any type of information that is obtained from a regular camera can be obtained from the scene camera of a HMGT.

§ 1.4 Thesis Statement and Research Contributions

This section introduces the contributions of this thesis. The research questions of this thesis are described in more details and are addressed in the following subsections. Each subsection corresponds to collection of the papers included in the thesis and presents a brief background summary and motivation needed to connect the particular chapter into the broader sweep of the thesis.

1.4.1 Gaze Pointing

The information provided by the gaze tracker can be used in different ways for interacting with a system. However, pointing seems to be the most obvious use of gaze due to the fact that humans naturally tend to direct the eyes toward the target of interest [17]. Point of regard is used for pointing in almost all gaze-based interactive techniques. The typical use of gaze as a pointing mechanism is to control the cursor position on the screen while sitting in front of a computer monitor. Remote gaze trackers usually estimate the gaze point in a 2-dimensional screen that limits the interaction space to a stationary plane around the light sources. However, the estimated gaze point is exactly the location of the interaction object. Once the position of the gaze point inside a computer display is known, selection/activation commands can be executed by the other interaction modalities or the gaze-based activation strategies. However, as it was mentioned in the section 1.3.1, HMGTs don't estimate the gaze point directly in the world coordinate system. Therefore, knowing the object of interest in the environment is not as straight forward as stationary situations with remote gaze trackers.

When the HMGT estimates the user's line-of-sight in the eye coordinates system [7] (Figure 1.8), calculating the exact coordinates of the PoR in the world coordinate system, needs more knowledge about the geometry of the scene and the user (location and orientation of the user's head) in space. In this case, location of the interaction object is obtained by calculating the intersection between the gaze and the scene. Head-mounted gaze trackers that have a scene camera and estimate the gaze point in the scene image, can make use of the information about the scene provided by the scene camera. The exact coordinates of the point of regard in space is not actually needed in many gaze interactive applications, and basically we only need to know which object the user is pointing at. There are a variety of computer vision techniques [36] that can be used for recognizing the object in the scene image that the gaze point is on it.

In contrast to the conventional gaze interaction applications with RGT, mobile gaze interaction can go beyond just interacting with computer displays and can also be used for interaction with real objects in the environment. Therefore, interaction object in 3D can be either an item displayed in a display or a real object in the environment. Figure 1.10 shows some possible scenarios showing that gaze as a pointing mechanism can potentially be used for pointing to a virtual item in a computer display or a real object in the environment (e.g., a lamp or a robot). The subject can also use gaze for pointing to a target point in the space to where a vehicle (e.g., a mobile robot) is supposed to go. The present work investigates the possibility of using gaze in all these different scenarios.

Chapter 2 presents a method for mobile gaze interaction with computer

[7] The coordinates system attached to the center of the eyeball

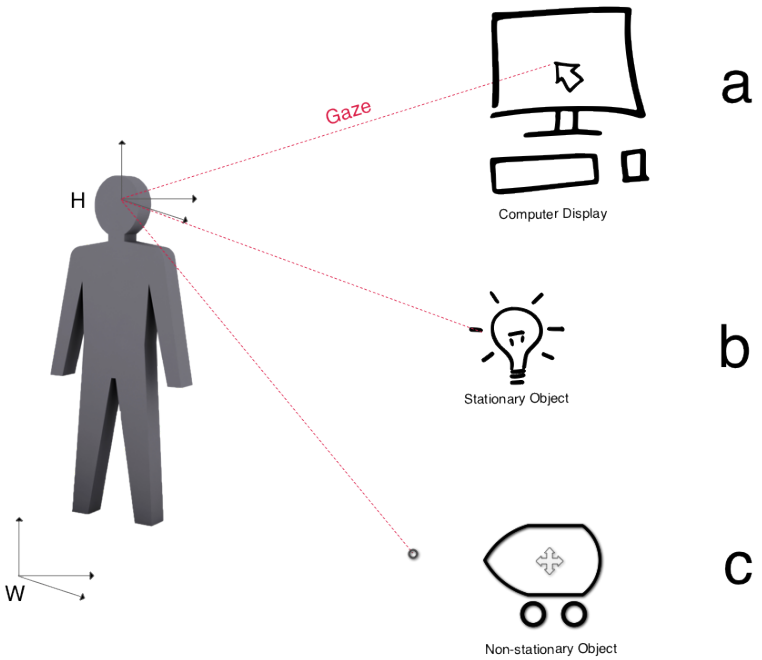


Figure 1.10: *Gaze interaction in 3D where the gaze is used for (a) pointing to an item or a point (that indicates the next position of the cursor) on a display, (b) pointing to an object of interest in the environment (stationary or non-stationary), (c) for determining the destination point of a remote vehicle in space*

displays (Figure 1.10.a) in a challenging situation where the exact position of the point of regard in space is unknown. The scene image of the HMGT is used as a resource for obtaining information about the surroundings. After detecting the display in the scene image the user's gaze point is mapped to the coordinate system attached to the display and therefore the system estimates the PoR inside the display. This allows HMGTs to be used for interaction with any planar display such as computer displays, mobile phones, and even projected displays. Additionally, an effective method for identifying the displays in the field of view of the user is presented using *temporary visual markers*. Identifying different displays in the field of view allows for interaction with multiple displays in the environment. The presented approach provides an efficient way of identifying displays and it is sufficiently general and scalable to situations with multiple gaze trackers and multiple displays located in individual networks (e.g. located over large distances).

Chapter 4 proposes a taxonomy of different situations that point of regard can be used for controlling a non-stationary object in the environment (Figure 1.10.c). This chapter discusses the possibilities and limitations of how gaze interaction can be performed for controlling vehicles in general challenging situations where the user and robot are mobile and movements may be governed by several degrees of freedom (e.g. flying).

1.4.2 Gaze-Based Activation

The gaze-based interactive applications typically employ gaze as a pointing modality. Although gaze is well suited as a pointing mechanism, the eyes alone lack a reliable selection mechanism [16]. The nature of the eye movement is completely different from hand motor control. The Midas-touch problem (the accidental selection of anything upon which a user rests his or her gaze) is one of the issues raised by this fact and has been mentioned as a limitation of gaze-based interaction in the literature. Information with which to supplement gaze point is needed to overcome the Midas-touch problem when using the eyes for activating an object. Various gaze interaction strategies obtain this extra information differently [22].
The term *gaze activation* is used when gaze is used in any form in the process of providing input information needed for selecting an object or executing an action command. There exist different gaze-based activation methods that can be used together with gaze pointing for enhancing the interaction with computer user interfaces (e.g., dwelling and gaze gestures). A comprehensive review of different gaze-based activation strategies is given in [22]. One important point here is that these conventional gaze-based activation methods are basically initiated to help people with severe motor impairments (e.g., ALS[8] patients that are only able to move their eyes) to interact

[8] Amyotrophic lateral sclerosis

with computer displays. However, gaze interaction with real objects in 3D by the general population may require different considerations than gaze interaction with computer graphical user interfaces by disabled people who can, for example, only move their eyes.

1.4.2.1 Gaze Activation Techniques for Interaction in 3D

There have been some studies that have applied the gaze interaction for controlling the real objects. Several gaze-based interfaces have been proposed in the literature that allow for controlling the objects in the environment (e.g., turning lamps on and off) by using the conventional gaze activation techniques [5],[32]. This is done through a graphical user interface and using a remote eye tracker, such that the individual first selects a symbol indicating a particular device from a menu on a computer monitor by gazing at it, then operates the device through the interface that is subsequently displayed [25].

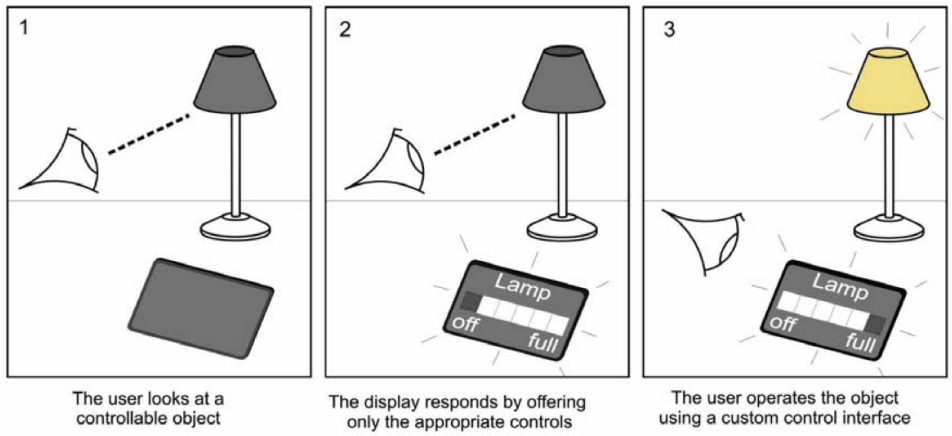


Figure 1.11: *Attention Responsive Technology (ART) proposed by [1]*

Controlling the objects in the environment using gaze has also been done through HMGTs. Gale et al. [1] used HMGT as a device for monitoring the user's object of interest in the environment. They developed Attention Responsive Technology (ART) system, i.e. the user first looks at a particular device in the environment; then the system recognizes the object of interest in the scene image; and then an interface dialogue appears on a display screen asking the user to make his/her control input by various means (Figure 1.11). Therefore, the control input is obtained through a GUI. This is a rather indirect and inconvenient way of interacting with the environment which is due to the limitations of the conventional gaze activation techniques (e.g., dwelling and gaze gestures) that make them unsuitable for controlling the real physical objects through a HMGT. One of the main goals of this thesis is

to investigate how we can employ HMGTs to interact with the environment in a more direct way without necessarily having a monitor and GUI. Chapter 3 presents an example scenario where HMGT is used for both detecting the object of interest and for detecting the action command (in this case a hand gesture) through the scene camera. A better example of gaze interaction with objects in 3D (Figure 1.10.b) is given in Chapter 4 where a gaze-based method is used for pointing and activating a non-stationary object in the space.

Chapter 5 presents a systematic review and a new taxonomy of gaze activation techniques that allows us to make a comparison between different existing gaze activation techniques, and to investigate the performance of each technique for interaction in 3D. The taxonomy presented in the Chapter 5 examines gaze activation strategies from the point of view of the source of information rather than the eye movements. The taxonomy is based on the way that information provided by a gaze tracker can provide applications with commands such as selections. Chapter 5 introduces the different conventional gaze activation techniques and categorizes them into 4 classes, and also describes their major limitations. The main limitations of the gaze activation techniques are summarized below:

Table 1.2: *The main limitations of gaze activation techniques (from the point of view of interaction in 3D)*

Method	Limitations
Blinking	- Having a limited range of commands. - Mistaking the natural blinks. Inconvenient long term use.
Dwelling	- Having a limited range of commands. - Needs for visual feedback and GUI.
Gaze Gestures	- Gaze is removed from the object. - It requires some pre-defined target points (e.g., off-screen targets) to help the user performing a desired gaze pattern. - Complex gaze gestures are unnatural and inconvenient.

The main considerations that should be taken into account when investigating the gaze interaction in 3D are:

1. It would be more convenient to keep the gaze on the object of interest while interacting with it.

2. The interaction does not need to be done only through the eye movements (considering the general population) and eye movements can be used together with the other modalities.

3. Interaction with different type of objects in the environment requires a wide range of interaction commands.

4. Interaction with the objects should not necessarily require the visual feedback and graphical user interfaces.

It can be seen from the Table 1.2 that none of the conventional gaze activation techniques addresses the four considerations above. However, Chapter 6 introduces a novel gaze-based activation technique which uses (voluntary) fixations on an object for pointing and involuntary eye movements (caused by VOR introduced in the Section 1.2.2.4) for activating the object. This new approach has been identified by looking at the way that eye movements are used in the previous techniques. The main difference between this new technique and the other conventional techniques is that it does not assume that moving the eyes necessarily changes the PoR even when the PoR is fixed in space. Therefore, it benefits both from the fixed gaze point and the eye movements.

1.4.2.2 The New Gaze-Based Activation Strategy

Eye-based head gesture (introduced in Chapter 6) is a novel method for enhancing gaze-based interaction through voluntary head movements. Eye-based head gesture is based on the fact that when the point of regard is fixed and the head moves, the eyes move in the opposite direction due to the vestibulo-ocular reflex introduced in section 1.2.2.4. The method allows the gaze position to remain fixed while the pupil position is changing over time. Both translational and rotational VOR can be detected through the eye movements. This method uses only the information obtained from the gaze tracker for measuring the head movements and to determine whether the PoR is fixed on the object of interest. The eye-based head gesture technique can be achieved with both remote and head-mounted gaze trackers and provides us a gaze-based interaction method for executing commands in remote and mobile situations. This technique has several advantages as a gaze-based selection/activation strategy:

1. Gaze is fixed on the object of interest during the interaction

2. Eyes are actively used only as a pointing mechanism not for motor control

3. Head-gestures seems to be more intuitive for communication than gaze gestures

4. Lower physiological and cognitive load compare to gaze gestures

5. In addition to discrete gestures, continuous head movements can be used for changing the continuous and analog interactive objects e.g.,

for scrolling, zooming, panning, dragging items, and adjusting the volume

6. The method can be applied to mobile interaction with objects in 3D without visual feedback and further it allows for more complicated interaction than just pointing and clicking

This new technique has been described in more details in Chapter 6.

1.4.2.3 Applications

Considering that the user is capable of rotating their head, the proposed technique can be applied to interaction in almost all gaze-based interactive applications. Since even very small head movements can be measured by the proposed technique, the head gestures can be very small not necessarily involving wide movements of the head.

Gaze pointing combined with head gestures provides a convenient way of interacting with the environment that requires little additional effort and causes less physical fatigue. Eye-based head gesture also allows for hands-free interaction with the virtual and real objects when the hands are occupied and cannot be used.

Chapter 6 presents two example applications showing the capability of the method for interacting with a screen at kitchen during cooking (and when the hands are occupied) and also for interacting with smart phones.

Chapter 4 shows the potential of this method for controlling the non-stationary objects and robots in 3D.

Chapter 7 shows the applications of the eye-based head gesture in the automotive context for interaction with the objects inside or outside the car, including heads-up displays. User interfaces in cars have become more complex with many new functionalities. It is important to find better ways of interaction with car user interfaces. A new way for interaction with objects inside the car has been suggested. The proposed approach involves three steps, first, the user looks at the object, and then fixating their gaze on a specific point on the windscreen and then they perform a head gesture. When the driver looks at an object in the car (e.g. the window) the gaze tracker recognizes that specific object and then the user can control that object using the eye-based head gestures while looking at a specific point on the windscreen. Since fixation on a target point on the windscreen (which can be detected by the gaze tracker) only occurs when the driver wants to interact with an object, natural head movements are not mistaken. This allows to use simpler head gesture vocabularies that require a lighter cognitive load. This method requires only a quick look at the object of interest which helps to minimize the amount of time that the driver's visual attention is away from the forward roadway. Chapter 7 shows how a video-based gaze trackers can potentially be used as a single multi-purpose device in the car.

It can be used for head gesture recognition, fatigue detection, monitoring the driver's visual attention as well as and gaze estimation.

Besides driving a car, in many other situations where losing the visual attention may increase the human risk (e.g., driving the wheelchair or in the high risk environments like the power plants control rooms), eye-based head gestures can be used for interaction without requiring the users to look away from their usual viewpoints. It can also be a way to interact with head-up displays in the automobile or aircrafts. This technique can also potentially be used for interacting with head-mounted displays. This has been discussed more in the Section 1.4.4.1 and a particular application example is presented in Chapter 12.

1.4.3 Parallax Error in HMGTs

When using a HMGT in 3D, the distance between the user and the fixation point in space varies. It may be very close (e.g., 0.05 m) or very far away from the subject. This requires the HMGT to estimate the gaze point in 3D with sufficient accuracy. However, HMGTs that have a simple design (monocular with one scene camera) and estimate the gaze point in the scene image have a common problem that they can only estimate the gaze point accurately for the calibration distance (the distance between the subject and the plane for which HMGT is calibrated). When the distance between the point of regard and the user is different from when the system was calibrated, they introduce gaze estimation errors which is called *parallax error*.

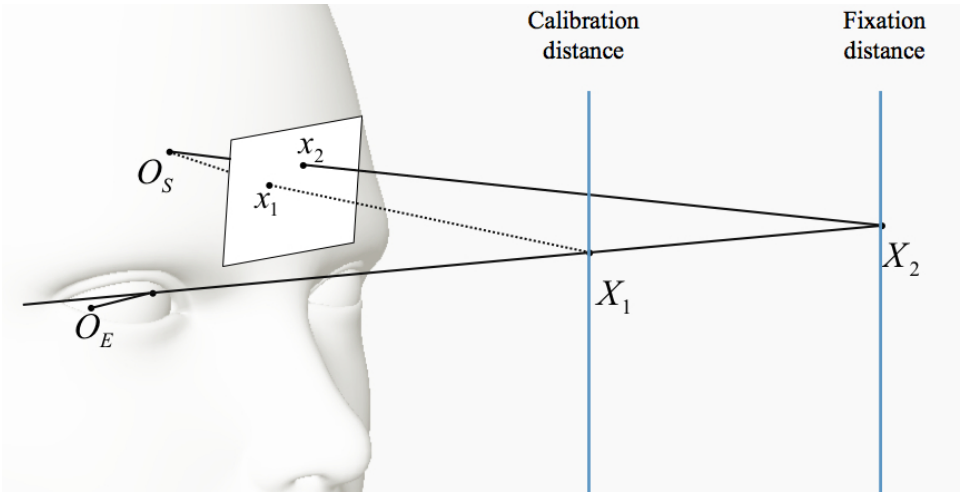


Figure 1.12: *Parallax error in a HMGT*

Figure 1.12 illustrates the parallax error in a HMGT when the user looks at two different points in different depths. When the user looks at the point X_1 in the calibration distance the estimated gaze point in the scene image is

the point x_1. When the user is looking at a further point X_2 (at the fixation distance) without changing the gaze direction, using the same calibration data still the point x_1 will be estimated as the PoR but the actual PoR in the image is the point x_2. This error happens because the the eye and the scene camera are not co-axial.

1.4.3.1 Describing the parallax error

Although the parallax between two views in a stereo camera system is a well known topic in computer vision, there has not been a comprehensive study on the error caused by the parallax between the eye and the scene camera in HMGTs. Li [21] investigated the parallax error behaviour in a simplified model of a HMGT, where the scene camera is mounted above the eye (only a vertical displacement). Bernet et al. [4] presented an extended description of parallax error based on the geometry for when the camera has one degree of rotation (pitch). However, the angle between the visual and optical axis is not considered in these analyses. Chapter 8 presents a detailed study of parallax error in HMGTs. Parallax error has been defined and described using the epipolar geometry for a general configuration. Furthermore, different parameters that change the error are introduced. One of the main outcomes of the presented study is that it shows that the difference between the visual and optical axes does not have a significant effect on the parallax error, therefore the error can be described using epipolar geometry. Parallax error has been simulated in HMGTs and the effect of different parameters has been shown. The results and the provided simulation code can help in finding the optimum configuration of the system when designing a HMGT. The relationship between the parallax error and the geometry of the system (in general configuration) has been also described in Chapter 9. Parallax error for any point in the scene image can be directly obtained as a function of system parameters. Furthermore, the error function allows us to better investigate the functional features and behaviour of the error.

1.4.3.2 Real-time compensation for parallax error

The standard method for dealing with parallax error is to calibrate the gaze tracker for a finite set of distances prior to use, and then apply the proper mapping function for gaze estimation in different distances. The approach is therefore most appropriate for off-line gaze analysis. In this case, the distance of the fixation plane (the working plane containing fixation points while using the system) should be set manually in the software before gaze estimation. Chapter 9 presents a new method for real-time compensation for the parallax error (Figure 1.13). The method presented there can be used in many mobile gaze tracking applications in which the fixation depth can be measured through the scene image. The method is based on the changes of

the error pattern at different depths. This method requires collecting some sample data prior to using the HMGT in order to approximate the error pattern in the scene image and at different distances. The next time that the system is used, the error of the PoR can be interpolated and compensated for by having the depth of the plane that the user is looking. A rational model has been suggested for approximating the error function at different depths, and it has been compared with the polynomial models.

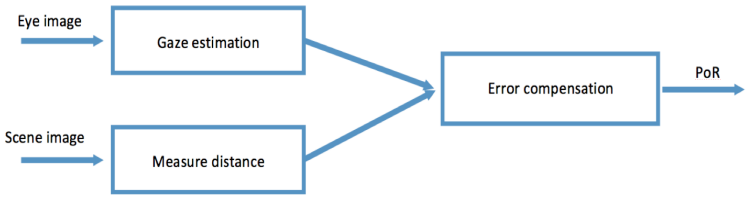


Figure 1.13: *Real-time compensation for parallax error*

1.4.4 HMGT, HMD & Wearable computers

HMGT contains much more information than just gaze and eye movements (Figure 1.9). Additionally, there is a strong link between gaze, attention, and cognitive processes [17]. Considering these facts, it seems obvious that HMGTs can become one of the main elements of the future wearable computing devices where gaze could be used as a pointing mechanism and as an interaction modality. Gaze interaction in natural settings allows for a wide range of applications. This thesis has shown that gaze can be used as an alternative input modality for hands-free interaction with a variety of virtual/real objects in the environment (e.g., in the home automation context or smart environments). Mobile eye tracking and gaze estimation can help create better mixed-reality personal computing systems. Chapter 10 discusses the the role of gaze in an egocentric interaction paradigm model through a situative space model (SSM) [28]. Furthermore, it extends the SSM model to better incorporate the role of gaze, and for taking advantage of emerging mobile gaze tracking technology. An interesting property of the SSM model is that unlike the classical Human-Computer/Machine interaction, it pays an equal attention to virtual and physical objects, circumstances, and agents. Besides, the model does not simply look at the gaze tracker as an input device but considers gaze as a visual modality for the user action and perception. It has been shown that gaze plays a fundamental role in defining the visual perception space for a given human (agent), and furthermore, in some cases provides enough data for making predictions [20] and detecting a set of objects (including objects across several existing SSM spaces and sets) that the given human agent is attending to. When looking at gaze tracking as a beneficial feature for wearable computers, its

interaction with head-mounted display (HMD) as the main visual output device of the wearable computers should be also investigated.

1.4.4.1 Fusion of HMGT and HMD

This subsection focus on the synergy between HMGT and HMD in a wearable computer. Although, gaze interaction in 3D can be done directly and without necessarily having a graphical user interface, providing visual feedback can enhance the functionality of gaze interaction in the environment. Besides interaction with the environment, gaze can potentially be used for interaction with head-mounted graphical user interfaces. On the other hand, using gaze information seems to be one of the natural ways for automatically filtering and mediating the contents displayed on HMDs. The description of actual systems combining both gaze and wearable technologies are still rare in the literature, however, there have been some attempts to show the potential benefits of combining the HMGTs and HMDs ([2], [15], [12], [27], and [29]). Chapter 11 discusses the synergy between gaze tracking and HMD technologies and the eventual melding of these two technologies that have two separate threads of development. Furthermore, the paper discusses the consequences of the gaze supported head-mounted displays, in terms of privacy, power relationships, and the social communications. Chapter 12 presents a possible application where gaze is used for filtering the contents displayed on a HMD. Three gaze-based interaction modes (including eye-based head gestures) are proposed for an augmented cognition application that provides information of the person being looked at. The system updates (and displays) the information about the person being looked at when for example a head gesture is performed. In this application, eye-based head gestures are shown to be an appropriate interaction modality whereas the other dwelling and gaze gesture activation strategies are less suitable. The main reason for this is that the eye-based head gesture technique uses the eye only for pointing and does not overload the eye with a control task.

- Chapter 2 -

Mobile Gaze Based Screen Interaction In 3D Environments

Mobile gaze-based screen interaction in 3D environments

Diako Mardanbegi
IT University of Copenhagen
Rued Langgaards Vej 7, 2300 KBH. S.
Phone: +45 50245776
dima@itu.dk

Dan Witzner Hansen
IT University of Copenhagen
Rued Langgaards Vej 7, 2300 KBH. S.
Phone: +45 72 18 50 88
witzner@itu.dk

ABSTRACT

Head-mounted eye trackers can be used for mobile interaction as well as gaze estimation purposes. This paper presents a method that enables the user to interact with any planar digital display in a 3D environment using a head-mounted eye tracker. An effective method for identifying the screens in the field of view of the user is also presented which can be applied in a general scenario in which multiple users can interact with multiple screens. A particular application of using this technique is implemented in a home environment with two big screens and a mobile phone. In this application a user was able to interact with these screens using a wireless head-mounted eye tracker.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Input devices and strategies, Interaction styles, Evaluation/Methodology; H.5.3 [**Group and Organization Interfaces**]: Collaborative computing.

General Terms

Human Factors

Keywords

Head-mounted eye tracker, Screen interaction, Gaze-based interaction, Domotics

1. INTRODUCTION

This paper presents a robust method to use head-mounted eye trackers for interaction with different screens in a 3D environment. Through this paper it is shown that gaze interaction can be generalized for usage in 3D environments where multiple screens and users can interact simultaneously thus allowing users to move around freely in a 3D environment while using gaze for interaction.

Eyes are meant for 3D navigation tasks, yet most gaze-aware applications are focused on 2D screen-based interaction. With the increasing number of displays (TVs, computer monitors, mobile devices and projectors) used ubiquitously in our 3D daily lives, and with the current developments in small high quality cameras that transmit data wirelessly it seems obvious that gaze-based interaction holds potential for more than tools aimed for limited user groups (e.g. disabled) and there is a long range of novel gaze-based applications waiting to be investigated with improved principles for gaze-based interaction in 3D environments.

Gaze interaction is mostly done with a single user sitting in front of a screen using a remote eye tracker. An attractive property of remote eye tracking is that it is quite accurate and allows for non-invasive interaction. Remote eye trackers are restricted by only allowing interaction with a single screen. Besides it only has a limited field of view. Multiple screen interaction can be obtained with multiple remote eye trackers but may induce high costs and it will despite the multiple eye tracker and novel synchronization schemes still not facilitate the user with a complete freedom to move. A high degree of flexibility can be obtained with remote eye trackers, where the eye tracker is mounted on the user and thus allows gaze to be estimated when e.g. walking and driving. Even though head mounted eye trackers have reported higher accuracies than remote eye trackers [11], head mounted eye trackers only give gaze estimates on the scene image and not on the object used for interaction e.g. the screen. Using head mounted eye trackers for screen-based interaction is also complicated by the fact that the screen may be viewed from multiple viewpoints. Head mounted eye trackers can be used with multiple screens without synchronization of eye trackers but requires some method for knowing which screen is in the field of view. Head mounted eye tracking may potentially allow multiple users share the same screen without additional requirements on the eye tracker.

This paper addresses the particular problem of using head mounted eye trackers for interaction with planar objects (such as screens and visual projections on planar surfaces). While the general problem of recognizing objects in images is challenging this paper presents a novel and effective method to determine which particular screen the user is looking at without heavy computational demands yet without cluttering the interaction space with tags attached to the objects. The proposed method also supports multiple users interacting with the screens simultaneously.

Section 2 describes previous work and section 3 gives a brief introduction to head mounted eye trackers. Section 4 describes the method for detecting and recognizing screens in the scene image and transferring gaze estimates from the scene image to the object space. Section 5 presents a particular application of using the generalized technique for a home environment and section 6 concludes the paper.

NGCA '11, May 26-27 2011, Karlskrona, Sweden

2. PREVIOUS WORK

A wide variety of eye tracking applications exist. These can broadly be categorized into diagnostic and interactive applications [8]. Interactive applications were initiated in the early 1980's [2] and further developed by [21]. A large body of novel applications has been proposed to use gaze information for improved interaction with screen-based applications.

Gaze interaction with screens is mostly done through remote eye trackers and significant attention has been given to applications that assist disabled people [13].

Eye interaction has also been used to control objects in the 3D environment, like turning lamps on and off via the monitor [5, 3], which is a rather indirect way of interacting with 3D objects. Head mounted eye trackers have been intended for environmental control. Gale [10, 20] proposes to use head-mounted eye trackers as a device for monitoring the attended objects in a 3D environmental control application. However, this work did not actually use the head mounted eye trackers for direct interaction with user interface and objects, and they relied on alternate sources to do the interaction e.g. remote eye trackers.

Some other applications include attentive user interfaces [14] (e.g., gaze contingent displays [7] and EyePliances [19]). Although remote eye trackers can be use for interaction with attentive user interfaces on public screens [1] or large screens, there are still the lack of mobility and multiple user interaction, and head mounted eye trackers may be better suited for this purpose. Eddy (2004) suggests using the head-mounted eye trackers together with a head-tracking device for monitoring the user's gaze when viewing large public displays [9], however head-mounted eye tracker was not used for gaze interaction.

Object identification can be done thorough visual markers and can either be visible [17] or invisible. Visible markers include QR-Codes, Microsoft color tag, and ArToolKit [15] and invisible tags can be obtained by using polarization [16] or infrared markers [18]. While being simplifying detection, the visual markers are limited by the need to place the markers on the objects.

3. HEAD MOUNTED EYE TRACKER

There are generally two types of video-based eye trackers: remote gaze trackers and head-mounted eye trackers [6]. Head mounted eye trackers (HMET) have at least one camera for capturing eye movements and another for capturing scene images (Figure 1-a). The cameras are mounted on the head to allow the user to move freely. This is in contrast to remote eye trackers that have only one camera located away from the user for capturing the eye image. Remote eye trackers estimate the point of regard on the screen while head-mounted eye trackers estimate the user's point of regard in the scene image (displayed in figure 1-b with a cross-hair).

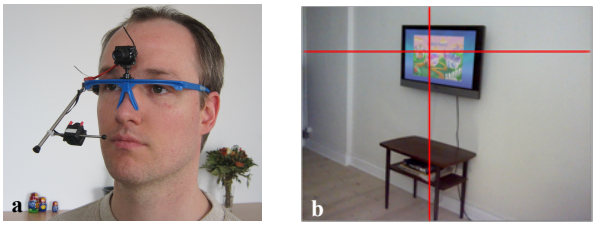


Figure 1.(a) HMET system and (b) scene image with a red cross-hair to indicate point of regard.

The head mounted eye tracker used in this paper is shown in figure 1 and was made by the authors. The eye tracker transmits image data to a server wirelessly for further processing.

4. FRAMEWORK

The general framework addressed in this paper contains several screens (clients), a server and one or more eye trackers. An example of a potential multi screen scenario is shown in figure 2. The user is wearing the HMET holding a mobile phone (screen) in the hand. There are two other screens in the background that could also be used for interaction.

Communication between system components (eye trackers, screens/clients and servers) builds upon TCP/IP. The purpose of the server is to facilitate communication between the eye tracker and the screens. Images from the eye tracker are sent wirelessly for further processing on a remote PC. The remote PC locates the screen, estimate gaze on the screen and subsequently sends the information to the server.

Figure 2. An example of a potential multi screen scenario with a user wearing a HMET, and able to interact with a TV screen (on the wall), a computer screen (on the table) and a mobile phone.

The following sections describe the screen detection method (section 4.1) and how the gaze coordinates from the scene image is transformed to screen coordinates (section 4.2).

4.1 Screen detection

The scene image is the prime resource for obtaining information about the surroundings in head mounted eye trackers unless other position devices are available. The eye tracker should potentially be able to detect and discern multiple screens. There is a multitude of image-based methods that could be used to detect a screen in the scene image. The ideal method is able to detect the screen in different light conditions and when the screen is turned on or off and should simultaneously be sufficiently fast to allow for real-time processing.

Another challenge is to be able to discern screens with identical appearance and when these are viewed from different angles. Fixed visual markers could be placed on the screen to allow for easy identification e.g. a QR-Code around the screen. The visual tag is only needed for identification of the screen and is not needed during interaction. Hence, fixed visual tags are needless most of the time and could be disturbing for the user while they also clutter the scene. Besides, fixed visual tags are not suitable for use when employing a large number of screens since someone needs to be placing the tags where most appropriate.

Potential screen candidates are detected using quadrilateral contour information (illustrated in figure 3). Whenever a quadrilateral, Q, appears in the scene image the eye tracker

notifies the server to show identification tags. For initialization, the server issues a command to all the screens to show their identification tag (similar to a QRCode) for short period of time. The tag is shown until the eye tracker has identified the tag in the scene image. The tag possesses information about screen identity and may contain other screen and application dependent information. The screen is tracked over time after identification, but the identification procedure is reinitiated when other screens appear in the scene image. During re-initialization the server only issues commands to the currently inactive screens.
This approach allows a low degree of maintenance and offers an efficient way of identifying screens. Notice that this approach is sufficiently general and scalable to situations with multiple eye trackers and multiple screens located in individual networks (e.g. located over large distances). Several users may even share the same screen.

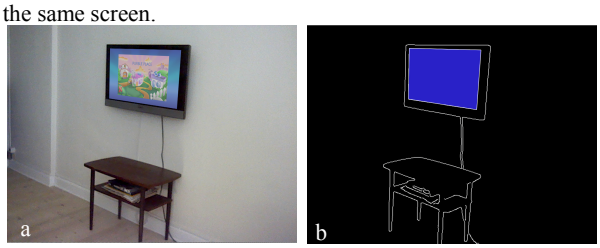


Figure 3. (a) Scene image with a screen (b) edge image and the detected screen

4.2 Mapping point of regard (PoR) to object space

The eye tracker provides only gaze estimates in the scene image, but what is needed is to be able to determine where on the screen the user is looking. This means that a mapping from the image coordinates, **s**, to the screen coordinates, **m**, are needed. In this paper we assume the objects used for interaction (the screens) are planar. Under these circumstance there is a homographic mapping, H_s^m from the screen in the scene image, to the screen coordinates [12]. H_s^m needs to be calculated in each frame since the position of the screen is not fixed in the scene image. The homography from the screen corners S_i to M_i (figure 4) is estimated in each time instance. Information about the screen dimensions are obtained from the visual tag during screen identification. The gaze point in the scene image is then mapped to the screen coordinates through $\mathbf{m} = H_s^m \cdot \mathbf{s}$. Figure 4 shows the mapping of the PoR (center of the red cross-hair) from the scene image to the screen plane and the real coordinates of the PoR in the screen by a black cross-hair (left image).

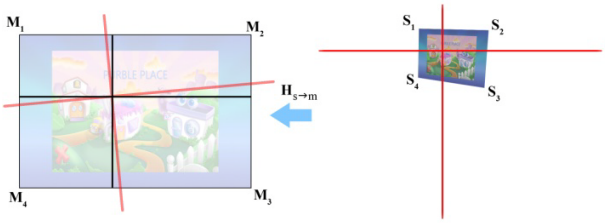


Figure 4. Mapping from the scene plane (right) to the real screen plane (left)

Eye trackers do not have pixel precision. Each gaze measurement in the scene image is therefore associated with an error. A convenient property of this approach is that the assumed precision and point of regard can be mapped to the screen image by mapping the uncertainty ellipse from the scene image to the screen image [12]. Figure 5 illustrates this process.

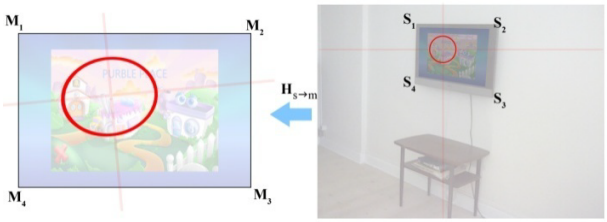


Figure 5. (left) The point of regard (cross hair) and the estimated uncertainty (ellipse) in the screen. (Right) The screen as viewed from the scene camera, the estimated point of regard (cross hair) and the assumed eye tracker precision.

5. EXPERIMENTAL APPLICATION

The experimental setup is intended for a home environment where the users are be able to communicate and interact with screens and control objects (e.g, fan, door, window and radio) via the screens.
Three screens are located in a house, each with a TCP/IP connection to the server. Two screens (S1 and S2) are 55" LG flat panel TVs. The third screen, S3, is a 4" Sony Ericsson Xperia X10 screen. Three different markers are used for identifying the screens. The applications are running on the screens, only allow single-user inputs and the experiments are therefore conducted with single user at the time. Each screen application is made to illustrate different applications of head mounted eye tracking for domotics [4] scenarios, namely controlling devices, the computer and small mobile devices.

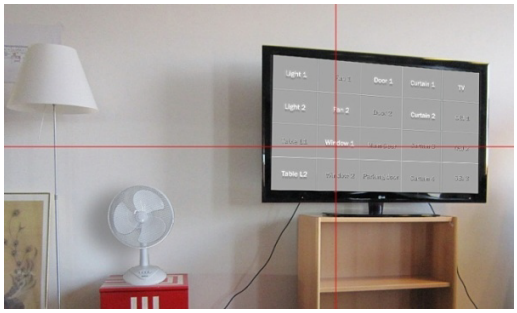


Figure 6. the user is interacting with S1

S1 is running an application which allows 20 devices to be turned on or off in the home environment. The user can control the devices by double blinking while gazing on the on-screen buttons. Each button spans a 17x24 cm rectangle on the screen and the user can see the status of each object by the changing of the color (figure 6). S2 is connected to a computer via RS232 port to allow the computer to communication with the functionalities of the TV. The user can change the channels up and down by double blinking on the left hand side corners and similarly for the volume (right hand side corners). Each of the corners regions is (20x20cm) on the screen. S3 is mobile phone screen and a java application is running on it that has 4 on screen-buttons and wirelessly connects to the server. Each button can be used to control a subset of the objects of S1 through double blinking. The eye tracker is feature-based using homographic mappings from the eye image to the scene image, thus requiring a 4point calibration procedure. The eye tracker runs at 15 fps on 640x480 images with an accuracy of about 1 degree of visual angle for the calibration distance. Screen

detection is done using quadrilaterals in the scene image based on the contour-based features.

Head mounted eye trackers are usually prone to errors when objects in the scene image are on different depths than calibration distance (due to the parallax of the scene camera and eye). When the calibration was performed at 1.5 meters, the eye tracker had an accuracy about 1.5° in the scene image when the user was at 4 meters from the screen, and about 3° when the user was at 40 cm. The inaccuracy of the eye tracker consequently propagates to the screen and is therefore dependent on the distance and angle between the user and screen (figure 5). However the accuracy on the screens was sufficient for interaction with mobile device and large screens (5 x 4 grid on the screen).

6. CONCLUSION

The background for this work was how interaction with multiple screens can be done with a head mounted eye tracker, We have presented a general framework that allows screens to be detected efficiently and identified without cluttering the scene or disturb the user significantly. The method is easily extendible to multiple locations, with many screens and is still easy to maintain. The low-cost head-mounted eye tracker that does not support parallax error, limits the difference between working plane and the calibration plane, however using the accurate systems calculate the point of regard accurately as the screen is viewed from close or far distances.

A significant limitation of our system, however, is that the current method for screen detection and mapping of the gaze point cannot be used when the screen is not completely inside the scene image (e.g. viewing the big screens from close distance). However with more advanced techniques this would be possible.

The method has been tested on 3 different applications intended for domotics using a low cost wireless head-mounted eye tracker. The users were able to interact with a TV and a computer screen located in different places in the home environment and with a mobile phone.

Through this work we have demonstrated that head mounted eye trackers can be used for interaction with the screens in 3D spaces.

7. REFERENCES

[1] Agustin, J.S., Hansen, J.P., Tall, M. 2010. Gaze-Based Interaction with Public Displays Using Off-the-Shelf Components. In Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UBICOMP2010), Copenhagen, Denmark. ACM, New York, pp. 377-378.

[2] Bolt, R.A. 1982. Eyes at the Interface. In Proc. of Human Factors in Computer Systems Conference, 360-362.

[3] Bonino, D.; Castellina, E., Corno, F. &Garbo, A. 2006. Control Application for Smart Housethrough Gaze interaction," Proceedings of the 2nd COGAIN Annual Conference onCommunication by Gaze Interaction, Turin, Italy.

[4] Bonino, D., Castellina, E., & Corno, F. 2008. The DOG gateway: enabling ontologybasedintelligent domotic environments. Consumer Electronics, IEEE Transactions on, 54 (4), 1656-1664.

[5] Castellina, E., Razzak, F., Corno, F. 2009. “Environmental Control Application. Compliant with Cogain Guidelines,” The 5h Conference on Communication by gaze interaction (COGAIN 2009).

[6] Duchowski, A.T. 2007. Eye Tracking Methodology: Theory and Practice. Springer, London. (2th edn)

[7] Duchowski, A. T., Cournia, N., and Murphy, H. 2004. Gaze-contingent displays: A Review. CyberPsychology and Behaviour, 7(6), 621‑634.

[8] Duchowski, A.T. 2002. A breadth-first survey of eye tracking applications. In Behavior Research Methods, Instruments, & Computers (BRMIC), 34(4), 455-470.

[9] Eaddy, M., Blasko, G., Babcock, J., and Feiner, S. 2004. My own private kiosk: Privacy-preserving public displays. In ISWC ’04: Proceedings of the Eighth International Symposium on Wearable Computers, 132‑135, Washington, DC, USA, IEEE Computer Society.

[10] Gale A.G., 2005. Attention Responsive Technology and Ergonomics. In Bust P.D. & McCabe P.T. (Eds.) Contemporary Ergonomics 2005, London, Taylor and Francis, 273-276.

[11] Hansen, D. W. and Ji, Q. 2010. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. IEEE Trans. Pattern Anal. Mach. Intell, 478-500.

[12] Hartley, R., and Zisserman, A. 2000. Multiple view geometry in computer vision. Cambridge University Press, Cambridge, UK.

[13] Hutchinson, T. E., White, K. P., Martin, W. N., Reichert, K. C., and Frey, L. A. 1989. Human‑computer interaction using eye-gaze input. Systems, Man and Cybernetics, IEEE Transactions on, 19(6),1527‑1534.

[14] Hyrskykari, A., Majaranta, P., and Raiha, K. J. 2005. From gaze control to attentive interfaces. In Proceedings of the 11th International Conference on Human–Computer Interaction (HCII 2005). IOS Press.

[15] Kato, H., and Billinghurst, M. 1999. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In Proc.IEEE International Workshop on Augmented Reality, 125–133.

[16] Koike,H., Nishikawa, W., Fukuchi, K. 2009. Transparent 2-D Markers on an LCD Tabletop System, ACM Human Factors in Computing Systems (CHI 2009), 163-172.

[17] Palmer, R.C. 2001. The Barcode Book, 4th edition, Helmers Pub.

[18] Park, H., and Park, J. 2004. Invisible marker tracking for AR. Proc. 3rd IEEE/ACM Int. Symp. on Mixed and AugmentedReality, 272–273.

[19] Shell, J. S., Vertegaal, R., and Skaburskis, A.W. 2003. Eyepliances: attention-seeking devices that respond to visual attention. In CHI ’03: Extended abstracts on Human factors in computing systems, pages 770‑771, New York, NY, USA. ACM Press.

[20] Shi, F., Gale, A.G. & Purdy, K.J. 2006. Eye-centric ICT control. In Bust P.D. & McCabe P.T. (Eds.) Contemporary Ergonomics 2006, 215-218.

[21] Ware, C. and Mikaelian, H.T. 1987. An evaluation of an eye tracker as a device for computer input. In Proc. of the ACM CHI + GI-87 Human Factors in Computing Systems Conference, 183-188.

- Chapter 3 -

Interacting with Objects in the Environment by Gaze and Hand Gestures

Interacting with Objects in the Environment by Gaze and Hand Gestures

Jeremy Hales
ICT Centre - CSIRO

David Rozado
ICT Centre - CSIRO

Diako Mardanbegi
ITU Copenhagen

A head-mounted wireless gaze tracker in the form of gaze tracking glasses is used here for continuous and mobile monitoring of a subject's point of regard on the surrounding environment. We combine gaze tracking and hand gesture recognition to allow a subject to interact with objects in the environment by gazing at them, and controlling the object using hand gesture commands. The gaze tracking glasses was made from low-cost hardware consisting of a safety glasses' frame and wireless eye tracking and scene cameras. An open source gaze estimation algorithm is used for eye tracking and user's gaze estimation. A visual markers recognition library is used to identify objects in the environment through the scene camera. A hand gesture classification algorithm is used to recognize hand-based control commands. When combining all these elements the emerging system permits a subject to move freely in an environment, select the object he wants to interact with using gaze (identification) and transmit a command to it by performing a hand gesture (control). The system identifies the target for interaction by using visual markers. This innovative HCI paradigm opens up new forms of interaction with objects in smart environments.

Keywords: Eye Tracking, Gaze Tracking, Head-Mounted Gaze Tracker, Eye Tracking Glasses, Mobile Interaction, Hand Gestures, Gaze Interaction, HCI, Gaze Aware Systems, Gaze Responsive Interface, Mobile Interaction

Introduction

Body language and gaze are important forms of communication among humans. In this work, we present a system that combines gaze pointing and hand gestures to interact with objects in the environment. Our system merges a video-based gaze tracker, a hand gesture classifier and a visual marker recognition module into an innovate HCI device that permits novel forms of interaction with electronic devices in the environment. Gaze is used as a pointing mechanism to select the object which the subject wants to interact with. A visual binary marker attached to the object is used for identification of the object by the system. Finally, a hand gesture is mapped to a specific control command that makes the object being gazed at to carry out a particular function.

Using gaze for interaction with computers was initiated in the early 1980s (Bolt, 1982) and further developed by (Ware & Mikaelian, 1987). Today, gaze interaction is mostly done using a remote eye tracker with a single user sitting in front of a computer display. However, head-mounted gaze trackers (HMGT) allow for a higher degree of mobility and flexibility, where the eye tracker is mounted on the user and thus allows gaze to be estimated when e.g. walking and driving. HMGT systems are commonly used for estimating the gaze point of the user in his field of view. However, the point of regard (PoR) obtained by head-mounted gaze trackers can be used for interaction with many different types of objects present in the environments during our daily activities. There has been some previous work done on using gaze for interaction with computers in mobile scenarios using head-mounted gaze trackers (Mardanbegi & Hansen, 2011). Despite the fact that gaze can be used as a mechanism for pointing in many interactive applications, eye information has been shown to be limited for interaction purposes. The PoR can be used for pointing, but not for yielding any additional commands. The main reason is that it is unnatural to overload a perceptual channel such as vision with a motor control task (Zhai, Morimoto, & Ihde, 1999). Therefore, other interaction modalities such as body gestures and speech together with gaze can be used for enhancing gaze-based interaction with computers and also with electronic objects in the envi-

This paper has been possible thanks to the CSIRO ICT Centre Undergraduate Vacation Scholarships Program. Corresponding author: jeremy.hales1@gmail.com

ronment. In this paper, we use hand gestures to circumvent the limitations of gaze to convey control commands. The combination of gaze and hand gestures enhances the interaction possibilities in a fully mobile scenario.

Automatic gesture recognition is a topic in computer science and language technology that strives to interpret human gestures via computational algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or the hands. An appealing feature of gestural interfaces is that they make it possible for users to communicate with objects without the need for external control devices. Hand gestures are an obvious choice as a mechanism to interact with objects in the environment. Automated hand gesture recognition is challenging since in order for such an approach to represent a serious alternative to conventional input devices, applications based on computer vision should be able to work successfully under uncontrolled light conditions, backgrounds and perspectives. In addition, deformable and articulated objects like hands represent added difficulty both for segmentation and shape recognition purposes. This paper does not intent to contribute significantly in the topic of hand gesture recognition methodology, but rather to suggest the combination of gaze and hand gestures as an alternative to the conventional methods that are used for gaze interaction such as: blinking (e.g., (MacKenzie & Zhang, 2008)), dwelling (e.g., (Jacob, 1991)), and gaze gestures (e.g., (Isokoski, 2000)). We use the scene image of the HMGT system for recognizing the hand gestures and for recognizing the visual markers attached to the gazed objects. The hand gesture recognition module we developed here is able to detect a hand in front of the scene camera of the HMGT and the number of fingers that the hand is holding up as well as its relative movements in 4 spatial directions.

In summary, this work represents a proof of concept for an innovative form of interacting with objects in the environment by combining gaze and hand gestures. Interaction is achieved by gazing at an object in the environment and carrying out a hand gesture. The hand gesture specifies a certain command and gazing at the object, and the visual marker associated to it, make only that specific object to respond to the subsequent hand gesture. The low cost off-the-shelf components used to build the hardware, and the open source nature of the algorithms used for gaze estimation and object recognition, make this form of interaction amenable for spreading among academic institutions and research labs to further investigate and stretch the possibilities of this innovative HCI paradigm.

The remaining of the paper is structured as follows. The *Related Work* section provides an overview of the literature on the topic of gaze and mobile interaction. The *System Overview* section delineates the main components of the system and their mutual interactions. The *Implementation* Section goes into a detailed description of each of the system's components. The *Application Example* Section describes a particular instantiation of our system to control 3 objects in an environment: an Arduino board, a computer and a robot. Finally, the *Discussion and Conclusion* Section elaborates in some of the issues we have found when trying out the proposed gaze and hand gestures based interaction as well as pointing out possible future research venues to continue exploring the innovative interaction modality proposed here.

Related Work

There has been substantial research in hand/body gestures used for human-computer interaction. There are many vision-based methods that by using video cameras as the input device, can detect, track and recognize hand gestures with various image features and hand models (Mitra & Acharya, 2007). Most of these approaches detect and segment the hand in the image using the skin color information (Argyros & Lourakis, 2004). In this paper we have used a color based hand gesture recognition method that is efficient and easy to implement. Hand gestures can be used as a mode of HCI that can simply enhance the human-computer interaction by making it more natural and intuitive. Some of the application domains where gestural interfaces have been comonly used is in virtual environments (VEs) ((Adam, 1993; Krueger, 1991)), augmented reality (Buchmann, Violich, Billinghurst, & Cockburn, 2004) and automatic sign language recognition (Rozado, Rodriguez, & Varona, 2012a, 2010) in which hand gestures are comonly used for manipulating the virtual objects (VOs) for interaction with the display or for recognition of sign language. The vision based hand gesture recognition devices can be worn by the user, providing the user with more flexibility and mobility for interaction with the environment (Starner, Auxier, Ashbrook, & Gandy, 2000; Amento, Hill, & Terveen, 2002).

More recently several authors have also investigated using gaze itself to generate gestures for control and interaction purposes (Istance, Hyrskykari, Immonen, Mansikkamaa, & Vickers, 2010; Rozado, Rodriguez, & Varona, 2012b; De Luca, Weiss, & Drewes, 2007; Rozado, Rodriguez, & Varona, 2011; Mollenbach, Lillholm, Gail, & Hansen, 2010; Drewes & Schmidt, 2007). While useful in many regards, by being very fast to perform and robust under low gaze estimation accuracy, gaze gestures also possess shortfalls in terms of risking to overload the visual channel which is intuitively perceived by users as just an input channel.

There is also a body of literature focused around gestures for multimodal interactions (Starner et al., 2000; Schapira & Sharma, 2001; Nickel & Stiefelhagen, 2003; Rozado, Agustin, Rodriguez, & Varona, 2012). For example, hand gestures in combination with speech provide a multimodal interactions mechanism that allows

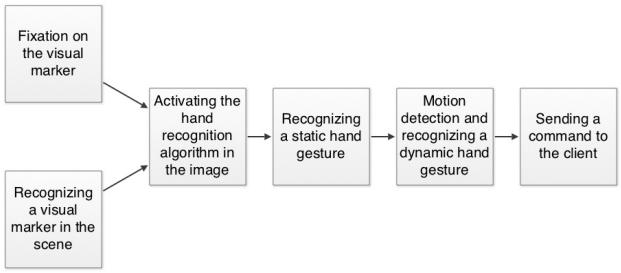


Figure 1. **Overview of the interaction modality proposed in this work.** The diagram describes the main components and actions involved in interacting with objects through gaze and hand gestures.

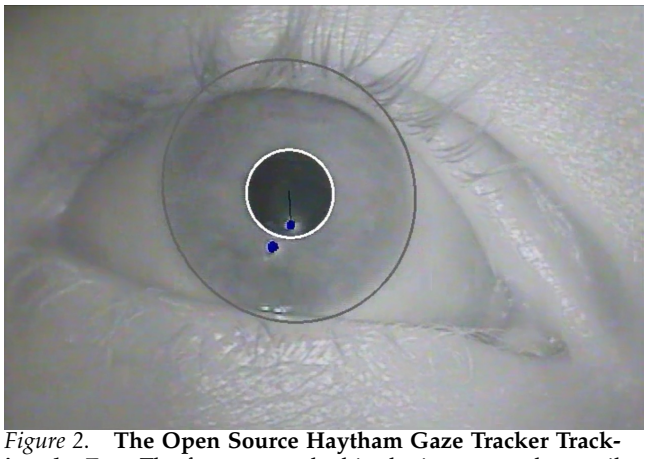


Figure 2. **The Open Source Haytham Gaze Tracker Tracking the Eye.** The features tracked in the image are the pupil center and two corneal reflections. These features are used by the gaze estimation algorithms to determine the PoR of the user on the scene camera.

the user to have an eyes-free interaction with the environment. Body gestures can also be combined with gaze in situations where the gazed context is the interaction object (e.g., looking at a lamp and turning the lamp on). In such cases, gaze acts as a complementary interaction modality and it is used for pointing. (Mardanbegi, Hansen, & Pederson, 2012) used head gestures together with gaze for controlling objects in the environment by gazing at the objects and then performing a head gesture. Authors used a mobile gaze tracker for gaze estimation and an eye-based method for measuring the relative head movements. They used the scene image for recognizing the objects and to ensure that the PoR is on the object during the gesture. In contrast, in this paper, we use gaze for pointing and hand gestures to execute a particular command using the scene camera of a head-mounted eye tracker for measuring the hand gestures, see Figure 1.

System Overview

In this section, different steps of the interaction process are introduced and the main elements of the system are described. In our system, a head-mounted gaze tracker estimates the gaze point in the user's field of view using an eye tracking camera and an scene camera. A simple method for recognizing the objects in the environment is used by detecting visual markers associated to them through the scene camera. When the subject carrying the gaze tracker looks at an object, the visual marker placed on the object is recognized by the system. When a visual marker has been detected, the hand gesture recognition algorithm will be activated in the scene image (for a short period of time) to detect the potential hand gesture that might be generated shortly after. A control command, associated to a specific hand gesture, will be send to the object if the gesture is detected. In this way, only that particular object in the environment gazed at will react to the hand gesture, while the rest of the objects in the environment susceptible to be controlled by gaze remain unresponsive.

The main hardware components of the system are introduced below:

a) A wireless mobile gaze tracker glasses with two cameras: one for tracking one eye and the other to capture the field of view of the subject.

b) Video receiver that is connected to a remote PC and receives the video streams of both the eye and the scene camera.

c) Visual markers attached to the target objects of interaction.

d) Interaction objects (e.g, robot, lamp, computer display).

The processing units of the system can be conceptually divided into two groups: the server and the clients, see Figure 3. The server processes the eye and the scene images. Eye tracking, gaze estimation, and recognizing the visual markers and the hand gestures are done in the server application running on a remote PC. The output of the application will be sent to the client application controling a specific object using the TCP/IP protocol. The client applications facilitates the connection between the server and the objects in the environment and undergoes the local processing needed for controlling the objects.

Gaze Tracking Depending on the hardware configuration of the different components, gaze tracking systems can be classified as either *remote* or *head-mounted*. In remote systems, the camera and the light sources are detached from the user and normally located around the device's screen, whereas in head-mounted systems the components are attached to the user's head. Head-mounted eye trackers can be used for mobile gaze estimation as well as gaze interaction purposes. The head-mounted gaze trackers have two cameras: one for recording the eye image and one for recording the

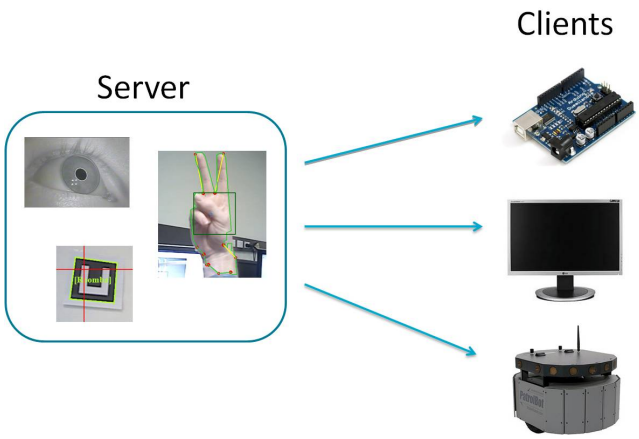


Figure 3. **System Diagram.** Several smart objects *clients* connect to a centralized servers that handles the gaze tracking and estimation, the visual marker recognition and the hand gesture recognition. The server dispatches the appropriate commands to a given client when a combination of gaze fixation on the object visual marker and hand gesture is detected.

scene image. In this work, we have used a head-mounted gaze tracker for gaze estimation on top of which, we have build a hand gesture recognition module. The point of regard and the coordinates of the gaze point in the scene image are measured by the system.

Object recognition Visual markers provide a simple solution for recognizing the objects in the scene allowing us to concentrate on illustrating the potential of the proposed interaction method. Visual marker recognition systems consist of a set of patterns that can be detected by a computer equipped with a camera and an appropriate detection algorithm (Middel, Scheler, & Hagen, n.d.). Markers placed in the environment provide easily detectable visual cues that can be associated to specific objects for identification purposes. Once a visual marker is recognized in the vicinity of the user's gaze, the hand gesture recognition algorithm will be activated.

Hand Gesture A skin color-based method is used for detecting the hand in the scene image. The hand gesture recognition worked well for natural skin color, but using a latex glove of a color not present in the environment improves the performance. Hand gestures are defined as holding the hand with a preset number of fingers for a predefined dwell time of 1 second (a static hand gesture) and moving it in a particular direction (a dynamic hand gesture): up, down, left or right. Therefore, the hand recognition part consists of two steps: detecting a static shape of the hand and then a dynamic hand gesture that ends by taking the hand outside the image.

Figure 4. **Low Cost Gaze Tracking Glasses.** The wireless camera on the top left of the figure is what we refer to in this work as the scene camera. The scene camera approximately captures the field of view of the user. The camera on the bottom left of the figure is the gaze tracking camera that monitors the user's gaze movements. The Haytham software uses the video stream provided by that eye camera to calculate the PoR of the user and superimposes the gaze estimation coordinates over the video stream generated by the scene camera. The top right of the figure shows the battery that is used to provide energy to the wireless cameras.

The gesture alphabet can be named using a combination of the number of fingers held up, x, and one of the four spatial directions that the hand is supposed to move to generate the gesture, D, in a pattern such as xD. For example 4Up, refers to a gesture consisting of the hand holding four fingers up and an upwards movement.

Implementation

The presented method has been implemented in a real scenario for controlling a remote robot, an Arduino, and a computer display. In this section, implementation and the hardware/software components of the system are introduced briefly.

Gaze Tracking System

We have build a low-cost head-mounted gaze tracker using off-the-shelf components (Figure 4 and Figure 5). The system consists of safety glasses, batteries, and the wireless eye/scene cameras. The wireless eye camera is equipped with infrared emitting diodes that permit the gaze tracking software to monitor the position of the pupil and the glint in the image. These

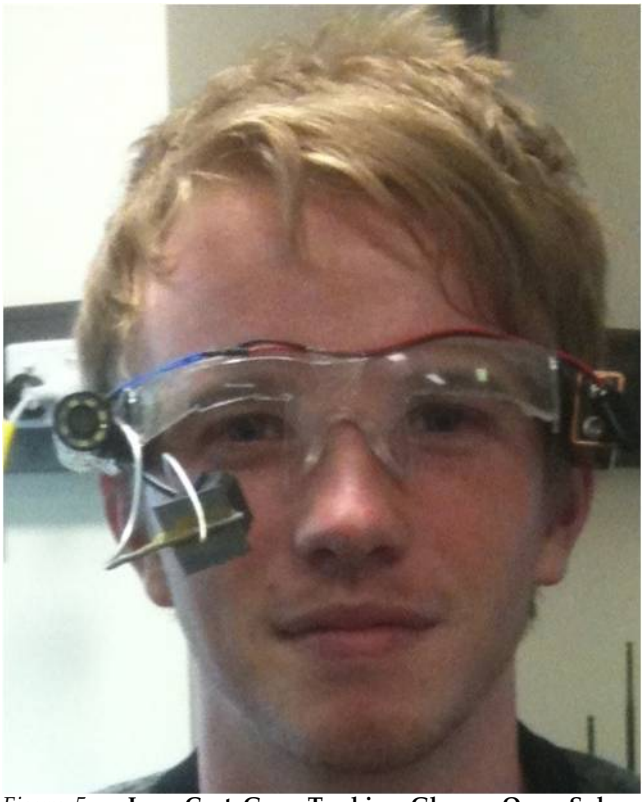


Figure 5. **Low Cost Gaze Tracking Glasses On a Subject.** This figure shows how the low-cost head-mounted gaze tracking system looks while being used by a subject.

features are used by the gaze estimation algorithm to estimate the PoR. Infrared light improves image contrast and produces a reflection on the cornea, known as corneal reflection or glint. A calibration procedure needs to be done to build a user specific model of the eye. The calibration procedure consists on the user looking at a number of points on the environment and marking them on the scene image while the user fixates on them. Once the calibration procedure is completed, the gaze estimation algorithm is able to determine the point of regard of the user in the environment. Figure 2 shows a screenshot of an eye being tracked by the open source gaze tracker (Mardanbegi et al., 2012) used in this work. In the figure, the center of the pupil and two corneal reflections are the features being tracked.

Making the head-mounted eye tracker glasses.

Figure 4 shows a prototype of the eye tracking glasses built for this work. An area was traced onto the lens of a pair of safety glasses where the eyes will be approximately located when the user puts on the glasses. Tin snips were used to cut away the plastic parts of the lenses bounded by the previously traced areas. It is important that the majority of the lenses of the glasses is left intact to preserve the structural integrity of the frame. Tin was cut to the size and shape of the infrared camera using the tin snips. Steel wire was used to attach the camera to the frame of the glasses. The wire was cut to a size of 25cm and attached to the piece of tin using araldite. Double sided tape was used to secure the tin to the back of the camera. The wire was bent into an 'L' shape and firmly attached to the right hand side of the glasses (frame) using tape. The infrared camera runs on a 9V battery that also needed to be mounted to the glasses. The connecting wires from the battery to the camera were extended and the battery was attached to the left hand side of the glasses. This distributes the weight of the components over the frame. Utilising the Haytham software, the position of the camera was checked to ensure the camera was capturing the entire eye. It was found that the best position of the eye camera is below the glasses so it doesn't obstruct the user's vision. The scene camera was firmly mounted to the right side of the glasses using tape as close as possible to the eye in order to minimize the parallax error, see Figure 5.

Gaze tracking software.

We used the Haytham[1] open source gaze tracker to monitor user's gaze. The Haytham gaze tracker provides real-time gaze estimation in the scene image as well as visual marker recognition in the scene camera video stream. Figure 6 shows a recognized marker from the scene video stream and the gaze point measured by the gaze tracker represented as a cross hair.

Implementing hand gestures recognition

Static hand gesture recognition algorithm.

An open source hand gesture recognition software[2] developed by Luca Del Tongo was modified for use in detecting the number of fingers raised by the hand. There are two options for analysing the images captured by the scene camera: colour or skin detection. To detect the skin of the hand the image was transformed to the Ycc colour space; upper and lower bounds were set for the Cr and Cb channels. To detect a coloured latex glove, the image was transformed to the HSV colour space; upper and lower bounds were set for the hue and saturation channels. Pixels that satisfied the bounding conditions are identified as potential sections of the hand. Two measures were implemented to reduce false detection caused by noise or objects with similar colours to skin or the coloured gloves. The blob with the largest contour area is designated as the hand and all blobs that are lower than a set area are removed from the image, including the blob that has been designated as the hand. This removes the possibility that small blobs (noise) are identified as the hand of the user. The convex hull is extracted from the hand and

[1] http://itu.dk/research/eye/

[2] http://blogs.ugidotnet.org/wetblog/Default.aspx

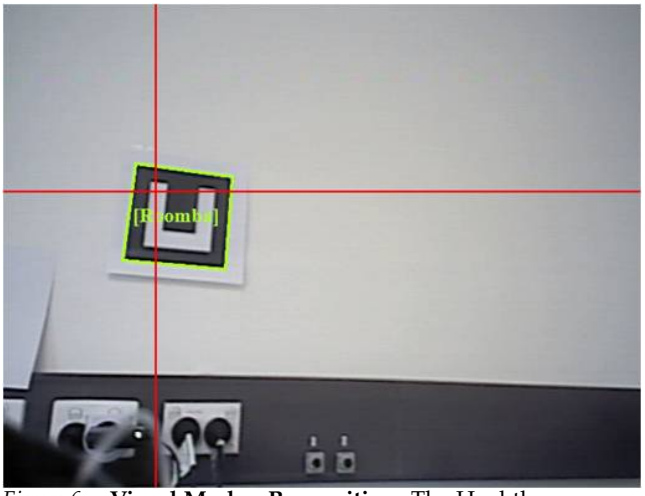


Figure 6. **Visual Marker Recognition.** The Hayhtham gaze tracker uses the Aforge glyph processing library (GRATF) for visual marker recognition in the scene image. This figure shows the identified marker and the user's gaze point (cross hair) in the scene image. When a subject position its gaze on a visual marker that identifies an object, the system interprets this as a pointing action and sends the subsequent recognized hand gestures to the specific object represented by the visual marker.

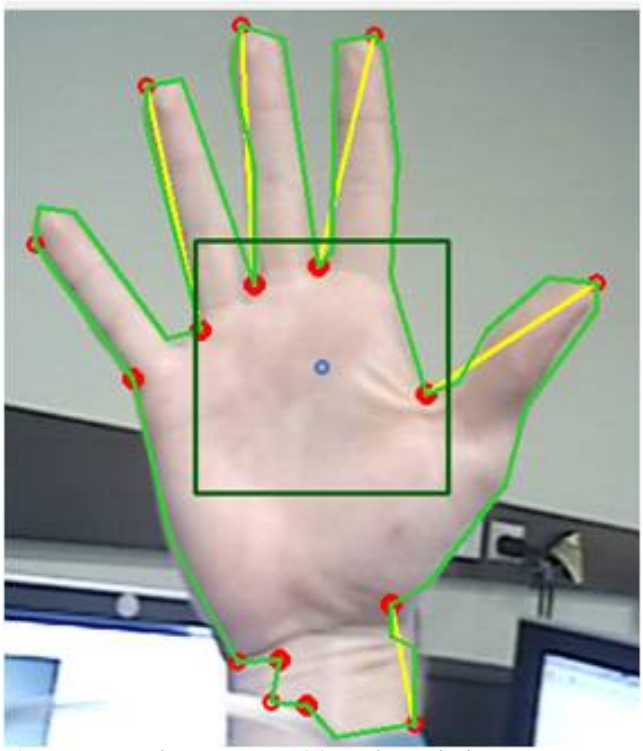


Figure 7. **Hand Pose Recognition Through the Scene Camera.** The figure shows a hand with five fingers held up as recognized through the scene camera by the hand pose recognition routine. The light green line outlines the convex hull of the hand and the dark green box represents the boundary for a classified movement.

the convexity defects are determined (Figure 7). Two parameters of the defects were used; the start and the end points are the points on the hull that mark where the defect starts and end. Three conditions were defined to determine whether a defect is a raised finger. They are: the start point of a defect must be higher than the end point, either the start or end points must be higher than the centre of the hand and the magnitude of the start and end points must be greater than the scaled down length of the hand. Each defect is checked and the total number of fingers identified is the sum of defects that satisfy the aforementioned conditions.

Dynamic hand gesture recognition algorithm.
The centroid of the hand contour is determined and an initial boundary box of size 20x20 pixels set. If the centroid doesn't move outside of the boundary box for 1.5 seconds, the current position of the hand is identified as the reference point and a new boundary box of size 60x60 pixels is set. If the centroid of the hand moves outside of the box it is classified as a movement. The location of the centroid when it moves outside the box designated the direction of movement: above the box is up, below the box is down, left of the box is left and right of the box is right. The program samples and averages the number of fingers shown. This helps to eliminate false identification of the number of fingers due to noise. When a movement is identified, the average number of fingers is sent to the client with the direction of movement.

Clients

A client program was developed to communicate with the devices in the environment. The program connects to the server (Haytham) using the TCP/IP protocol. Haytham sends commands to the client detailing specifics such as: the marker that has been recognised, the number of fingers raised and the direction of movement of the hand.

The proposed method is used for controlling a patrol robot, controlling an Arduino, and for interaction with a computer display (Figure 3) as described below.

The patrol robot connects to the computer using an Ethernet cable. The number of fingers determines the magnitude of movement and the hand movement controls the direction of movement (e.g. 2Up will move the robot forward with a magnitude of 2 and 3Left will rotate the robot counter-clockwise). An Arduino is connected to the client program via serial connection and is used to control 3 leds on a breadboard. The interaction with the computer display is done by minimizing or maximizing the windows in the display through use of the sendMessage function.

Application Example

We carried out a small pilot study to test the functionality and performance of the system. We decided to

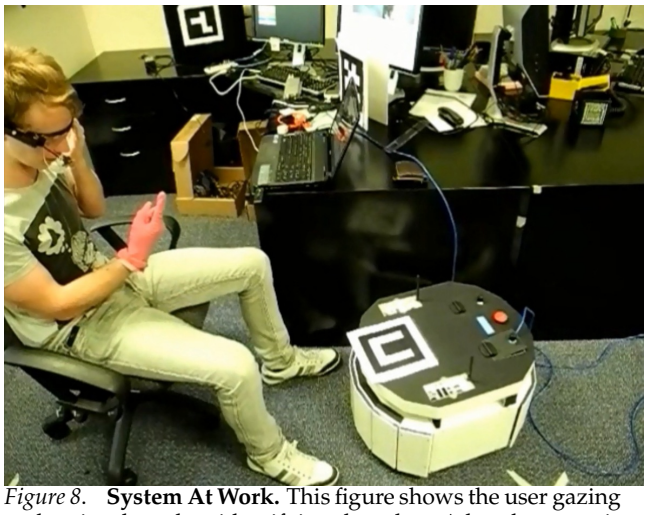


Figure 8. **System At Work.** This figure shows the user gazing at the visual marker, identifying the robot. A hand gesture is performed to transmit a movement command to the robot.

test the system in a environment where 3 "smart" objects could be controlled by the system simultaneously: a computer, a set of leds in a breadboard and a robot. The hand gesture recognition module could recognize 5 different states of the hands as defined by the number of fingers being held up: 1, 2, 3, 4 and 5. A gesture was defined as one of these 5 states plus one of four spatial directions: up, down, left and right.

The breadboard responded to users commands just by turning the infrared leds on and off. Two fingers being held up and an upward movement would turn the leds on. Four fingers being held up and a movement to the right would turn them off.

The same hand gestures were used to control the computer. The upward movement of the hand with two fingers being held up was mapped to a command in the operating system that minimizes all the current open windows on display in the computer GUI. Four fingers being hold up and a movement to the right gesture was mapped to a command that brings all the minimized windows back up. This particular set of gestures and control commands were not selected specifically for any particular reason other than as a proof of concept. Any other type of gestures associated to different control commands could be envisioned and implemented.

The robotic control example was the most elaborated one. The robot could be made to move forward or backward and to turn right or left. The numbers of fingers being held up with the hand indicated, either the speed for forward and backward movements or the amount of turn to be made for right and left movements.

The hand gestures could be done with bare hands, but we noticed that in environments where the color of the walls could resemble the skin hue, hand gesture recognition performance would suffer. Using a glove with a distinctive color, not present in the rest of the environment, enhanced hand recognition performance.

This manuscript's associated video[3] provides a good visual overview of the system at work and how it is being used by two different users to interact with a computer, a breadboard with a set of light emitting diodes and with a robot.

Discussion and Conclusion

In this work we have shown how to interact with objects in the environment through an innovative combination of gaze and hand gestures using a set of gaze tracking glasses and a hand gesture recognition module. The method is easily extensible to multiple objects in the environment and to a wide array of hand gestures.

The low-cost head-mounted eye tracker used and the gaze estimation algorithms employed do not compensate for parallax error, i.e. the inability to differentiate between the working plane and the calibration plane (Mardanbegi & Hansen, 2012). This limits the ability to alternate interaction with objects at a distance and objects up close. Nonetheless, since the scene camera used in the glasses is relatively close to the eye being tracked, see Figure 4, the parallax error was minimized. Furthermore, we noticed that during the calibration, using calibration points situated at different distances (from 1 to 10 meters) would achieve a compromise between objects far away and objects up close and would generate good gaze estimation for all sort of distances. We noticed that gaze estimation accuracy was never an issue for our system. Only over time, if the glasses would move slightly from their position during calibration, due to sweat on the skin or drastic head movements that would cause the glasses to slide slightly, would gaze estimation degrade marginally.

We did notice problems with the skin detection algorithms when the hand was position within the field of view of the scene camera. This was markedly noticeable, when the colors of the background were similar to the skin color. Usage of more sophisticated skin detection algorithms could help to solve this issue.

An important issue of the system was the fact that the user wearing the glasses did not have any sort of feedback signal in terms of where within the field of view of the scene camera the hand was placed when it was about to initiate a hand gesture. This was due to the lack of a display on the glasses to provided visual feedback in terms of how the hand is positioned within the field of view of the scene camera. We implemented an auditory feedback signal to indicate that the system had found the hand holding a number of fingers up within the field of view of the scene camera and it was therefore ready to receive a gesture. We

[3] http://youtu.be/SGqF1Mi6JGI

found that this helped the user but still did not provide real time feedback to carry out small corrections of hand positioning for proper positioning within the field of view of the scene camera. This issue was due to the usage of a scene camera with a relatively narrow field of view. Using a scene camera with a wider field of view should prevent the need of feedback for hand positioning with high granularity precision since the hand would always fall within the field of view of the scene camera as long as the arm was stretched in front of the user.

Further work should strive to carry out an extensive quantitative analysis of the performance of the system within a large user study and in comparison to alternative modalities of gestures based interaction with objects in the environment through gaze alone, gaze and voice, and gaze and head gestures.

More sophisticated hand gestures that the ones described here can also be envisioned. However, complex gaze gestures generate a cognitive and physiological load on the user. Cognitively it is difficult for users to remember a large set of complex gestures, and physiologically it is tiring and challenging to complete them. Finding the right trade-off between simple and complex hand gestures is therefore paramount to successfully use hand gestures as a control input device.

More reliable hand tracking technologies that use depth sensor such as infrared laser projections to be combined with monochrome CMOS sensor, able to capture video data in 3D under any ambient light conditions, would greatly enhance the robustness of the hand recognition algorithms, making our system as a whole more reliable.

The preliminary results obtained in this pilot work shows promise for this form of interaction with objects in the environment. The combination of gaze and hand gestures to select an object and emit a control command are both natural to potential users and fast to carry out liberating users of the need to carry control devices in their hands. The richness of hand gestures potentially available suggests that this form of interaction can be used for sophisticated and complex environments requiring a large set of control commands while allowing the user to remain mobile in the environment.

References

Adam, J. A. (1993). Virtual reality is for real. *Spectrum, IEEE*, *30*(10), 22–29.

Amento, B., Hill, W., & Terveen, L. (2002). The sound of one hand: a wrist-mounted bio-acoustic fingertip gesture interface. In *Chi'02 extended abstracts on human factors in computing systems* (pp. 724–725).

Argyros, A. A., & Lourakis, M. I. (2004). Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *Computer vision-eccv 2004* (pp. 368–379). Springer.

Bolt, R. A. (1982). Eyes at the interface. In *Proceedings of the 1982 conference on human factors in computing systems* (pp. 360–362).

Buchmann, V., Violich, S., Billinghurst, M., & Cockburn, A. (2004). Fingartips: gesture based direct manipulation in augmented reality. In *Proceedings of the 2nd international conference on computer graphics and interactive techniques in australasia and south east asia* (pp. 212–221).

De Luca, A., Weiss, R., & Drewes, H. (2007). Evaluation of eye-gaze interaction methods for security enhanced PIN-entry. In *Proceedings of the 19th australasian conference on computer-human interaction: Entertaining user interfaces* (pp. 199–202). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/1324892.1324932` doi: http://doi.acm.org/10.1145/1324892.1324932

Drewes, H., & Schmidt, A. (2007). Interacting with the computer using gaze gestures. In *Proceedings of the 11th ifip tc 13 international conference on human-computer interaction - volume part ii* (pp. 475–488). Berlin, Heidelberg: Springer-Verlag. Retrieved from `http://portal.acm.org/citation.cfm?id=1778331.1778385`

Isokoski, P. (2000). Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 symposium on eye tracking research & applications* (pp. 15–21).

Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., & Vickers, S. (2010). Designing gaze gestures for gaming: an investigation of performance. In *Proceedings of the 2010 symposium on eye-tracking research & applications* (pp. 323–330). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/1743666.1743740` doi: http://doi.acm.org/10.1145/1743666.1743740

Jacob, R. J. K. (1991). The use of eye movements in human-computer interaction techniques: what you look at is what you get. {*ACM*} *Trans. Inf. Syst*, *9*(2), 152–169.

Krueger, M. W. (1991). *Artificial reality ii* (Vol. 10). Addison-Wesley Reading (Ma).

MacKenzie, I. S., & Zhang, X. (2008). Eye typing using word and letter prediction and a fixation algorithm. In *Proceedings of the 2008 symposium on eye tracking research & applications* (pp. 55–58).

Mardanbegi, D., & Hansen, D. W. (2011). Mobile gaze-based screen interaction in 3d environments. In *Proceedings of the 1st conference on novel gaze-controlled applications* (pp. 2:1–2:4). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/1983302.1983304` doi: 10.1145/1983302.1983304

Mardanbegi, D., & Hansen, D. W. (2012). Parallax error in the monocular head-mounted eye trackers. In *Proceedings of the 2012 acm conference on ubiquitous computing* (pp. 689–694).

Mardanbegi, D., Hansen, D. W., & Pederson, T. (2012). Eye-based head gestures. In *Proceedings of the symposium on eye tracking research and applications* (pp. 139–146).

Middel, A., Scheler, I., & Hagen, H. (n.d.). Detection and identification techniques for markers used in computer vision. In *Visualization of large and unstructured data sets-applications in geospatial planning, modeling and engineering* (Vol. 19, pp. 36–44).

Mitra, S., & Acharya, T. (2007, May). Gesture Recognition: A Survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *37*(3), 311–324. doi: 10.1109/TSMCC.2007.893280

Mollenbach, E., Lillholm, M., Gail, A., & Hansen, J. P. (2010). Single gaze gestures. In *Proceedings of the 2010 symposium on eye-tracking research & applications* (pp. 177–180).

Nickel, K., & Stiefelhagen, R. (2003). Pointing gesture recog-

nition based on 3d-tracking of face, hands and head orientation. In *Proceedings of the 5th international conference on multimodal interfaces* (pp. 140–146).

Rozado, D., Agustin, J. S., Rodriguez, F. B., & Varona, P. (2012, January). Gliding and saccadic gaze gesture recognition in real time. *ACM Transactions on Interactive Intelligent Systems*, *1*(2), 1–27. Retrieved from http://dl.acm.org/citation.cfm?id=2070719.2070723 doi: 10.1145/2070719.2070723

Rozado, D., Rodriguez, F. B., & Varona, P. (2010). Optimizing Hierarchical Temporal Memory for Multivariable Time Series. In K. Diamantaras, W. Duch, & L. Iliadis (Eds.), *Artificial neural networks - icann 2010* (Vol. 6353, pp. 506–518). Springer Berlin / Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-15822-3_62

Rozado, D., Rodriguez, F. B., & Varona, P. (2011). Gaze Gesture Recognition with Hierarchical Temporal Memory Networks. In J. Cabestany, I. Rojas, & G. Joya (Eds.), *Advances in computational intelligence* (Vol. 6691, pp. 1–8). Springer Berlin / Heidelberg.

Rozado, D., Rodriguez, F. B., & Varona, P. (2012a, March). Extending the bioinspired hierarchical temporal memory paradigm for sign language recognition. *Neurocomputing*, *79*(null), 75–86. Retrieved from http://dx.doi.org/10.1016/j.neucom.2011.10.005 doi: 10.1016/j.neucom.2011.10.005

Rozado, D., Rodriguez, F. B., & Varona, P. (2012b, August). Low cost remote gaze gesture recognition in real time. *Applied Soft Computing*, *12*(8), 2072–2084. Retrieved from http://dx.doi.org/10.1016/j.asoc.2012.02.023 doi: 10.1016/j.asoc.2012.02.023

Schapira, E., & Sharma, R. (2001). Experimental evaluation of vision and speech based multimodal interfaces. In *Proceedings of the 2001 workshop on perceptive user interfaces* (pp. 1–9).

Starner, T., Auxier, J., Ashbrook, D., & Gandy, M. (2000). The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *Wearable computers, the fourth international symposium on* (pp. 87–94).

Ware, C., & Mikaelian, H. H. (1987). An evaluation of an eye tracker as a device for computer input2. *ACM SIGCHI Bulletin*, *18*(4), 183–188.

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. In *Chi '99: Proceedings of the sigchi conference on human factors in computing systems* (pp. 246–253). New York, NY, USA: ACM. doi: http://doi.acm.org/10.1145/302979.303053

- Chapter 4 -

Gaze Based Controlling a Vehicle

Gaze-Based Controlling a Vehicle

Diako Mardanbegi
IT University of Copenhagen
Rued Langgaards Vej 7,
DK-2300 Copenhagen S
dima@itu.dk

Dan Witzner Hansen
IT University of Copenhagen
Rued Langgaards Vej 7,
DK-2300 Copenhagen S
Witzner@itu.dk

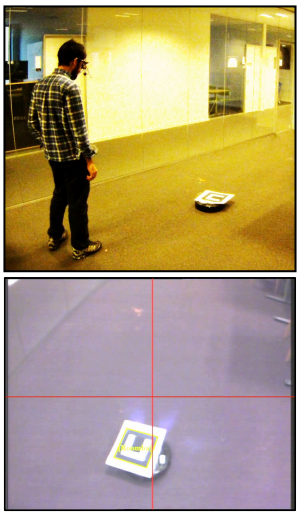


Figure 1: Controlling a Roomba vacuum cleaner by gaze in a mobile situation. The user's PoR is shown (cross hair) in the scene image of the mobile gaze tracker in the bottom image.

Abstract
Research and applications of gaze interaction has mainly been conducted on a 2 dimensional surface (usually screens) for controlling a computer or controlling the movements of a robot. Emerging wearable and mobile technologies, such as google glasses may shift how gaze is used as an interactive modality if gaze trackers are embedded into the head-mounted devices. The domain of gaze-based interactive applications increases dramatically as interaction is no longer constrained to 2D displays. This paper proposes a general framework for gaze-based controlling a non-stationary robot (vehicle) as an example of a complex gaze-based task in environment. This paper discusses the possibilities and limitations of how gaze interaction can be performed for controlling vehicles not only using a remote gaze tracker but also in general challenging situations where the user and robot are mobile and the movements may be governed by several degrees of freedom (e.g. flying). A case study is also introduced where the mobile gaze tracker is used for controlling a Roomba vacuum cleaner.

Author Keywords
Gaze-based interaction; robot; vehicle; craft; head-gestures; eye tracking; driving

ACM Classification Keywords
H.5.2. User Interfaces — Input devices and strategies.

Introduction
Gaze interaction can be generalized for usage in 3D environments where it can be used for interaction with many different types of objects present in our daily activities. This paper focuses on possibilities of using gaze trackers for controlling remote robots in 3D environment. Most approaches to gaze-based vehicle control are focused on using remote eye trackers and the point of regard on a monitor. This paper discusses how gaze interaction can be performed in more challenging situations where the eye tracker/user is mobile and where the vehicle movements may be governed by several degrees of freedom (e.g. flying). The paper categorizes different approaches for gaze-based controlling vehicles using the readily available data in eye trackers, and it discusses limitations and possibilities for the different approaches.

The rest of the paper is organized as follows. First the basic concepts in gaze controlling a vehicle are

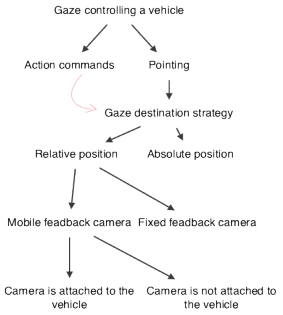


Figure 2: Gaze for controlling a vehicle.

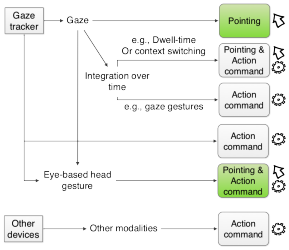


Figure 3: Gaze as a tool for pointing and for sending commands.

proposed in the following section. Then in the second section different situations and approaches are categorized based on the basic descriptions. Finally an implementation case study is presented and then we conclude the paper.

Basic Concepts and Overview

The fact that we often look into the direction of the next move when walking or driving, tells us that the gaze can somehow be used for enhancing the process of driving a car or a remote robot. Using gaze for controlling vehicles (e.g., wheelchairs and remote robots) has been studied in [2, 7, 8]. This paper studies the fundamental principles of gaze interaction with remote robots. There are many factors that influence the way that gaze can be used for controlling a vehicle (figure 2) specially in the challenging situations where: (a) mobile gaze trackers are used instead of remote gaze tracker, (b) when the user and robot can move relative to each other, (c) or when vehicle have more than two degrees of freedom (e.g. flying vehicles). In order to be able to discuss how gaze may be used in different situations, first these different factors are introduced in this section.

Controlling a Vehicle

In this paper, a vehicle is defined as a rigid body object located in 3D space with the ability of moving between two points inside its movement space. The movement space is a space of possible position pairs that the vehicle can travel between. The movement space may be one, two, or three dimensional (curve, surface, or a 3d space). In this paper, vehicles with each of these three types of movement spaces are termed as 1D, 2D or 3D vehicles.

A vehicle may have different conditions in terms of degrees of movement (translation and rotation). A 2D vehicle or a car can reach any point on the ground by having only one rotation (turning) and one translation (forward/backward), or by only having 2 translational degrees of movement (forward/backward and right/left). A vehicle may need to have more degrees of movements to be able to get any orientation in its movement space.

In this paper, controlling a vehicle is defined as below: "Sending at least one bit of information (*input information*) to a vehicle in order to start, stop, or changing the direction or velocity of the movement in at least one of the degrees of movement of the vehicle."

Consequently, when gaze is used in any form (directly or indirectly) in the process of providing the input information by the machine, we can say that vehicle is controlled by the gaze. In gaze-base controlling approaches, gaze can be used only for pointing, or for both pointing and sending commands together. This has been described more in the following.

Gaze for Interaction

Using gaze for interaction on screen-based interfaces is well known [6]. The point of regard and pupil position are locations in space. The point of regard (PoR) can at a specific time instance and due to the Midas-touch problem only be used for pointing and not be used to yield any additional commands. More information is needed to define commands e.g. to make a selection. A common way to achieve more information is to integrate the eye and gaze information over the time. Dwell-time activation is obtained when the gaze and pupil positions are fixed over time. The limitations of

Examples of two

Gaze control strategies

Gaze action strategy:

- The user looks at the "left" button on the computer display for moving the vehicle to the left.
- The user looks at the vehicle for moving it or stopping it.
- The user is sitting inside a car and looks at a point shown on the windscreen and perform a head gesture for changing the direction.

Gaze destination strategy:

- The user looks at a point on the floor (space) and sends a command, and then the robot goes to that point on the floor (space). In this case, the user may be looking at the scene through a display.
- The user is sitting inside a craft and looks at a target point and the craft goes toward that point.
- The user is looking at a button called "kitchen" and then the vehicle goes to the kitchen.

dwell-time activations have already been investigated thoroughly [4]. The principles of gaze gestures [3] is based on the pupil and point of regard (in space) are both changed over time. A known limitation of gaze gestures is that they do not allow the user to keep the gaze at object - i.e. gaze is removed from its context in which the interaction is intended. Context switching [9] turns this limitation into to an advantage by defining at least two contexts and let the transition of gaze and eye positions act as the defining principle for a command. Eye-based head gesture is a novel method for enhancing gaze-based interaction through voluntary head movements [5]. The method allows the gaze position to remain fixed while the pupil position is changing over time. Eye-based head gesture is based on the fact that when the point of regard is fixed and the head moves, the eyes move in the opposite direction due to the vestibulo-ocular reflex.
Since, eye-based head gesture technique can be achieved with both remote and head-mounted gaze trackers and provides us a gaze-based interaction method for executing commands in remote and mobile situations, in this paper it has been considered as the default tool for sending action commands to a remote robot (figure 3). However, measuring the head movements through the eye movements may be challenging in situations where the gazed object moves very fast while performing the head gesture. In these challenging situations in which the eye-based head gestures are not measurable, other interaction modalities for example the head movements measured by the other devices can be used for sending the action commands and gaze is only used for pointing.

Gaze Control Strategies

The PoR is usually a point on a 2D surface [1], however gaze may also be estimated as a 3D point or as a direction. This paper assumes that whenever the gaze is estimated as a direction, it should be intersected with a surface in order to become useful for controlling a vehicle. In this paper, possible approaches of using gaze point for controlling a vehicle are classified into two strategies: 1) gaze action strategy 2) gaze destination strategy.
Gaze action strategy uses the gaze for sending an action command to the system for starting or stopping the movement of the vehicle in one of its degrees of movement. Gaze destination strategy is when the gaze is used for giving information about where the movement should be stopped (desired or destination point). In the gaze destination strategy, the user may be directly looking at a desired point in space or in an image, or he/she may look at a context that contains information about the destination position.
The gaze destination strategy in which robot goes to reach a gazed destination point, involves different approaches based on the knowledge of the system about the exact position of the vehicle, the user and the gaze point in the world coordinates system. This has been addressed in the following.

Relative or Absolute Positions

When the gaze is used for pointing the destination of the next movement of vehicle, the navigation method would be different based on the information that the system has about the locations of the gaze point and the vehicle in the world coordinates system. When the system have enough information for obtaining the absolute position of the gaze point and the vehicle in the world coordinates system, it can easily determine

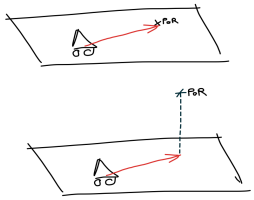


Figure 4: Minimizing the distance between the PoR and the 2D vehicle position on a plane

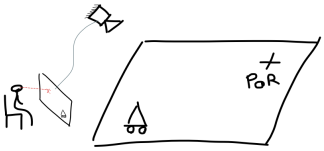


Figure 5: The user is sitting in front of a display showing the image from the feedback camera and a remote gaze tracker is used for gaze estimation.

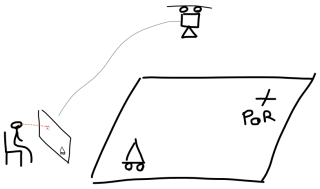


Figure 6: the feedback camera is mobile and the image is shown on a remote display

the path that the vehicle should follow in its movement space in order to reach the destination.
When the system does not have enough information about the exact position of the vehicle in the world coordinates system, but the relative positions of the vehicle and the gaze are accessible in an image, the navigation still can be done through a feedback loop. In this approach, the system measures the relative distance between the gaze point and the vehicle position in each time instance and always tries to minimize this distance.

Methods

In this section, we investigate different gaze-based methods of controlling a vehicle in different conditions.

Using Gaze action strategy

Hemin et. al [2] have followed this approach by introducing the TeleGaze interface overlaid on top of the video stream from the robot camera shown on a computer display, and used gaze for controlling a remote robot. Eye-based head gestures can be used for sending the action commands (e.g., left, right, forward, and backward) to a vehicle. It may be done by interacting with a graphical user interface or by looking at the objects in the real world. A case study of using eye-based head gestures for interacting with a mobile robot is introduced in the section in the following.

Using Gaze destination strategy

Different approaches of this strategy are categorized based of the knowledge of the system about the position of the vehicle and the PoR in the world coordinates system.

ABSOLUTE POSITIONS

This method is very straightforward when the position of the vehicle and the gaze point is known in the world coordinates system. The system infers the destination point from the gaze and then moves the vehicle toward the gazed point. In case of using mobile gaze trackers, estimating the gaze point in the world coordinates system varies based on the type of the eye tracker that is used and some times requires information about the position of the user in space. However, the destination coordinates can also be obtained indirectly from the gaze. For example when the destination is the point A, user can look at a button called "A" on the computer display, or a real object in the real world signed as "A", and the system infers the destination coordinates through the context of that object. The desired destination point may be inside or outside the movement space of the vehicle, and the system always tries to minimize the distance between the vehicle the destination point (figure 4). The way that the system navigates the vehicle is out of the scope of this paper.

RELATIVE POSITIONS

When the absolute position of the vehicle is unknown, in some situations, the gaze tracker may still be used for navigating the vehicle. It should be noted that here we assume that the gaze is pointing to a desired position (PoR) and the system moves the vehicle toward that point. Therefore, when we want the vehicle to follow our gaze point continuously (like when the cursor follows the gaze on the screen), at each instance of time, we are actually defining a destination point for the next movement. Consequently, we only discuss one step of controlling the vehicle, where we point to one specific point in space. When the system does not have any information about the absolute position of the

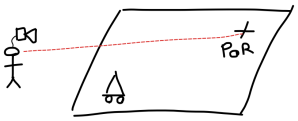


Figure 7: the feedback camera is the scene camera of a mobile gaze tracker.

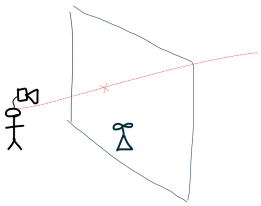


Figure 8: This figure shows the same concept of the figure 3 but with a 3D robot (craft) that be moved inside a virtual plane by the gaze. The movement in the third dimension (along the gaze direction) is controlled by looking at the craft and through continues head movements.

vehicle in the real world, but the relative positions of the destination point (gaze point) and the vehicle are known, navigation can be done through a feedback loop. This relative position may be measured by the system visually through a camera (feedback camera). The image of the camera should contain enough information about the PoR position in the image (x,y), and the posture of the vehicle (for all degrees of movement). When we only want to move the vehicle to the PoR and the final orientation is not important for us, the system only needs to be able to measure the changes of the minimum degrees of movement of the vehicle between each two frames. Two situations may happen for the feedback camera. The camera can be fixed or mobile in the world coordinate system while vehicle is moving in the feedback loop. Different approaches that can be used in these two situations are described below:

Fixed Feedback Camera

When the camera is fixed, one time sampling the gaze point in the camera image is enough and the system only needs to get feedback from the posture of the vehicle (changes in degrees of movement). If the user wants to change the destination point, he/she looks at another point and the system needs to update the PoR estimated by the gaze tracker. One example of this situation is shown in figure 5. It is obvious that when the PoR and the vehicle are along the optical axis of the camera, the system needs some extra information to move the vehicle in that direction.

Mobile Feedback Camera

In this situation, the user's gaze may move in the image when the feedback camera is moving. Therefore, the position of the destination point may be changed in each frame. Many computer vision techniques can be applied for detecting or tracking the destination point (PoR) in the image while the camera in moving. It means that the user does not need to keep gazing at the destination point while moving the vehicle, and one time sampling is enough. Two situations are shown in figure 6 and figure 7 where the feedback camera is mobile and it moves independently from the vehicle. When we have a 3D vehicle (craft) and the camera is not attached to it, the same approach can be used (figure 8). However, the craft can only move in a 2-dimensional plane unless the third degree of movement of the vehicle is activated by a command (e.g., pressing a button, or blinking, or a gesture). The eye-based head gestures can be very useful in this situation, because it allows the user to even control the third degree of movement by the continuous head movements while looking at the vehicle (craft). Furthermore, it does not require an extra device for controlling the third dimensional movement.

Figures 9 shows a situation where the feedback camera is attached to the vehicle. The image of the camera can then be transferred to a remote user outside the vehicle. The main important object that has to be considered here is that at least the axis of one of the translational movements of the vehicle should be visible in the image (either as a point or a line). In the feedback loop, the system tries to minimize the relative distance between the gaze point and the projection of the translational (forward/backward) movement in the image (adjusting the locomotion of the robot). However, the forward/backward movement cannot be controlled by the gaze point, and another modality should be used for controlling the movement in that direction. Eye-based head gesture can be used for this

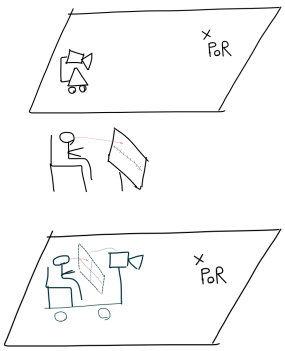


Figure 9: The feedback camera is attached to the vehicle and the image is shown on a remote desktop

Video demonstrations of this experiment can be accessed at http://youtu.be/6O2gYjRymyg.

More information about the open source Haytham gaze tracker is used for gaze tracking, eye-based head gestures, and interaction with the robot can also be accessed at http://eye.itu.dk.

purpose while the user is looking at a point. Tall et.al [8] have implemented the situation shown in figure 9. The camera may also be attached to a 3D craft in space and the gaze can be used in a same way for controlling it. However, the navigation method during the feedback loop varies based on the degrees of movement of the vehicle.

Case Study

An implementation case study has been conducted for controlling a Roomba vacuum cleaner using a mobile gaze tracker. Two strategies are tested in the experiment. The first was the gaze action strategy where the user looks at the robot ant controls the robot using the head gestures. The second, Roomba is following the user's gaze in the scene image and only the action commands (e.g., clean, turn off, and turn on) were sent using the head gestures. Figure 7 shows the mobile situation and the scene image of the mobile gaze tracker. A visual marker is used for detecting the robot in the image. This case study is an example of the situation using a mobile feedback camera in a relative position method.

Conclusion

The fundamental principles for controlling non-stationary robots have been studied. Different approaches for controlling vehicles using gaze are categorized based the knowledge of the system about the PoR and position of the vehicle in 3D space. A case study is introduced in which two of the approaches are implemented. This case study shows the potential of using only a mobile gaze tracker for controlling a remote robot in a 3D environment.

References

[1] Hansen, D.W., and Ji, Q., 2010. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. IEEE Trans. Pattern Anal. Mach. Intell, pages 478-500.

[2] Hemin, O. L., Nasser, S. & Ahmad L. (2008) Remote Control of Mobile Robots through Human Eye Gaze: The Design and Evaluation of an Interfacel, SPIE Europe Security and Defence 2008.

[3] Isokoski, P. 2000. Text Input Methods for Eye Trackers Using Off-Screen Targets. In Proceedings of the ACM symposium on Eye tracking research & applications ETRA '00. ACM Press.

[4] MacKenzie, I. S., & Zhang, X. (2008). Eye typing using word and letter prediction and a fixation algorithm. In ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications.

[5] Mardanbegi, D., Hansen, D. W., and Pederson, T. Eye-based head gestures. In Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA'12, ACM Press.

[6] Morimoto C., H., Koons, D., Amir, A., Flickner, M., and Zhai, S. 1999. Keeping an Eye for HCI. In Proceedings of the XII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '99). IEEE Computer Society.

[7] Matsumoto Y., Ino, T. & Ogasawara, T. (2001). Development of Intelligent Wheelchair System with Face and Gaze Based Interface, Proceedings of 10th IEEE Int. Workshop on Robot and Human Communication.

[8] Tall, M., Alapetite, A., Agustin J.S., Skovsgaard, H., Hansen, J. P., Hansen, D.W., and Møllenbach, E. 2009. Gaze-controlled driving. In CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09).

[9] Tula, A. D., Campos, F., and Morimoto, C.H. 2012. Dynamic context switching for gaze based interaction. In Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12), Stephen N. Spencer (Ed.).

- Chapter 5 -

Gaze Activation Techniques

Eye Activation Techniques

Diako Mardanbegi
IT University of Copenhagen
Rued Langgaards Vej 7,
DK-2300 Copenhagen S
dima@itu.dk

Dan Witzner Hansen
IT University of Copenhagen
Rued Langgaards Vej 7,
DK-2300 Copenhagen S
Witzner@itu.dk

ABSTRACT
This paper provides a systematic review of eye-based activation techniques. It suggests a taxonomy of eye activation techniques based on the way that information provided by a gaze tracker is used for making selection or in general for sending activation commands in gaze interaction scenarios. Conventional techniques are introduced based on the presented taxonomy and their limitations are described. Also, a comparison between different eye activation techniques for the purpose of interaction in 3D has been presented.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Presentation]: User Interfaces – Evaluation/methodology; Input devices and Strategies

General Terms
Performance, Experimentation, Human Factors.

Keywords
Gaze selection, Gaze activation, Eye activation, Gaze tracking, Eye tracking, Gaze interaction

1. INTRODUCTION
This paper focuses on explicit gaze-based interaction scenarios where a system acts on explicit commands measured by a gaze tracker. Pointing seems to be the most obvious use of gaze, however, interaction with objects is more than just pointing, and the ability of selecting an item or even issuing more commands is needed in many applications. Therefore, the explicit gaze interaction is divided into pointing and activation. In this paper the term *eye activation technique* is used rather than *gaze selection technique* when eye is used in any form in the process of providing input information needed for selecting an object or executing an action command.

Møllenbach [15] has suggested a taxonomy of eye activation techniques based on eye movements characteristics and visualization. In contrast, this paper presents a different taxonomy for eye activation techniques based on how an activation command can be measured by the gaze tracker. This taxonomy looks at the eye activation strategies from the point of view of the source of information rather than the eye movements. Knowing and classifying how information derived from the eyes are used for interaction can potentially help us to create new techniques for gaze interaction.

There exist different eye-based activation methods that can be used together with gaze pointing for interaction with computer user interfaces. These traditional eye-based activation methods

are basically initiated to help people with severe motor impairments to interact with computer displays. This paper has a broader perspective and classifies different gaze supported activation techniques that can also be used by general people. Gaze-supported multimodal interaction techniques can potentially be used in the future for general use in our daily life (e.g., controlling wearable computing devices).

2. EYE ACTIVATION
In general gaze trackers can provide an abundance of information about the subject (e.g., gaze, eye features, and eye movements) and the environment (e.g., object recognition). Different types of eye-related information can be obtained from the eye camera which is a common element between remote gaze trackers (RGT) and head-mounted gaze trackers (HMGT). On the other hand, HMGTs can yield information from both the eye camera and the scene camera. Each may serve different purposes when used for interaction. Therefore a distinction is made here between *eye-related information* and *non-eye-related information* obtained from a gaze tracker. The term eye-related information is used for any type of information that is provided by the gaze tracker and is somehow related to the eye (e.g., eye movements, eye features, and gaze related data). The term non-eye-related information is used for other type of information that is obtained from the gaze tracker but it is not related to the eye (e.g., information obtained from the scene camera or a gyroscope). The gaze tracker can measure an activation command through these two types of information. Eye activation techniques that are the main focus of this paper use the eye-related information for measuring an action command executed by the user. Gaze trackers may also measure an activation command through non-eye-related information. For example, body gestures can be detected through the scene image of a HMGT and be used for activating an object while looking at the object [3].

The first group in Figure 1 includes those techniques that complement the gaze pointing with action commands measured through non-eye-related information or other input devices. Different conventional eye activation strategies have been categorized into 3 classes. This categorization has been illustrated in Figure 1 (group 2-4) and each group is described in the following.

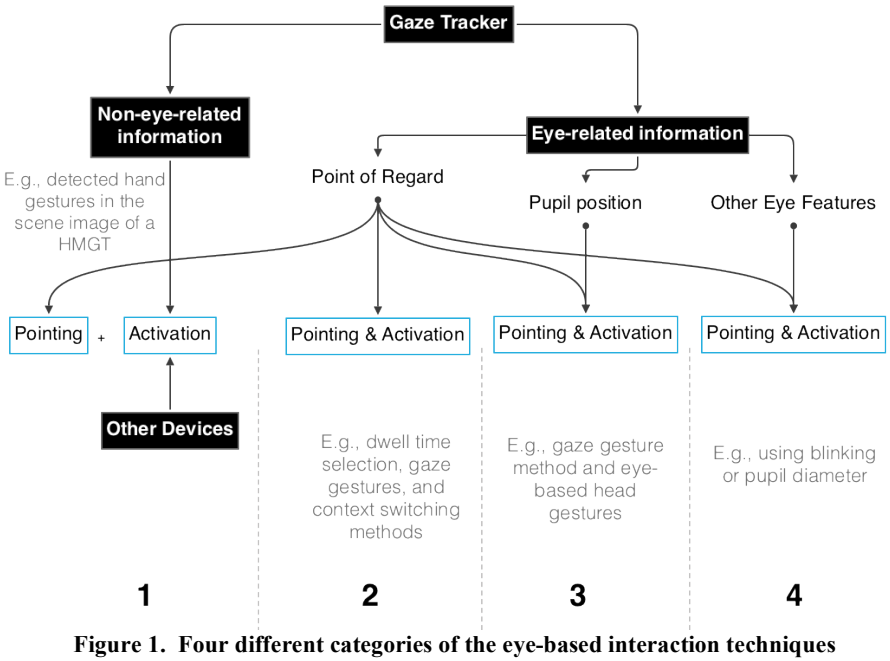


Figure 1. Four different categories of the eye-based interaction techniques

Among the eye activation techniques, are some that use eyes actively both as a pointing and an activation mechanism (e.g., dwelling). There are also some techniques in which the subject uses the eye actively only for pointing, and then executes the activation commands by other modalities (e.g., head gestures) that will be measured through the eyes by the gaze tracker. These techniques are described more in the following.

2.1 Using the point of regard

One of the eye-based information obtained form a gaze tracker is the point of regard (PoR) which is used as a pointing mechanism. PoR can at a specific time instance and due to the Midas-touch problem (the accidental selection of anything that is looked at) only be used for pointing and not be used to yield any additional commands (e.g. to make a selection). However, there are some methods that use only the PoR for both pointing and executing commands. A common way to achieve more information from PoR is to integrate it over the time. Dwell-time activation (dwelling), gaze gestures and the context switching methods are examples of this approach.

Dwelling

Dwell-time is when the gaze is fixated on an object for a duration of time (a dwell). The limitations of dwell-time activations have already been investigated thoroughly [12], [9]. A single dwell selection is when the activation occurs on the initial fixation. The need for long dwell duration time is one of the main limitations of the single dwell selection.

The amount of unintended selections will be increased when dwell duration is too short, whereas long dwells will decrease the user's performance will be annoying for experienced users that are working in a familiar surround [4]. On the other hand, the complex dwell[1] selection [14] (e.g., two-step dwelling) requires a visual feedback of the selection process, therefore it is limited by screen space, and require a high level of precision pointing by the user [14]. Dwelling can be used for sending one bit of information that makes the "selection" possible. Interaction with the computers and graphical user interfaces involves more than just pointing and selection. Two-step dwell [11] may become useful for issuing secondary commands such as right-clicking. However, This technique is not sufficient when more commands are needed to accomplish a task.

Gaze Gestures

Gaze gestures [6] are based on the changes of the location of the pupil center or point of regard over time. Istance et al. [7] define the gaze gesture as: *"A definable pattern of eye movements performed within a limited time period, which may or may not be constrained to a particular range or area, which can be identified in real time, and used to signify a particular command or intent"*.

This definition of the gaze gesture might become clearer by changing the “eye movements” to “gaze”, because it defines the gaze gesture by an eye movement pattern, whereas the eye may be moving even when the gaze is fixed. This occurs when rotating the head while looking at an object (due to the vestibulo-ocular reflex).

The first obvious problem with gaze gesture activation is that it uses a perceptual channel such as vision for motor control that may be considered unnatural. In terms of interaction gaze gestures are also facing several limitations. Mollenbach [14] has studied the characteristics, advantages, and limitations of the single stroke gaze gestures. However, single stroke gaze gestures only constitute a limited number of interactions. Furthermore, natural eye movement patterns may not be easily distinguished from the simple gaze gestures. Therefore, complex gaze gestures that are composed of more multiple strokes are needed for robust results. Making the gesture patterns more complex requires the user to remember specific eye movement patterns and their consequence while forcing the eyes to be used actively. This takes the focus away from the actual interaction task, and increases the cognitive load.

The main limitation of the gaze gestures is that when performing a gaze gesture, the point of regard leaves the object of interest while interaction.

Although gaze gesture uses changes of gaze for executing commands, it can be combined with gaze pointing by considering one of the fixation points along the gesture for pointing. For example, the fixated object at the beginning or at the end of the gesture may be considered as the interaction object.

Context Switching

[1] Complex dwell selection is when the activation occurs after multiple fixations

Context switching technique [17] defines two contexts and let the transition of gaze and eye positions from one context to another context acts as the defining principle for a command. When the PoR is on an item in a context and then the PoR jumps to the other context, it triggers the selection of the item that was under focus. The main two advantages of this method are the user can freely explore one context without worrying about the Midas touch, and the user can see the interaction context after the saccade. Although the context switching has been shown to be a good alternative for the gaze gesture activation in some applications (e.g., eye typing), it might not be practical in some gaze interactive applications that have only one interaction context (e.g., controlling a real object in the environment).

2.2 Using the pupil position

Just as using the changes of the PoR in the first category, the pupil position (the center of the pupil) which is just a point in the eye image, can be used for activation by integrating its change over time.

When the PoR changes, the pupil position in the eye image changes as well. Therefore, the gaze gesture technique described before can be also measured through the pupil position instead of the PoR.

Eye-based head gestures

Eye-based head gesture is another eye activation technique that uses the changes of the pupil position (through voluntary head movements) while the PoR is fixed [13], [16]. Therefore, this method allows the gaze position to remain fixed while the pupil position changes over time. This technique makes use of VOR movements of the eye caused by the head movements. Since the main assumption of this method is that the PoR is fixed, it may not work when the object of interest is moving in the field of view of the user. The performance of this method for the moving objects has not been studied yet, but separating the VOR movements from the natural eye movements may be challenging in situations where the gazed object moves very fast while performing the head gesture.

2.3 Using the other eye features

The gaze tracker can provide more eye-based information than PoR and the pupil position. For example, blinking (detected by the eye tracker) has been used as a selection mechanism [8]. Double-Blink (Blinking twice quickly) may also become useful for issuing more commands such as right-clicking. Achieving more information from this technique maybe uncomfortable (e.g. by combining them in a sequence similarly to Morse code).

Blink normally happen about 10 times per minute [1], and therefore these natural blinking have to be separated from the intentional blinking for object selection, otherwise, some natural blinks may be mistaken for activations. One solution to differentiate between the intentional and natural blinks for activating events is to make the duration of intentional blinks longer than the average length of a natural blink which is about 300-400 ms [10]. Making the blinks longer may influence the speed of interaction and more important natural changes of the vergence of the eyes that occur during prolonged blinks [4], may move the gaze point from interaction object. Furthermore, repetitive blinking for long-term use may become tiring for the user.

The voluntary pupil dilations have been also studied [2] as an activation mechanism in some interactive applications. However, not many people can control their pupil, and besides that there are many parameters that in effect the pupil dilation.

3. DISCUSSION

This paper presents a systematic way of categorizing the conventional eye activation techniques based on the source of information used in these techniques. Looking at the gaze interaction from the point of view of information gives us a different perspective and may reveal some potential activation techniques that could be the subject of future research.

Different conventional eye activation techniques are introduced and their major limitations are described in this paper. As it was mentioned before, most of the eye activation methods are initiated for use in mono-modal interactive situations (using the eyes actively for both pointing and activation) with the purpose of helping disable people interacting with the computer displays. With head-mounted gaze tracking technology getting smaller and easier to use, it is likely that in the near future HMGT functionality will be compact enough to fit into wearable computing devices. Therefore, gaze interaction moves more toward interaction with the 3D environment in mobile situations. Although, there may be many other different sensors embedded in the wearable computers that can sense our body gestures and actions, still a proper eye activation technique can enhance our interaction with the environment. However, gaze interaction with real objects in 3D may require different considerations than gaze interaction with computer graphical user interfaces. This involves some constraints on the way that eye information is used in the eye activation technique. For example, a convenient property would be that the activation techniques should not preferably require the gaze to be removed from the object. This leaves out the use of gaze gesture technique for interaction with the environment. Another reason why it is not practical to use gaze gestures for interaction in 3D is that this method requires some pre-defined target points (e.g., off-screen targets) to help the user performing a desired gaze pattern. Having a set of pre-defined targets around different objects in the environment is not always possible unless we have a head-mounted display where we can display some fixation targets around each object. Another consideration that needs to be addressed when interaction with the real objects in the environment is that more than one activation command may be needed for controlling different objects. For example, dwelling technique may be used for making selection or sending an ON and OFF command to an object but this method cannot easily provide more information especially when there is no graphical user interface and visual feedback.

Based on the discussion above, eye-based head gesture technique seems to be the most convenient method for interaction in 3D, among the other conventional eye activation techniques. It allows the user to keep the gaze fixed on the object of interest while controlling it. It also provides a variety of activation commands that can be used for interaction with different types of objects in the environment. Furthermore, in addition to discreet gestures, continuous head movements can be used for changing the continuous and analog interactive objects e.g., for scrolling, zooming, panning, dragging items, and adjusting the volume.

4. REFERENCES

[1] Doughty, M. J. Further assessment of gender- and blink pattern- related differences in the spontaneous eyeblink activity in primary gaze in young adult humans. Optometry and vision science: official publica- tion of the American Academy of Optometry 79, 7 (July 2002), 439–447.

[2] Ekman, I., Poikola, A., Mäkäräinen, M., Takala, T., and Hämäläinen, P. Voluntary pupil size change as control in eyes only interaction. In Proceedings of the 2008 symposium on Eye Tracking Research & Applications, ETRA '08, pages 115–118, New York, NY, USA, 2008. ACM.

[3] Hales, J., Rozado, D., Mardanbegi, D. "Interacting with

Objects in the Environment by Gaze and Hand Gestures" 3rd International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction (PETMEI2013) at 17th European Conference on Eye Movements (ECEM 2013), Lund, Sweden, 2013.

[4] Huckauf, A., and Urbina, M. H. Object selection in gaze controlled systems: What you don't look at is what you get. ACM Trans. Appl. Percept. 8, 2 (Feb. 2011), 13:1–13:14.

[5] Huckauf, A., and Urbina, M. H. Object selection in gaze controlled systems: What you don't look at is what you get. ACM Trans. Appl. Percept. 8, 2 (Feb. 2011), 13:1–13:14.

[6] Isokoski, P. Text input methods for eye trackers using off-screen targets. In Proceedings of the 2000 symposium on Eye tracking research & applications (New York, NY, USA, 2000), ETRA '00, ACM, p. 15–21.

[7] Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., and Vickers, S. Designing gaze gestures for gaming: an investigation of performance. In Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications (New York, NY, USA, 2010), ETRA '10, ACM, p. 323–330.

[8] Jacob, R. J. K. Eye movement-based human-computer interaction techniques: Toward non-command interfaces. In IN ADVANCES IN HUMAN-COMPUTER INTERACTION (1993), Ablex Publishing Co, p. 151–190.

[9] Jacob, R. J. K. The use of eye movements in human-computer inter- action techniques: what you look at is what you get. ACM Trans. Inf. Syst. 9, 2 (Apr. 1991), 152–169.

[10] Kaufman, P. L., Levin, L. A., Alm, A., Nilsson, S. F., and Ver Hoeve, J. Adler's Physiology of the Eye. Mosby, 2011.

[11] Lankford, C. Effective eye-gaze input into windows. In Proceedings of the 2000 symposium on Eye tracking research & applications (New York, NY, USA, 2000), ETRA '00, ACM, p. 23–27.

[12] MacKenzie, I. S., and Zhang, X. Eye typing using word and letter prediction and a fixation algorithm. In Proceedings of the 2008 sym- posium on Eye tracking research & applications (New York, NY, USA, 2008), ETRA '08, ACM, p. 55–58.

[13] Mardanbegi, D., Hansen, D. W., and Pederson, T. Eye-based head gestures. In Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA'12, ACM Press.

[14] Mollenbach, E. Selection Strategies in Gaze Interaction. PhD thesis, Loughborough University, 2010.

[15] Møllenbach, E., Hansen, J. P., & Lillholm, M. (2013). Eye Movements in Gaze Interaction. *Journal of Eye Movement Research*, *6*(2), 1-1.

[16] Oleg Špakov and Päivi Majaranta. 2012. Enhanced gaze interaction using simple head gestures. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (UbiComp '12). ACM, New York, NY, USA, 705-710.

[17] Tula, A. D., de Campos, F. M. S., and Morimoto, C. H. Dy- namic context switching for gaze based interaction. In Proceedings of the Symposium on Eye Tracking Research and Applications (New York, NY, USA, 2012), ETRA '12, ACM, p. 353–356.

- Chapter 6 -

Eye Based Head Gestures

Eye-based head gestures

Diako Mardanbegi
IT University of Copenhagen
dima@itu.dk

Dan Witzner Hansen
IT University of Copenhagen
Witzner@itu.dk

Thomas Pederson
IT University of Copenhagen
tped@itu.dk

Abstract

A novel method for video-based head gesture recognition using eye information by an eye tracker has been proposed. The method uses a combination of gaze and eye movement to infer head gestures. Compared to other gesture-based methods a major advantage of the method is that the user keeps the gaze on the interaction object while interacting. This method has been implemented on a head-mounted eye tracker for detecting a set of predefined head gestures. The accuracy of the gesture classifier is evaluated and verified for gaze-based interaction in applications intended for both large public displays and small mobile phone screens. The user study shows that the method detects a set of defined gestures reliably.

CR Categories: H.5.2 [Information interfaces and presentation]: User Interfaces ─Input devices and strategies, Interaction styles

Keywords: Head Gestures, Gaze interaction, Eye tracker, Interaction

1 Introduction

Gaze-based interaction has so far been restricted to interaction with computer screens using remote eye trackers. Gaze-based applications are still waiting to be investigated with improved principles that can even be used for gaze-based interaction in 3D environments as well as with virtual objects on screen. This paper proposes a novel method for enhancing gaze-based interaction through both voluntary head movements and vestibulo-ocular reflexes. Contrary to previous research this information is obtained only through eye and gaze information using an eye tracker. The method is shown to be useful for both gaze-based screen interaction and 3D environmental control.

Gaze interaction has been shown to be useful for many applications but eye information has been shown to be limited for interaction. The point of regard only posses information about position and does not provide sufficient information to make selections (a.k.a Midas touch). Extra information is needed to make convey other pieces of information such as clicks. Dwell-time selection, eye blinks and gaze-gestures [Jacob 1993; Isokoski 2000] have been typical ways of extending the capabilities of eye trackers with methods for communicating with interfaces (e.g. making selections on a screen).

Gestures are commonly used for interaction and are used to signify a particular command or intent. For eyes there are two types of gestures, namely eye and gaze gestures. Eye gestures such as wink and blinks make use of movements of the eyelid and eyebrows. However, interaction with eye gestures and blinking especially repetitive blinking for long-term use may create a feeling of nervous eye muscles [Drewes 2010]. Gaze gestures, on the other hand, are definable patterns of eye movements performed within a limited time interval [Istance et al. 2010]. Simple gaze gestures are not distinguishable from natural eye patterns and make unintended interaction similar to the Midas-touch problem. Complex gaze gestures consist of several simple gaze gestures are therefore needed for robust results. Such use may be considered unnatural as a perceptual channel is used for motor control [Zhai et al. 1999]. Besides, it may be physically straining and requires the user to memorize combinations of gaze gestures. This increases the cognitive load while forcing the eyes to be used actively, and therefore takes the focus away from the actual interaction task. In terms of interaction gaze gestures are facing severe limitations, for example, gaze gestures are not intuitively applicable for user interaction on e.g. Icons or objects in 3D space since the point of regard possibly leaves the object while interacting thus may confuse the user as well as significantly complicates the algorithmic design.

Head nods and shakes are widely used in our daily conversation as a gesture to fulfill a semantic function and as conversational feedback (e.g., nodding instead of saying yes) [Darwin 1872; Morris 1994]. People are more used to making deliberate movements of the head compared with similar patterns of eye movements. Basic head gestures such as nod and shake are relatively easy to measure from full-face images and have been also used for interaction with user interfaces [Toyama 1998; Kjeldsen 2001]. Methods for video-based head gesture recognition deal with three main problems: First localizing and identifying the face region in the image (which may have a cluttered background) using a fixed camera located in front of the head and works only when the face is in the field of view of the camera. The second problem is to extract the feature set that represents the head movements. And then classifying the feature set into a number of head gestures. These methods are not able to separate

the head gestures from the natural head movements and most of them are only limited to detect some specific gestures like head nods and shakes. On the other hand, real time detection of head nods and shakes is difficult, as the head movements during a nod or shake are small, fast and jerky.

This paper suggests using head gestures measured by the gaze trackers, as a convenient way of interaction when using the gaze trackers. Eye image alone is not sufficient for detecting the head gestures, but by in combination with gaze information it is possible to measure the head gestures. This paper describes a novel approach for detecting head movements using only eye images and the point of regard. Having the point of regard allows for distinguish between the visual eye movements (eye-movements that are associated with vision) and the non-visual eye-movements that are associated with vestibulo-ocular reflex (VOR) and are caused by the head movements. This work is meant for gaze-based interaction and is related to gaze gestures in the sense that eye movements are used to signal gestures. However, the user does not move the eyes voluntarily, but eye movements are an effect of vestibulo-ocular reflexes when the user fixates on the interaction object and does head gestures.

This paper shows that it is possible to detect a relatively large amount of both large and small head gestures, using gaze trackers thus minimizing the need to make very complex gestures. The main advantage of this method is that attention remains fixed on the object of interaction while executing gestures.

The rest of the paper is organized as follows. Section 2 presents related work, and section 3 presents an overview of the method. The head gestures are introduced in section 4, and the algorithm used for recognizing the gestures are described in section 5. Section 6 describes the experimental applications in which the method is tested for interaction. Section 7 presents the experimental results and section 8 concludes the paper with future work.

2 Previous work

A comprehensive review on gaze gestures is given in [Møllenbach 2010]. Research on gaze gestures was initiated by Isokoski for text input using off-screen targets. The eye gaze has to visit the off-screen targets in a certain order to select characters. Off-screen targets force the gesture to be performed in a fixed location and with a fixed size [Isokoski 2000]. Drewes and Schmidt [2007] made a comprehensive research on gaze gestures and presented some scalable gaze gestures which could be performed in any location on screen, and used them for interacting with computers and devices with smaller displays. Wobbrock et al. proposed a similar idea to gaze entry of letters using Edge-Write gestures when the user could map out letters by combining the four corners of a square in various ways [Wobbrock et al. 2007]. The idea of using the gaze gestures for text input was continued later [Porta and Turina 2008; Bee and Andre 2008].

Many video-based methods for head gesture recognition have been proposed. Some attempts have been made to use eye information (e.g., eye location) for head gesture recognition. Davis and Vaks presented a prototype perceptual user interface for a responsive dialog-box agent. They used IBM PupilCam technology for only detecting the eye location in the image and used together with anthropometric head and face measurements to detect the location of the user's face. Salient facial features are then identified and tracked between frames to compute the global 2-D motion direction of the head. A Finite State Machine incorporating the natural timings of the computed head motions was employed for recognition of head gestures (nod=yes, shake=no) [Davis and Vaks 2001]. Kapoor and Picard introduced an infrared camera synchronized with infrared LEDs to detect the position of the pupils, and used it as the feature. A HMM based pattern analyzer was used to detect the nods and the shakes [Kapoor and Picard 2002]. Recognition of head gestures had been demonstrated by tracking eye position over time. They presented a real-time nod/shake head gesture detector. However, their system used complex hardware and software and had problems with people wearing glasses and with earrings. Nonaka [2003] used Eye-mark recorder and FASTRAK motion tracking system to track the eye movements and head movements respectively and proposed a communication interface working by eye-gaze and head gesture. Nonaka tried to use the eye tracker for detecting the fixed point of regard during the head gestures. Beside the complex hardware of the system, FASTRAK head motion tracker only worked in the range of its magnetic transmitter (max 3 meter). Fixation of eye gaze and also the gestures of "Shaking Head", "Nodding Head", and "Inclining Head" (assigned to "no", "yes" and "undo" respectively) are detected using successive dynamic programming (S-DP) matching method with their reference patterns. However this system was not always able to identify even these three gestures correctly.

3 VOR-based detection of head movements

Eye movements can be caused by the head movements while PoR is fixed (*fixed-gaze eye movements*) or by changing the PoR when the head is fixed (*fixed-head eye movements*). This paper investigates the fixed-gaze eye movements. When the point of regard is fixed and the head moves, the eyes move in opposite direction and with the same speed as the head movement. The eye movements are due to the vestibulo-ocular reflexes (VOR), which are used to stabilize the image on the retina. Figure 1 illustrates a user looking at an object but in two different situations, one when head is up and the other when head is down. The eye image is different in each posture even though the PoR is fixed.

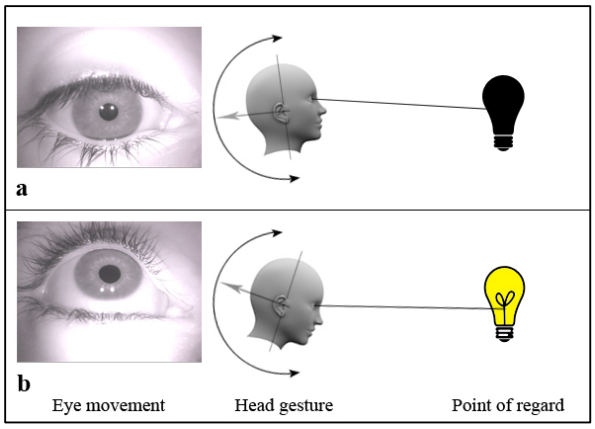


Figure 1 *Eye image when POR is fixed and (a) head is up or (b) down*

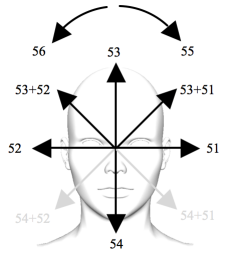
The eye trackers are able to distinguish between *fixed-gaze eye movements* and *fixed-head eye movements* since they measure both eye movements and estimate the point of regard. The term *eye-based head gestures* will in the following denote a predefined pattern of head movements measured through eye movements but where the PoR is fixed on a given object.

This paper focuses on measuring head gestures from a head mounted eye tracker. Head mounted eye trackers move with the head movements and there is therefore no information about the world reference frame in the eye image. Eye movements caused by VOR or by changing the gaze direction cannot be determined unless additional information is available. Head-mounted eye trackers have a scene camera that captures the user's field of view through which the PoR is determined. So, by the ability of recognizing a known reference point in the scene image, fixed-gaze eye movements can be recognized through the point of regard and the reference point when the user fixes the gaze and moves the head.

4 Head Gestures

This section describes head gestures, their relation to eye movements and how these can be measured in an eye tracker.

Ekman and Friesen [1978] developed a common standard to systematically categorize and encode human facial expressions. There are 44 action units (AU) that account for change in facial expressions and orientations. 8 action units correspond to head orientation (shown in figure 2). Some movements such as diagonal downwards movements (AU54+52 and AU54+51 in figure 2 (left) are uncomfortable to perform and are usually made in conjunction with head tilts (AU55-56). These head movements would not be suitable for interaction and are therefore disregarded in this paper.

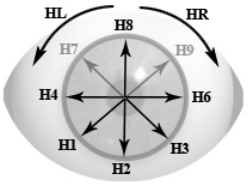


AU Number	Movement
51	Head turn left
52	Head turn right
53	Head up
54	Head down
55	Head tilt left
56	Head tilt right

Figure 2 *The basic movements of head gestures and their AU number*

When a user keeps the gaze on a specific point in space, the vestibulo-ocular reflex makes it possible to measure head movements through eye movements, but where eye and head movements are in opposite directions. Consequently, head movements are measurable indirectly by eye trackers, even in close-up images.

The basic eye movements associated with a given head movement (when the PoR is fixed), is shown in figure 3. The VOR has both rotational (AVOR) and translational (TVOR) aspects [Panerai1998]. When the head tilts (AU55-56), AVOR can be seen as the iris rotates around LoS axis. These movements are termed as rotational eye movements. For the other head movements (AU51-54), we see a translation of the pupil center in the eye image, which is termed as linear eye movements. In the following, HL and HR denote left and right rotational eye movements and H1-H9 denote the linear eye movements



Eye movement	Head movement
H1	53+51
H2	53
H3	53+52
H4	51
H6	52
H8	54
HR	56
HL	55

Figure 3 *Basic reflexive movements of the iris/pupil and the corresponding head movements*

Measuring sequences of eye movement are usually influenced by noise. We define a *character,* C_i, as a sequence of N eye movements where the majority of movements are the same e.g. $C_i = H_i^1..H_i^N$, (defined in figure 3). *Head gestures* are either *discrete* or *continuous.* A *discrete head gesture*, $G = C_1..C_T$, consists of a repeatable and recognizable sequence of characters, C_i. Discrete gestures and characters can conceptually be related to words and letters, when writing text. *Simple gestures*, $Gij = C_iC_j$ are 2 character words and *continuous gesture*, G_H, are sequences of eye movements H along an axis.

There are in total (8*8) 64 simple gestures but only 14 of these are considered here since executing and distinguishing gestures that are orthogonal or neighboring is hard. A simple gesture is denoted *sweep gesture* when the characters in the gesture $Gij = C_iC_j$ are different ($i \neq j$) and is denoted *repetitive* when the characters are identical ($i = j$). In this paper repetitive gestures consist of two linear movements separated by a short break, C_B. Figure 6 shows examples of sweep gestures (top row) and a repetitive gesture (bottom row). Gestures are in this paper well described by regular expressions and thus by a finite state machine.

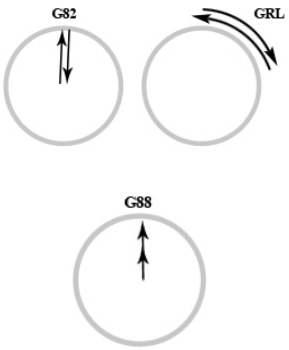


Figure 4 *Examples of sweep gestures (top row) and a repetitive gesture (bottom row). The arrows indicate eye movement. The actual head movement is in the opposite direction.*

An example of continuous gestures is shown in Figure 5 where the gesture is used to continuously change the value e.g. the volume of a loudspeaker.

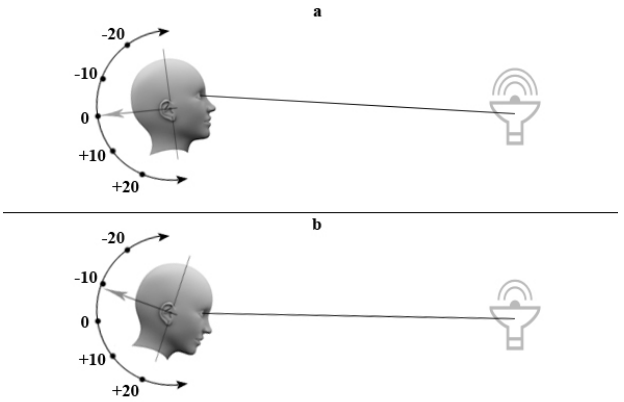


Figure 5 *A continuous gesture moving the head downwards while the eyes move upwards.*

5 Gesture recognition

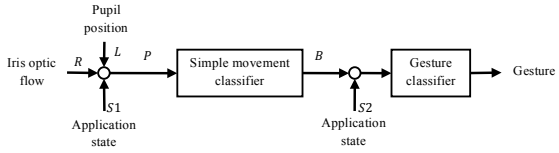


Figure 6 *Overview of the gesture recognition method.*

The method consists of a classifier to detect the basic eye movements and a gesture classifier based on regular expressions and is shown in figure 6. The length of the sequences of head movements defining a character and the dictionary of gestures are found experimentally.

Basic eye movements H_i^t at time t are estimated through feature vectors $\{P_1, \ldots, P_t\}$ measured from the images. Each feature vector is for clarity separated into 3 subsectors $P_i = [L, R, S_1]$, where $L = [l_1, l_2]$ are features needed for detecting the linear movements, $R = [r_1, .. r_8]$ are features needed to estimate rotational movements and S_1 is the current application state. The pupil center (l_1), and its velocity (l_2) between frames define the feature vector L. Feature vectors $r_1, .. r_k$ are sampled in regions $A_1, .. A_k$, where r_i is the mean optic flow quantized into 8 directions. The regions A_i and the corresponding feature vector r_i are shown in figure 7. The location of each patch A_i is defined relative to the pupil diameter to ensure the regions A_i are stabilized within the normalized region between iris and pupil.

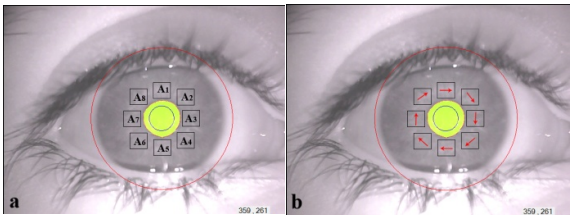


Figure 7 *Measurements of rotational head movements with (left) the regions A_i and the corresponding feature (right) measured during a rotation of the head.*

6 Experimental applications

The proposed method has been implemented for use with a head mounted eye tracker. Two experimental applications have been developed.

iRecipe, is an application to read and follow recipes when the hands are occupied or in a state that is not recommended for touching the computer. The second application is called *iiPhone* which is an iPhone emulator running on the screen that can be controlled by head gestures to show the potential of the proposed method for the mobile devices.

For both applications, the screen contour is detected and tracked within the scene image of the eye tracker. The eye tracker provides only gaze estimates $s = (x_s, y_s)$ in the scene image, but we need to determine where on the screen the user is looking. A homography [Harley and Zisserman 2000] from the screen corners S_i to M_i (figure 8) is estimated in each time instance. The gaze point in the scene image coordinates is then mapped to the screen coordinates through $(x_m, y_m) = H_s^m \cdot \mathbf{s}$. Figure 8 shows the mapping of the PoR (center of the red cross-hair) from the scene image to the screen plane and the real coordinates of the PoR in the screen by a black cross-hair (left image) [Mardanbegi and Hansen 2011].

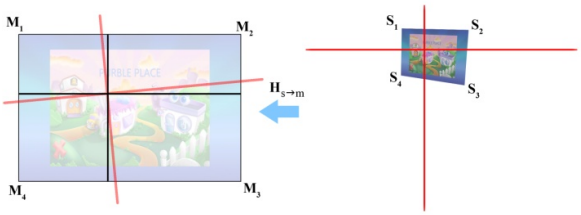


Figure 8 *Mapping from the scene plane (right) to the real screen plane (left)*

6.1 iRecipe application

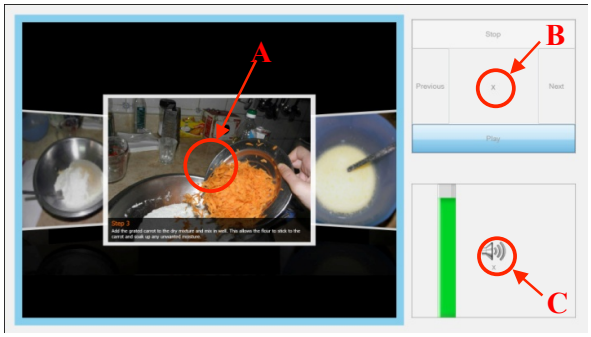


Figure 9 *iRecipe interface with slides frame in the left side, music player at the top right corner and the volume frame at the below right corner. A, B and C are predefined regions in each frame that user should look at them while doing the gestures.*

The iRecipe application is intended for a hands-free interaction with a recipe when cooking. The user interface of iRecipe is shown in figure 9. The interface consists of three areas: the recipe slides frame, a simple music player and the volume frame. The interface is operated by looking at predefined regions (A, B or C) while doing the gestures. Each gesture is interpreted differently based on the gazed object. Therefore the same gestures might have different meanings depending on the PoR.

Four different sweep gestures including Up, Down, Left and Right ($G46, G64, G28, G82$) together with the continuous vertical head movements were used for controlling the application as below:

I. Changing the slides by looking at the region "A" and doing the right or left head gestures.

II. Changing the music files by Left/Right gestures and stopping and playing by Up or Down gestures when looking at the center of the player (B)

III. Changing the volume had 3 steps. First enabling the volume by looking at the icon(C) in the volume window for 1 second (dwell-time activation). The color of the icon will be changed when the volume is active, and then user can change the volume by vertical head movements (as it is shown in figure 8). Then the volume can be disabled simply by looking at another part of the screen or by closing the eye. Changing the volume will be indicated by showing a number (0-100) below the icon during the vertical movements and adjusting the volume.

There is no cursor shown on the screen but the user has visual feedback on the interface during the interaction as the regions become highlighted when the PoR is inside that.

6.2 iiPhone application

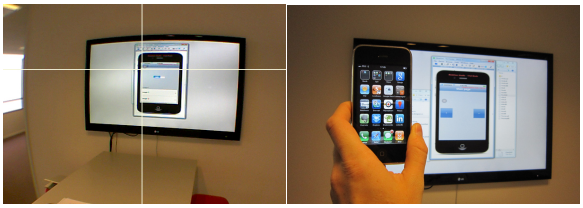


Figure 10 *Interaction with iphone emulator. Left image shows the scene image with the gaze point (white cross), and the right image shows the visual appearance of a real iphone and the emulator in the user's field of view are about the same.*

The second application was to interact with an iphone emulator using the head gestures. The application had 4 different pages with different buttons and list-boxes. User is able to press or select an item by looking at the item and performing the corresponding gesture that was showing on the items. (e.g., left gesture for the back button).

7 Experimental results

A classifier test has been done before the applications. In this section the results of the classifier test and the performance of the users during the iRecipe and iiPhone applications will be presented.

The classifier test has been conducted for testing the accuracy of the implemented algorithm on a head mounted eye tracker. Simple gestures introduced in section 6, were tested in the classifier test, however we restrict the repetitive gestures (Gii) to only the linear movements ($i \neq RorL$). 14 simple gestures were shown on the screen by a simple figure, two times one by one and randomly. The shown gesture remains on the screen until the user performs the same gesture or pressing a key in the case when the user was not able to perform that gesture.

8 participants (6 male and 2 female, mean=35.6, SD=9.7) are used in the experiments. 7 participants were unfamiliar with this method. The method and gestures were introduced to participants and they had the chance of practicing the gestures for 10 minutes before the experiments. The experiments on each participant lasted about 50 minutes.

In all the experiments, a 55" LG flat panel screen was used as a display. The users wearing a head mounted eye tracker were able to move around the screen during the task and at the same time interact with the screen.

The head mounted eye tracker with the accuracy of about 1° made by the authors was employed in the experiments. The eye tracker consists of two webcams, and both eye and scene images with the resolution of 640x480 are processed at 25 frames per second in real time. A feature-based method has been used for pupil detection, and a homography mapping has been use for gaze estimation.

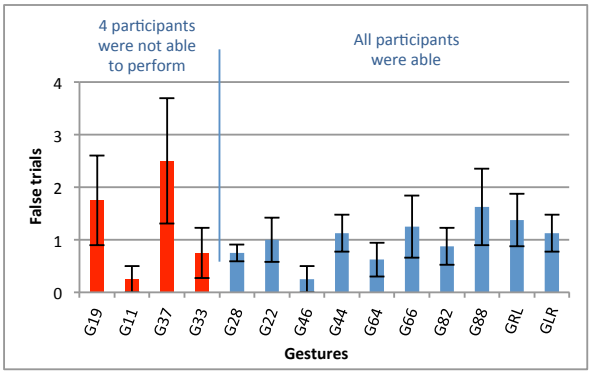


Figure 11 *Average number of false trials per each gesture. Error bars show the standard error of the mean.*

Figure 11 shows the results of the classifier test and the average number of false trials of all participants for each gesture. Each gesture has been shown 16 times (2 times per participant). Each time that a participant performs a gesture but it is not recognized correctly, it will considered as a false trial. Ideally the number of the false trials should be 0, it means that the participant only performs the gesture one time after displaying the gesture on the screen which is detected correctly by the classifier. 4 participants were not able to perform the diagonal gestures ($G19, G11, G37, G33$), and these gestures are shown at the left side of the graph indicated by the red color.

The results show that the diagonal gestures were difficult for some of the participants. Among the other linear gestures which were more convenient for the participants, sweep down gesture ($G28$) had a more average of false trials. It means that it was not easy for participants to turn the head down. Repetitive down gesture ($G88$) is even more inconvenient and has the highest number of false trials in the right side of the graph, and it is because of the user needs to divide the down movement into two steps.

In the classifier test, it was also observed that even smallest movements of the head (<2°) can be detected by the system which is not possible to detect by the other methods introduced in the previous work.

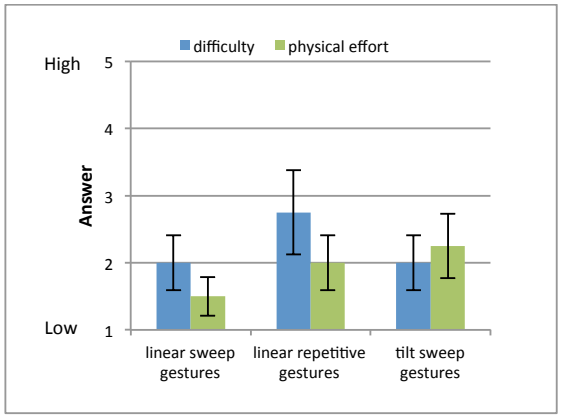


Figure 12 *Results of the questionnaire, showing the physical effort and the level of difficulty for three types of gestures*

After the test, the participants were given a questionnaire consists of questions with the range of the answers from 1 to 5 to investigate the participants experience in terms of physical effort and the level of difficulty. The participant answered each question for three different gestures. Figure 12 shows the results of the questionnaire.

In the classifier test, the target was only a marker on the screen and sometimes the participants were moving the gaze point together with the head movements meaning that the gesture was not correct. However, it has been observed that it is easier for the users to keep the gaze point fixed during the head movements, when the object is something that they want to control it by a gesture. For example, during the iRecipe and iiPhone applications, participants were trying to press the buttons or controlling some items on the screen, and the average number of false trials of the simple sweep gestures, was less than classifier test. The false detection may also occur when the classifier is not able to detect the head gestures, meaning that the classification method requires some improvements. In addition, some of the defined gestures were difficult to perform and the users need more practice in order to be able to look at an object and do the movements.

All the participants were able to control iRecipe and iiPhone applications and do the tasks by head gestures. Each application took approximately 10 min. Some of the participant found the volume control more convenient than controlling the other parts of the recipe application. This was because of the real time feedback of the interface, both by showing the volume gain number at the gazed object and by hearing the changes of the volume. It shows that the small visual changes in the gazed object (e.g., changing the color) during the head movements can help the user to keep focus on the object during the gesture. However, any change in the appearance of the gazed area that leads to loos the visual attention from the object should not be done during the gesture. Using the sound as feedback, can also be a good choice in some cases. For example for controlling the objects in the 3D environment, when the visual feedback is not possible, sound feedback can be used during the gesture whenever the system detects a basic head movement, or before the gesture just to show that the gazed object is ready for control.

The accuracy of the eye tracker allowed the user to interact comfortably with 4×2 regions on the mobile display. The size of the emulated display shown on the 55" screen was about the same visual angle as real iphone display (figure 10). 5 of the participants were already using the iPhone and it was so easy for them to interact with the emulator by the head gestures.

Even thought the used gestures in the applications were simple (double characters), no unintended command was observed during the tasks.

8 Discussion & Future work

A novel method for detecting the head gestures in combination with gaze was suggested and tested on a mobile eye tracker. The proposed method shows that head gestures can be measured through eye image based on vestibulo-ocular reflex and by having the gaze point. Many video based methods have been used so far for detecting the head gestures. In contrast, the presented method in this paper, allows for identifying a wide range of head gestures even the small gestures accurately and in real time, by only using an eye tracker.

Head gestures together with fixed gaze point can be used as a method for gaze based interaction by eye trackers instead of complex gaze gestures. It can be used when the user is able to slightly move the head. The main advantage of this method with compare to the gaze gestures is that the user does not lose the visual attention on the object during the interaction.

This method has been implemented on a head mounted eye tracker for detecting a set of 14 simple gestures and the algorithm was evaluated. The method was also tested on two applications one to show the capability of the method for interacting with a screen at kitchen during cooking and when the hand is occupied. The other application was to interact with an emulated iPhone. The results showed the possibility of this method for interaction with the screens and even small displays like the mobile devices.

Future work

We have already shown that the presented method can be used for interaction with screens. This method has also a high potential to be a direct way of communication and controlling the objects (looking at the objects and doing a simple gesture). As a future work, we are trying to use this method and the developed head mounted eye tracker for interaction and controlling the objects in the home environment.

The proposed method allows for very simple and intuitive way of interaction that can be used either with head mounted gaze trackers or remote gaze trackers. However, how to determine whether the gaze is fixed during the head gestures differs for remote and head-mounted eye trackers. We are trying to implement this method on a remote eye tracker, since the mobile devices are predicted to embed increasingly capable eye gaze tracking technology. Eye-based head gestures can be used alone or together with finger gestures for operating the interfaces of tablets and mobile phones. Most of the remote video-based eye trackers use the Pupil-Corneal Reflection (P-CR)" to determine the point of regard (PoR) [Hansen and JI 2010]. The light sources are always in the field of view of the eye and the reflections in the eye image provide a reference at the world reference frame. It makes the detection of fixed gaze easier in the remote eye trackers with compare to the head mounted eye trackers.

When for some reason, the hands cannot be used, (e.g. due to the object being too far away; the hands are already occupied with other things; the hands cannot be adequately controlled due to disease or impairment) or even when the hands are free, proposed method can be used as a fast way of interaction with objects. Besides, in some applications that loosing the visual attention may increase the human risk (e.g., driving the vehicles, driving the wheelchair or in the high risk environments like the power plants control rooms), eye-based head gestures can be used for interaction without requiring the users to look away from their usual viewpoints. It can also be a way to interact with head-up displays in the automobile or aircrafts.

9 References

Bee, N. and Andre, E. 2008. Writing with Your Eye: A Dwell Time Free Writing System Adapted to the Nature of Human Eye Gaze. *In Perception in Multimodal Dialogue Systems, LNCS 5078/2008*, Springer, pages 111 – 122.

Darwin, C. 1872/1998. *The Expression of the Emotions in Man and Animals*, third edition. New York, Oxford University Press.

Davis, J.W., and Vaks, S. 2001. A Perceptual User Interface for Recognizing Head Gesture Acknowledgements. *In Proceedings Workshop on Perceptive User Interfaces.*

Drewes, H. 2010.Eye Gaze Tracking for Human Computer Interaction. PhD thesis, Faculty of Mathematics, Computer Science and Statistics, LMU München.

Drewes, H., and Schmidt, A. 2007. Interacting with the Computer using Gaze Gestures. *In Proceedings of Human-Computer Interaction - INTERACT 2007*, Springer LNCS 4663, pages 475 – 488.

Ekman, P., and Friesen, W., 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto.

Gale A.G., 2005. Attention Responsive Technology and Ergonomics. *In Bust P.D. & McCabe P.T. (Eds.) Contemporary Ergonomics, Proceedings of the Ergonomics Society Annual Conference*, Hatfield University, Hertfordshire, pages 273-276.

Hansen, D.W., and Ji, Q., 2010. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. IEEE Trans. Pattern Anal. Mach. Intell, pages 478-500.

Hartley, R., and Zisserman, A. 2000. *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge, UK.

Isokoski, P. 2000. Text Input Methods for Eye Trackers Using Off-Screen Targets. *In Proceedings of the ACM symposium on Eye tracking research & applications ETRA '00*. ACM Press, pages 15 – 21.

Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., and Vickers, S.2010. Designing Gaze Gestures for Gaming: an Investigation of Performance. *In Proceedings of the ACM symposium on Eye tracking research & applications ETRA '10,* ACM Press, New York, NY March.

Kapoor, A., and Picard, R.W. 2002. A real-time head nod and shake detector. Technical Report 544, MIT Media Laboratory Affective Computing Group.

Kjeldsen, R. 2001. Head gestures for computer control. In *Proceedings Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real time Systems*, pages 62–67.

Mardanbegi, D., Hansen, D.W. 2011. Mobile gaze-based screen interaction in 3D environments, Proceedings of the 1st Conference on Novel Gaze-Controlled Applications (NGCA2011), Blekinge Institute of Technology, Karlskrona, Sweden.

Møllenbach, E. 2010. Selection strategies in gaze interaction. PhD thesis, Innovative communication group, IT University of Copenhagen.

Morris, D. 1994. *Body talk: The Meaning of Human Gestures*. Crown Publishers, New York.

Nonaka, H. 2003. Communication interface with eye-gaze and head gesture using successive DP matching and fuzzy inference.

Journal of Intelligent Information Systems 21(2): pages 105–112.

Panerai, F. and Sandini, G. 1998. Oculo-Motor Stabilization Reflexes: Integration of Inertial and Visual Information. Neural Networks, 11, pages 1191–1204.

Porta, M., and Turina, M. 2008. Eye-S: a Full-Screen Input Modality for Pure Eye-based Communication. *In Proceedings of the ACM symposium on Eye tracking research & applications ETRA '08*. ACM Press, pages 27 – 34.

Shi, F., Gale, A.G. & Purdy, K.J. 2006.Eye-centric ICT control. *In Bust P.D. & McCabe P.T. (Eds.) Contemporary Ergonomics, Proceedings of the Ergonomics Society Annual Conference*, pages 215-218.

Toyama, K. 1998. Look, ma–no hands! Hands free cursor control with real-time 3D face tracking. *In Proceedings Workshop on Perceptual User Interfaces (PUI'98)*, pages 49–54.

Wobbrock, J. O., Rubinstein, J., Sawyer, M., and Duchowski, A. T. 2007. Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures. *In Proceedings of COGAIN '07*, pages 61 – 64.

Zhai, S., Morimoto, C., and Ihde, S. 1999. Manual And Gaze Input Cascaded (MAGIC) Pointing. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '99*, ACM Press, pages 246 – 253.

- Chapter 7 -

Eye Based Head Gestures for Interaction In The Car

Eye-based head gestures for interaction in the car

Diako Mardanbegi
IT University of Copenhagen
Rued Langgaards Vej 7, DK-2300
Copenhagen S
+45 72 18 53 72
dima@itu.dk

Dan Witzner Hansen
IT University of Copenhagen
Rued Langgaards Vej 7, DK-2300
Copenhagen S
+45 72 18 50 88
witzner@itu.dk

ABSTRACT
In this paper we suggest using a new method for head gesture recognition in the automotive context. This method involves using only the eye tracker for measuring the head movements through the eye movements when the gaze point is fixed. It allows for identifying a wide range of head gestures that can be used as an alternative input in the multimodal interaction context. Two approaches are described for using this method for interaction with objects inside or outside the car. Some application examples are described where the discrete or continuous head movements in combination with the driver's visual attention can be used for controlling the objects inside the car.

Categories
H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces – Interaction styles (e.g., commands, menus, forms, direct manipulation).

General Terms
Human Factors; Measurement.

Keywords
Gaze tracking; Head gesture; Interaction; Eye movements

1. INTRODUCTION
In the last decades, automotive user interfaces have become more complex with much new functionality. Besides controlling the vehicle and operating the primary tasks (maneuvering the car e.g. controlling the speed or checking the distance to other cars), drivers need to interact with a variety of digital devices and applications in the car when driving. However, driver's focus on driving, is still the primary task, and should have the highest priority. The other tasks should be as minimally distracting as possible for the safety reasons [11]. New interaction techniques like speech, touch, gesture recognition, and also gaze have found their way to be used for interaction with user interfaces in a multifunctional space like car. This paper proposes using *eye-based head gestures* as a potential technique for interaction with automotive user interfaces. Eye-based head gesture [13] is a technique for recognizing head gestures. It uses the driver's gaze and eye tracking data for a) distinguishing the gestures from the natural head movements, b) for measuring the head gestures, and c) for using the driver's intention in interaction with objects.

Among the new interaction methods that have so far been studied in the automotive context, techniques like speech and head gestures have the advantage of providing a way for hands-free interaction. However, speech, and head gesture recognition often require a short explicit command like pushing a button before they can be used. Therefore, they can be used in multimodal interaction systems combined with the other input modes and help to minimize the amount of time that the driver's hand is off the steering wheel.

Associated level of physical, visual, and mental workload should be considered when designing a user interface and thinking about the interaction with an automotive user interface [3]. There have been some studies that report that certain kinds of voice-activated interfaces impose inappropriately high cognitive loads and can negatively affect driving performance [5, 6]. The main reason is that we are still far from achieving high-performance automatic speech recognition (ASR) systems. There are also some tasks like controlling radio volume, opening the window just slightly, continuously zoom or scrolling the map which are not intuitive operations to perform solely via speech-based interaction. Speech input cannot also be used when the environment is too noisy. In contrast, head gesture recognition is more reliable and can be a good alternative to speech input. Even if the number of different detected gestures is relatively small, they can be used as both continuous and discrete commands. Interaction by head gestures involves less driver's cognitive load as it can use the natural human communication skills. However the head gesture recognition has been mostly concentrated on detecting head shakes and nods to communicate approval or rejection and as an intuitive alternative in any kind of yes/no decision of system-initiated questions or option dialogs.

On the other hand, much work has been done in driver fatigue detection, and a fatigue monitoring device have been studied as a tool that allow for implicit interaction between the car and the driver to improve driving safety [16]. Eye and the visual behaviors measured by a video-based eye tracker provide significant information about driver's attention [14, 15] and the state of drowsiness and vigilance [18]. A video based eye tracker can also be used for recognizing head gestures using the eye and gaze information. It is possible to detect a wide range of head gestures as well as nods and shakes, which can be used for interaction. Head gestures can also be interpreted as different interaction commands by using the other modalities like gaze and intention proving an inferred interaction.

The paper is organized as follows. Some related works are described in the next section. Then, eye-based head gesture and the interaction method are described. Some application scenarios

AutomotiveUI'12, October 17-19, Portsmouth, NH, USA.
Adjunct Proceedings.

of using the method for interaction with objects in the car are described in a subsequent section and finally we conclude in the last section.

2. RELATED WORK

Many methods for gesture recognition have been proposed and some of them are applied to the automotive environment for detecting the head and hand gestures. Among the non video-based methods, an interesting work was done by Geiger [7], in which a field of infrared distance sensors is used to locate the hand and the head of the driver and sensing the movements. Although the sensor array does not achieve the resolution of a video-based methods, but his system is evaluated to be highly robust in measuring the simple directional gestures. Here, our focus is on the video-based methods for head gesture recognition. Many video-based techniques have been proposed for tracking the user's head and mostly are based on head/face detection and tracking. For example, Althoff [1], developed a system for detecting the head nod and shake using a near infrared imaging approach for interaction in the vehicle. In general, video-based techniques use some features of the face for detecting the head position in 2-D image space [12, 19], or some of them work by fitting a 3D model to the face in each image of the video to provide estimates of the 3D pose of the face [2]. However, these methods are not usually robust enough to strong illumination changes, and usually not accurate and fast enough to be useful for interactive environments.

On the other hands, some attempts have been made to use eye image for head gesture recognition. Concentrating on head gesture recognition methods that use the eye features, Davis and Vaks [4] presented a prototype perceptual user interface for a responsive dialog-box agent. They used IBM PupilCam technology for only detecting the eye location in the image and used together with anthropometric head and face measurements to detect the location of the user's face. A Finite State Machine incorporating the natural timings of the computed head motions was employed for recognition of head gestures (nod=yes, shake=no). Kapoor and Picard [10] introduced an infrared camera synchronized with infrared LEDs to detect the position of the pupils. Recognizing the head gestures had been demonstrated by tracking the eye position over time and a HMM based pattern analyzer was used detecting the nod/shake head gesture in real-time. However, their system used complex hardware and software and had problems with people wearing glasses and with earrings. The most relevant work to this paper is conducted by Ji and Yang [8, 9]. They have proposed a camera-based real-time prototype system for monitoring driver vigilance. An infrared imaging system and the bright/dark pupil effects (similar to PupilCam) is used for detecting the pupil position. They investigated the relationships between face orientation and these pupil features and so that the 3D face (head) pose have been estimated from a set of seven pupil features: inter-pupil distance, sizes of left and right pupils, intensities of left and right pupils, and ellipse ratios of left and right pupils. They have also estimated the driver's gaze and average eye closure speed having the eye images. However, their gaze estimation was limited into nine areas: frontal, left, right, up, down, upper left, upper right, lower left and lower right. Head movements were not measured accurately and what they were interested was to detect if the driver head deviates from its nominal position/orientation for an extended time or too frequently. The same idea for detecting the limited head movement and the rough gaze estimation using the eye images (with different methods) had been also presented before in [17].

3. EYE-BASED HEAD GESTURES

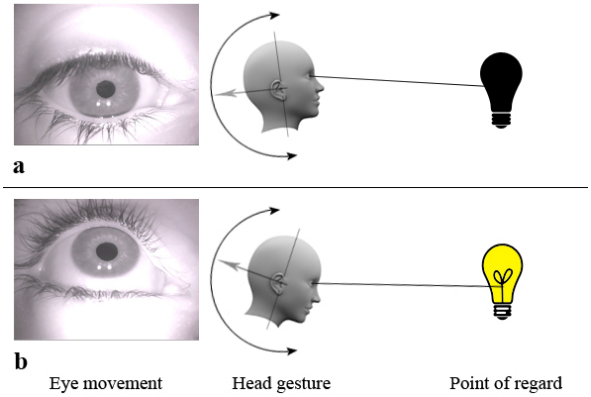


Figure 1: When the gaze point is fixed the head movements can be measured through the eye movements

Eye movements can be caused by the head movements while point of regard (PoR) is fixed or by changing the PoR when the head is fixed. When the point of regard is fixed and the head moves, the eyes move in the opposite direction and with the same speed as the head movement. These eye movements are due to the vestibulo-ocular reflexes (VOR), which are used to stabilize the image on the retina. Figure 1 illustrates a user looking at an object but in two different situations, one when the head is up (Figure 1.a) and the other when the head is down (Figure 1.b). The eye image is different in each posture even though the PoR is fixed. Since the eye trackers measure the eye movements and estimate the point of regard, they are able to measure the head movements when the PoR is fixed. In this paper, the term eye-based head gestures, denotes a predefined pattern of head movements measured through eye movements but where the PoR is fixed on a given object, and the term fixed-gaze target denotes the object that PoR is fixed on it. This method is able to measure a wide range of the head movements (including the head roll) and even though they are very small. The head roll can be detected by measuring the optic flow of the iris pattern and the yaw/pitch movements by tracking the pupil center. Figure 2 shows the basic roll, yaw and pitch movements of the head and the corresponding eye movements in the eye image.

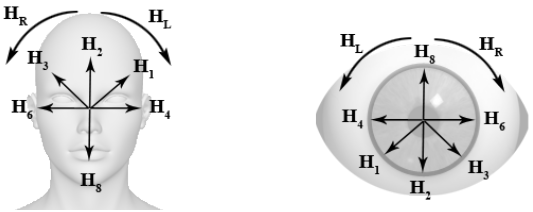


Figure 2: The basic head movements and their corresponding eye movements

This method is independent of the type of the eye tracker and where the data come from and it can be used for head gesture recognition whenever the gaze point and the eye image are available.

Head gestures together with fixed gaze point can be used as a method for gaze based interaction. A combination of fixed gaze and head gestures can be used for interaction with both the objects inside the car and also outside of the car. Two different methods

are presented in this section for interaction with objects inside or outside the car. The main reason of separating these two is that fixating the gaze on the objects inside the vehicle during performing the head gesture is not acceptable, and we are interested to minimize the amount of time that the driver's visual attention is away from the forward roadway.

3.1 Interaction with the roadway objects:

For interaction with the objects on the roadway (e.g. getting information about the signs), the driver can simply keep the gaze fixed on the object and then perform a gesture. The eye tracker will recognize the gazed object even though the object and the driver may have a relative movement. When the object has a velocity less than $15°s^{-1}$ in the field of view, the eyes have a slow movement called smooth pursuit. Above this speed the smooth pursuit will be accompanied by saccades. Therefore, these eye movements need to be differentiated from the eye movements caused by the head gestures according to their range of speed. However, in this case, the head rolls can be easily detected by measuring the iris torsion, and can be used as gestures.

3.2 Interaction with the objects inside the car:

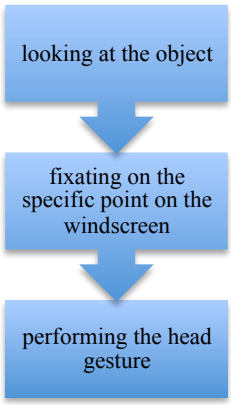


Figure 3: The main 3 steps for interacting with objects inside the vehicle

Interacting with the objects by looking at the object, fixating the gaze on the object, and then performing a head gestures can be useful for some tasks. However, when the task is more complex, this method would not be a safe approach for interaction (e.g. adjusting the side-view mirror in the car). With the method described below, we minimize the time that the gaze is away from the roadway by transferring the fixed-gaze target from a point on the object to a specified point on the windscreen. This point can be indicated by a small dot located on the windscreen in front of the driver. When the target is shown on the windscreen allows the driver to maintain attention to events happening on the road. Therefore, Interaction with the objects inside the car can be done by looking at the object, and then fixating the gaze on a specific point on the windscreen and performing the head gesture. This method uses the driver's visual attention as an implicit interaction modality, so that when the driver looks at an object in the car (e.g. the window) the eye tracker recognize that specific object and then waits for the next step. Once the user fixates on the specific point on the windscreen, the system waits for the user's head gesture for controlling the last intended object.

While performing the gesture, eye tracker measures the eye movements and tracks the gaze point. The distance between the windscreen target and the eye is basically less than 1 meter and therefore the driver's eyes converge during the gesture. The eye tracker can detect this convergence by measuring the distance between the two pupils. Therefore, the convergence of the eyes can be used as an indicator that the driver is performing a gesture.

4. APPLICATION SCENARIOS

Some example applications of using eye-based head gestures in the automotive context are described in this section.

Head gestures have a great potential to be used as an intuitive alternative in any kind of yes/no decision when a system initiated questions or option dialogs. As an example, when the mobile phone is ringing, the incoming calls can be accepted or denied by the head gestures. These simple vertical head gestures can also be used for flipping the rear-view mirror down or up.

The left and right head movements can be used for shortcut functions enabling the user to control the music player and to skip between individual cd-tracks or radio stations.

This method can also be used as a way for interacting between the driver and the head-up display (HUD), enabling the driver to do selecting and for switching between different submenus in a more intuitive way compared to standard button interactions.

Continuous vertical movements of the head can be useful for changing the volume, adjusting the air conditioning temperature, opening and closing the window, and continuously zoom or scrolling the map. In these examples, visual or audio feedback through HUD or speakers can help the driver to perform the task more efficiently. The visual feedback can be a highlight color or even displaying the image of the object. For example, when the driver wants to adjust the side-view mirrors, he/she looks at the mirror and then the eye tracker recognize the mirror as the attended object and then the system shows the real-time image of the mirror in the head-up display. Now, the driver can see the mirror image in front of the windscreen and therefore can easily adjust the mirror by the head movements.

5. Conclusion

In this paper, we suggested to use eye-based head gestures for interaction in the automobile. This method uses only the information extracted from the eye image for measuring the head movements. One of the advantages of this technique is that even very small head movements can be measured through the eye movements. Another advantage is that a video-based eye trackers can potentially be used as one multi-purpose device in the car for head gesture recognition as well as for fatigue detection, monitoring the driver's visual attention, and gaze estimation. Some example applications are described where the gaze and head gestures are used together for controlling some objects in the car. In general, whenever the head gestures are used so far in the automotive context, the new method for head gesture recognition can be applied, too.

6. REFERENCES

[1] Althoff, F., Lindl, R., and Walchshaeusl, L. Robust Multimodal Hand- and Head Gesture Recognition for controlling Automotive Infotainment Systems. In VDI-Tagung: Der Fahrer im 21. Jahrhundert, Braunschweig, Germany, November 22-23 2005.

[2] Black, M.J. and Yacoob, Y. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In ICCV95, pages 374–381, 1995.

[3] Burnett, G.E, Designing and evaluating in-car user-interfaces. In J. Lumsden (Ed.) Handbook of Research on User-Interface Design and Evaluation for Mobile Technology, Idea Group Inc. 2008.

[4] Davis, J.W., and Vaks, S. 2001. A Perceptual User Interface for Recognizing Head Gesture Acknowledgements. In Proceedings Workshop on Perceptive User Interfaces.

[5] Garay-Vega, L., A. Pradhan, G. Weinberg, B. Schmidt-Nielsen, B. Harsham, Y. Shen, G. Divekar, M. Romoser, M. Knodler, and D. Fisher. Evaluation of different speech and touch interfaces to in-vehicle music retrieval systems. Accident Analysis & Prevention, 42(3): 913–920, May 2010.

[6] Gartner, U., Konig, W., and Wittig, T. Evaluation of manual vs. speech input when using a driver information system in real traffic. In International driving symposium on human factors in driver assessment, training and vehicle design, 2001.

[7] Geiger, M. Beruhrungslose Bedienung von Infotainment-Systemen im Fahrzeug. PhD thesis, TU Mu¨nchen, 2003.

[8] Ji, Q., Yang, X. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance, in: Rea Time Imaging, pages 357-377, 2002.

[9] Ji, Q. and Bebis, G. “Visual Cues Extraction for Monitoring Driver’s Vigilance.” Procs. Honda Symposium, pp.48-55,1999.

[10] Kapoor, A., and Picard, R.W. 2002. A real-time head nod and shake detector. Technical Report 544, MIT Media Laboratory Affective Computing Group.

[11] Kern, D., Schmidt, A. Design space for driver-based automotive user interfaces. In Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI ’09). ACM Press (2009), 3-10.

[12] Kjeldsen, R. Head gestures for computer control. In Proc. Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, pages 62–67, 2001.

[13] Mardanbegi, D., Hansen, D.W., and Pederson, T. “Eye-based head gestures: Head gestures through eye movements”. Proceedings of the ACM symposium on Eye tracking research & applications ETRA '12, ACM Press, California, USA, 2012.

[14] Pomarjanschi, L., Dorr, M., and Barth, E. Gaze guidance reduces the number of collisions with pedestrians in a driving simulator. ACM Transactions on Interactive Intelligent Systems, 1(2):8:1-8:14, 2012.

[15] Pomarjanschi, L., Rasche C., Dorr M., Vig E., Barth E. 2010, "Safer driving with gaze guidance" Perception 39 ECVP Abstract Supplement, page 83.

[16] Schmidt A (2000) Implicit human computer interaction through context. Pers Ubiquit Comput 4(2):191–199.

[17] Smith, P., Shah, M., and da Vitoria Lobo, N. "Monitoring. Head/Eye Motion for Driver Alertness with One Camera', The. Fifteenth IEEE ICPR. Nov. 2000.

[18] Wang, Q., Yang, J., Ren, M., Zheng, Y. “Driver Fatigue Detection: A Survey”, in Proc Intelligent control and Automation, Dalion, China, pp 8587- 8591, 2006.

[19] Wren, C.R., Azarbayejani, A., Darrell, T.J., and Pentland, A.P. Pfinder: Real-time tracking of the human body. PAMI, 19(7):780–785, July 1997.

- Chapter 8 -

Parallax Error In The Monocular Head-Mounted Eye Trackers

Parallax error in the monocular head-mounted eye trackers

Diako Mardanbegi
IT University of Copenhagen
Rued Langgaards Vej 7,
DK-2300 Copenhagen S
dima@itu.dk

Dan Witzner Hansen
IT University of Copenhagen
Rued Langgaards Vej 7,
DK-2300 Copenhagen S
Witzner@itu.dk

ABSTRACT
This paper investigates the parallax error, which is a common problem of many video-based monocular mobile gaze trackers. The parallax error is defined and described using the epipolar geometry in a stereo camera setup. The main parameters that change the error are introduced and it is shown how each parameter affects the error. The optimum distribution of the error (magnitude and direction) in the field of view varies for different applications. However, the results can be used for finding the optimum parameters that are needed for designing a head-mounted gaze tracker. It has been shown that the difference between the visual and optical axes does not have a significant effect on the parallax error, and the epipolar geometry can be used for describing the parallax error in the HMGT.

Author Keywords
Head-mounted gaze tracker, Parallax error, Mobile gaze tracker, epipolar geometry

ACM Classification Keywords
I.4.1 Image processing and computer vision: Digitization and Image Capture

General Terms
Measurement, Performance

INTRODUCTION
Head mounted gaze trackers (HMGT) are used for estimating the PoR in the user's field of view and are widely used for diagnostic applications. They have also been used for interaction in virtual [4, 5] or real [3, 6] environments. Head mounted gaze trackers have a scene camera for capturing the scene and another camera for capturing the eye image. HMGT is also called mobile gaze tracker because it is mounted on the user's head and can be used when the user is fully mobile. HMGT can potentially obtain a high degree of flexibility and mobility. However, most of the HMGT systems do not still allow for estimating the gaze point accurately in wide range of distances. A common problem with Head-mounted gaze trackers is that they introduce gaze estimation errors (a.k.a. *parallax error*) when the distance between the point of regard and the user (fixation distance) is different than when the system was calibrated. This error is due to the scene camera and the eye are not co-axial. Parallax error limits the use of head-mounted gaze trackers into a certain range of depth (a.k.a. *effective depth*).

There is a physical solution for removing the parallax between the scene camera and the eye. When the projection center of the scene camera coincides with the eyeball center, there is no parallax error. This can be done by using a visor (half mirror) in front of the eye and transferring the field of view of the eye to the scene camera. Head-mounted gaze trackers that do not have the scene camera mounted co-axial with the eye, require an indirect way of compensating for the parallax error. In order to be able to compensate for the parallax error, it important to know more about the error behavior and the main parameters that change the error. There are two main questions here: first, how do the scene camera orientation and position influence the parallax error? And second, with a fixed camera configuration, how does changing the calibration and fixation distances change the error? This paper investigates the answers of these two questions. The answer of the first question helps for having a better and optimum design for the head-mounted gaze trackers that have less error in gaze estimation. The answer of the second question helps for understanding the behavior of the parallax error when the fixation distance changes. It may help for estimating a function that calculates the parallax error given the fixation distance, which can be used for compensating for the error.

The remainder of this paper is organized as follows. Some related works about the parallax error in HMGT are briefly mentioned in the next section. We then introduce

UbiComp' 12, Sep 5 – Sep 8, 2012, Pittsburgh, USA.

the parallax error and describe the HMGT setup as a stereo camera setup. Then we describe in details how to calculate the parallax error in HMGT. Then we present the results of calculating the parallax error in the image and fixation planes with a summary.

PREVIOUS WORK

Velez et.al [7] at 1988 introduced a method for direct compensating for parallax error in the head-mounted eye trackers, using a transparent visor in front of the eye, which reflects the eye image towards the eye camera and the scene image towards the scene camera making a parallax free scene camera configuration. This method is a direct way for eliminating the parallax error and the eye tracker works quite accurate for different depths using only one time calibration. However, not all the head-mounted gaze trackers have such design today. Li [2], investigated the parallax error behavior in a simplified model of a HMGT, where the scene camera is mounted above the eye (only a vertical displacement). The angle between the visual and optical axis was not also considered in the analysis.

PARALLAX ERROR

The problem of parallax error can be simplified into two dimensions, which is visualized in Figure 1.

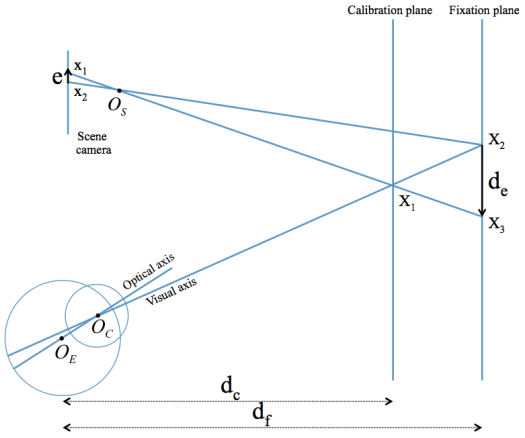


Figure 1: 2D parallax error

Suppose that the center of rotation of the eye is the point O_E, and the point O_C, is the center of the cornea where the visual and optical axes of the eye intersect. The scene camera is shown as a pinhole camera with a vertical image plane. Suppose that the system is calibrated for a plane (calibration plane) at a given distance d_c and the user fixates on a plane (fixation plane) at a further distance d_f. The visual axis of the eye intersects the calibration plane at the point X_1 and the fixation plane at the point X_2. The projections of these two points are not coincident on the image plane. When the user is looking at the point X_1 in the calibration plane, the estimated gaze point on the scene image would be the point x_1. When the user is looking at the point X_2, the visual axes and subsequently the eye image would be the same as for point X_1 and therefore the estimated gaze point would be the same point as x_1. The projection of the gaze point X_2, is the point x_2, however since the eye image has not been changed[1], the gaze tracker cannot compensate for this error. The parallax error can be defined as a vector in the scene image (x_1-x_2), which is corresponding to the vector X_3-X_2 in the fixation plane.

The relationship between the parallax error and the geometry of the system can be described in the general condition by epipolar geometry in a stereo camera system. Figure 2 shows a scene camera mounted on the head modeled as a pinhole camera with an optical center located at the point O_S and the focal length of f. The general transformation matrix $[R|t]$ represents the translation (t) and orientation (R) of the camera coordinate system relative to the fixed coordinate system. The fixed head coordinate system (X_E,Y_E,Z_E) is a right-handed 3D cartesian coordinate system located at the center of the eyeball (O_E), such that the Z_E axis is pointing forward, X_E is pointing to the left and Y_E is upward. This coordinates system, is considered as the fixed world coordinates system. Both calibration and fixation planes are assumed to be two planar surfaces in front of the head and parallel to the anatomical frontal plane of the body (X_E-Y_E plane).

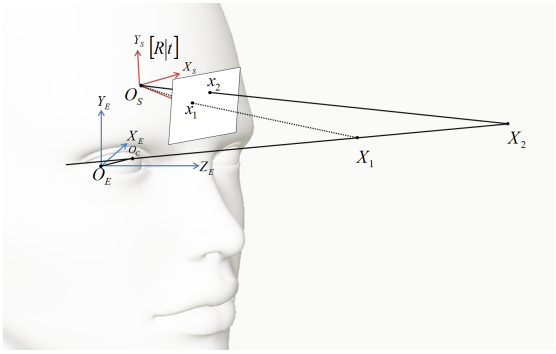


Figure 2: General configuration

Transformation from the world coordinate into the camera coordinate system can be done by the scene camera matrix which can be defined as the matrix C=K[R|t] where R and t are the external parameters, and the matrix K is the internal parameters of the camera. For a normal CCD camera with the focal length of f (in meter) and the principal point at the center of the image, the matrix K can be described by:

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

[1] The lens thickness would be different for points X_1 and X_2 as its focusing distance varies. However, it cannot be observed by a regular camera.

Assuming that the scene camera has a translation of ${}^{E}_{S}\mathrm{Tr} = (tx, ty, tz)$ relative to the fixed coordinate frame and three rotations around the fixed axes, the scene camera can be described by:

$$ {}^{E}_{S}T = \begin{bmatrix} {}^{E}_{S}\mathrm{R} & {}^{E}_{S}\mathrm{Tr} \\ 0 & 1 \end{bmatrix} \qquad (2) $$

Where ${}^{E}_{S}\mathrm{R}$ and ${}^{E}_{S}\mathrm{T}$ define the camera coordinate frame (S) relative to the fixed coordinate frame (E) in the homogeneous form. The rotation matrix ${}^{E}_{S}\mathrm{R}$ is the multiplication of three rotations:

$$ {}^{E}_{S}\mathrm{R} = \mathrm{RZ}(\gamma_z)\mathrm{RY}(\gamma_y)\mathrm{RX}(\gamma_x) \qquad (3) $$

where $\mathrm{RX}(\gamma_x)$ is the rotation by γ_x around the $\mathrm{X_E}$ axis, $\mathrm{RY}(\gamma_y)$ is the rotation by γ_y around the $\mathrm{Y_E}$ axis, and $\mathrm{RZ}(\gamma_z)$ is the rotation by γ_z around the $\mathrm{Z_E}$ axis. For simplicity, in the following of this paper the orientation of the camera is shown by the angles as R= (γ_x, γ_y, γ_z). The external parameters of the camera can be calculated by:

$$ [\mathrm{R|t}] = {}^{E}_{S}T^{-1} = \begin{bmatrix} {}^{E}_{S}\mathrm{R}^{\mathrm{T}} & -{}^{E}_{S}\mathrm{R}^{\mathrm{T}}\,{}^{E}_{S}\mathrm{Tr} \\ 0 & 1 \end{bmatrix} \qquad (4) $$

Knowing the camera matrix (C=K[R|t]), we can project any point in the field of view (X) into the camera image by multiplying the point by the camera matrix (x=CX). The parallax error is defined as the vector x_1-x_2 which is the projection of vector X_1-X_2. The parallax error may be different for each point in the fixation plane, and for each point the error can be considered as a function of the calibration distance (dc), geometrical parameters of the camera (R, Tr, f) and the coordinates of that point in the fixation distance (xf,yf,df).

When the fixation point (X) goes further away from the calibration plane, the projection (x) moves along a line in the scene image called epipolar line. If we assume that the point of regard is along the optical axis, then the eye and scene camera can be considered as a stereo setup. The projections of the points of an optical axis onto the scene image are all along a line called epipolar line. Changing the angle of the optical axis change the epipolar line, however, all epipolar lines intersect at a point called the epipole, which is the projection of the eyeball center into image plane. Considering the difference between the optical and visual axes and the fact that the point of regard is along the visual axis, the result would be slightly different.

In this paper, the displacement of the fovea from the optical axis is taken into account and the visual axis has been used instead of the optical axis.

CALCULATING THE PARALLAX ERROR

For calculating the parallax error in the image plane, we choose a point in the scene image (e.g. x_2), and find the visual axis passes through the point X_2 of the fixation plane. Then, the error vector can be calculated by having the projection of the point X_1, which is the intersection of the visual axis and the calibration plane. In order to find the visual axis, first the selected point x_2 on the image is back-projected on the fixation plane:

$$ X_2 = \begin{bmatrix} x_f \\ y_f \\ df \end{bmatrix} = C^{-1}x_2 \qquad (5) $$

Then the visual axis can be calculated as a line that passes through the points O_C and X_2. Figure 3 shows the points X_1 and X_2 in the fixed coordinate system. The eye rotates around the center of the eyeball (O_E) and it changes the direction of the optical axis (O_E-O_C). The visual axis intersects the optical axis at the center of the cornea (O_C), which is also the nodal point of the eye. The orientation of the optical axis can be described by the horizontal (pan) angle θ and the vertical (tilt) angle ϕ. The point O_C can be described by these angles as below:

$$ O_C = d \begin{bmatrix} cos\varphi\ sin\theta \\ sin\varphi \\ cos\varphi\ cos\theta \end{bmatrix} \qquad (6) $$

where the parameter d is the distance between the center of cornea and the center of eyeball (O_E).

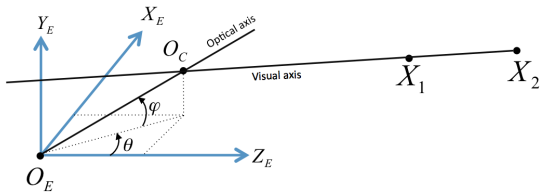


Figure 3: Showing the visual and optical axes in the fixed coordinate frame

The orientation of the visual axis can be expressed by the pan angle θ+α and the tilt angle φ+β where the α and β are the horizontal and vertical angles between the visual and optical axes. Therefore, any point on the visual axis can be expressed by:

$$ \mathrm{X} = O_c + k \begin{bmatrix} \cos(\varphi+\beta)\sin(\theta+\alpha) \\ sin(\varphi+\beta) \\ cos(\varphi+\beta)\ cos(\theta+\alpha) \end{bmatrix} \qquad (7) $$

Where the scalar k defines the distance from the point O_C.

Given the known point X_2, the three unknown parameters (φ, θ, and k) of the equation (7) can be obtained, and by knowing these parameters, we can calculate the point X_1, which is on the calibration distance (dc). Finally, the point x_1 can be obtained by projecting the intersection of the visual axis and the calibration plane (X_1).

The parallax error can be both represented as a vector in the scene image (x_1-x_2), or as a vector in the fixation plane (X_3-X_1). The error in the fixation plane can be obtained by having the point X_1 and the point X_3 (figure 1) which can be obtained by back-projecting the point x_1 onto the fixation plane.

In the next two sections, the parallax error has been calculated for different camera positions and distances and the simulation has been performed based on the equations above.

ERROR IN THE IMAGE PLANE

In this section, we measure the parallax error for different points of the scene image and it is shown how the angle and magnitude of the error vector will be influenced by changing the calibration and fixation distances, and also the camera position and orientation. It has been observed that by considering the visual axis, the epipolar lines do not intersect in exactly one point, however, there is not a significant difference in the overall distribution of the error directions in the image. In order to provide a better understanding of distribution of the error in the scene image, the error is measured in meter in the image plane instead of visual angle. However, in the next section when the error is presented in the fixation plane, it has also been measured in visual degree. Therefore, the unit meter is used for the focal length (f) in this section, and wherever the error is measured in the image, it will be in meter. If the focal length of the camera is known in pixel (f_{Pixel}), the error can be obtained in pixel by multiplying the error value to f_{Pixel}/f.

We start with the vertical translation of the camera (*ty*), and show the error changes by changing the fixation and calibration distances. Then we investigate the other transformations of the camera. The typical values of the eye parameters (α=±5°, β=1.5°[1], d=5.3mm[9]) have been used for calculation in the following, and all the calculations are done for the right eye (α=+5°). The range of the eyeball rotation ~70°×~70°[8] has been used to define the user's field of view and the size of the fixation plane in different distances, however, this angle is not used in practice, and the actual range of the eye movements is less than 50°×50°.

Figure 4 shows the magnitude of the parallax error in the center of the scene image for three different calibration distances (dc= 1.5, 3 and 5m), and the fixation distance from 0.4m to 10m. The camera parameters are R=(0,0,0), Tr=(0,0.05m,0) and f=0.005m. The field of view of the camera is considered to be 50°.

It can be seen that the parallax error is zero when the fixated and calibrated distances are equal and then increases as they diverge. The parallax error is larger for the closer distances and rises a bit faster as the fixation distance falls behind the calibration distance.

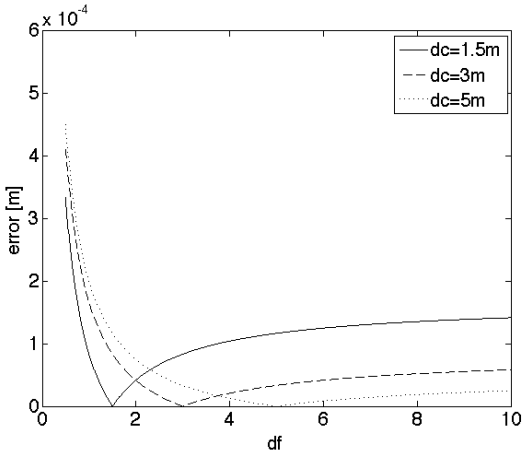


Figure 4: Changes in parallax error by changing the fixation distance

Figure 4, can give us an idea about how to choose the calibration distance when the gaze tracker is supposed to be used in a certain distances. For example for the range of 2m-10m, the calibration distance around 5m results less average error in the range of use. The error shown in the figure 4 is almost the same for all points in the scene image. The small variance has been observed for different points which is because of the angle between the visual and optical axes and is not significant. Generally, in a stereo setup, when the camera images are parallel to the baseline, the epipoles in the images are at infinity. When the epipole in the scene image is at infinity, the parallax error (magnitude and direction) is the same for all the points in the image. Therefore, by translating the camera horizontally or vertically (*tx,ty*) or rotating the camera around its optical axis the parallax error would still be the same for all the points in the image and can be described by one vector.

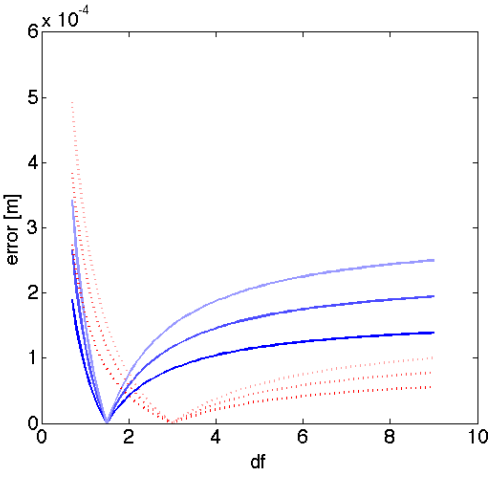


Figure 5: The effects of vertical and horizontal translations of the camera on the parallax error

Increasing the vertical distance between the camera and the eye, increase the level of the error curve shown in figure 4. Figure 5 shows these changes for two calibration

distances 1.5m (blue curves) and 3m (red dotted curves). Three different curves can be seen for each calibration distance. The curves with the lower levels are for the ty=0.05m, the curves in the middle are for the ty=0.07m and the upper curves are for ty=0.09m.

In general, the parallax error can be shown as a function of both calibration distance and fixation distance (figure 6), when the image plane in parallel to the fixation plane and the error is the same for all points in the image.

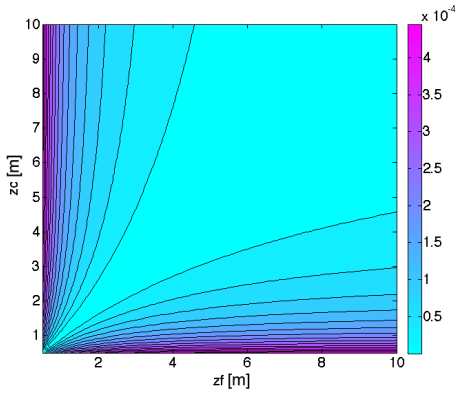


Figure 6: 2D error diagram for showing the error changes by changing the calibration and fixation distances

The results shown above for the magnitude of the parallax error, are the same when the camera translation is along the X_E axis instead of Y_E. The only difference is the changes in the direction of the error vectors. However, difference between the visual and optical axes, makes small differences in the magnitude of the error within the image, but is not significant. Figure 7, shows the vector field of the parallax error in the scene image for camera translations of Tr={(0,0.05m,0), (-0.05m,0,0), (0.05m,0,0), (0.05m,0.05m,0)} with the calibration distance of *dc*=2m and fixation distance of *df*=0.5m, without considering the difference between optical and visual axes.

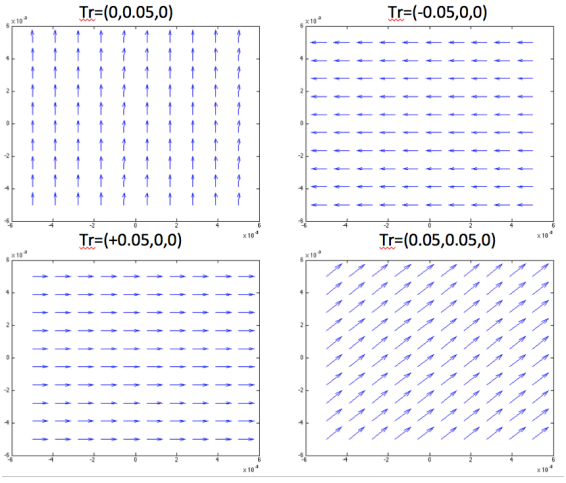


Figure 7: The vector field of the error in the scene image when the camera has the vertical and horizontal translations

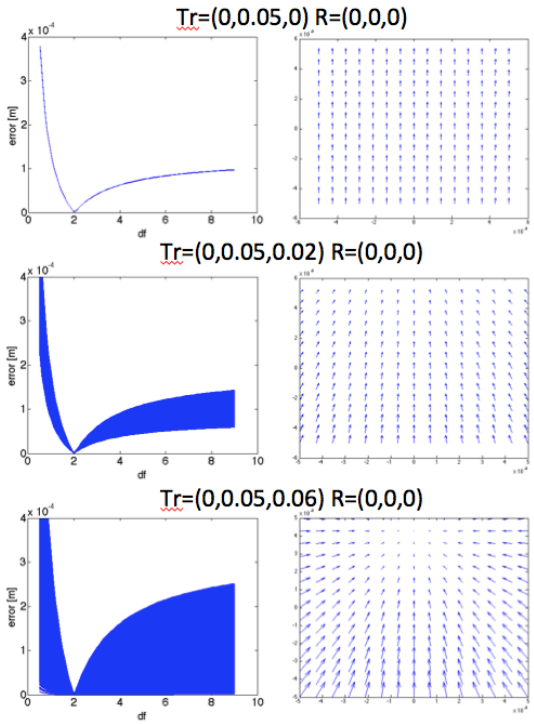


Figure 8: The effects of moving the camera along the Z-axis on the error

Translating the camera in the *Z* direction, moves the epipoles from the infinity toward the center of the image, and it changes the uniformity of the error within the image. It increases the error in some points and decreases the error in some other points. When the epipole is inside the image, the parallax error is zero for the epipole. Generally when the epipole is not at infinity, the *tx* and *ty* move the epipole horizontally and vertically respectively.

Figure 8 shows the parallax error for three different camera positions of Tr={(0,0.05m,0), (0,0.05m,0.02m), (0,0.05m,0.06m)} when the calibration distance is 2m. The graphs on the left side, show the changes in error magnitude for the different points of the image when the fixation distance is changing. It can be seen in this figure that how moving the camera in the Z direction changes the upper and lower bound of the error in the image. The vector field of the error for the fixation distance of 0.8m has been also shown in the right side.

Regarding the camera rotation, we show the effect of two important rotations pan and tilt. Usually the scene cameras do not have the roll rotation. When the roll angle is zero the pan (horizontal) rotation and tilt (vertical) rotation move the epipole horizontally and vertically respectively. It means that for example when the translation tz moves the epipole from the infinity to the center, the rotation Rx can translate it back again to the

infinity. Therefore, direction of the error vectors would be the same but their magnitudes are different. Figure 9 shows the error for the camera with a vertical rotation of $\gamma_x = 20°$ and a translation Tr=(0,0.05m,0.02m). It can be compared to the second row of the figure 8.

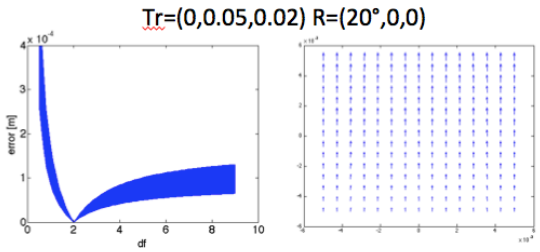


Figure 9: Moving the epipole to infinity by tilting the camera

As it can be seen in figure 9, the direction of the error vectors is uniform in the image but the range of the error size has not been changed too much after the rotation.

ERROR IN THE FIXATION PLANE

Sometimes it is useful to know the size of the corresponding error in the fixation plane. Figure 10 shows the magnitude of the parallax error in the fixation plane with the same configuration as for figure 4 and three different calibration distances (dc= 1.5, 3 and 5m). Figure 11 shows the error in visual angle for different points in the fixation plane when dc=3 and df=6.

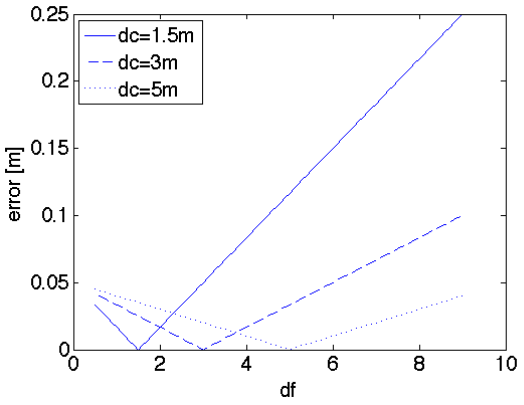


Figure 10: The actual error size in the fixation plane

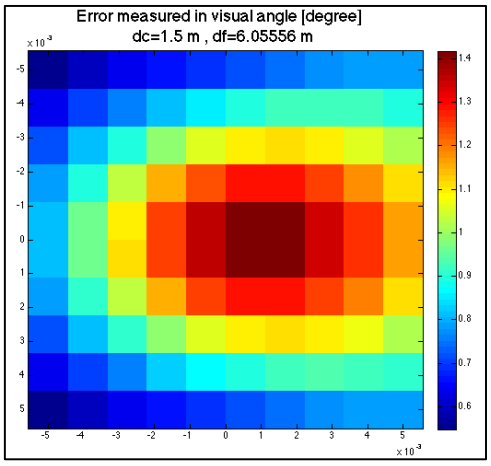


Figure 11: Error in a fixation plane in visual angle

CONCLUSION

In this paper, the parallax error in the head-mounted gaze trackers has been defined and described using the epipolar geometry in a stereo camera setup. The effect of changing the calibration and fixation distances on the parallax error has been investigated. It has been shown that the effective range of the gaze estimation with less parallax error is larger when the distance between the user and calibration plane is larger. The changes in the parallax error for different positions of the scene camera have been investigated. Camera translation and rotations relative to the eye, change the distribution of the error size and the direction of the error vectors in the image. The optimum configuration can be chosen based on the method that will be applied for compensating for the parallax error. It has also been shown that the difference between the visual and optical axes does not have a significant effect on the parallax error.

REFERENCES

1. Gale, A. G. "A note on the remote oculometer technique for recording eye movements," *Vis. Res.*, vol. 22, no. 1, pp. 201–202, 1982.
2. Li., D. Low-Cost Eye-Tracking for Human Computer Interaction. Master's thesis, Iowa State University, Ames, IA., Techreport TAMU-88-010, 2006.
3. Mardanbegi, D., and Hansen, D.W. Mobile gaze-based screen interaction in 3D environments. In Proc. Novel Gaze-Controlled Applications (NGCA '11). Blekinge Institute of Technology, Karlskrona, Sweden, 2011.
4. Nilsson, S., Gustafsson, T., Carleberg, P.: Hands Free Interaction with Virtual Information in a Real Environment. In: Proc. COGAIN 2007, Leicester, UK, pp. 53–57, 2007.
5. Park, H. M., Lee, S. H., Choi, J.S. Wearable augmented reality system using gaze interaction, Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, p.175-176, September 15-18, 2008.

6. Shi, F., Gale, A.G. & Purdy, K.J. Eye-centric ICT control. In Bust P.D. & McCabe P.T. (Eds.) Contemporary Ergonomics, 215-218, 2006.
7. Valez, J., Borah, J. D. "Visor and camera providing a parallax-free field of view image for a head-mounted eye movement measurement system." U.S. Patent 4 852 988, Aug. 1, 1989.
8. Wandell, B. A. *Foundations of vision.* Sinauer Associates Inc; USA, 1 edition, 1995.
9. Young, L. R., and Sheena, D. "Methods and designs—survey of eye movement recording methods," *Behav. Res. Meth. Instrum.*, vol. 7, no. 5, pp. 397–429, 1975.

- Chapter 9 -

Real-Time Compensation for Parallax Error

(TO BE SUBMITTED)

Real-time compensation for parallax error in head-mounted gaze trackers

Diako Mardanbegi
IT University of Copenhagen
Rued Langgaards Vej 7,
DK-2300 Copenhagen S
dima@itu.dk

Dan Witzner Hansen
IT University of Copenhagen
Rued Langgaards Vej 7,
DK-2300 Copenhagen S
Witzner@itu.dk

ABSTRACT
A new method has been presented for correcting the parallax error in head-mounted gaze trackers. The method estimates the parallax error for each point in the scene image using data samples collected prior to use of the gaze tracker. The paper shows that the same data samples can be reused for when the gaze tracker is calibrated for different distances. It has also been shown that the same data can be reused for different subjects when the scene camera position relative to the eye does not change significantly. The main assumption of the method is that the distance between the eye and the fixation point in space is known and for example can be obtained through the scene image.

Categories and Subject Descriptors
I.4.1 [**Image processing and computer vision**]: Digitization and Image Capture

General Terms
Measurement, Performance

Keywords
Head-mounted gaze tracker, Parallax error, Gaze estimation.

1. INTRODUCTION
A monocular head-mounted gaze tracker (HMGT) that has a scene camera, uses a function for mapping eye features extracted from the eye camera to a point in the scene image indicating the point of regard (PoR). In order to find the mapping function, a calibration procedure is needed prior to using the system. While calibrating the gaze tracker, the user is asked to look at certain points on a fronto-parallel plane in a certain distance (calibration plane). These types of HMGTs have a common problem of introducing gaze estimation errors when the distance between the point of regard and the user is different from when the system was calibrated. This gaze estimation error is called "Parallax Error" and it is due to the scene camera and the eye are not co-axial. Because of this error HMGTs can only estimate the gaze point accurately in a limited range of distance. In practice, the effective range of the gaze estimation (with less parallax error) is larger when the distance between the user and calibration plane is larger [2] (Figure 1).

The standard method for dealing with parallax error is to calibrate the gaze tracker for a finite set of distances prior to using it, and then apply the proper mapping function for gaze estimation in different distances. Therefore, the distance of the fixation plane (the working plane containing fixation points while using the system) should be set manually in the software before gaze estimation. The approach is therefore most appropriate for off-line gaze analysis.

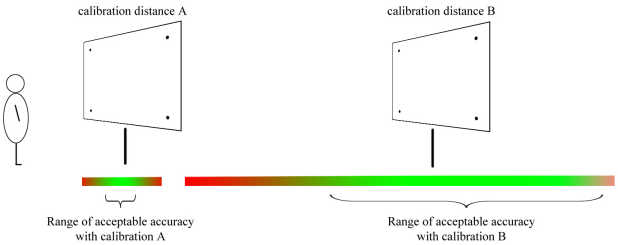


Figure 1. Limited range of use of HMGT for different calibration distances

This paper, presents an interpolation method for real-time compensation for the parallax error. Instead of using different mapping functions for gaze estimation, the same mapping function will be used for different distances and the error will be corrected for each gaze point by estimating the compensation value. The presented method estimates the error for any point in space based on a prior knowledge about the behavior of the parallax error and by knowing the distance from the fixation plane. During a "depth calibration" that it needs to be done prior to using the system, a set of sample data is collected from different distances. This data will be used later for estimating the compensation values. The distance between the user and the fixation plane is measured automatically through the scene image of the calibrated scene camera and having some extra information about the scene. In many gaze tracking applications, it is possible to estimate the distance from the fixation planes by detecting some items in the plane such as: visual markers that are used for recognizing the objects, infrared tags that are used for detecting the fixation plane [e.g., Tobii Glasses [5]), or the computer display that the user is interacting with [3].

The parallax error is described in Sec. 2. The possibility of measuring the fixation depth in HMGTs is discussed in Sec. 3. The real-time compensation method and the results of the simulation are presented in sections 5. Section 0 includes the conclusion and the future work.

2. Parallax Error
Figure 2 shows the parallax error in a general configuration of the system. When the HMGT is calibrated for a distance (dc) and the user's point of regard ($X2$) is in a distance closer or further than the calibration distance, the estimated gaze point in the scene image ($x2$) will have an offset from the actual gaze point ($x1$). The vector between the estimated gaze point and the actual gaze point in the scene image (e) is defined as parallax error.

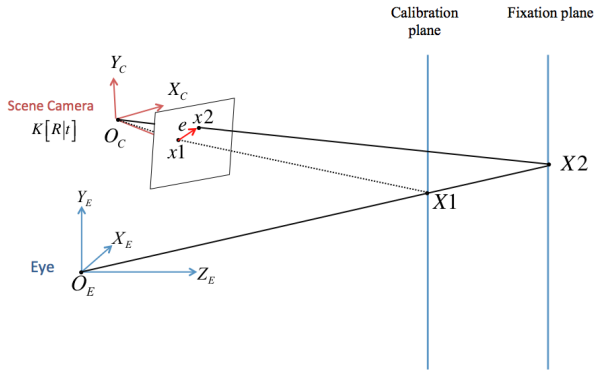


Figure 2. Parallax error in a geometrical model of a HMGT

This error changes by changing the fixation distance. Depending on the configuration of the scene camera, error may also be different for different gaze directions (non-uniform error inside the scene image). [2] shows that difference between the visual and optical axes of the eye does not have a significant affect on the parallax error. Therefore, eye can be considered as a pinhole camera and the fovea displacement can be ignored when describing the parallax error in HMGTs.

A full description of the parallax error using the epipolar geometry in a stereo camera setup has been presented in [2, 1]. In the following, the relationship between the parallax error and the geometry of the system (general configuration) has been described as a function that allows us to investigate the functional features and behavior of the error.

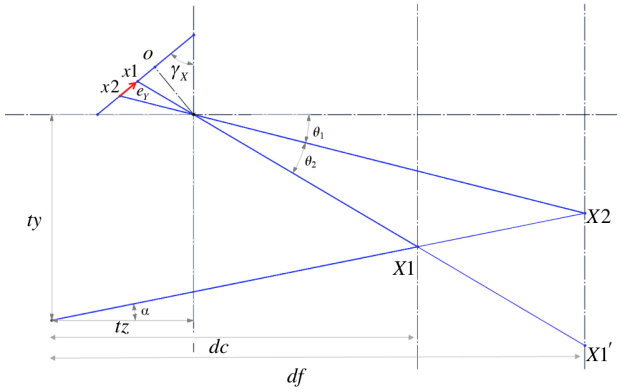


Figure 3. Vertical component of the error in the side view

Figure 3 shows the side view (along the X-axis) of the system shown in the Figure 2. All the measurements are relative to the center of the eyeball. The scene camera is modeled as a pinhole camera with the focal length of f and the principal point at the center of the image. It has rotations ($^{E}_{C}R$) around its axes X_C, Y_C, Z_C with angles of $\gamma_z, \gamma_y, \gamma_x$, and a translation of $^{E}_{C}Tr = [tx, ty, tz]$ relative to the eye center. The camera matrix can be described as:

$$P = K[R|t] = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{E}_{C}R^{T} & -{}^{E}_{C}R^{T}\,{}^{E}_{C}Tr \\ 0 & 1 \end{bmatrix} \quad (1)$$

The vertical component of the parallax error in the image plane is defined as $e_Y = x1 - x2$ where $x1$ and $x2$ are defined as:

$$x1 = f.tan(\gamma_x - (\theta_1 + \theta_2)) \quad (2)$$

$$x2 = f.tan(\gamma_x - \theta_1) \quad (3)$$

The two angles θ_1 and θ_2 can be obtained from the other known parameters as below:

$$tan(\theta_1 + \theta_2) = \frac{ty - (dc + tz) \cdot tan(\alpha)}{dc} \quad (4)$$

$$tan(\theta_1) = \frac{ty - (df + tz) \cdot tan(\alpha)}{df} \quad (5)$$

Where α is the vertical angle of the gaze. This angle can be obtained from the point of regard $X2 = [X2_x \quad X2_y \quad df]^T$ which is the actual gaze point in space:

$$\tan(\alpha) = \frac{X2_y}{df} \quad (6)$$

Therefore, the vertical component of error in the image can be expressed as (From Eq. 1-5):

$$e_Y = f.\tan\left(\gamma_x - \tan^{-1}\left(\frac{ty - (dc + tz) \cdot \left(\frac{X2_y}{df}\right)}{dc}\right)\right) - f.\tan\left(\gamma_x - \tan^{-1}\left(\frac{ty - (df + tz) \cdot \left(\frac{X2_y}{df}\right)}{df}\right)\right) \quad (7)$$

Expression 7 shows the relationship between the vertical component of the error in the image, the geometrical parameters of the system, and the calibration and fixation distances.

Likewise, the horizontal component of the error can be obtained from the top view of the Figure 2, as below:

$$e_X = f.\tan\left(-\gamma_y - \tan^{-1}\left(\frac{tx - (dc + tz) \cdot \left(\frac{X2_x}{df}\right)}{dc}\right)\right) - f.\tan\left(-\gamma_y - \tan^{-1}\left(\frac{tx - (df + tz) \cdot \left(\frac{X2_x}{df}\right)}{df}\right)\right) \quad (8)$$

Expressions 7 and 8 calculate the error vector in the image for any gaze point in space ($X2$). They can also be used for calculating the error of a given point in the scene image. This can be done by back-projecting the point of regard $X2$ into the camera image that returns the actual gaze point in the scene image:

$$x2 = P^{-1}\, X2 \quad (9)$$

Therefore, the parallax error can be calculated for any point in the scene image by knowing the system configuration and the calibration and fixation distances.

3. Changing the calibration distance

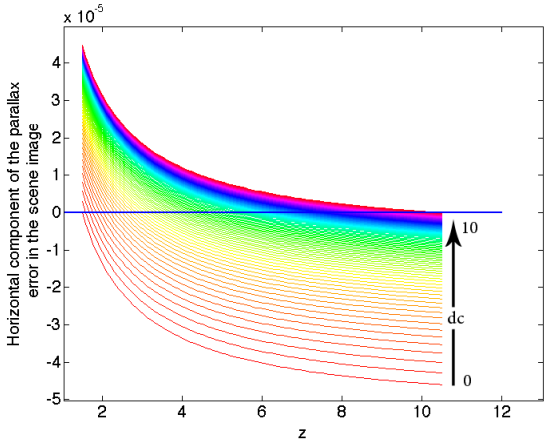


Figure 4. Translation of the function $e_X(df)$ for a given point in the image by changing the calibration distance.

By looking at a HMGT as a stereo camera system, we can deduce that changing the calibration and fixation distances does

not change the location of the epipole in the scene image. In another word, changing the calibration distance only change the magnitude of the error vectors and does not change the direction of the vector in the scene image. This can also be seen in Eq. 7 and 8 that for any given point in the image, changing the calibration distance (dc) only moves the graph of the function $e_x(df)$ up or down (Figure 4).

4. Measuring the fixation distance

The parallax error can be compensated for when the distance between the subject and the point of regard is known. With the HMGTs, the fixation distance in some situations can be obtained indirectly through the scene image. This requires an assumption that PoR is in a plane (fixation plane) that is recognizable in the scene image. Without this assumption, the fixation distance cannot be found especially when the estimated gaze point has an offset and the system has no information about the actual PoR in space. Therefore, the idea would be to find the transformation and the orientation of the fixation plane relative to the camera and then obtaining the fixation distance from that. Before describing the way of measuring the distance to the fixation plane, one important issue bears mention. When the fixation plane is fronto-parallel (Figure 5.a), the fixation distance is the same as the distance to the fixation plane. However, when the user is viewing the fixation plane not straight ahead and the plane is not fronto-parallel (Figure 5.b), the distance is not the same for all points in the plane. Therefore, the actual fixation distance cannot be found and it can only be estimated approximately based on the location of the invalid gaze point in the scene image. However, this error is only significant for the extreme viewing angles or for the areas that the parallax error is large (e.g., very close distances).

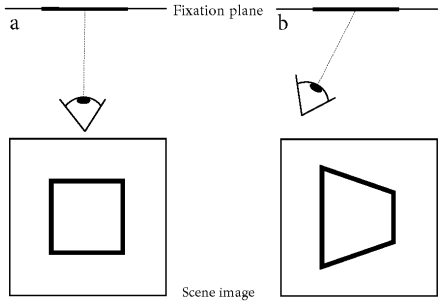


Figure 5. Viewing the fixation plane (the rectangle) from different angles and the scene images.

A single camera is not enough to measure distances unless other cues are used such as: size, shape or motion. When the scene camera of the HMGT is calibrated, the distance between the camera and a known size object may be obtained by recognizing the object in the scene image. Several pose estimation algorithms have been presented in the literature and can be used for obtaining the distance between the camera and the primitives. Most of the analytical and iterative algorithms [4] which are developed for camera pos estimation, can be used for obtaining the depth of the fiducial vertices, based on the geometrical extraction of primitives which allow the matching of 2D features (points or lines) extracted from the image with known 3D features of an object.

In many mobile gaze-tracking applications, it is possible to recognize the fixation plane in the scene image and to find at least 3 points inside the plane. By having the depth of at least 3 points in the working plane, depth for any other point within the plane can be obtained.

5. Compensation Method

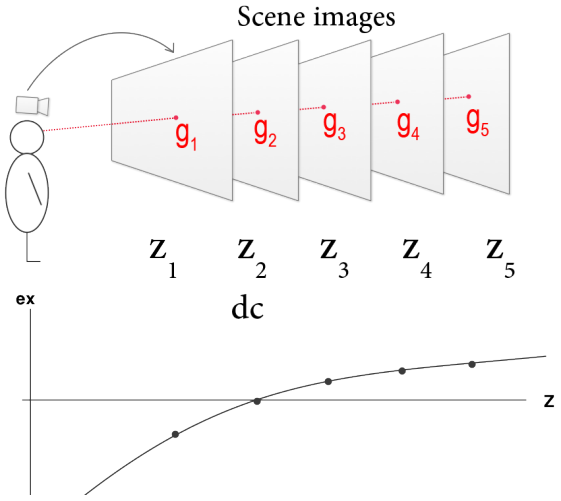


Figure 6. Interpolating the parallax error inside each plane and along distances

The presented method consists of the following steps:

Prior to use:

1. Calibrating the gaze tracker for a distance dc.
2. Measuring the compensation vectors for some sample points in different planes (at different distances).
3. Interpolating the compensation vectors inside each plane.

While using the system:

4. Measuring the fixation distance in real-time.
5. Finding the compensation vector (Figure 3) for the estimated gaze point by interpolating the vectors obtained in step 3 along the all taken distances in the step 2.
6. Correcting the estimated gaze point by the compensation vector (Figure 4)

These steps are described in detail in the following.

The head-mounted gaze tracker is calibrated for a distance (dc). After calibrating the gaze tracker, the parallax error will be measured for some sample points in different distances. While the subject is looking at a target point in a fixation plane, the actual gaze point will be detected in the scene image. The compensation vector (v) is defined as the vector between the estimated gaze point and the actual gaze point (Figure 7) in the scene image which is actually the parallax error defined before but in opposite direction. This will be repeated for a set of finite fixation planes at different depths.

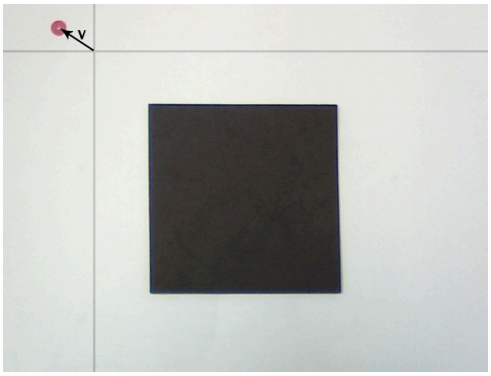


Figure 7. The scene image of the HMGT showing the compensation vector and the estimated gaze point (cross-hair) while the subject is looking at a sample point (upper-left dot marker).

Distribution of the compensation vector in each sample distance can be obtained by interpolating the sample data collected from each distance (Figure 8). In this paper, the distribution of the parallax error is modeled by a 2D first-order polynomial in the scene image and 4 points will be taken in each plane, allowing us to obtain the polynomial coefficients:

$$\begin{cases} v_x = a_1 + a_2 x + a_3 y \\ v_y = b_1 + b_2 x + b_3 y \end{cases} \quad (10)$$

Where (x, y) are the coordinates of a point inside the image.

Figure 6 illustrates the method and shows 5 different scene images for 5 different fixation planes. This figure is only for schematic illustration in which the scene images are back-projected onto the fixation planes. It shows the horizontal component of the compensation vector for the same point in the 5 scene images. The calibration distance in the example shown in the Figure 6 is z_2 which has the error of zero in the scene image (dc=z_2).

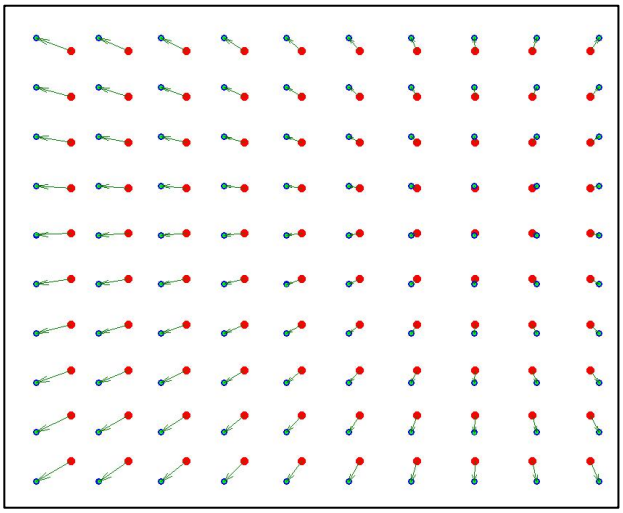


Figure 8. Interpolating the compensation vectors inside the image. The estimated gaze points (red dots) in the scene image have been corrected (blue dots) by the compensation vectors.

Having the distribution of the compensation vector in some depths allows us to estimate the compensation vector for any point and in any distance while using the HMGT. In this paper, estimating the compensation vectors for a given fixation distance (df) has been done by interpolating the sample data over depth (Z axis). This can be done by fitting a rational function to the sample data in different depths:

$$V(z) = \frac{a_3 z^3 + a_2 z^2 + a_1 z \quad + a_0}{b_2 z^2 + b_1 z \quad + b_0} \quad (11)$$

Although this is not a perfect model but it gives a good approximation of the actual trigonometric function of the parallax error [Eq. 7 & 8]. The main advantage of this model compare to a simple polynomial interpolation is that it can also be used for extrapolation and approximating the data outside the range of the sample depths. However, at least 6 different sampling distances are needed for estimating the unknowns in the Eq. 11.

The same sample data collected prior to using the HMGT can be also used for different calibration distances. Changing the calibration distance only translate the function up or down. By knowing the new calibration distance (dc_2), the rational function can be translated along the vertical axis. This translation can be done by the equation below:

$$V_{dc_2}(z) = V_{dc_1}(z) - V_{dc_1}(dc_2) \quad (12)$$

Where dc_1 is the calibration distance of when the data samples are collected.

6. Simulation and Results

A simulation has been carried out to test the compensation method described above. The results of this simulation are presented in this section.

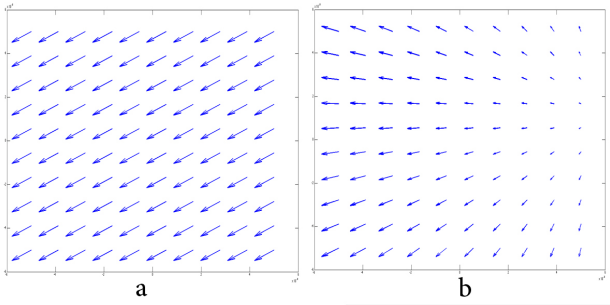


Figure 9. Two different types of error pattern in the image. (a) the epipole in at infinity, (b) the epipole is inside the image.

In the simulation, we have used a camera configuration that creates a non-uniform error pattern that has epipole inside the scene image (Figure 9).

The following parameters are used for the simulation:

The scene camera is located 3cm to the left, 1cm to the up, and 5cm to the front relative to the left eye (Tr=(0.03m,0.01m,0.05m)). The field of view of the camera is 90º (both horizontally and vertically). The camera has no rotation and the scene image is parallel to the fixation plane. The calibration distance (dc) is 2m. 6 fixation planes are used for taking the sample data which are located at zf=[0.5 1.5 2 3 4 5] m. 4 sample points at the corners of the scene image are used for each plane. After calibration and data sampling, the performance of the method has been measured for different distances (test distances). For each test distance, the compensation vector has been estimated for 10x10 points in the image. The estimated gaze point and the corrected gaze point have been back-projected to the fixation plane and the gaze estimation error is measured in degrees of visual angle.

The performance of the method is measured for 20 different distances. For each test distance, the average error of 100 points in the image has been calculated before and after compensating for the parallax error. These averages are shown in Figure 10.

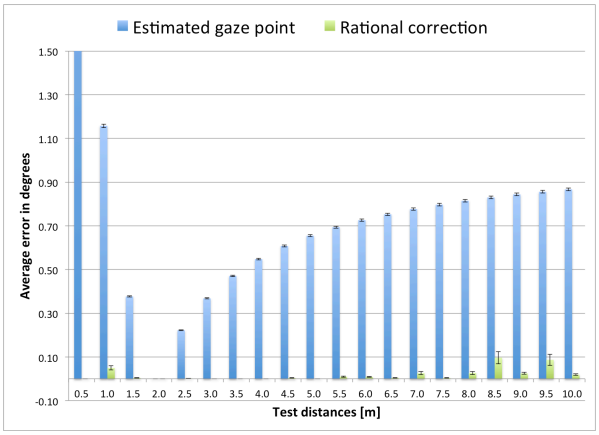


Figure 10. Average error after and before compensating for the parallax error

Figure 11 shows the comparison between the rational model presented in this paper and the polynomial models to fit to the

data in 5 different distances. As we can see in Figure 11 and also in Figure 10, the rational function makes a good approximation of the data outside the sampling distance range (>5m).

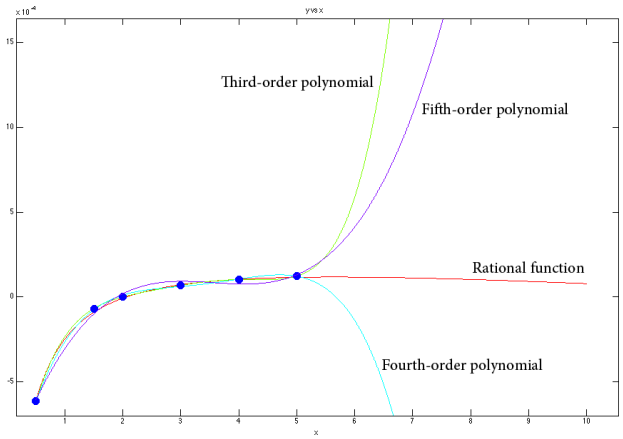


Figure 11. Fitting the rational and polynomial functions to 6 data samples in different depths.

The same data samples used in the first test have been used for when the gaze tracker is calibrated for a different distance ($dc_2 = 5m$). The rational function has been translated vertically (using the Eq. 12) after fitting to the previews data samples and it is used for correcting the parallax error in the scene image. Figure 12 shows the average error before and after applying the method measured for 20 different depths.

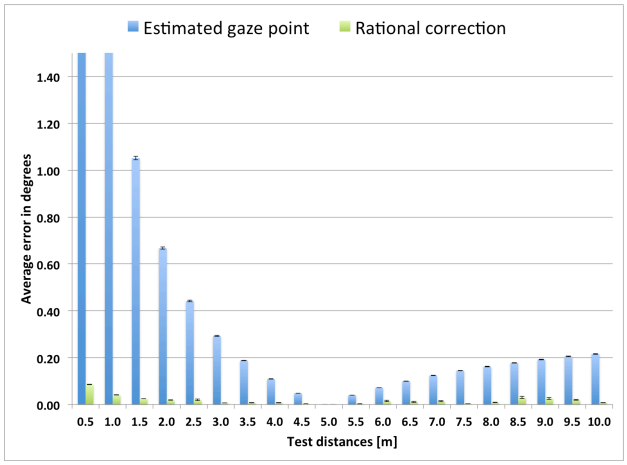


Figure 12. Average error when using the same data samples for a different calibration distance ($dc_2 = 5m$).

Changing the configuration of the scene camera in a HMGT changes pattern of parallax error in the scene image, however, most of the time, almost the same configuration will be used for different users (especially when the HMGT is in a glasses form). Although, the same data samples cannot be reused for different camera configurations, but it may still be used when the camera position relative to the eye slightly changes. It may occur because of the differences in geometry of the head between subjects when using the same system for different users. Figure 8 shows the results of using the same data samples for different camera configuration of Tr=(0.025m,0.005m,0.045m) which is 5mm closer to the eye in all three dimensions.

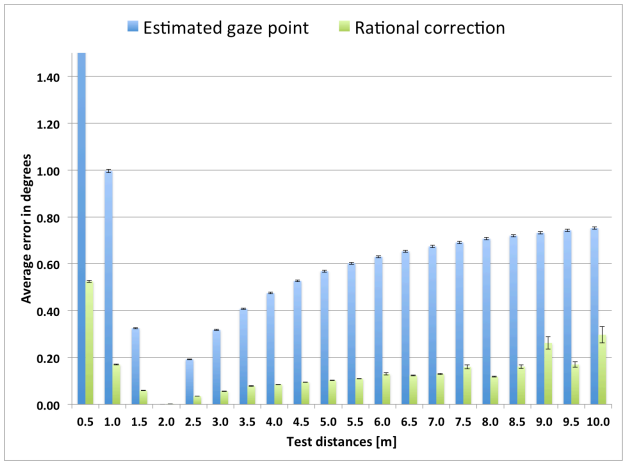


Figure 13. Average error when using the same data samples for a different subject where the camera is 5mm closer to the eye in all three dimentions.

7. Conclusion

The parallax error in the monocular HMGTs has been described as a function of different parameters of the system in a general configuration. Furthermore, a new method for compensating for parallax error has been introduced based on the assumption that the depth of PoR is know through the scene image. A prior data sampling is needed in order to find the pattern of the error for the scene camera configuration. The effectiveness of the method has been shown by simulating the HMGT. The results show that the parallax error can be corrected while using the HMGT. It has been shown that the same data samples can be reused for when the HMGT is calibrated in a different distance. Reusing the same model on different subjects is also possible unless the position of the camera relative to the eye changes significantly.

As a future work, the effectiveness of the method should be investigated for when the fixation plane is not fronto-parallel and for the extreme viewing angles (Figure 5).

8. REFERENCES

[1] Bernet, Sacha, Christophe Cudel, Damien Lefloch, and Michel Basset 2013Autocalibration-based Partioning Relationship and Parallax Relation for Head-mounted Eye Trackers. Machine Vision and Applications 24(2): 393–406.

[2] Mardanbegi, D., and Hansen, D.W. “Parallax error in the monocular head-mounted eye trackers” In Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12). ACM, New York, NY, USA, 689-694.

[3] Mardanbegi, D., Hansen, D.W., “Mobile gaze-based screen interaction in 3D environments”, Proceedings of the 1st Conference on Novel Gaze-Controlled Applications (NGCA2011), Blekinge Institute of Technology, Karlskrona, Sweden, 2011.

[4] Maidi, D., Didier, J.Y., Ababsa, F., and Mallem, M. A performance study for camera pose estimation using visual marker based tracking. Machine Vision and Applications, IAPR International Journal, Springer, 2008.

[5] Tobii Technology. tobii eye tracking research. http://www.tobii.com/en/eye-tracking-research/global/, June 2012

- Chapter 10 -

Investigations of the Role of Gaze in Mixed Reality Personal Computing

Investigations of the Role of Gaze in Mixed-Reality Personal Computing

Thomas Pederson, Dan Witzner Hansen, and Diako Mardanbegi
IT University of Copenhagen
Rued Langgaards Vej 7
2300 Copenhagen, Denmark
{tped, witzner, dima}@itu.dk

ABSTRACT
This short paper constitutes our first investigation of how eye tracking and gaze estimation can help create better mixed-reality personal computing systems involving both physical (real world) and virtual (digital) objects. The role of gaze is discussed in the light of the situative space model (SSM) which determines the set of objects a given human agent can perceive, and act on, in any given moment in time. As a result, we propose to extend the SSM in order to better incorporate the role of gaze, and for taking advantage of emerging mobile eye tracking technology.

Author Keywords
Interaction paradigm, gaze tracking.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
The design of interactive systems that involve more than one computer device and also a range of everyday physical objects, demands us to extend the classical user-centered approach in HCI [3]. One challenge is that both system and human needs to continuously establish an understanding of what parts of the physical and virtual worlds that currently make up the "user interface" as devices and interaction modalities change with context. The egocentric interaction paradigm [5] proposes a change in view of a) the role of digital interactive devices in relation to the information they provide access to, and b) to generalize the HCI input/output concept to make room for multiple parallell interaction channels as well as interaction with objects in the real world (physical objects).

Virtual Objects and Mediators Instead of Interactive Devices
Input and output devices embedded in digital appliances are viewed as *mediators* through which virtual objects are accessed. Virtual objects are assumed to be dynamically assigned to mediators by an *interaction manager* software component residing on body-worn hardware. The purpose and function of mediators is that of expanding the *action space* and *perception space* of a human agent (Fig. 2).

Action and Perception Instead of Input and Output
In the egocentric interaction paradigm, the modeled human individual is an agent moving about in a mixed-reality environment, not a "user" interacting with a computer. Also the HCI concepts input and output are reconsidered: (device) "input" and "output" are replaced with (human agent) "action" and "perception". Note that object manipulation and perception are processes that can take place in any modality: tactile, visual, aural, etc. In this paper, we focus on *visual* modalities for perception and action.

HUMAN ACTIVITY AND GAZE
Eye movements are versatile and play an important role in everyday activities [2]. It is well known that human eye movements are governed by our interests and intentions [6], and

IUI 2011, February 13–16, 2011, Palo Alto, California, USA.

humans tend to look at the object that they want to act on prior to any motor control. The sequences of fixations, trackable by emerging mobile tracking technology [1] in some cases provide enough data for making predictions [2].

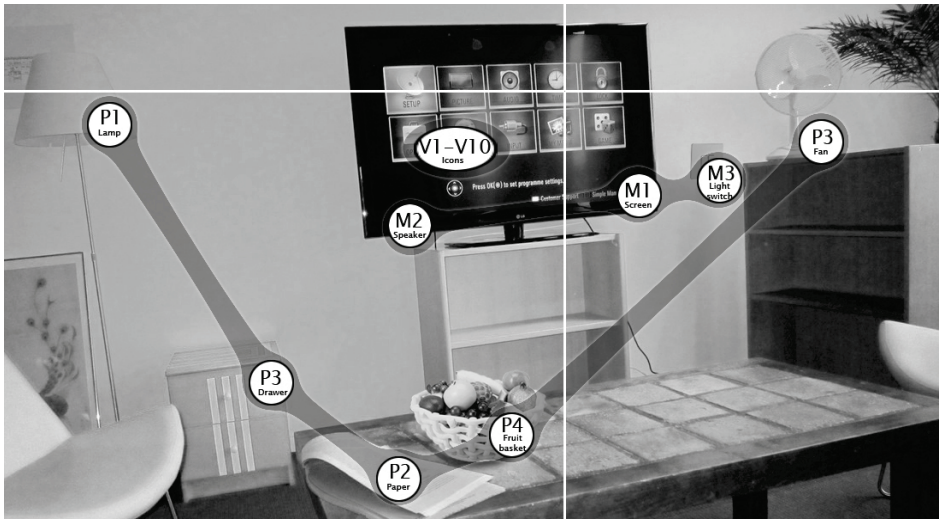


Fig. 1. A living room environment as seen by a human agent. Some physical objects (P1-P5), virtual objects (V1-V10) and mediators (M1 and M2) are labelled for illustrative purposes. The gaze direction of the human agent is indicated by the hair cross.

A SITUATIVE SPACE MODEL

The situative space model (SSM) [4] is intended to model what a specific human agent can perceive, reach and operate, at any given moment in time. This model is intended to be the emerging egocentric interaction paradigm equivalent of what the virtual desktop is for the PC/WIMP (Window, Icon, Menu, Pointing device) interaction paradigm: more or less everything of interest to a specific human agent is assumed to, and supposed to, happen here. Fig. 1. shows a typical situation which the SSM is intended to formalise and capture: a living room environment inhabited by a human agent.

In the following, we will discuss the role of gaze in the light of SSM definition excerpts from [5].

Perception Space (PS)

The part of the space around the agent that can be perceived at each moment. Like all the spaces and sets defined below, it is agent-centered, varying continuously with the agent's movements of body and body parts. Different senses have differently shaped PS, with different operating requirements, range, and spatial and directional resolution with regard to the perceived sources of the sense data. Compare vision and hearing, e.g.

Within PS, an object may be too far away to be possible to recognize and identify. As the agent and the object come closer to each other (either by object movement, agent movement, or both) the agent will be able to identify it as X, where X is a certain *type* of object, or possibly a unique individual. For each type X, the predicate "perceptible-as-X" will cut out a sector of PS, the distance to the farthest part of which will be called *recognition distance*. [5]

Naturally, gaze direction plays a fundamental role in defining the visual PS for a given human agent. Any object directly hit by the vector anchored in the fovea and passing through the center of the lense of an eye (that is, the line of sight, LoS) is a top candidate member of the PS since it is only along this vector human agents literally see clearly. However, other components of the human visual perception system "expands" this single vector of visual impression so that visual attention in practice typically is directed to a larger area than just a point in 3D space. Let us call this 2-dimensional expanded area – with the LoS hitting its center – the field of view (FoV). Then, very simplified, the 3D space created by the union of the two eye's FoV, let us call it the 3DFoV, forms the basis for the visual PS (again, with the help of complementary parts of the human perception system, dealing with angular calculations and objects obstructing each other, etc.). All objects in the 3DFoV (not just the object in LoS) should be included in PS for a given agent.

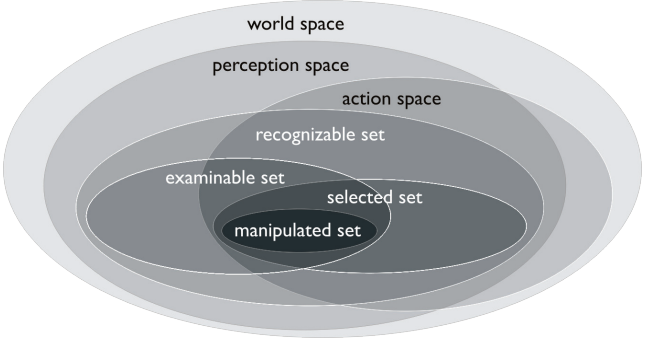


Fig. 2. A Situative Space Model. [4]

Recognizable Set (RS)
The set of objects currently within PS that are within their recognition distances.

The kind of object types we are particularly interested in here are object types that can be directly associated with activities of the agent – ongoing activities, and activities potentially interesting to start up – which is related to what in folk-taxonomy studies is known as the basic level.

To perceive the status of a designed object with regard to its relevant (perceivable) states (operations and functions as defined by the designer of the artifact) it will often have to be closer to the agent than its recognition distance: the outer limit will be called *examination distance*. [5]

Examinable Set (ES)
The set of objects currently within PS that are within examination distances. [5]

The visual RS and ES in the SSM (motivated by the potential value for an egocentric interaction system to know in what detail objects can be analysed by a human agent) raises gaze tracking questions. Can gaze estimation be used for determining whether an object is examinable, recognizable or just perceivable? Eye movement pattern categorization over time and object types could, potentially, help determining whether a visually perceivable object belongs to RS or ES.

Action Space (AS)
The part of the space around the agent that is currently accessible to the agent's physical actions. Objects within this space can be directly acted on. The outer range limit is less dependent on object type than PS, RS and ES, and is basically determined by the physical reach of the agent, but obviously depends qualitatively also on the type of action and the physical properties of objects involved; e.g., an object may be too heavy to handle with outstretched arms. Since many actions require perception to be efficient or even effective at all, AS is qualitatively affected also by the current shape of PS.

From the point of view of what can be relatively easily automatically tracked on a finer time scale, it will be useful to introduce a couple of narrowly focused and highly dynamic sets within AS (real and mediated). [5]

The visual AS is limited: Few actions that change the state of physical or virtual objects can be performed using eyes alone. However, gaze activity is often part of actions executed using other parts of the body such as the hands.

Selected Set (SdS)
The set of objects currently being physically or virtually handled (touched, gripped; or selected in the virtual sense) by the agent.

Physical selection is almost always preceded by visual selection: before grabbing anything, we visually fixate the object. Without dwelling into the reasons, this fact means that by tracking gaze, computer systems can do heuristical guesses for what object, among all the objects in AS, that is about to get manipulated next.

Manipulated Set (MdS)
The set of objects whose states (external as well as internal) are currently in the process of being changed by the agent. [5]

All these spaces and sets, with the obvious exception of the SdS and the MdS, primarily provide data on what is *potentially* involved in the agent's current activities. Cf. the virtual desktop in the PC/WIMP interaction paradigm.

Like object selection, also object manipulation can involve gaze. While visual feedback is crucial for certain kinds of physical object manipulation (e.g. hand writing), it is probably less important for most. For manipulation of virtual objects, the situation is different. One of the most prevailing critisisms of today's user interfaces is in fact the heavy reliance on visual feedback. Contrary to actions in the real world, most user interfaces *rely* on continuous visual attention also during object manipulation.

EXAMPLE SITUATION
Fig. 1. shows a living room environment. If we assume that the area covered by the photo approximately corresponds to the field of view of a given human agent, objects in the photo can be categorized using the SSM as follows:

Physical objects
The physical object P1 (the floor lamp) belongs to the examinable set since the human agent can determine whether the lamp is on or off. The paper document P2 is not in the examinable set because from this position, the human agent can not likely determine what the document is about, see what page that is on top, let alone read the text of it. P2 is however in the recognizable set because it is indeed clear that the object is a paper document. The drawer P3 belongs to the examinable set because it is possible to see whether it is open or closed. The fruit basket P4 is examinable: it is possible to determine whether it is empty or full and even the kind of fruit that it contains. The desk fan P5

is also examinable – it is possible to see whether its rotor blades are turning or if they are still.

Mediators

The TV embeds two mediators: The screen (M1) and the speaker (M2). The screen M1 is in the examinable set since the human agent can determine what is shown on it, i.e. the virtual objects that it currently mediates. The TV speaker M2 is not in the visual perception space at all since the case design of the TV hides its presence. (It is true that it is in the aural perception space – virtual objects can be sufficiently sonified from this distance – but we limit our analysis to the visual perception space.) The light switch M3 is in the perception space but not examinable: the human agent cannot determine its state from this distance.

Virtual objects

The icons shown on the screen M1, modeled as virtual objects V1-V10, are all examinable because their state (selected/not selected) can be determined from the position of the h. agent.

Action space

With respect to action space, most of the objects labelled in Fig. 1. are outside of that space. The human agent cannot, from her/his current positon manipulate them. The exception might be the paper document P2 or the fruit basket P4 which might be just about reachable. If we imagine the human agent to hold the TV remote control in her/his hands (a physical object embedding mediator buttons) however, also the 10 icons V1-V10 enter action space since that would allow her/him to manipulate them.

The hair cross in the picture simulates the gaze direction of the human agent, currently examining one of the 10 icons on the TV.

CONCLUSION

In this paper we have taken our initial steps in modeling gaze within the situative space model (SSM). Gaze turns out to be a defining factor for to which space an object belongs, potentially altering an object's location within the model rapidly. To fully exploit the information in eye and gaze movements, the SSM might benefit from the incorporation of something like an "attended-to" set of objects (Fig. 3.), including objects across several existing SSM spaces and sets that the given human agent is attending to. Among many open issues related to gaze and human attention is that a person may attend to objects that they can see but not recognize. At the same time, an object may be recognizable but not really attended to.

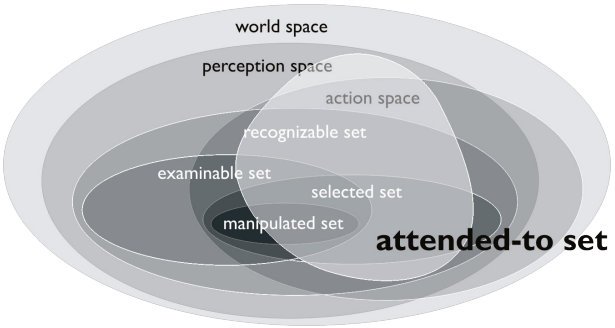


Fig. 3. Future work: extending the situative space model with an "attended-to" set.

REFERENCES

1. Hansen, D. W., Ji, Q., In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 3*, 2010,478–500.
2. Land M.F., Tatler B.W., *Looking and Acting: Vision and eye movements in natural behaviou*r. Oxford; New York: Oxford University Press, 2009.
3. Norman, D. & Draper, S. (Eds.) *User centered system design*. Erlbaum, Hillsdale, NJ, 1986.
4. Pederson, T., Janlert, L-E., Surie, D., Setting the Stage for Mobile Mixed-Reality Computing - A Situative Space Model based on Human Perception. *IEEE Pervasive Computing Magazine* (to appear), 2011.
5. Pederson, T., Janlert, L-E., Surie, D., Towards a Model for Egocentric Interaction with Physical and Virtual Objects. *Proceedings of NordiCHI'10*, ACM Press, 2010, 755-758.
6. Yarbus, A. L., *Eye Movements and Vision*. New York: Plenum Press, 1967.

- Chapter 11 -

Synergies Between HMDs and HMGTs

Synergies between head-mounted displays and head-mounted eye tracking: The trajectory of development and its social consequences

by

Rich Ling, Professor, IT University of Copenhagen (rili@itu.dk)

Diako Mardanbeigi, PhD Fellow, IT University of Copenhagen (dima@itu.com)

Dan Witzner Hansen, IT University of Copenhagen (witzner@itu.dk)

Abstract

Introduction

Gaze tracking has moved from being unwieldy and intrusive to being simple and discreet. It has moved from being a technology that is complex to use and reliant on the care and prodding of highly trained engineers and scientists to becoming non-invasive and relatively straightforward to use. It has also moved from being a technology with only marginally practical applications to being one that has an increasing number of use areas. This is not to say that head mounted gaze tracking (HMGT) is a mainline technology. There are however significant areas where the technology can enhance data collection and can assist in the execution of important tasks. In this paper we are interested to look at this technology in the context of head-mounted displays and the consider the likely trajectory of development.

As with many other electronic devices, HMGT technology has become smaller and more agile. Early in their history, eye tracking devices often involved elements attached directly to the eye and the need to stabilize the head (read: fix it into place with various frames and straps). By contrast, contemporary eye tracking technology can disappear into simple, lightweight mobile devices. This development has been seen on many technical fronts. Indeed we are on the cusp of another

transition; namely mobile head mounted displays that will have the ability to retrieve information and to help us mediate our communication.

It is likely that in the near future HMGT functionality will be compact enough to fit into wearable displays such as POV devices including Google Glasses that replicate an individual's field of vision. The current crop of these devices allow the capture video of, for example, a person as they parachute out of a plane or a law enforcement person as they go their rounds. The image captured, however, replicates their the broad field of vision and not a particular point of gaze. In many cases, this broader image is what is best, however, we contend that there are also situations where a more specific focal point is also of interest.

There has been limited discussion of HMGT and heads-up displays in the literature. In the work that exists they have been examined as extensions virtual reality and immersive computing and as a way of apportioning attention. HMGT has also been examined in terms of its impact on social interaction in a laboratory context . Thinking somewhat more broadly HMGT-enhanced head mounted displays (HMD),such as we see in the Google Glass project, we will have the ability to further specify our point of attention and eventually transmit this to others or make it available for later examination. HMGT will tell us, for example that a user is looking at a particular individual and not a crowd, a particular product in the shelf in the grocery store and not the whole shelf or a particular part of the PC screen and not the whole screen. This can change the way that we can interact with our environment. In this paper we consider how HMDs and HMGT can fuse into a single platform. Because of this development it is likely that HMGT will find new applications. In this process, we also see that it there are consequences in relation to privacy and power relationships.

We will first go through the development and application of wearable computing. We follow this with a short history of history and affordances of gaze tracking. We next discuss the melding of HMGT and heads up display technologies and the potential for using this when it facilitates interacting with information that is embedded in the local context. This touches on issues such as the so-called "the internet of things." Finally we look into the eventual applications for HMGT enhanced wearable displays both in terms of the possibilities and the threats that they represent for at the personal and the social levels.

Head-mounted display and wearable computing technology

Technical development of HMD and wearable computing

Wearable devices that enhance our interaction with the world might be traced back to, for example the development of glasses. Following this line of thought, the watch was carried on the body (often in a well protected pocket) from the 1600's and in the case of women on the wrist often as a piece of jewelry. The wristwatch made its appearance with males during the First World War since it was awkward for pilots to dig out pocket watches. Moving to head mounted electronic devices, earphones have been a part of the technical landscape since the early period of the radio and the idea of a HMD was first was patented by McCollum and as a stereoscopic television HMD by Heilig (1960). Because of the technical limitations at that time, the idea of HMD was more focused on giving the user a virtual experience by showing a video on a HMD. The first video see-through augmented reality system was made in the 1960's by the Bell Helicopter Company, which was a servo-controlled camera-based HMD. This provided the pilot with an augmented view captured by an infrared camera under the helicopter that it was useful for landing at night. Since the early 1970s, the U.S. Air Force has carried out research on HMD systems as a way of providing the aircrew with a variety of flight information and and also interacting with the airplane and user interfaces. In the 1980s we began to see the use of HMDs where the user is able "see-through" the device either optically or based on a video image. The user can see for example 3D computer- generated objects superimposed on his/her real-world view. The optical and the video approaches for HMD hardware design merge and superimpose the virtual view onto the real views of the world either via a semi-transparent mirror as with optical see-through HMDs, or via video cameras mounted on the head as with video see-through HMDs.

The most recent of HMD project, and the one that seems to have garnered the greatest general interest, is the Google Glass project that includes an augmented reality head-mounted display for public. As of this point, Google Glasses includes a heads-up display in addition to an embedded POV scene camera, microphone, different types of radio-based communication (Wifi - 802.11b/g and Bluetooth), GPS functionality, an accelerometer and "bone conduction" in lieu of speakers. Voice control is used to operate the device including using taking pictures/video, sending messages, getting directions etc. Google glasses, and some other smart glasses (e.g., Vuzix M100), show that HMDs can potentially be used as the visual interface of the mobile devices and the next generation of smart phones wearable rather than mobile. HMDs can become a common display for various devices that we use such as mobile phones, tablets and even laptops.

Applications of the head-mounted computing technology

HMDs have been used in many different application fields such as: military (e.g., air force and navigation) governmental (e.g., police), civilian (e.g., engineering, medicine, and computer-guided

surgery) , video gaming, sports, and simulation (e.g., driving and flight). Perhaps the most promising future uses of HMDs are those in which the display allows for an enhanced virtual environments (e.g. enhanced reality) rather than replacing real environments as in virtual reality.

Head mounted displays provide the ability to use context sensitive information such as weather reports, incoming text messages, public transportation schedules, route finding, sharing information with others, etc. Additional functionality will likely include pattern recognition perhaps similar to that in Google Goggles that references libraries of photos taken by by others in addition to GPS data to search for further information on the item in question.

Gaze Tracking Technology

Parallel with the development of wearable computing and head mounted displays, there is also a development in the area of gaze tracking. Gaze tracking monitors and records the point of regard (i.e. where a person is looking). The point of gaze generally constitutes approximately 3-5 degrees of the total field of vision which is about the size of the thumb nail at arms length. In this section, a short history of the gaze tracking technology in terms of technical development and then different application areas of this technology are briefly described. At the end of this section, some of the limitations of the gaze trackers are described.

A short history of gaze tracking

The functioning of the eyes and the interaction between gaze and cognition has long been the subject of interest. The people who have contributed to our understanding of vision include some of the luminaries of science such as Kepler and Descartes. People have been developing ways of mechanically tracking eye movement for over 100 years. Seen from our remove, many of the early systems were quite draconian. The earliest devices were physical “contact lenses” that were attached to the eye using either an adhesive or suction to hold them in place. This contact lens was sometimes attached to a mechanical lever in order to track the movement of the eye. It goes without saying that this hindered natural observations. As Jacob notes “This method is obviously practical only for laboratory studies, as it is awkward, uncomfortable and interferes with blinking.” An early researcher, Edmund Huey described his approach to recording the movement of a subject’s eyes:

> I arranged apparatus as follows: A plaster of Paris cup was moulded to fit the cornea accurately and smoothly, sand-papered until it was very light and thin, and placed upon the front surface of the eye, the cup adhering tightly to the moist cornea. No inconvenience was felt, as the corneal surface was made insensitive by the use of a little holocain, or sometimes

> cocaine. A round hole in the cup permitted the observer to read with this eye, the other eye was left free. A light tubular level of celloidin and glass connected the cup to the aluminum pointer, flat and thin, which responded instantly to the slightest movement of the sys; and, suspended over the smoked-paper surface of a moving drum cylinder, the aluminum point traced a record of the eye's movement as the observer read.

The system of tracking eye movement became progressively less invasive as the technology for observation developed. The use of film cameras eased the burden on (and presumably the irritation of) subjects. Shortly after the turn of the last century, researchers attached a simple "white speck of material" to "the eye of a subject and filmed it as the individual read." Researchers began to photograph the light reflected from the cornea. In 1901, Dodge and Cline developed what they called the "Dodge Photochronograph" that is seen as the progenitor of today's eye reflection tracking systems that have since dispensed with attaching anything to the eye. This is not to say, however that the gaze tracking systems were not bulky. They might take up whole sections of the laboratory. Buswell's 1935 device, for example, was a rambling collection of tubes, monitors, electronics, struts, lights and frames with which to stabilize the subject's head. It filled a large desk and spilled over onto area behind. It was nothing if not voluminous.

As with many other areas of research, the rise of computerization dramatically changed the way that we were able to gather and analyze gaze tracking information. The equipment for tracking eye movement has undergone a radical reduction in size and devices have seen a similarly radical increase in processing power, accuracy and responsivity. With time, researchers developed head mounted devices that allowed the subjects greater freedom of movement.

Early eye tracking systems used retrospective analysis of film or other recording material. Starting in the 1960's computers gave researchers the ability to digitally gather gaze tracking information, process the data and provide feedback in real time. These developments mean that gaze trackers can be used as a computer pointing device, they can also be used for sending commands (e.g. making selections on a screen).[1]

Gaze interaction with computers has, until now, mostly been applied to the situation of a single, stationary user is sitting in front of a screen. It has used a camera often mounted on or near the PC

[1] Dwell-time selection, eye blinks, gaze-gestures, and context switching have been typical ways of extending the capabilities of eye trackers for gaze-based interaction. Gaze as a pointing modality can also be used together with some other interaction modalities such as body gestures, and speech. Eye-based head gesture is a novel technique for enhancing gaze-based interaction through voluntary head movements. Gaze and head gestures measured by the gaze trackers provide a gaze-based method for interacting with computers and objects in the environments.

screen to first calibrate and then to track the user's gaze (a remote eye tracker). Recent work has moved in the direction of head mounted devices where, as the name suggests, the camera that captures the individual's eye movement is mounted on the person's head using either a helmet, a headband or glasses. This has extended the domain of gaze-based interaction into the mobile situations which allow the user almost complete freedom of head movement as well as mobility.

Compared to the previous generation of gaze trackers, HMGT devices afford an unheard of degree of mobility. The developments in camera technology and miniaturization mean that it is now possible to move away from the desk-bound notion of eye tracking. Indeed, we are entering a period where head-mounted eye trackers have become much smaller, lighter and thus easier to integrate with other mobile devices. Further, the integration of a variety of input possibilities (gaze, haptics, gestures, etc.) mean that HMGT is becoming more flexible and more suitable for mobile, gaze-based interactive applications.

HMGT is currently at a stage where size and quality allow seamless integration of eye trackers into normal glasses. HMGT software is, to a large extent, also equal to an increasing number of gaze tracking tasks.[2] As we will discuss below, this also expanded the areas of use of gaze tracking.

Gaze tracking applications

Gaze tracking applications can broadly be divided into two categories: *diagnostic applications* where the eye tracker provides objective and quantitative evidence of the user's visual and attentional processes or neurological disorders (e.g., Identification of neurological disorders by studying the diagnostic data provided by properties of saccades and fixations, and applications in psychology, cognitive linguistics, and product design), and *interactive applications* where the eye tracker is used as an input device of an interactive system, and the system responds to the user's gaze [Duchowski 2007].

Diagnostic applications

The earliest questions that used gaze tracking considered the interaction between gaze and tasks such as reading and looking at a picture. The research questions revolved around the interaction between vision and comprehension. Yarbus and Riggs, for example, recorded people's gaze as they

[2] We currently have cameras that are only several millimeters in size. In addition the use of infrared light sources in glasses mean that glasses mounted eye-trackers are not a significant technological challenge. Clearly, several issues remain that will improve eye trackers even further e.g. robustness to large and rapid light changes. A general problem for most current eye trackers is their need to be calibrated to the individual. While this is a current problem with most commercial eye trackers, there exist several possible techniques that could limit explicit per session calibration (see Witzner & Ji,2010)

looked at an image when there was no particular task required of the viewer and then when the viewer was asked to retrieve different types of information from the image (i.e. the the number of people in the image or the type of clothes they are wearing). In other cases, gaze was recorded when people were asked to synthesize information from the image such as the class status of the people. In each case, Yarbus and Riggs recorded different patterns of eye movement.

Eye tracking has also been used when examining how people read. Just and Carpenter, for example, have used eye tracking to measure the time (in milliseconds) that subjects looked at words in sentences. They suggest that the time to integrate gaze and comprehension depends on the frequency of a words general use and its thematic importance. There is also a pause at the end of a sentence. The research also shows that eye movement differs when a person is reading aloud and silently. Also the research has indicated that as the complexity of the material becomes more difficult we spend a longer time on each word and we have a narrower field of focus.

A similar application has been to study the use of gaze in how people carry out everyday tasks such as simple food preparation and how people handle different situations that arise in driving in traffic (often examined using driving simulators). In the case of the simple tasks, the research as been concerned with the role of gaze when going through a sequence of actions. The findings show that gaze often anticipates the next physical action of an individual. When we are making a sandwich we look at the butter immediately before we move our hands to retrieve it. In the case of driving, while this is a dynamic situation as compared to the static analysis of reading or viewing a photograph, it has a common thread in that gaze tracking is used to understand the how the eyes focus on certain things and perhaps ignore other items that may also have importance.

Gaze tracking has been applied to usability studies. In a classic study Fitts et al. used a film camera to record the gaze of pilots as they landed airplanes. This has been extended later with other dimensions of flying order to better understand where to place the instruments. This type of research has been further applied to the placement of other arenas. Researchers have been interested to understand, for example the best arrangement of items on a web page or in printed material. It is often the case that the diagnostic applications have not relied on real-time feedback. Rather the data is captured and analyzed later.

Another area of research has been to control how people carry out various types of visual analysis. This includes questions of, for example X-ray inspection, production control inspection and photo interpretation (e.g. in the case of astronomy or national security).

A question that has been broached in this context is the connection between seeing and cognition.

According to Jacob and Karn:

> Psychologists who studied eye movements and fixations prior to the 1970s generally attempted to avoid cognitive factors such as learning, memory, workload, and deployment of attention. Instead their focus was on relationships between eye movements and simple visual stimulus properties such as target movement, contrast, and location. Their solution to the problem of higher-level cognitive factors had been "to ignore, minimize or postpone their consideration in an attempt to develop models of the supposedly simpler lower-level processes, namely, sensorimotor relationships and their underlying physiology (Kowler 1990, 1".

Perhaps as an attempt to address this issue, the next step in this line of research was to combine eye tracking with brain activity as recorded with functional Magnetic Resonance Imaging (fMRI). This development has provided a new tool for with which to study the the interaction between reading or looking and cognition. The research generally shows the correlation between eye fixation and brain activity. This approach allows us to better understand the way that cognition works as we access different types of information in our brains. A related question is the interaction between vision and cognition for populations that are not able to communicate or have only fundamental communication capacity, e.g. newborn children. This is another area where gaze tracking has been applied. The research has investigated how newborn children fixate on various shapes such as images with faces vs. more abstract images. This provides insight into the bonding process.

Interactive applications

As noted above the development of computing capacity meant that gaze tracking provided for immediate feedback. This led to the use eye-movement as a pointing device for computer-based user interfaces. The most common application of this capability has been to allow disabled persons who cannot use their hands to control a mouse or keyboard by using their gaze (Jacob & Karn, 2003).

We are now seeing that the gaze-tracking devices are becoming smaller, more robust and less in need of the careful goading and maintenance of engineers and scientists. Since they are no longer leashed to large computing devices. This means that the uses of gaze tracking can move into more natural settings. To the degree that they can be used in natural settings we can begin to consider a broader range of applications. In addition to the traditional uses of cognition research, usability studies and as aids for disabled persons, it is possible to develop gaze tracking applications for more quotidian purposes. This is a discussion to which we will return below.

The synergies of HMGT and wearable computers

Limitations of the head-mounted display/computer

The current implementation of the Google Glass project, as well as various POV "action video cameras,"[3] have the ability to capture, in a broad sense, what the individual is looking at. Many of the face mounted devices replicate the users' field of vision. However, the field of view for these video-based applications (often about 170 degrees) is broader than our active field of vision (which is about 135 degrees vertically and 160 degrees horizontally degrees). The most sensitive part of the eye is actually a small part of the total organ. The field of vision is divided into three different areas of decreasing sensitivity, the foveal (about 1-2 degrees of vision), parafoveal (about 3-5 degrees of vision), and peripheral region (everything beyond about about 6 degrees). The foveal area stands for about half of the information that is eventually sent to the brain from the eyes. The peripheral area is only able to register movements and contrasts as it has very poor visual acuity.

When we are looking at a scene before us, we focus on only a small portion of the total information. We continually scan a scene in order to gather further information. In some cases we can move our attention to the peripheral areas of vision albeit not with the same natural ease. Within the brain a large portion of the cerebral cortex is devoted to processing the visual information. Indeed, a large percent of our total brain processing capacity is used when we carefully look at something. Thus, while POV scene cameras can capture the broad sweep of visual information, they do not allow us to know the specific point of gaze. The wide frame captured by a many POV video system does not reflect the foveal point of our vision.

The affordances of the current HMGT

Head-mounted eye trackers integrated with a POV scene camera can indicate the point of gaze. Additionally, we can use computer vision techniques for recognizing the objects in the scene and also for reconstructing the environment around the user. When the apparatus is attached to the user's head, it is also possible to use it know the direction and the speed of the movements of the user's head.

Gaze tracking can provide an abundance of information about the subject and the environment. The

[3] These include the GoPro, Contour+, Ion Air Pro Drift HD, Panasonic HX-A100, AXON flex and car-mounted video devices and an increasing number of other devices that are moving into this space.

eye image recorded by today's gaze trackers can be used for measuring the eye movements and fixations (e.g., The number of fixations, the amount of time in each area, the number of times returned to a point etc.) and also estimating the gaze. In addition it can also provide other types of eye-based information such as the pupil diameter (e.g., as an indicator of the cognitive load), different eye features like iris pattern (e.g., used as a biometric), the frequency of blinking, the behavior of the eye muscles (e.g, as one of the indicators of the user's fatigue), and the reflection of the environment on the surface of the cornea. In addition, the vestibulo-ocular reflex that coordinates eye movements relative to head movements makes it possible to even measure the head rotations (roll, tilt, and pan) through the eye movements.

By looking at the future interactive applications of wearable computers, and different ways of interaction with the head-mounted graphical user interfaces, we see that gaze as a pointing mechanism will likely be an early functionality to head mounted computing devices. In addition, speech and gestures will also likely be added as mechanisms for sending commands (e.g., doing selection). Other technologies such as, haptic, accelerometers, electroencephalography (EEG), and perhaps other biosensors (e.g., EMG, SC, and GSR) may also be used to give more functionalities to wearable computers.

Applications of gaze-enhanced head mounted computing devices

There are a wide range of applications that are possible with gaze enhanced head mounted computing devices. It is possible to imagine systems that allow for extremely detailed interaction between users. Indeed, when the gaze of one person is transmitted to another, the second person can specifically understand what the first person is looking at and, by inference, where their attention is directed.

It is possible to think of using gaze enhanced devices when teaching people to react to visually specific clues, e.g. the investigation of x-ray images or when learning to drive. Using this functionality it is possible to imagine, for example, that a technician can call to a remote expert and be "talked through" exceedingly detailed procedures. It is also possible to conceive of these technologies being used to deploy and direct remotely located workers across a broader geographical area. Gaze tracking can facilitate logistical systems if delivery people visually check the stocks of items on the shelves and gaze tracking can "check off" on the QR codes of the existing stock. This might be enhanced to give the delivery person a visual cue for out of date items. Shared gaze tracking can help us help one another to to focus in on relevant (and very detailed) information when navigating in unfamiliar areas. Alternatively, if an individual is lost he/she can track on a sign showing the name of the street (or perhaps another sign such as a local restaurant) and this will help the system locate the

individual.

As is probably obvious, the integration of HMGT and heads up display technology has many applications for individuals such as better specification of the interaction afforded by HMDs that, at this point, in many cases relies on voice input. However, combining HMDs and HMGT, we also move beyond applications for single individuals. As with many other technologies, we suggest that the first users will likely be larger institutions, particularly those where there is a need for central coordination and mutual understanding of one another's situation. However, we suggest that with time the technology will be further diffused for use by less formal social clusters such as families and groups of friends. That said, the likely areas of adoption will be niche applications in the near future. This is a theme to which we will return below.

Social consequences of HMGT and digital artifacts

As noted above there have been several phases in the development of gaze tracking. These have included the basic understanding of eye movement, the application of this basic understanding to both the study of cognition and to usability and most recently, the use of gaze tracking with live video and sophisticated computing power to control computers. We are now entering a phase when gaze tracking is moving out of the sheltered environment of the laboratory and moving “into the wild.” As noted above, the devices are becoming easy enough to use that they can be imbedded in other head-mounted gadgets such as POV video devices and heads up display units. The technology is available. This means that HMGT is becoming available for the development of a variety of applications that were not possible when it was bound to specific locations by the bulkiness of the equipment.

However, the very mobility of the equipment also means that there are several new questions that arise. Two important ones are first the question of privacy and the issues of recording the social interactions. The other question focuses on the degree to which HMGT will become embedded in the structure of social interaction.

Privacy and legal issues of HMGT

The head-mounted POV scene cameras are a common element in computing glasses with HMD (e.g. Google Glasses) just as they are common in head mounted gaze trackers. The privacy issues of the HMGT, are on the one hand associated with the scene camera and on the other hand related to the gaze data and the eye information.

Use of video equipment raises question with regards our rights to gather photographic information and our rights with regards being photographed. The use of photographic equipment is well trod. As soon as photography became common the question of our right "to be let alone" was an issue. Warren and Brandeis wrote in 1890 that "Instantaneous photographs . . . have invaded the sacred precincts of private and domestic life; and numerous mechanical devices threaten to make good the prediction 'what is whispered in the closet shall be proclaimed from the house-tops.'" The context in which Warren and Brandeis were discussing privacy was an era when photography was largely practiced by professional news photographers previous to the popularization of smaller personal cameras and more than a century before digital photography became standard. With time, the development of closed-circuit television and a variety of other digital recording systems adds unheard of dimensions to"shouting from the house-tops." In many cases, however, there has been and continues to be a power differential between those who record and those who are recorded. It is the local convenience store or gas station that has the security cameras. These were used in the context of protecting their private property. The ability of individuals to record material in these private settings is different from the right of the property owner to do the same. This question has been brought into the public discussion by the so-called McDonalds incident with Steve Mann. A short synopsis of the incident was that Mann entered a McDonalds in France wearing his "eye tap" device. This is, among other things, a forward mounted video camera mounted in a glasses frame wherein the video camera covers one eye. According to Mann's version of the incident one of the employees tried to tear the glasses off his face and Mann was eventually pushed out the door.[4]

Among the other issues that the incident touches on there are questions are associated with who is allowed to capture video in a particular situation. In the case of the commercial establishments, they often have the right to have surveillance. Also, since it is considered their domain, they can to some degree set other conditions with regards who they will serve. Clearly the incident raises the question of the conditions for video capture both on the part of establishments as well as with customers. The incident has been couched in terms of power to surveille and be surveilled as a function of power. A somewhat parallel question arises with the equipping of police with eye-mounted video cameras as in Rialto, California.[5] In this case, the local police department realized a major reduction in the number of complaints against officers. There is the idea that words and comments are no longer ephemeral, but they become a digital artifact.

There is, however, another issue associated with eventually wearing a digital recording device in the normal flux of daily life (as seen in, for example the idea of Memex, MyLifeBits and in so called

[4] http://www.slashgear.com/broken-glass-father-of-wearable-computing-allegedly-assaulted-17238802/

[5] http://www.nytimes.com/2013/04/07/business/wearable-video-cameras-for-police-officers.html?_r=0

lifecasting); namely that it imposes a dimension on the situation that has not hereto been a part of our understanding of a social situation. A tacit idea associated with social repartee is the idea that the interaction is not recorded, it is ephemeral. The imposition of a record on the interaction eventually changes the way that we are willing to commit ourselves to the situation. It eventually raises the spectre of being accountable for our comments and our actions in a way that we are not accountable when they are fleeting.

The development of HMGT in natural settings ratchets up the issue of privacy to yet another level since the technology not only records what is happening in a particular situation, but where the gaze of one of the actors in the situation is resting at any given moment. To be the subject of others' digital gaze and to know that it is recorded means that the scene takes on a different social character. I will eventually be held responsible for my comments and actions in a way that was not possible heretofore. We may also find that our own HMGT record incriminates us in ways that were previously difficult to document. If the gaze tracked record of a car accident, for example, shows that I was adjusting the radio at the time of the crash, it has implications for the apportionment of responsibility.

There are yet other dimensions to this issue. HMGT could eventually record the individuals that we see or the items we look at in a store. In this latter case the collection of QR codes that we gaze at can be valuable information for marketing purposes. The question then arises as to ownership of that data and how that data might be used by marketers to form a profile of the individual. Since HMGT is far more specific than simple POV devices (or GPS information) ownership and use of the information presents an important unsolved issue. Another question revolves around the potential for the system to distract us when we are engaged in various activities such as driving. HMGT can provide important feedback to a driver such as monitoring eye activity and sensing when they are in need of a rest stop. However, gaze tracking could also be used against a driver if it finds that their gaze was not on the appropriate place when they were involved in an accident. Thus there are potentially some difficult unresolved questions that need to be settled.

HMGT as a social mediation technology

Another issue associated with the eventual development of HMGT is the degree to which it can become embedded in the flux of social interaction. There are a range of technologies and systems that take on dimensions of being Durkheimian social facts. Mechanical timekeeping, telecommunication and dimensions of the internet can been seen in this context on a broad social

level. In addition, in more restricted groups, technologies such as calendaring systems and, in its time, the network of fax machines are examples of social mediation technologies.

The characteristics that are common for these technologies are that they have a critically large number of users, their adoption is supported by an ideology that legitimates their position in society (we feel safer by having a mobile phone with us), they have arranged the social landscape to the exclusion of alternative systems that provide approximate the same function (e.g. the clock displaced the sun dial) and perhaps most importantly, there is a recripocal expectation that that others will also either operate based on the edicts of the system (everyone needs to respect time and timekeeping) or be mutually available via a particular mediation form. This is not to say that all technical developments become social mediation technologies. There are many that have become thoroughly embedded in society in spite of not being used for social mediation. Refrigeration is an example of a technology that has made dramatic changes in the social ecology. It is not, however, used for the mediation of social interaction.

The question here is whether HMGT, or for that matter HMDs, will become a technology of social mediation. It is indeed difficult to make the case that this will happen. As we have noted there is undeniable functionality that is provided by HMGT. As we have noted, the trajectory that is perhaps most likely is that HMGT will be implemented in a future heads-up devices. In this trajectory it will be developed for special applications such as remotely mediated group work where the detailed knowledge of one anothers focus is important, i.e. coaching of detailed repairs. It might be that teams of repair personnel might be linked to one another as they carry out a distributed repair task and can thereby interact with one another to facilitate their common work. It might be that we use gaze tracking when discussing detailed co-editing of documents with one another so that we can tacitly see where our co-authors are looking. Other applications might extensions of the inspection functionality noted above where, for example delivery people will need to gaze at particular points in a store where they deliver products to insure that they are displayed properly.

This suggests, however, that video recording and also the more specific use of gaze tracking may find a niche when used in formalized settings for well-defined purposes. When thinking of personal uses of HMGT it is possible to imagine people using gaze to access specific types of information in specific setting. It might, for example, be useful to have detailed gaze tracking while shopping so that we can read in barcodes or QR codes to gather information about products like their nutritional value as compared to our favorite diet or eventually that the item is on sale at a store down the street. As noted above, however, there are a variety of questions that need to be addressed before this is universally accepted.

However, it is more difficult to understand how either HMGT or HMDs will quickly become a part of the general flux of social interaction. While there is a begrudging acceptance of surveillance in society and there has been the development of sousveillance, i.e. people below observing those above, there is not a major discussion of what is termed veillance where there is not a power differential between the individuals involved. This has been a sphere based on trust and forgiveness. The insertion of digital recording and more specifically gaze tracking into this context will likely not be as simple as it raises a broader set of questions. The point here is that HMGT can and likely will become a part of the broader digital landscape, but that the first applications will not be associated with social interaction but with commercial situations.

In a similar way, we will also likely develop norms of when we are explicitly NOT looking at the activities of others. We will develop the sense that it is not appropriate to have on our HMGT unit when another person is using their PIN code. We may need to have a function that shows the recorder is not on, or we will take off the HMGT device, much as we take off sun glasses, as a sign of courtesy.

Conclusion

In this paper we have considered the eventual melding of HMGT with heads-up display technology. We see that heads-up devices are moving into the diffusion process. The commercialization of devices such as Google Glasses indicate that there is a certain interest in this direction. At this point, HMGT and heads-up technology are two separate threads of development.

HMGT technology is technically available. The cameras that will provide for gaze-tracking, the computing capacity and the batteries are already available. It is very likely that gaze tracking will enter become a feature of POV video devices such as the Google Glasses. This may well come as a part of the “feature creep” that is often associated with these types of gadgets. We will certainly see that it is applied to various types of “niche” applications such as those noted above.

- Chapter 12 -

Towards Wearable Gaze Supported Augmented Cognition

Towards Wearable Gaze Supported Augmented Cognition

Andrew Toshiaki Kurauchi
University of São Paulo
Rua do Matão 1010
São Paulo, SP
kurauchi@ime.usp.br

Carlos Hitoshi Morimoto
University of São Paulo
Rua do Matão 1010
São Paulo, SP
hitoshi@ime.usp.br

Diako Mardanbegi
IT University, Copenhagen
Rued Langgaardsvej 7
2300 Copenhagen
dima@itu.dk

Dan Witzner Hansen
IT University, Copenhagen
Rued Langgaardsvej 7
2300 Copenhagen
witzner@itu.dk

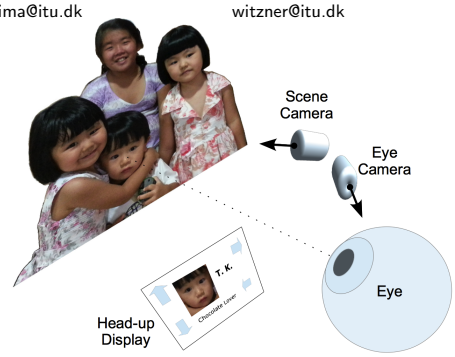


Figure 1: The user has instant and up-to-date information about a person and can interact using gaze alone, gaze and a button, and gaze and head gestures.

Abstract
Augmented cognition applications must deal with the problem of how to exhibit information in an orderly, understandable, and timely fashion. Though context have been suggested to control the kind, amount, and timing of the information delivered, we argue that gaze can be a fundamental tool to reduce the amount of information and provide an appropriate mechanism for low and divided attention interaction. We claim that most current gaze interaction paradigms are not appropriate for wearable computing because they are not designed for divided attention. We have used principles suggested by the wearable computing community to develop a gaze supported augmented cognition application with three interaction modes. The application provides information of the person being looked at. The continuous mode updates information every time the user looks at a different face. The key activated discrete mode and the head gesture activated mode only update the information when the key is pressed or the gesture is performed. A prototype of the system is currently under development and it will be used to further investigate these claims.

Author Keywords
gaze interaction; wearable computing; augmented cognition

ACM Classification Keywords
H.5.2 [Information interfaces and presentation: user interfaces]: .

Introduction
In this paper we explore how gaze interaction might enhance the usability of wearable computers by creating simpler interaction mechanisms and show how such mechanisms can be applied in applications for cognitive augmentation. But first we will discuss some design issues to better understand the benefits of gaze interaction for wearable computing applications,

Time and space are important to define the physical context but gaze may yield aspects of attention.

Wearable computing devices such as the EyeTap [7] combine a scene camera and a head-up display (HUD) to enable mediated reality, the ability to computationally augment, diminish, or alter our visual perception. The EyeTap configuration allows the camera to capture the same image as it would be captured by the eye, providing very realistic visual effects and life logging data that can be shared and used as the user's extended memory [5].

Similar but simpler configurations such as the "Memory Glasses" by DeVaul [4], may place a wearable HUD to (or instead of) the lens of the eye glasses. In such configuration, the useful display area covers just part of the visual field of view of one of the user's eyes, reducing the quality of the mediated reality experienced. Based on the announced Google Project Glass and the Vuzix M100, the next generation of smart phones is moving from mobile to wearable by using an HUD for "hands free" constant information and communication access.

Constancy is an important characteristic of wearable computers. Because the applications can always be on and available, having information popping up at any time may distract the user and become a hazard in particular situations, such as competing for (or even obstructing) the user's attention when crossing a street.

Therefore, the design of wearable applications must consider different design issues than desktop applications. In particular, as pointed out by Rhodes [10], typical WIMP interfaces require fine motor control and eye-hand coordination on a large screen, while many typical wearable computing applications are secondary tasks (e.g. reminders) or support a complex primary task. Even when the wearable application is the primary task (such as text editing), the environment might intrude and, therefore, there is a need to design for low and divided attention.

Bulling and Gellersen [1] provide a recent discussion on the current state of mobile gaze trackers and describe ways of using them in mobile applications. Due to the developments in wearable eye trackers that have just recently become more portable and easy to use, it is not surprising that there are only a few wearable computer systems that use gaze information.

For example, [2] suggests the eye movements data from a EOG eye tracker to determine context information to wearable applications, but does not use gaze information. Data input is carried out using a chord keyboard. One concrete example of a wearable augmented reality system using gaze interaction was described by Park et al. [9]. Their system relies on scene markers to position virtual objects. Gaze information is used to point and objects can be selected by dwell-time.

Because most of the work on gaze interaction has been done assuming a desktop or mobile device scenario, we discuss next different principles that can be used to design gaze supported wearable computing applications.

Current gaze interaction applications are not designed for low or divided attention.

Augmented cognition should be effortless.

Interaction with wearable computers

Because wearable computers provide support while the user is performing other activities, freeing the hands (or at least one hand) from computer interaction is an important feature. Typically, chord keyboards are used as input devices with the HUDs. Though chord keyboards can be very efficient for data entry, becoming an efficient typist might required a great effort [4]. To overcome this difficulty, speech and hand gestures have also been used.

Due to its ability to augment and mediate reality, wearable computing applications can provide support for complex real-word activities, with applications areas such as military, education, medical, business, and many others. But as identified by many wearable computing researchers, augmented cognition applications will be a major factor in the development of wearable computers. Augmented cognition applications can help the user to perform mental tasks. Because wearable computers are always on and available, they can be incorporated by the user to act like a prosthetic and become an extension of the user's mind and body.

Examples of augmented cognition applications are described in [6, 4]. Mann [6] gives examples of how diminished reality, i.e., removing clutter from the scene such as advertising and billboards, can help the user by avoiding information overload. The use of an EyeTap facilitates the substitution of planar patches of the scene by virtual cues. Another possible application is to place virtual name tags on each person within the field of view.

DeVaul [4] proposes the use of software agents to provide just-in-time information based on the user's local context. Using an HUD with a chord keyboard, his system (called Memory Glasses) was able to present short text messages on the HUD related to personal annotations typed using the chord keyboard, helping the user to remember related issues stored in the system. Today, with the current state of mobile computing, related information could be searched in the Internet.

During the development of the Memory Glasses, DeVaul [4] defined the following principles of low-attention interaction for wearable computing:

1. Avoid encumbering the user, both physically and perceptually, referring to the hardware, peripherals and interface.
2. Avoid unnecessary distractions, by minimizing the frequency and duration of the interactions, and using appropriate context information.
3. Design interfaces that are quick to evaluate, so the user, even when interrupted, is always in control.
4. Simplify the execution as much as possible, but no further. Easy things should be easy, hard things should be possible.
5. Avoid pointers, hidden interface states, non-salient interface changes, and never assume the wearable interface has the user's undivided attention.

These principles will be used in the design of a cognitive augmented application for memory aids, described next.

Gaze Supported Augmented Cognition

The "extended mind" conjecture of Clark and Chalmers [3] states that not all cognitive processes are in the head. The claim is based on the idea of epistemic actions, i.e., actions that alter the world to help cognitive processes. Because gaze, attention, and cognitive processes are so interrelated, it seems natural to use gaze information to

automatically filter, control, and mediate the contents of wearable computing application, but the description of actual systems combining both gaze and wearable technologies are still rare in the literature. As an initial effort to combine previous experiences from both areas, we have followed the principles proposed by DeVaul [4] for low-attention interaction, to design three interaction modes for a gaze supported augmented cognition applications.

The objective of the application is to provide the user information about the person currently being observed.

The objective of the application is to provide the user information about the person currently being observed, similar to the automatic name tag application proposed by Mann [6], but using a simpler setup. The basic components of the system are shown in Figure 1. Two cameras are required for the wearable gaze tracker, one pointing to the scene and a second looking at the eye. An HUD is used to display relevant information to the user. Observe that it is also possible to use gaze information for interaction with the HUD.

Due to the low resolution screen of the HUD, when multiple people are seen by the scene camera, presenting information about every person at once might be confusing, since it might be difficult to associate a name to a given face. Following DeVault's first principle, to avoid encumbering the interface, our system is designed to provide information about a single person at a time, corresponding to the face being looked at.

To minimize the frequency and duration of the interactions, the information about the person can be updated every time the user's gaze lies on a new face. We will call this interaction mode continuous (C). Because the information is always presented in the same location on the HUD, this information can be easily ignored by the user.

We are also developing a discrete (D) mode, that updates the information on the HUD after a key press to determine if continuous updates are distracting. A third discrete mode controlled by head gestures (G) is also being developed. The head gesture mode allows for completely hands-free operation, while not overloading the eye with a control task. Because the head can perform simple gestures independently of the eye natural behavior, head gestures are more appropriate than eye gestures for wearable computing.

These three modes follow the simplicity of execution principle for the task of associating names to faces. For more complex tasks, e.g., to show more information about the person, the D and G modes could facilitate the interaction because they can be easily extended, using a double click or a different yet simple head gesture. Because the HUD can also be used for gaze interaction, a point and click (or point and gesture) interface will also be developed.

For the continuous mode, dwell-time and eye gestures could also be used for interaction with the HUD, but because these interaction modes would require longer interaction times and require full attention of the user, they would not be appropriate. Also for the C mode, to avoid the information to change when the user is looking at the HUD in case a person is positioned in that direction, its region is masked out, so no face is detected within the HUD region. As pointed by DeVaul's, context information could be used to improve the quality and timing of the information, and it should clearly be considered in a real application. The use of context information is not though the focus of this paper.

DeVaul's third principle states that the interface should be quick to evaluate. Designing for divided attention also

Figure 2: Low cost wearable head mounted eye tracker.

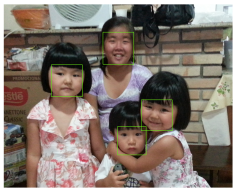


Figure 3: Faces detected using the Viola-Jones algorithm.

requires the user to be reminded of the last face seen, in case of distraction. Therefore the information is presented with a cropped region computed automatically by the face detector algorithm, showing the detected face. This feature also allows the user to avoid detection errors by the system. The last principle is a list of things to be avoided and that has been followed by our design.

System implementation

Figure 2 shows a low cost wearable head mounted eye tracker being used in our experiments. It uses two USB webcams, one pointing towards the scene and the second looking at the eye. The eye camera has two IR leds to provide robustness to illumination conditions. Both cameras are mounted on a baseball cap. The gaze tracking software is based on the open source Haytham gaze tracker (available at eye.itu.dk), that has been ported to run on a Linux platform. A 4 point calibration is used to compute a homographic transformation.

The wearable gaze tracker has not been integrated with an HUD yet, so the proposed memory aids methods will be demonstrated on videos projected on a large screen. The projected videos will be scaled to show the faces close to their actual size. Though this is not an ideal situation, we expect the video to cover the field of view of the user, so the HUD display can be simulated as part of the projected screen, and placed somewhere on the lower left of the video.

Faces are automatically detected using live video from the scene camera of the wearable gaze tracker using the Viola-Jones algorithm [11]. A result from this algorithm is shown in Figure 3. Once the user's gaze is detected within a face region, an estimator based on our gaze-to-face mapping algorithm is used to recognize the face and information about the person is displayed according to the current interaction method (C, D, or G). For the D mode, the left button of a wireless mouse is being used.

For the recognition of head gestures in the G mode, we are using the method introduced by Mardanbegi et al. [8]. Their method uses a combination of head gestures and a fixed gaze location for interaction with applications running in large displays and small mobile phone screens. Because the head gestures are estimated directly from the eye movements without the need of extra sensors such as accelerometers, the whole gaze interaction system can be made very light and comfortable to wear, as seen in Figure 2. Figure 4 show two images of the eye when fixating at a target and performing a vertical head movement (initially down and moving upwards, while looking forward). When a user keeps the gaze on a specific target, the vestibular-ocular reflex makes it possible to measure head movements because the eye moves in the opposite direction of the head. Therefore, head movements are measured indirectly from the eye movements detected from the eye camera.

Conclusion

A typical wearable computing application is always on and available, so it must be designed for divided attention. Gaze based applications, on the other hand, have been mainly developed for desktop computing. Therefore, the direct port of gaze based applications to wearable computing is not recommended since gaze and attention are so interrelated. More importantly, the use of most gaze interaction paradigms, such as dwell-time and gaze gestures are not appropriate for wearable computing, since they not only require full attention by the user to interact, but they misappropriate the natural behavior of the user's gaze.

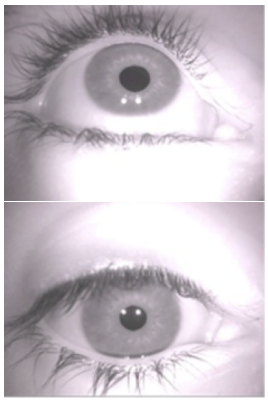


Figure 4: Images of the eye when looking at a target and performing a head gesture.

Nonetheless, we do believe gaze can revolutionize the way we interact with wearable computers. For that purpose, we have described our on going research on wearable gaze supported augmented cognition. By applying design principles learned from the wearable computing community, we proposed three gaze-based interaction modes that are appropriate for low and divided attention. The continuous mode updates information at every new event (such as looking at a different face), a key activated discrete mode, and a head gesture activated mode.

Though speech and gestures have also been used to interact with wearable computers, gaze interaction offers more privacy and discreteness and we expect it to offer faster interaction (though not faster than a chord keyboard, but definitely easier to learn). Maybe the most important characteristic is that gaze can potentially be used to interact with scene objects (with the help of computer vision algorithms), besides the head mounted display. A prototype of the system is currently under development and it will be used to further investigate these ideas.

References

[1] Bulling, A., and Gellersen, H. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing 9*, 4 (2010), 8–12.

[2] Bulling, A., Roggen, D., and Tröster, G. Wearable eog goggles: eye-based interaction in everyday environments. In *CHI Extended Abstracts* (2009), 3259–3264.

[3] Clark, A., and Chalmers, D. The extended mind. *Analysis 58*, 1 (1998), 7–19.

[4] DeVaul, R. *The memory glasses: wearable computing for just-in-time memory support*. PhD thesis, Massachusetts Institute of Technology, April 2004.

[5] Ishiguro, Y., Mujibiya, A., Miyaki, T., and Rekimoto, J. Aided eyes: eye activity sensing for daily life. In *Proceedings of the 1st Augmented Human International Conference*, AH '10, ACM (2010), 25:1–25:7.

[6] Mann, S., and Fung, J. Videoorbits on eyetap devices for deliberately diminished reality or altering the visual perception of rigid planar patches of a real world scene. In *Proceedings of the Second IEEE International Symposium on Mixed Reality* (2001), 48–55.

[7] Mann, S., Fung, J., Aimone, C., Sehgal, A., and Chen, D. Designing eyetap digital eyeglasses for continuous lifelong capture and sharing of personal experiences. In *Proc. CHI 2005 Conference on Computer Human Interaction* (2005).

[8] Mardanbegi, D., Hansen, D. W., and Pederson, T. Eye-based head gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA'12*, ACM Press (2012), 139–146.

[9] Park, H. M., Lee, S. H., and Choi, J. S. Wearable augmented reality system using gaze interaction. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, IEEE Computer Society (2008), 175–176.

[10] Rhodes, B. The wearable remembrance agent: a system for augmented memory. *Personal Technologies Journal Special Issue on Wearable Computing , Personal Technologies*, 1 (1997), 218–224.

[11] Viola, P. A., and Jones, M. J. Robust real-time face detection. In *ICCV* (2001), 747.

- Chapter 13 -

Discussion & Future Work

This chapter summarizes the work presented in the thesis including a discussion in relation to the research questions, as well as a consideration of future research possibilities. The most important research result of this work from the point of view of gaze interaction, is to show that HMGTs can provide a broad range of interactive applications in 3D. Conventional gaze interactive applications were mostly limited to situations where a remote (table-mounted) gaze tracker allows the subject to interact with a computer display. This thesis has revealed that gaze interactive applications can be extended to mobile situations for interacting with virtual and real objects in the environment. It has been shown that the information provided by a gaze tracker alone is enough to provide an intuitive gaze-based interaction with the environment by taking the nature of eye movements into account. The conclusions with respect to each of the four groups of research questions introduced in the Section 1.1.2 are presented in the following sections.

§ 13.1 Gaze Pointing

It has been shown that an ordinary HMGT that estimates the gaze point in the scene image allows the user to interact with computer displays without need for estimating the gaze in 3D or having the position sensors. There are two possible improvements for future work in regard to the method that has been presented:

1. Instead of detecting the display in every frame of the video sequence, it can be tracked in the image after the first time that the display is being detected and recognized in the image. This allows for interaction with multiple displays when more than one display is in the scene image.

2. The method presented here that maps the gaze point from the scene image to the display in the environment requires the display to be en-

tirely visible inside the scene image. This limitation can be addressed in future work such that the system can estimate the PoR in the display coordinate system even when only a portion of the display is within the field of view of the scene camera. This is possible by applying more advanced computer vision techniques for detecting the rectangular shape and also tracking the display inside the image. The information about the aspect ratio and the resolution of the display that is needed for this purpose can be transferred to the system through the temporary visual markers (introduced in the Section 1.4.1).

§ 13.2 Activation Strategy

It has been shown that the conventional gaze activation strategies that have mostly been initiated to help some group of disabled people to interact with computer displays, are not suitable for interaction in 3D. A systematic way of categorizing the gaze activation techniques and a new taxonomy has been proposed which is based on how the information obtained from the gaze tracker is used for activation. The taxonomy presented here reveals a new technique in which in contrast to the conventional techniques differentiates between the eye movements and the gaze movements. The new technique proposed here allows for measuring a wide variety range of head rotations (roll, pitch, and yaw) using only the information provided by the gaze tracker. The technique provides a multimodal interaction mechanism that uses the gaze for pointing and head-gestures (measured through the eye movements) for activating an object. The eye-based head gesture technique has two main requirements:

1. The gaze point should be fixed on an object while performing the head gesture and the system should be able to determine whether gaze was fixed.

2. The system should be able to separate the VOR movements from the natural movements of the eye.

In the following two subsections, these two requirements are briefly discussed for different conditions.

13.2.1 Detecting the Fixed-Gaze

The fixed-gaze requirement can be detected in both remote and head-mounted gaze trackers. A remote gaze tracker estimates the gaze point in a 2D/3D space which is stationary relative to the camera and the light sources (e.g., a 2-dimensional computer screen on a table). The PoR will be estimated

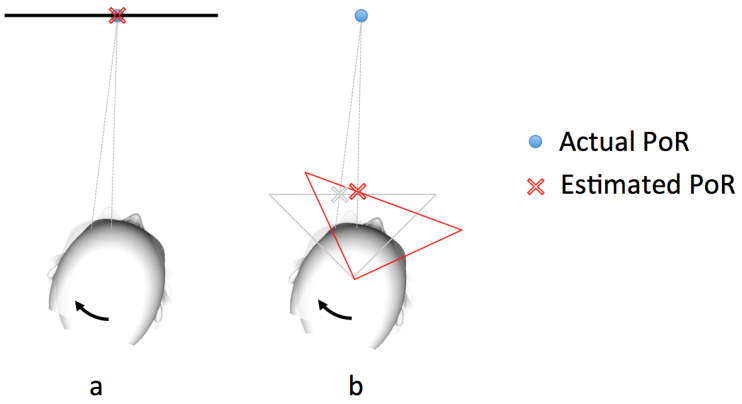


Figure 13.1: *(a) Gaze estimation in a RGT is independent of head rotations, (b) when the actual PoR is fixed the estimated gaze point in the scene image of a HMGT changes by rotating the head*

independently of head rotations (Figure 13.1.a). Therefore, the first requirement can be easily checked when eye is moving in the eye image. Checking the fixed-gaze condition is also straightforward with the HMGTs that use position sensors and estimate the absolute position of the PoR in the space independent of head movements. In contrast, estimating the PoR in HMGTs that use a scene camera is not independent of head movements, meaning that even when the subject is looking at a fixed point in the space, the estimated gaze point in the scene image changes when the head is rotated. These type of HMGTs actually estimate the intersection between the gaze (the visual axis) and the image plane of the scene camera that is attached to the head (Figure 13.1.b). Therefore, implementing the eye-based head gesture technique is more challenging with HMGTs that have a scene camera. This is because more information is needed to check whether the PoR is fixed while moving the head. One solution is to use computer vision techniques to recognize the gazed object in the scene image. By measuring the distance between the gaze point and the object in the image, we can determine whether or not the gaze is fixed during the head rotations. Another solution is to use template matching [6] and determine whether or not the image patch around the gaze point moves together with gaze while rotating the head. Another solution is to estimate the motion of the entire scene image and compare that to the movement of the estimated gaze point in the image during the head rotation. When the scene image moves in the opposite direction and with the same speed as the gaze point, it may be an indication that the gaze is fixed on an object while the head is rotating.

13.2.2 Separating the VOR from the Natural Eye Movements

Eye-based head gesture technique is only applicable when the VOR movements can be separated from the natural movements of the eye. This can be done easily when the gazed object is fixed, because only the VOR movements will occur while performing a head gesture. An interesting question that arises is whether this technique is applicable when the object of interest is in motion?

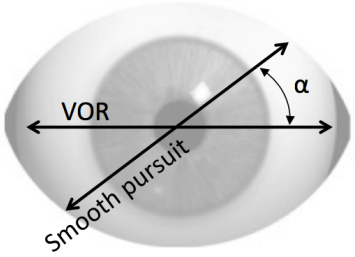


Figure 13.2: *The direction of the VOR movements caused by the head yaw and the direction of the smooth pursuit movements caused by fixating on a moving target*

The main assumption of the eye-based head gesture technique is that the object of interest is stationary while the head is moving. However, this technique may also be applicable for non-stationary objects. When the gazed object is moving and the head is rotating, the VOR movements function in conjunction with smooth pursuit. The eye-based head gesture technique can be applied when it is possible to separate the VOR movements of the eye from smooth pursuit (SP) movements. The interaction of smooth pursuit eye movements and vestibulo-ocular reflex is still not well understood [33]. There have been some studies that have investigated the linear interaction (summation) of the neural driving signals of smooth pursuit eye movements and VOR [33]. Figure 13.2 shows a situation when VOR and SP are acting simultaneously along different axes. For example, when the head is rotating horizontally (head yaw) and the gazed object has a motion in space that causes a smooth pursuit in a different direction than the VOR. Classifying the eye movements measured by the eye tracker would be easier when the angle between the VOR and SP (α) is larger. For example, measuring horizontal head movements would be easier when the gazed object is moving vertically in front of the eye. When the angle α is small, measuring the head movements would be more challenging depending on the speed of the head movements and the target (gazed object) in space. Figure 13.3 shows a user performing a head gesture while looking at an object which in motion.

It illustrates an object that is moving to the right while the user's head is rotating to the right and then to the left. The right-left head gesture creates a left-right VOR pattern (indicating the gesture). In this case, the VOR and the smooth pursuit movements observed in the eye image are along the same axis and the angle α is nearly zero. Therefore, the smooth pursuit movement will change the left-right pattern of the VOR movements. Figure 13.3 shows the summation of the VOR and the SP movements of the eye. The SP movement in the right direction may completely cancel the left element of the VOR pattern. However, it seems possible to filter out the slow SP movements from the VOR movements when the head movements are fast enough (faster head gestures). For the target velocities less than about $15°s^{-1}$ (of visual angle[1]) the smooth pursuit is not saccadic and it is probably easier to filter out the smooth pursuit from the eye movements.

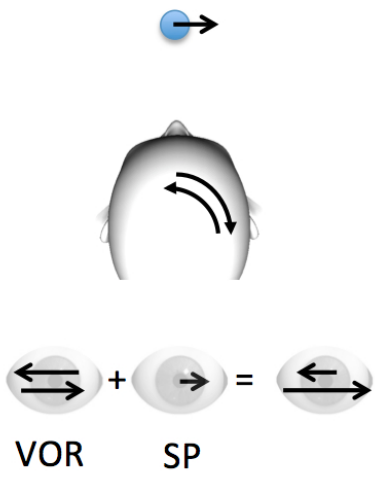


Figure 13.3: *Sum of the VOR and SP movements and the final eye movements*

Measuring the head movements and recognizing the gesture pattern through the eye movements may become very challenging in situations where the gazed object moves very fast while performing the head gesture. In this case, the eyes track the object with a saccadic movement instead of smooth pursuit.
The head movements can not be measured through the eyes in the situations where the object is fixed relative to the head. For example, when the user is looking at an object that is attached to the head (e.g. an item shown in a head-mounted display), and moves the head, the gaze does not change rela-

[1]The maximum speed of non-saccadic smooth pursuit. It has been described in the Section 1.2.2.3

tive to the head while the head is moving and therefore, VOR will not occur. Investigating whether it would be practically possible to use the eye-based head gesture technique for interaction with a moving object (with different range of speed and different types of movement), is the subject for future research.

13.2.3 Why VOR?

One question may arise when implementing the eye-based head gestures for HMGTs:

- In a situation where the scene image moves as a consequence of moving the head with an attached scene camera, can the head movements be measured directly through the scene image instead of the VOR?

One answer is that measuring the motion in the scene image (e.g., by using phase correlation [18] or optical flow [3] techniques) relies very much on the texture and light conditions of the image. For example, the method may not work properly when the light conditions change or the image lacks the texture. Measuring the fast head movements through the scene image is also challenging unless there is a high frame rate camera and a fast processor. The blurry image caused by the camera moving makes the head movement detection challenging and it may require better cameras with lower exposure time that implies less motion blur. Measuring the head movements through the eye movements does not have this limit. In addition, the head yaw and pitch already have been measured by the gaze tracker (through pupil position) and it is not necessary to measure this movements again through the scene image. Furthermore, determining the direction of the head movements from the scene image requires more information as to the orientation of the camera on the head.

§ 13.3 Parallax Error

Parallax error in head-mounted gaze trackers is defined and described using the epipolar geometry. It is shown that the angular offset between the optical and visual axis does not have a significant effect on the changes in parallax error, and therefore, the eye can be considered as a pinhole camera when studying the parallax in a HMGT. Looking at the eye and scene camera of a HMGT as a stereo camera setup allows us to describe and formulate the relationship between the parallax error and the geometry of the system. The description presented describing the error enables us to investigate the functional features and behaviour of the error. It also allows estimation of different parameters such as: the calibration/fixation distance, camera configuration, and the viewing angle. The results may be highly useful for

optimum design of a HMGT that leads to the minimum parallax error. A new method for compensating for parallax error has been introduced based on the assumption that the distance between the user and the PoR in space is known (e.g., through the scene image). This method estimates the parallax error for each point. Future work should investigate the performance of this method when the fixation plane is not fronto-parallel and when there are extreme viewing angles.

§ 13.4 HMGTs & HMDs

This thesis has shown the potential use of HMGTs for interaction in 3D. As HMGTs are becoming smaller, more robust and easier to use, gaze tracking may become a standard feature in future wearable computing devices. As discussed in Chapter 11 many different applications can be imagined when HMGTs are coupled with HMDs. Gaze as an input mechanism may find many meaningful applications once these two technologies are integrated in a wearable computing device. Interaction with a see-through HMD is different than interaction with a regular computer display. Gaze-based interaction with HMD will undoubtedly be an interesting research topic for future work.

Bibliography

[1] Alastair G. Gale, K. P. The ergonomics of attention responsive technology. Full research report, 2007. 16, 129

[2] Beach, G., Cohen, C. J., Braun, J., and Moody, G. Eye tracker system for use with head mounted displays. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on* (1998), vol. 5, p. 4348–4352. 23

[3] Beauchemin, S. S., and Barron, J. L. The computation of optical flow. *ACM Comput. Surv. 27*, 3 (Sept. 1995), 433–466. 122

[4] Bernet, S., Cudel, C., Lefloch, D., and Basset, M. Autocalibration-based partioning relationship and parallax relation for head-mounted eye trackers. *Machine Vision and Applications 24*, 2 (Feb. 2013), 393–406. 21

[5] Bonino, D., Castellina, E., Corno, F., Garbo, A., and Pellegrino, P. Control application for smart house through gaze interaction,". In *Proceedings of the 2nd COGAIN Annual Conference on Communication by Gaze Interaction, Turin, Italy* (2006). 16

[6] Brunelli, R. *Template matching techniques in computer vision: theory and practice*. Wiley, Chichester, U.K, 2009. 119

[7] Carpenter, R. H. S. *Movements of the eyes*. Pion, 1988. 8

[8] Crawford, J. D., and Vilis, T. Axes of eye rotation and listing's law during rotations of the head. *Journal of neurophysiology 65*, 3 (Mar. 1991), 407–423. PMID: 2051188. 9

[9] Duchowski, A. *Eye Tracking Methodology: Theory and Practice*, 2nd ed. Springer, July 2007. v, 2, 8

[10] "eye muscles". http://one.aao.org/, May 2012. 6, 129

[11] "EYEBALL". eyeball (anatomy) – britannica online encyclopedia, May 2012. 5, 129

[12] HANDA, S., AND EBISAWA, Y. Development of head-mounted display with eye-gaze detection function for the severely disabled. In *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2008. VECIMS 2008. IEEE Conference on* (2008), p. 140–144. 23

[13] HANSEN, D., AND JI, Q. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 3 (2010), 478–500. 9, 10

[14] HENNESSEY, C. *Point-of-gaze estimation in three dimensions.* PhD thesis, The University Of British Columbia, 2008. 10

[15] HUA, H., KRISHNASWAMY, P., AND ROLLAND, J. P. Video-based eyetracking methods and algorithms in head-mounted displays. *Optics Express 14*, 10 (2006), 4328–4350. 23

[16] JACOB, R. J. K. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Trans. Inf. Syst. 9*, 2 (Apr. 1991), 152–169. v, 15

[17] JAMES, W. *The Principles of Psychology, Vol. 1*, reprint ed. Dover Publications, June 1950. 13, 22

[18] KUGLIN, C., AND HINES, D. The phase correlation image alignment method. *IEEE Conference on Cybernetics and Society* (1975), 163–165. 122

[19] KWON, Y.-M., AND SHUL, J. K. Experimental researches on gaze-based 3D interaction to stereo image display. In *Technologies for E-Learning and Digital Entertainment*. Springer, 2006, p. 1112–1120. 10

[20] LAND, M., AND TATLER, B. *Looking and Acting: Vision and eye movements in natural behaviour*, 1 ed. Oxford University Press, USA, Oct. 2009. 7, 8, 22

[21] LI, D. *Low-cost eye-tracking for human computer interaction.* Master's thesis, Iowa State University, 2006. 21

[22] MOLLENBACH, E. *Selection Strategies in Gaze Interaction.* PhD thesis, Loughborough University, 2010. 15

[23] MORIMOTO, C., KOONS, D., AMIT, A., FLICKNER, M., AND ZHAI, S. Keeping an eye for HCI. In *XII Brazilian Symposium on Computer Graphics and Image Processing, 1999. Proceedings* (1999), pp. 171–176. 2

[24] Nishino, K., and Nayar, S. The world in an eye [eye image interpretation]. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004* (2004), vol. 1, pp. I–444–I–451 Vol.1. 12

[25] Novák, P., Stepankova, O., Uller, M., Novakova, L., and Moc, P. Home and environment control. *COGAIN 2009 Proceedings. Lyngby: Technical University of Denmark* (2009), 35–38. 16

[26] Panerai, F., and Sandini, G. Oculo-motor stabilization reflexes: integration of inertial and visual information. *Neural Networks 11*, 7 (1998), 1191–1204. 9

[27] Park, H. M., Lee, S. H., and Choi, J. S. Wearable augmented reality system using gaze interaction. In *Proceedings of the 7th IEEE/ACM international Symposium on Mixed and Augmented Reality* (2008), p. 175–176. 23

[28] Pederson, T., Janlert, L.-E., and Surie, D. Towards a model for egocentric interaction with physical and virtual objects. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (2010), p. 755–758. 22

[29] Perez, K. S., Vaught, B. I., Lewis, J. R., Crocco, R. L., and Kipman, A. A.-A. Enhancing an object of interest in a see-through, mixed reality display device, Feb. 2013. U.S. Classification 348/47, 348/E13.74, 345/633; International Classification G09G5/00, H04N13/02. 23

[30] Pomplun, M., and Sunkara, S. Pupil dilation as an indicator of cognitive workload in human-computer interaction. In *Proceedings of the International Conference on HCI* (2003). 12

[31] Purves, and Dale and Augustine, George J and Fitzpatrick, David and Katz, Lawrence C and LaMantia, Anthony-Samuel and McNamara, James O and Williams, S Mark and others. Eye movements and sensory motor integration. In *Neuroscience*, 2 ed. Sinauer Associates, 2001, p. 457. 8, 9

[32] Razzak, F., Castellina, E., and Corno, F. Environmental control application compliant with COGAIN guidelines. *Politecnico di Torino* (2009). 16

[33] Schweigart, G., Maurer, C., and Mergner, T. Combined action of smooth pursuit eye movements, optokinetic reflex and vestibulo-ocular reflex in macaque monkey during transient stimulation. *Neuroscience letters 340*, 3 (2003), 217. 120

[34] Sereno, S. C., and Rayner, K. Measuring word recognition in reading: eye movements and event-related potentials. *Trends in Cognitive Sciences 7*, 11 (Nov. 2003), 489–493. 8

[35] Singh, H., Bhatia, J. S., and Kaur, J. Eye tracking based driver fatigue monitoring and warning system. In *2010 India International Conference on Power Electronics (IICPE)* (2011), pp. 1–6. 11

[36] Treiber, M. *An introduction to object recognition: selected algorithms for a wide variety of applications*. Advances in pattern recognition. Springer, London ; New York, 2010. 13

[37] Ukmar, M. Cranial nerves: anatomy, pathology, imaging. *Clinical Imaging 35* (2011), 163–164. 9

[38] Watson, N. V. *The mind's machine: foundations of brain and behavior*. Sinauer Associates, Sunderland, Mass, 2012. 9, 129

List of Figures

1.1 The application domains of the head mounted gaze trackers have been limited to the diagnostic applications. Curtesy of Tobii Technologies[2] 2

1.2 (left) Gaze-based interaction with one computer display using RGT (right) mobile gaze-based interaction with objects in the environment using HMGT. 3

1.3 Human eyeball. Adapted from [11] 5

1.4 Eye image and the reflection of the light on the cornea 6

1.5 Extraocular muscles. Adapted from [10] 6

1.6 Head rotations and the corresponding rotational and translational reflexive movements of the iris/pupil. The image in the center is adapted from [38]. 9

1.7 (left) RGT estimates the gaze point in a fixed two-dimensional space e.g., computer display. (right) In contrast, HMGT estimates the gaze in the user's FoV. 10

1.8 HMGTs that don't have a scene camera, usually estimate the line of sight in the head coordinates system (H) which can move relative to the world coordinates system (W) 11

1.9 Many different types of information can be obtained from the eye and the scene cameras of a HMGT. Curtesy of Tobii Technologies[3] 12

1.10 Gaze interaction in 3D where the gaze is used for (a) pointing to an item or a point (that indicates the next position of the cursor) on a display, (b) pointing to an object of interest in the environment (stationary or non-stationary), (c) for determining the destination point of a remote vehicle in space 14

1.11 Attention Responsive Technology (ART) proposed by [1] . . . 16

1.12 Parallax error in a HMGT 20

1.13 Real-time compensation for parallax error 22

13.1 (a) Gaze estimation in a RGT is independent of head rotations, (b) when the actual PoR is fixed the estimated gaze point in the scene image of a HMGT changes by rotating the head 119
13.2 The direction of the VOR movements caused by the head yaw and the direction of the smooth pursuit movements caused by fixating on a moving target 120
13.3 Sum of the VOR and SP movements and the final eye movements 121

List of Tables

1.1 Eye muscles 7
1.2 The main limitations of gaze activation techniques (from the point of view of interaction in 3D) 17