Event Detection in Eye-Tracking Data for Use in Applications with Dynamic Stimuli

Larsson, Linnéa

2016

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

Event Detection in Eye-Tracking Data for Use in Applications with Dynamic Stimuli

Linnéa Larsson

LUND
UNIVERSITY

Doctoral Dissertation, March 4, 2016

Department of Biomedical Engineering
Lund University
P.O. Box 118, SE-221 00 LUND
SWEDEN

ISBN: 978-91-7623-663-5 (print)
ISBN: 978-91-7623-664-2 (pdf)
ISRN: LUTEDX/TEEM - 1101 - SE
Report No. 2/16

Printed in Sweden by *Tryckeriet i E-huset*, Lund.
February 2016.

To my family

"Learn from yesterday, live for today, hope for tomorrow.
The important thing is not to stop questioning."
Albert Einstein

Populärvetenskaplig sammanfattning

Det sägs att ögonen är själens spegel och att man genom att titta på någons ögon kan säga något om personens sinnesstämning och hur personen mår. Forskning kring ögonrörelser har visat att man genom att mäta ögats rörelser kan tolka hur den visuella informationen som vi tar in genom ögonen har behandlats. Eftersom det är hjärnan som styr muskulaturen runt ögat som i sin tur kontrollerar ögats rörelser, så kan man genom att studera ögonrörelser dra slutsatser om hjärnans funktion i de delar som styr ögats muskler. Ögonrörelser mäts idag genom att en videokamera filmar ögat och med hjälp av bildbehandling skattas blickens position. En sådan utrustning kallas för en eye-tracker eller ögonrörelsemätare. Idag används ögonrörelsemätare bland annat för att analysera relationen mellan våra ögonrörelser och motsvarande kognitiva processer i hjärnan, till exempel då vi läser en text. För att kunna analysera och förstå denna relation behöver den inspelade ögonrörelsesignalen delas in i olika typer av ögonrörelser. De vanligaste typerna av ögonrörelser är fixeringar, sackader, och mjuka följerörelser. Problemet med nuvarande metoder för klassificering av ögonrörelser är att de oftast är utvecklade för att användas till signaler som är inspelade när en person tittar på statiska bilder, vilket medför att metoderna inte kan hantera mjuka följerörelser. Dessutom saknas standardiserade metoder för att jämföra och utvärdera existerande metoder.

Denna avhandling handlar om att utveckla metoder för att dela upp och klassificera segment av den inspelade ögonrörelsesignalen i de vanligaste typerna av ögonrörelser oberoende av vilken typ av stimuli som har använts vid inspelningen. Avhandlingen behandlar även olika sätt att utvärdera metoder för klassificering av ögonrörelser. I det första arbetet har en metod utvecklats för att klassificera sackader i signaler som är inspelade när personer tittar både på bilder och rörliga videoklipp. Förutom sackader, så klassificeras även så kallade post-sackadiska oscillationer (PSO), som är snabba oscillerande rörelser som följer direkt efter vissa sackader.

PSO ses ofta som en störning och om de är ögonrörelser eller inte är fortfarande inte helt klarlagt. Denna nya metod gör det möjligt att studera sackader i signaler som är inspelade under rörliga videoklipp, där tidigare metoder haft problem att hitta och avgränsa sackader.

I det andra arbetet har en metod utvecklats som delar upp intervallen mellan de detekterade sackaderna och möjliga PSO i fixeringar och mjuka följerörelser. För att separera de två typerna av rörelser beräknas signalens spatiala utbredning och riktning i både långa och korta tidsskalor. Metodens prestanda utvärderas med fem olika mått som beskriver både generell och detaljerad prestanda för klassificeringen.

Ofta vid inspelning av ögonrörelser spelas signaler från båda ögonen in, men det finns få algoritmer som drar nytta av informationen från båda ögonen. I det tredje arbetet har en metod utvecklats som använder signaler från båda ögonen för att bättre kunna separera mjuka följerörelser från fixeringar. Genom att använda signalerna från båda ögonen kan synkroniseringen mellan ögonen studeras och motverka att drift under fixeringar blir felaktigt klassificerade som mjuka följerörelser. Utöver en ny metod för klassificering av ögonrörelser föreslås i det tredje arbetet även en ny utvärderingsmetod. Utvärderingsmetoden baseras på automatiskt detekterade rörliga objekt i de videoklipp som används vid inspelningen av ögonrörelserna. Genom att jämföra tidpunkter då ögat rör sig samstämmigt med något rörligt objektet kan mjuk följerörelse utvärderas utan att tidskrävande manuella annoteringar behöver användas.

I de tre första arbetena används en stationär ögonrörelsemätare med hög samplingsfrekvens. I det fjärde arbetet används istället en mobil ögonrörelsemätare i form av ett par glasögon. När en mobil ögonrörelsemätare används kan personen fritt röra huvudet. I det fjärde arbetet har därför en metod utvecklats för att kompensera för huvudrörelser i den inspelade ögonrörelsesignalen. Huvudrörelserna mäts med hjälpa av en sensor (IMU) som placeras på personens huvud och skattar dess orientering. Den kompenserade ögonrörelsesignalen används sedan tillsammans med automatiskt detekterade objekt från scenvideon för att detektera sackader, fixeringar, och mjuka följerörelser.

Totalt utgör de fyra delarna i avhandlingen en metodplattform för robust detektering av olika typer av ögonrörelser vid dynamisk stimulus, dvs. när man tittar på rörliga bilder eller en rörlig scen. I plattformen ingår även metoder för utvärdering av algoritmerna som är oberoende av om en stationär eller en mobil ögonrörelsemätare har använts.

Abstract

This doctoral thesis has signal processing of eye-tracking data as its main theme. An eye-tracker is a tool used for estimation of the point where one is looking. Automatic algorithms for classification of different types of eye movements, so called events, form the basis for relating the eye-tracking data to cognitive processes during, e.g., reading a text or watching a movie. The problems with the algorithms available today are that there are few algorithms that can handle detection of events during dynamic stimuli and that there is no standardized procedure for how to evaluate the algorithms.

This thesis comprises an introduction and four papers describing methods for detection of the most common types of eye movements in eye-tracking data and strategies for evaluation of such methods. The most common types of eye movements are fixations, saccades, and smooth pursuit movements. In addition to these eye movements, the event post-saccadic oscillations, (PSO), is considered. The eye-tracking data in this thesis are recorded using both high- and low-speed eye-trackers.

The first paper presents a method for detection of saccades and PSO. The saccades are detected using the acceleration signal and three specialized criteria based on directional information. In order to detect PSO, the interval after each saccade is modeled and the parameters of the model are used to determine whether PSO are present or not. The algorithm was evaluated by comparing the detection results to manual annotations and to the detection results of the most recent PSO detection algorithm. The results show that the algorithm is in good agreement with annotations, and has better performance than the compared algorithm.

In the second paper, a method for separation of fixations and smooth pursuit movements is proposed. In the intervals between the detected saccades/PSO, the algorithm uses different spatial scales of the position signal in order to separate between the two types of eye movements. The algorithm is evaluated by computing five different performance measures, showing both general and detailed aspects of the discrimination performance. The performance of the algorithm is compared to the performance of a velocity and dispersion based algorithm, (I-VDT), to the per-

formance of an algorithm based on principle component analysis, (I-PCA), and to manual annotations by two experts. The results show that the proposed algorithm performs considerably better than the compared algorithms.

In the third paper, a method based on eye-tracking signals from both eyes is proposed for improved separation of fixations and smooth pursuit movements. The method utilizes directional clustering of the eye-tracking signals in combination with binary filters taking both temporal and spatial aspects of the eye-tracking signal into account. The performance of the method is evaluated using a novel evaluation strategy based on automatically detected moving objects in the video stimuli. The results show that the use of binocular information for separation of fixations and smooth pursuit movements is advantageous in static stimuli, without impairing the algorithm's ability to detect smooth pursuit movements in video and moving dot stimuli.

The three first papers in this thesis are based on eye-tracking signals recorded using a stationary eye-tracker, while the fourth paper uses eye-tracking signals recorded using a mobile eye-tracker. In mobile eye-tracking, the user is allowed to move the head and the body, which affects the recorded data. In the fourth paper, a method for compensation of head movements using an inertial measurement unit, (IMU), combined with an event detector for lower sampling rate data is proposed. The event detection is performed by combining information from the eye-tracking signals with information about objects extracted from the scene video of the mobile eye-tracker. The results show that by introducing head movement compensation and information about detected objects in the scene video in the event detector, improved classification can be achieved.

In summary, this thesis proposes an entire methodological framework for robust event detection which performs better than previous methods when analyzing eye-tracking signals recorded during dynamic stimuli, and also provides a methodology for performance evaluation of event detection algorithms.

Preface

The doctoral thesis comprises an introduction and four parts describing methods for detection of common types of eye movements in eye-tracking data and strategies for evaluation of such methods. The four parts are based on the following papers:

[1] Linnéa Larsson, Marcus Nyström, and Martin Stridh, "Detection of saccades and postsaccadic oscillations in the presence of smooth pursuit," in *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 9, pp. 2484–2493, 2013.

[2] Linnéa Larsson, Marcus Nyström, Richard Andersson, and Martin Stridh, "Detection of Fixations and Smooth Pursuit Movements in High-Speed Eye-Tracking Data," in *Biomedical Signal Processing and Control*, vol. 18, pp. 145–152, April 2015.

[3] Linnéa Larsson, Marcus Nyström, Håkan Ardö, Kalle Åström, and Martin Stridh, "Smooth Pursuit Detection in Binocular Eye-Tracking Data with Automatic Video-Based Performance Evaluation," Submitted for publication.

[4] Linnéa Larsson, Andrea Schwaller, Marcus Nyström, and Martin Stridh, "Head Movement Compensation and Multi-Modal Event Detection for Mobile Eye-Trackers," Submitted for publication.

In Papers I-IV, the author of this thesis designed the experiments, recorded the data, developed and implemented the algorithms, and prepared the manuscripts. Parts of the work have been presented at the following conferences:

[5] Linnéa Larsson, Martin Stridh, and Marcus Nyström, "Event detection in data with static and dynamic stimuli," *16th European Conference on Eye Movements*, Marseille, France, pp. 37, August, 2011.

[6] Linnéa Larsson, Martin Stridh, and Marcus Nyström, "Detection of fixations and smooth pursuit eye movements using local and global properties of the eye-tracking signal," Book of Abstracts of the 17th European Conference on Eye Movements, in Lund, Sweden. *Journal of Eye Movement Research*, 6(3), pp. 250, August, 2013.

[7] Linnéa Larsson, Marcus Nyström, and Martin Stridh, "Discrimination of fixations and smooth pursuit eye movements in high-speed eye-tracking data," in *Proc. 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Chicago, USA, pp. 3797–3800, August, 2014.

[8] Linnéa Larsson, Andrea Schwaller, Kenneth Holmqvist, Marcus Nyström, and Martin Stridh, "Compensation of head movements in mobile eye-tracking data using an inertial measurement unit," in *Proc. of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1161–1167, Sept., 2014.

Acknowledgments

There are many people who I would like to thank for these five years and who have in different ways contributed to the work that is summarized in this thesis.

First of all, I would like to express my gratitude to my two supervisors, Martin Stridh and Marcus Nyström, who have supported and encouraged me throughout these five years that we have worked together. Thank you Martin for your enthusiasm for trying new methods and new ideas on how to solve algorithmic problems, and for our daily chats about general things in life. Thank you Marcus for your expertise in eye-tracking and for sharing your enthusiasm for this technique. To both of you, I see very good friends in you. I give you my best wishes for the continuation of this project and I hope we keep in touch also in the future.

Thanks to the Lund University Humanities Laboratory for giving me the opportunity to use your facilities and equipment when performing my experiments. Thanks also to the eye-tracking group for sharing your knowledge and experience regarding eye-tracking with me.

Thanks to current and former colleagues in the signal processing group, Mattias, Hamid, Egle, Mikael S, Mikael H, Ulrike, Nedo, Frida, Martin, Bengt, and Leif, for all your help and fruitful signal processing discussions.

Thanks to my colleagues at Department of Electro- and Information Technology, especially Anders J, Nafishe, and Isael, for the first three years of my PhD-studies. Thanks to all colleagues at Department of Biomedical Engineering for joyful discussions during coffee breaks during the last two years. A big thank you to the administrative staff at both departments for all your help with practical issues. During the last six months I got the opportunity to share office with Roger; thank you for your always so positive and optimistic view of life.

Thanks to the participants of my experiments and to the eSSENCE project for the funding that made this thesis possible.

Thanks also to my two loyal training companions, Mattias and Oskar, for sharing the triathlon and rollerski experiences with me and for getting me to the training

sessions even when I am not in the mood for training.

Thanks to my parents, Irene and Bosse, my brother Anton, and my sister Maja, for the support you always give me.

Finally, I would like to say thank you Oskar. I am forever grateful for your patience, your love, and your support, during the last four years.

Linnéa Larsson

List of Acronyms and Abbreviations

AR Autoregressive

BIT Binocular Individual Threshold algorithm

C-DT Covariance Dispersion Threshold algorithm

CR Corneal Reflection

CWT-SD Continous Wavelet Transform Saccade Detection

DOF Degrees of Freedom

DPI Dual-Purkinje-image eye-tracker

EHM Eye- and Head Movements

EM Eye Movements

EOG Electro-oculography

HM Head Movements

I-DT Identification with Dispersion Threshold

I-HMM Identification with Hidden Markov Model

IMU Inertial Measurement Unit

I-PCA Identification with Principle Component Analysis

I-VDT Identification with Velocity and Dispersion Threshold

I-VMP Identification with Velocity and Movement Pattern

I-VT Identification with Velocity Threshold

I-VVT Identification with Velocity and Velocity Threshold

OKN Optokinetic Nystagmus

PSO Postsaccadic Oscillations

POR Point-of-Regard

RMSE Root Mean Square Error

SD Standard Deviation

SLAM Simultaneous Localization And Mapping

VGM Video-Gaze Model

VOG Video-oculography

VOR Vestibular Ocular Reflex

Contents

Populärvetenskaplig sammanfattning **v**

Abstract **vii**

Preface **ix**

Acknowledgments **xi**

List of Acronyms and Abbreviations **xiii**

I Introduction **1**

1 Thesis Introduction **3**

2 The Human Eye **5**
2.1 Anatomy of the eye 5
2.2 Eye Movements 6

3 The Eye-Tracker **11**
3.1 Video-oculography (VOG) 11
3.2 Other methods 18

4 Event Detection **21**
4.1 Events 21
4.2 Event detection algorithms 26
4.3 Performance evaluation 35
4.4 Available databases and algorithms 38

5 Analysis of Mobile Eye-Tracking Data **41**
5.1 Systems to track head- and body movements 41
5.2 Analyzing mobile eye-tracking data through the scene video 45

6 Summary of the Included Papers **47**

References **56**

II Included Papers **67**

PAPER I – Detection of Saccades and Postsaccadic Oscillations in the Presence of Smooth Pursuit **71**
1 Introduction 73
2 Methods 76
3 Experiment and database 86
4 Results 89
5 Discussion 94
6 Conclusions 96
References 96

PAPER II – Detection of Fixations and Smooth Pursuit Movements in High-Speed Eye-Tracking Data **101**
1 Introduction 103
2 Methods 104
3 Experiment and database 110
4 Results 110
5 Discussion 118
6 Conclusions 121
References 122

PAPER III – Smooth Pursuit Detection in Binocular Eye-Tracking Data with Automatic Video-Based Performance Evaluation **127**
1 Introduction 129
2 Methods 131
3 Experiment and database 142
4 Results 145
5 Discussion 150
6 Conclusions 151
References 152

PAPER IV – Head Movement Compensation and Multi-Modal Event Detection for Mobile Eye-Trackers **157**
1 Introduction 159
2 Methods 160
3 Experiment and database 175
4 Results 178
5 Discussion 188
6 Conclusions 190
References 190

Part I

Introduction

Chapter 1
Thesis Introduction

This doctoral thesis deals with the development of methods for event detection in eye-tracking signals, and strategies for evaluation of such methods. An eye-tracker is a tool for estimation of where a person is looking, i.e., the point of gaze. Automatic detection of different types of eye movements in the eye-tracking signal, so called events, forms the basis for researchers that use eye-tracking to understand the relationship between eye movements and the corresponding processes in the brain. Eye-tracking is used in as diverse fields of research as psychology, cognitive science, neurology, medicine, engineering, and economics, to mention some. Eye-tracking research is thus often highly inter-disciplinary, which is also reflected in how eye-tracking hardware and software have been developed over the years.

Eye-trackers have at the same pace as other electronics become smaller, lighter, and cheaper, and have therefore become more easily accessible for a larger group of researchers during recent years. Most of the hardware in today's eye-trackers are quite mature in the sense that they can record eye movements in various environments and for different participants. A major bottleneck for the continued progress in eye-tracking research is the need for improved algorithms to perform the analysis of the recorded eye-tracking signals. Especially, there is a need for analysis software for recordings where dynamic stimuli are used. Dynamic stimuli refer to that the objects that the user is looking at are moving either in the environment, when a mobile eye-tracker is used, or on a computer screen in a stationary setup.

A majority of the available algorithms are developed to detect two of the most common types of eye movements which occur when data are recorded using a stationary eye-tracker and when viewing static stimuli. Recently, however, the interest in using dynamic stimuli has grown, both for stationary and mobile eye-tracking. When dynamic stimuli are used, additional types of eye movements occur in the data. The problem with many of the current algorithms is that, since they are not developed for this purpose, they may behave unreliably or even erroneously for more complex data.

Event detection in eye-tracking data is associated with many challenges. One of those is that many different types of noise and disturbances may occur in the recorded signals which originate both from the eye-tracker and from individual differences among the users. This variability between measurements and individuals may create signals that are difficult to analyze. The challenge is therefore to develop algorithms that are flexible enough to be used for signals that contain various types of events and disturbances, and that can handle both different individuals and different types of eye-trackers. An additional challenge in event detection of eye-tracking signals is how to evaluate and compare different algorithms. In many signal processing applications, the algorithms are evaluated by calculating the performance for simulated signals. It is, however, a challenge to construct eye-tracking signals that capture the variations and the disturbances in real signals to such an extent that they are authentic and are useful for performance evaluation. Without a standard procedure for how to perform the evaluation, it is also difficult to compare the performances of algorithms from different research groups.

In the present thesis, four papers are included which in different ways deal with event detection in eye-tracking signals. The central themes of the papers are:

I. Detection of saccades and post-saccadic oscillations in high-speed eye-tracking data when static and dynamic stimuli are used.

II. Detection of fixations and smooth pursuit movements in high-speed eye-tracking data when static and dynamic stimuli are used.

III. Detection of smooth pursuit movements in binocular eye-tracking signals combined with automatic performance evaluation based on objects detected in the stimuli videos.

IV. Compensation of head movements and multi-modal event detection in signals recorded using a mobile eye-tracker.

In the following chapters, an introduction to the eye-tracking research field is provided. In Chapter 2, an overview of the anatomy and physiology of the eye is given. Chapter 3 contains a description of the principles of an eye-tracking system and how the gaze of the user is estimated. The current state-of-the-art of event detection algorithms is summarized in Chapter 4, and in Chapter 5 an overview of methods for analyzing mobile eye-tracking data is given. Finally, in Chapter 6, the included papers and the main contributions are summarized.

Chapter 2
The Human Eye

2.1 Anatomy of the eye

The sensory systems in the human body consist of sensory receptor cells that are stimulated from internal or external sources in the body, neural pathways that transfer the sensory information to the brain, and parts of the brain where the sensory information is processed [1]. In the human body, there are several sensory systems, e.g., the auditory for hearing, the vestibular for balance, and the visual system for vision. The visual system is the sensory system that makes it possible for us to process visual information that we capture through our eyes [1]. It comprises the eyeball, the muscles surrounding it, and the neural pathway transmitting the signals to the brain. The function of the eyes in the visual system is to focus light from objects around us to the rear part of the eyeball and convert the light to electrical signals that are transmitted to the brain for further processing [2]. The eye is a liquid-filled ball that is enclosed by a white surface called the sclera. An illustration of the human eye is shown in Fig. 2.1. From the outside, parts of the sclera are seen together with the colored iris and the black pupil. The sclera surrounds the eyeball except for its most anterior surface which is the thin transparent and protective layer called the cornea. The cornea is the first medium of the eye to reflect and refract incoming light, before it passes through the pupil and further to the lens where the light refracts once more [3]. The size of the pupil changes with the ambient light conditions and controls the amount of light entering the eye and the lens. When the light refracts in the lens, fine adjustments are performed before the light continues through the liquid filled globe to the rear parts of the eyeball, the retina. The retina is a thin layer of tissue that covers most of the inner walls of the eyeball. It is sensitive to light and consists mainly of photoreceptor cells, nerve cells, and glial cells [4]. There are two types of photoreceptive cells: rods and cones. These two types of photoreceptive cells have different functions; cones enable color vision and high visual acuity, while the rods are important for night vision and for detection

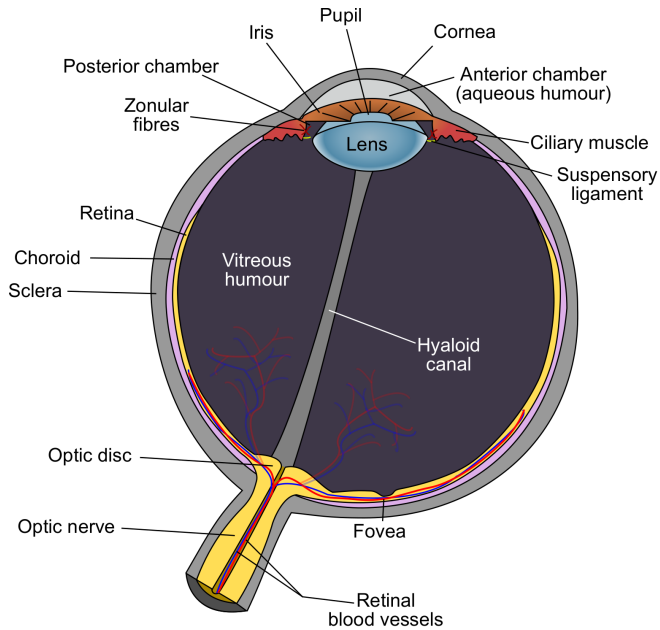


Figure 2.1: An illustration of the anatomy of the human eye, from Wikimedia Commons.

of motion. In the retina, there are about 100 millions of rods and 5 millions of cones [4]. The density of cones and rods are unevenly distributed over the retina, see Fig. 2.2. Fovea centralis is the spot on the retina where the concentration of cones is the highest. Moving only a few millimeters outside the fovea centralis the concentration of cones decreases and the concentration of rods increases. Since it is only in regions with high concentrations of cones where objects can be clearly seen, it is only when light hits the fovea that we are able to see an image with high resolution. The rods and cones absorb the photons and convert them to electrical signals that are transmitted via the optical nerve to the brain.

2.2 Eye Movements

The main purpose of eye movements is to direct the eyes towards the object of interest or to keep the object of interest at the center of the fovea in order to provide a clear vision of the object. In order for the eye to move, three pairs of muscles are attached to each of the eyeballs. These muscles make it possible for the eye to move, within its orbit, vertically, horizontally, and torsionally [5]. The movements of the eye are divided into seven functional classes, see Table 2.1. Each functional class is described further in the following subsections.

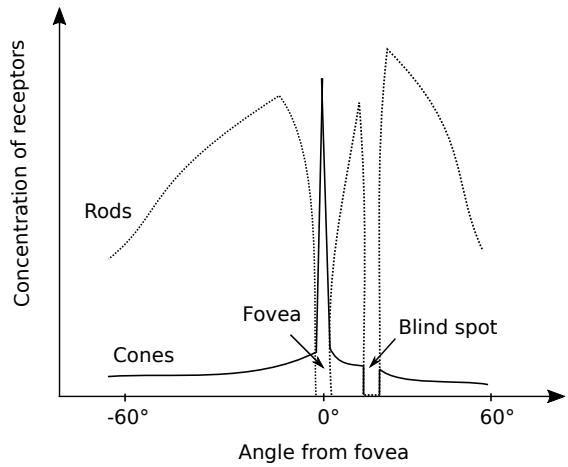


Figure 2.2: An illustration of the distribution of the cones and rods on the retina, adopted from Wikimedia Commons.

2.2.1 Fixations

A fixation is a movement when the eye is more or less still and focuses on an object. The function of a fixation is to stabilize the image on the fovea, so that it can be seen clearly. The small movements during a fixation can be divided into three types: tremor, slow drift, and microsaccades [6]. Tremor is a small wave-like motion of the eye, with an amplitude around $0.01°$ and a frequency below 150 Hz [7]. The function of tremor is still largely unknown [6]. Drift is a slow motion of the eye, which occurs simultaneously with tremor [6]. It was for a long time believed that drift was a random motion of the eye due to instability in the oculomotor system. Later, it was found that drift has a compensatory role to maintain visual acuity during fixations, when there are not sufficiently many microsaccades [6]. A microsaccade is the fastest of the fixational eye movements and has a duration of about 25 ms [6]. Microsaccades occur around 1-2 times per second, depending on the task [8]. Even though the amplitude of a microsaccade is lower than for a normal saccade, they share many properties. Recently, it was found that microsaccades may be voluntary movements when performed during natural tasks [8].

Table 2.1: The functional classes of eye movements, inspired by [7].

Class of Eye Movement	**Main Function**
Fixation	Holds the image of a stationary object on the fovea
Saccade	Brings images of objects of interest onto the fovea
Smooth Pursuit	Holds the image of a moving target on the fovea
Vergence	Moves the eyes in opposite directions so that the images of a single object from the two eyes are placed or held simultaneously on their respective fovea
Vestibular	Holds images of the seen world steady on the retina during brief head rotations or translations
Optokinetic	Holds images of the seen world steady on the retina during sustained head rotations
Nystagmus quick phases	Resets the eyes during prolonged rotation and directs gaze towards the scene that will come

2.2.2 Saccades

The saccade is the fastest of the eye movements and its main purpose is to change the gaze from one object of interest to the next. A typical saccade has a duration between 30 and 80 ms and a velocity between 30°/s and 500°/s [9]. A relationship exists between the duration, amplitude, and velocity of a saccade, which suggests that larger saccades have larger velocities, and last longer [10]. The latency in the saccadic system is around 200 ms, and corresponds to the time from the onset of the stimulus to the initiation of the eye movement. This includes the time it takes for the central nervous system to determine whether a saccade should be initiated or not, and, if this is the case, calculate the distance that the eye should move, and transmit the neural pulses to the muscles that move the eyes. A common assumption is that a saccade is a straight line between point A and point B. However, in reality, a saccade is seldom a straight line, instead it most often has a slightly curved trajectory [11].

2.2.3 Smooth pursuit movements

A smooth pursuit is performed when the eyes track a moving object, e.g., follow a bird that flies across the sky. A smooth pursuit movement can only be performed when there is a moving object to follow [12]. The latency of the smooth pursuit system is about 100 ms, which is slightly shorter than for saccades [9]. The latency of the smooth pursuit system corresponds to the time it takes for the eye to start moving from the onset of the target motion. A smooth pursuit movement can broadly be divided into two stages: open-loop and closed-loop [13]. The open-loop stage is the pre-programmed initiation stage of the smooth pursuit where the eye accelerates in order to catch up with the moving target. The closed-loop stage starts when the eye has caught up with the target and follows it with a velocity similar to that of the target. In order to be able to follow the moving target in the closed-loop stage, the velocity of the moving target is estimated and compared to the velocity of the eye. If the velocity of the two are different, e.g., the eye lags behind the moving target, a movement known as a catch-up saccade is performed in order to catch up with the target again. The human eye can follow a target at velocities up to 100°/s [14]. The higher the velocity of the moving target, the more catch-up saccades are needed in order for the eye to be able to follow the target. However, most often smooth pursuit movements have velocities below 30°/s. If the stimulus only consists of one moving target that moves in a predictable way, the eye will be able to follow it more accurately, with fewer catch-up saccades [7].

2.2.4 Vergence eye movements

When the two eyes point at the same object, the eyes need to be directed in slightly different directions. This is due to the fact that the two eyes are separated by a few

centimeters. The movements that the eyes perform when they either move towards each other, convergence, or away from each other, divergence, are often referred to as vergence eye movements [7]. Each eye needs to be controlled separately, in order to keep the same object on the fovea of both eyes. This is especially important for objects that are at a close distance. In order for the brain to be able to combine the images of the object from the two eyes into one, the object must lie on the corresponding spot on the retina of each eye. The maximum visual angle that the object can be apart on the retina is called Panum's area [7]. This means that if the object seen from the two eyes are within this area the images are combined into one, and if not, the object will be interpreted as two and double vision will occur [7].

2.2.5 Vestibular eye movements

The function of vestibular eye movements is to stabilize the image on the fovea in order to sustain clear vision during head rotations [7]. Since the vestibular eye movements respond to signals from the vestibular system, the latency for these eye movements is shorter than for eye movements initiated by the visual system. The latency of the system can be as short as $7 - 15$ ms, compared to about 200 ms for the saccadic system. The eye movement vestibular ocular reflex, VOR, responds to both translational and rotational head movements, which both are natural movements in everyday life. The translational head movements are performed when the head moves from left to right, up and down, or forward and backward, with the nose pointing in the same direction. For rotational head movements, the head can rotate around three axes: horizontally, vertically, and torsionally. Horizontal rotation corresponds to shaking the head, vertical rotation corresponds to nodding, and torsional rotation corresponds to lying the head against one of the shoulders. The size of the compensatory eye movements that are needed to keep the image on the fovea during these head movements is larger for closer objects than for distant objects.

2.2.6 Optokinetic and Nystagmus quick phase

Optokinetic eye movements are similar to vestibular eye movements in the sense that they are initiated in order to keep the image on the fovea and compensate for head movements. Optokinetic eye movements respond to sustained head rotations, e.g., when sitting in a spinning chair. In order for the eye not to get stuck in the outer part of the eye socket in the opposite direction of the rotation and not be able to make any movements during sustained head rotations, the eye needs to quickly move in the same direction as the rotation, referred to as the quick phase of nystagmus [7].

Chapter 3
The Eye-Tracker

Originally, an eye-tracker referred to an equipment that was used to measure the orientation of the eye, while a gaze-tracker was used to estimate where a person was looking. Over time, these two terms have been used interchangeably, and in the following the term eye-tracker is used to refer to the equipment that tracks the movements of the eyes and that estimates the direction and position of gaze. Over the last 100 years, several different types of measurement techniques have evolved. This chapter gives an overview of different types of eye-trackers, with the emphasis on the most widely used eye-tracker today, video-oculography.

3.1 Video-oculography (VOG)

The type of eye-tracker that is most commonly used today is camera-based and is referred to as video-oculography (VOG). There are three main types of VOG-systems: tower mounted, remote, and mobile [9]. The tower mounted and the remote systems are stationary setups where the user typically sits in front of a computer screen, or a larger monitor, while a mobile eye-tracker is a wearable setup which can be used in a setting outside the laboratory. The technology behind a video-based eye-tracker can be divided into two subparts: *eye detection* and *gaze estimation*. In the eye detection part, the eye is detected and tracked in the images captured by the camera, and in the gaze estimation part, the direction of the gaze is estimated.

3.1.1 Eye detection

With very few exceptions, VOG-systems consist of one or several cameras that record the eye and one or several infrared light sources directed towards the eye. Since the light is infrared, it is not visible to the human eye and will therefore not distract the user [15]. The infrared light sources give rise to reflections in the eye, referred to as Purkinje images. The first Purkinje image, is the reflection in the cornea,

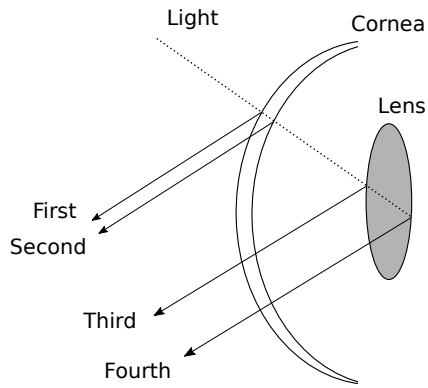


Figure 3.1: An illustration of the four Purkinje images.

and is therefore called the corneal reflection (CR). In the eye, there are four changes in medium that may reflect the incoming light and give rise to Purkinje images. An illustration of these four reflections is shown in Fig. 3.1. To detect and track the eye in the image captured by the camera is a challenging task, e.g., due to occlusion of the eye, the degree of openness of the eye, variation in size, reflections, viewing angle, head pose, eye color, light conditions, and variation in eye shape [15]. Several methods have been proposed in order to overcome these challenges. The methods are divided into three main categories: shape-based, appearance-based, and hybrid methods. The shape-based methods use either models that rely on local features or contours of the eye. A wide range of models have been used, from simple elliptic models to more complex models that take both the shape of the eye and the structure that surrounds it into account. The more simple methods are not robust to changes in light and focus of the camera, and to occlusion, i.e., periods when the user closes the eyelid. On the other hand, the more complex models suffer from being computationally demanding, in need of high resolution images with high contrast, sensitive to changes in pose, and also to occlusion of the eye [15]. The appearance-based methods are based on templates that detect and track the eye based on the distribution of color or responses from a filter bank that enhance desired features in the image. The weaknesses of appearance-based methods are that they are not invariant to scale and rotation of the eye, and since a template is used, it is difficult to capture all variations of human eyes. The hybrid models combine shape-based methods with the appearance-based methods in order to overcome the limitations of each method. One such method uses part-based modeling, which attempts to build a general model out of smaller parts of the image [15]. One limitation with this type of method is that a specific model needs to be built for each person [15].

3.1.2 Gaze Estimation

The goal of the gaze estimation part of the eye-tracker is to convert the information extracted from the image of the eye into a gaze direction or the position of gaze [15]. Most gaze estimation methods are based on features, which means that they extract features such as the contours of the eye and the pupil, and different reflections in the surfaces of the eye and based on these features calculates the direction of gaze [15]. Feature-based methods can be divided into two main categories: interpolation-based methods and model-based methods. Characteristic for the interpolation-based methods is that the extracted features from the image are mapped to gaze coordinates by a mapping function that most often is a parametric function, e.g., a polynomial function. Other nonparametric functions may also be used, e.g., a neural network [16]. In the interpolation-based methods, the gaze position is explicitly calculated without previous calculation of the direction of the gaze. The model-based methods are based on a geometric model of the eye and the objects that are being viewed, and the gaze direction is estimated based on the features extracted from the image of the eye. The position of gaze, referred to the point-of-regard (POR), is estimated as the intersection between the gaze direction and the nearest viewed object, e.g., the monitor.

The simplest VOG eye-tracker is based on one camera and one light source. The idea behind this type of setup is that when the eye moves the pupil moves with it. Instead of measuring the movement of the eye directly, it is indirectly measured by the motion of the pupil in the recorded image. This setup assumes that the CR does not move much when the eye moves, and because of that, the CR can be used as a reference position in the recorded image. Thus, when the user looks in different directions, the relationship between the CR and the pupil changes. By asking the user to look at a number of predefined positions on a monitor, referred to as calibration points, a relationship between the relative positions of the CR and the pupil, and the positions on the monitor can be established. This setup works well when the head is fixated, e.g., for the tower mounted setup. In order to be able to perform gaze estimation in front of a computer screen when the head is allowed to move, e.g., when using a remote system setup, the number of light sources and/or the number of cameras needs to be increased. By using a setup with one camera and multiple light sources the setup is made invariant to head pose, e.g., by placing four IR-light sources on the corners of the monitor that the test person is facing, and by calculating the projection of the light sources on the surface of the cornea, the gaze can be estimated [17]. The method is head pose invariant, but sensitive to changes in depth, i.e., if the distance between the user and the monitor changes.

In the setup with one camera and multiple light sources, there is often a trade off between a wide angle camera that allows for large head movements and an image of the eye with high enough resolution and contrast in order to be able to detect and track the eye in the image. In order to solve this problem multiple cameras

and multiple light sources can be used. In a multiple camera setup, one wide angle camera and one narrow angle camera that is directed towards the eye may be used. When using multiple cameras, the cameras need to be calibrated in order to avoid problems when matching images from the two cameras to each other, and in addition, it is more data to process. For a complete review of eye detection and gaze estimation methods, see [15].

3.1.3 Types of VOG systems and their data

Tower mounted eye-trackers

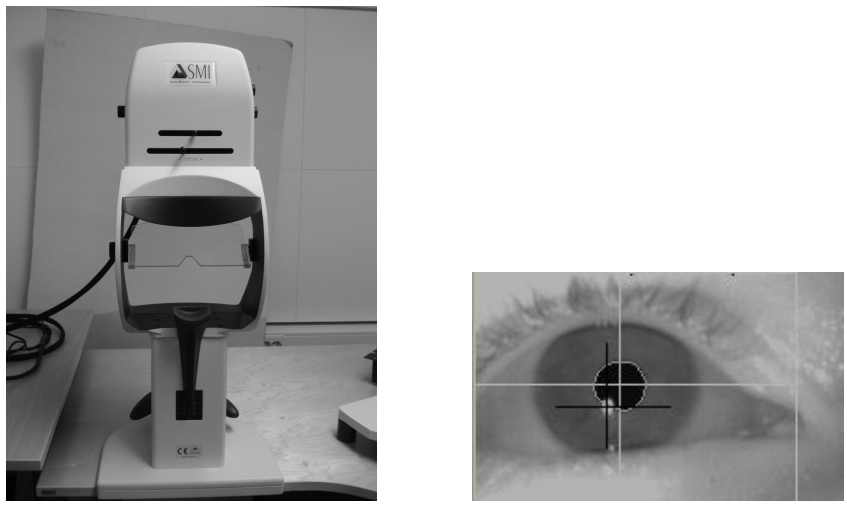
A tower mounted eye-tracker, see Fig. 3.2a, consists of a pillar where the user places the head. At the top of the pillar a camera and an infrared light source pointing downwards are attached. The camera films the eye through a mirror, which is placed in front of the user's eyes. The camera is typically a high speed camera with a frame rate of 1000 – 2000 frames/s. The infrared light source which is directed via the mirror towards the eye, gives rise to the CR. In the image of the eye, captured by the camera, the CR and the pupil are detected, see Fig. 3.2b. The tower mounted eye-tracker requires that the head of the user is fixated in order to record data with high quality.

Because of its size and that the setup requires the user to have the head fixated, a tower mounted eye-tracker is typically used for laboratory experiments. A typical setup is that the tower mounted eye-tracker is placed in front of a computer screen where the stimuli are presented to the user. Eye-tracking data recorded from a tower mounted eye-tracker are shown in Fig. 3.3, where the user is reading a text.

Remote eye-trackers

In a remote eye-tracking system, illustrated in Fig. 3.4a, the camera or cameras are attached below a computer screen. In contrast to the tower mounted eye-tracking system where the image from the camera covers only the eye, the image from a remote camera covers larger parts of the face, see Fig. 3.4b. Since the user has the freedom to move, although within a limited range, the eye detection methods need to be more robust to larger movements of the eyes between frames, than the methods in the tower mounted eye-tracking system. The cameras in a remote system can either be integrated in the computer monitor or be a separate device that can be mounted on any monitor or laptop. Remote eye-trackers are available in many forms, from a low cost web-camera solution that samples at 25 Hz, to fully integrated systems with sampling frequencies up to 1000 Hz.

Since the user is able to move during the recordings, remote eye-trackers are very popular. But the freedom for the user to move during the recording comes at the cost of less precise and less accurate data compared to data recorded with the



(a) **(b)**

Figure 3.2: (a) An example of a tower mounted VOG-system. (b) An image of an eye captured with the camera of the system in (a), where the detected pupil and CR are marked with a white and a black cross.

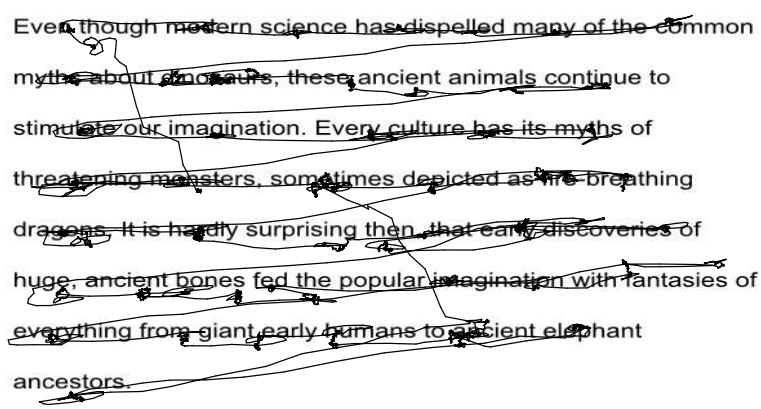
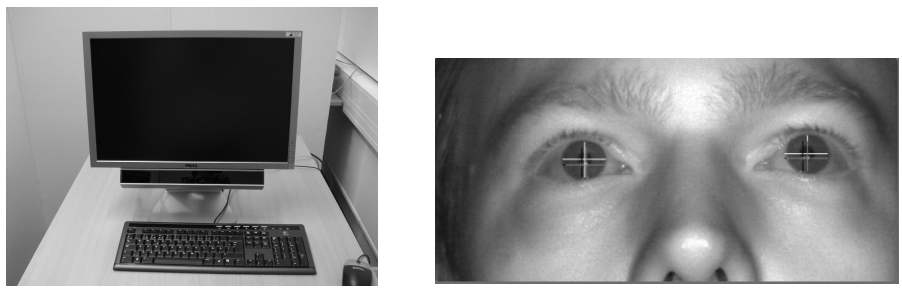


Figure 3.3: Eye-tracking data from a person reading a text. The data is recorded with a tower mounted system.



(a) (b)

Figure 3.4: (a) An example of a remote VOG-system. (b) An image of the eyes captured with a remote VOG-system, where the detected pupils and CRs are marked with crosses.

(a) (b)

Figure 3.5: (a) An example of a mobile VOG-system. (b) An image of an eye captured with the camera of the system in (a), where six CRs are visible in a circle around the pupil.

tower mounted eye-tracker. Examples of research where a remote eye-tracker has been used are infant studies [18] and recordings of school children [19], where a tower mounted eye-tracker often is not applicable.

Mobile eye-trackers

Both tower mounted systems and remote systems are used in laboratory experiments where a participant is seated in front of a computer screen. The third type of system is a mobile system, see Fig. 3.5a, where the eye-tracking equipment is attached to a helmet, a cap, or a pair of glasses. The mobile eye-tracking system typically consists of a camera that records the eye, some versions have one camera for each eye, and one camera, referred to as the scene camera, that captures the scene that the user is exploring. The camera(s) that films the eye can be placed either on the

Figure 3.6: Data point from a mobile eye-tracker that is mapped onto the scene video. Here the gaze is indicated with a red dot.

head of the user filming the eye through a mirror, or be integrated on the inside of the frame of a pair of glasses filming the eye directly. In Fig. 3.5b, an example of an image of the eye captured with a mobile eye-tracking camera is shown. Since the camera is placed on the user, the eye movements are recorded in relation to the movements of the head, referred to as eye-in-head motion. In order to record eye movements in relation to a world coordinate system, referred to as eye-in-space motion, the position of the head and the body needs to be measured with either external equipment or through the scene camera. An overview of such methods is given in Chapter 5. In many mobile eye-tracking systems, the recorded eye-in-head signal is given in the coordinate system of the scene camera, and the gaze is marked with a cross or a dot in the scene camera video, an example is shown in Fig. 3.6. The sampling frequency of the recorded eye-tracking signal presently ranges from 25 Hz up to 100 Hz.

The flexibility of mobile eye-trackers opens up for experiments outside the laboratory when studying, e.g., decision making in the supermarket [20] or eye movements during sports activities [21].

3.2 Other methods

3.2.1 Electro-oculography (EOG)

In the eye, there is a potential difference between the positive frontal part, the cornea, and the negative rear part, the retina, referred to as the corneo-retinal potential [22]. This potential difference of about 1mV is utilized in the electro-oculography (EOG) method. Assuming that the difference in the corneo-retinal potential is stable, the eye can be modeled as a dipole. When the eye moves, e.g., from left to right, the orientation of the dipole changes and causes a change in the electric field. The change in the electric field is measured by placing electrodes at both sides of the eyes, and one additional electrode as reference. The measured potential is small and is in the range of 15-200μV [22]. One of the advantages of using the EOG is that eye movements can be measured when the eyes are closed, and one disadvantages is that the measurements may be disturbed by noise from the action potentials of the surrounding muscles which are several magnitudes larger than the potential difference caused by the movements of the eyes.

3.2.2 Scleral search coils

One of the most accurate and precise methods for measurement of eye movements is the scleral search coil method [22]. The coils consist of a lens, where a magnetic or an optical object is placed [23]. Robinson [24], was the first to introduce magnetic wires in the coils. The user was placed in two magnetic fields that were perpendicular to each other and when the eye moves, a current is induced in the coil. The voltage between the coils caused by this current is the measurement of the eye movement. Even though the scleral search coils have high spatial and temporal resolutions, it is a very invasive method which is highly uncomfortable for the user [25]. Often the user can only wear the coils for maximally 30 minutes, even when anesthesia is used, which limits the maximal time of the experiment [25].

3.2.3 Dual-Purkinje-image eye-tracker (DPI)

The first dual-Purkinje-image eye-tracker, DPI, was introduced in [26], and later updated in [27]. An infrared light source is directed towards the eye and both the first and the fourth Purkinje images are tracked [26]. When the eye rotates, the first Purkinje image moves in the same direction as the eye, while the fourth Purkinje image moves in the opposite direction in relation to the optical axis. By measuring the difference between the two reflections, the movements of the eye can be determined. The DPI is known to be an eye-tracker with a low noise level, even though the recorded signals contain wobbling patterns in the beginning and in the end of saccades. These wobbling patterns in the signal are hypothesized to occur

since the attachment of the lens is elastic and the fast rotations of the eye cause the lens to wobble in relation to the eyeball [28].

Chapter 4
Event Detection

The purpose of an event detection algorithm in the context of eye-tracking is to discriminate between the different types of eye movements and other types of events in the recorded eye-tracking signal. The different types of eye movements have different functions and are controlled by different parts of the brain. By performing event detection in a recorded eye-tracking signal, the different eye movements are separated from each other, which allows researchers to study the cognitive function of each type of eye movement separately. One such example is in reading research where eye-tracking has a long history. In reading studies, the length, direction, and number of saccades, and the duration of fixations, are examples of characteristics that are used to interpret and quantify how someone is reading a text [29]. Similarly, the characteristics of smooth pursuit movements reflect the functionality in several different parts of the brain and can therefore be used as an indicator of disease [7].

This chapter covers the most common types of events that are detected in eye-tracking signals. An overview of existing event detection algorithms is provided together with a summary of performance evaluation strategies.

4.1 Events

Since eye-tracking signals do not only consist of different types of eye movements, but also blinks and noise from different sources, an event detection algorithm also needs to consider such events. This section describes the different types of events that appear in eye-tracking signals.

4.1.1 Eye movements

The most obvious types of events are the "true" eye movements. Out of the seven functional classes of eye movements presented in Table 2.1, fixations, saccades, and smooth pursuit movements are the ones that most often are considered by event de-

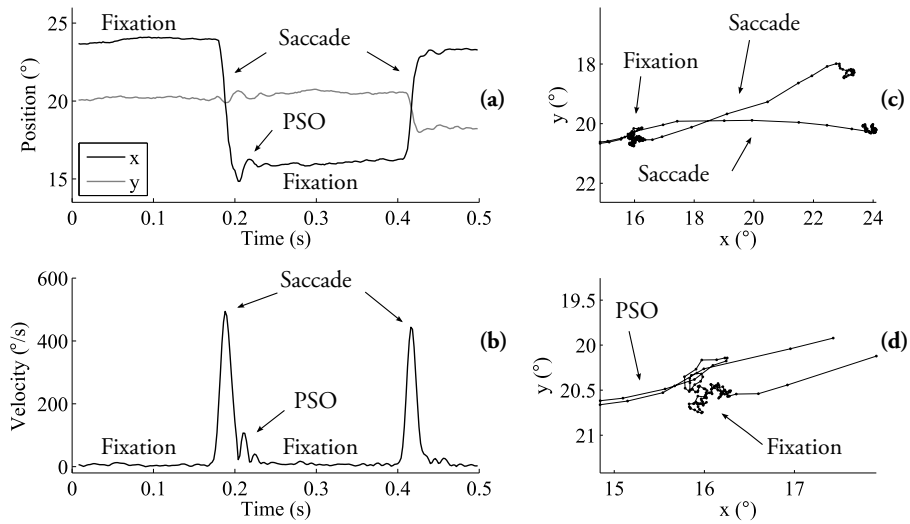


Figure 4.1: Example of saccades, fixations, and PSO. (a) Position over time, (b) velocity over time, (c) position in the spatial domain, and (d) PSO and fixation zoomed in.

tection algorithms. In Figs. 4.1 – 4.2, eye-tracking signals with indicated fixations, saccades, and smooth pursuit movements are shown.

4.1.2 Postsaccadic Oscillations (PSO)

Rapid oscillatory movements that may occur immediately after the saccade, are referred to as postsaccadic oscillations, (PSO). Similar movements have in the literature been referred to as dynamic overshoot [30, 31], and postsaccadic ringing [32]. Postsaccadic oscillations refer in this thesis to all rapid movements in the eye-tracking signals that occur directly after the saccade. The amplitudes of PSO range from 0.25° up to over 1° and there are large individual differences in both the amplitude and occurrence of PSO [33, 34].

The characteristics of the eye-tracking data originating from different recording systems may vary depending on the measurement principle, sampling frequency, and internal filters of the system. The appearances of most types of eye movements are the same, but the appearance of PSO is different for different recording systems. This fact has made researchers to start questioning whether PSO represent eye movements or are artifacts from the eye-tracker. PSO have been reported from search coil systems [31], DPI eye trackers [28] and VOG-systems [33, 32]. In [28], simultaneous recordings with search coils and a Dual Purkinje Image eye-tracker

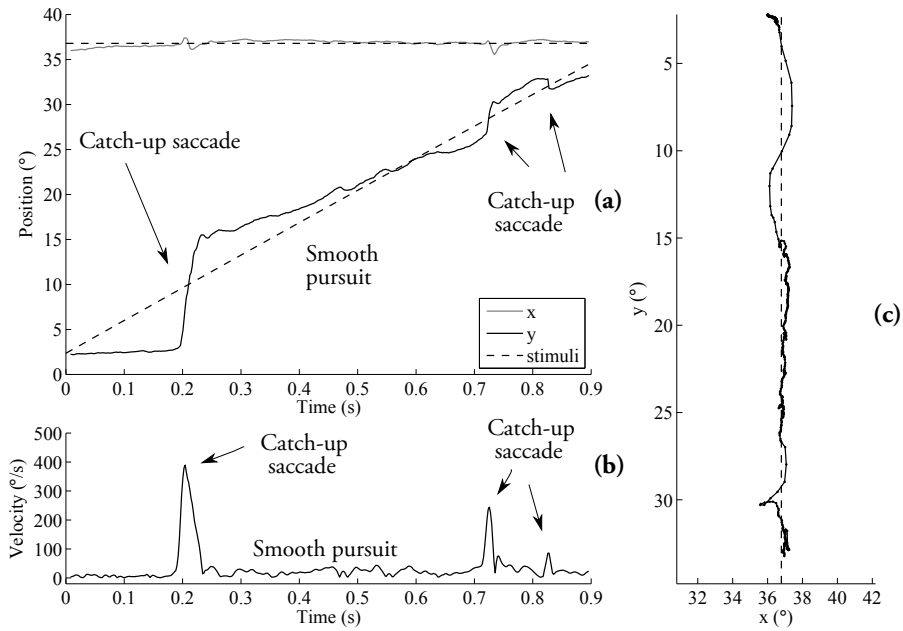


Figure 4.2: Example of smooth pursuit movements. (a) Position over time, (b) velocity over time, and (c) position in the spatial domain.

showed PSO in data recorded when using the DPI, but not in the data from the search coils. Therefore, Deubel and Bridgeman [28] concluded that PSO originate from lens wobbling and not from rotations of the eye. Recent research suggests that it is the pupil that is moving inside the iris, causing the PSO in data from video based eye-trackers [33]. Regardless of the origin of PSO, and their consequences for perception, the question of whether PSO should be classified as belonging to saccades, as belonging to fixations, or simply be removed from the recorded data remains unsolved. Examples of PSO recorded with a VOG-system are shown in Fig. 4.1.

4.1.3 Blinks

When recording eye movements using a VOG-system the signal is interrupted each time the user closes the eyelids. In order to not confuse blinks with other types of eye movements, it is important to detect the blinks properly. An example of an eye-tracking signal, recorded using a stationary VOG-system, where a blink is indicated,

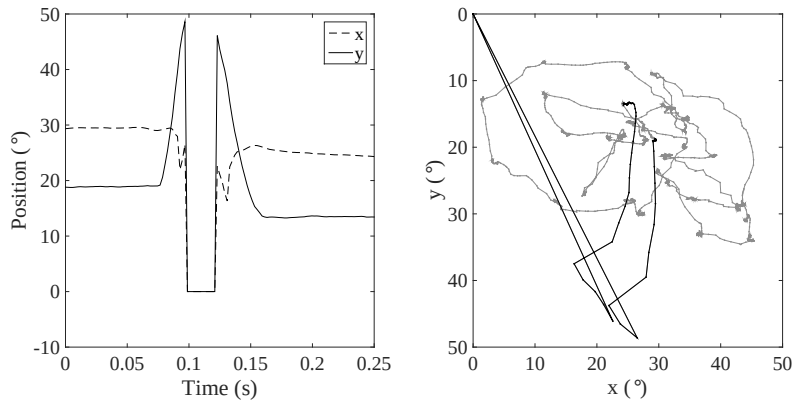


Figure 4.3: Example of the appearance of a blink. (a) Eye-tracking signal over time, where the blink is indicated as coordinates (0,0), corresponding to that the participant has closed the eye completely. (b) Appearance of a blink in the xy-plane. The segment of the signal corresponding to the signal shown in (a) is marked in black.

is shown in Fig. 4.3. In this specific VOG-system, the eye-tracking signal is set to zero when the eyelid is completely closed, and the drift-like movements before and after the actual blink correspond to that the eyelid is partly open.

4.1.4 Noise and Artifacts

Noise and artifacts in the eye-tracking signal may originate from several different sources and may have different appearances. Typically, the source of the noise is that the pupil or the CR are not correctly detected in the image of the eye. This may be caused by droopy eyelids, make-up, reflections in the sclera, reflections in glasses, light conditions, or movements of the users which cannot be handled by the eye-detection algorithm [9]. The resulting noise and artifacts in the eye-tracking signal have different appearances depending on type of problem and type of eye-tracker. A typical appearance of these types of artifacts, when using a VOG-system, is different types of spikes, referred to as one- and two-samples spikes [35]. An example of a signal that contains artifacts when the participant performs a smooth pursuit movement is shown in Fig. 4.4. In order to detect and remove noise, several filters and algorithms have been proposed [36, 37]. A survey of filters for real-time applications is given in [38].

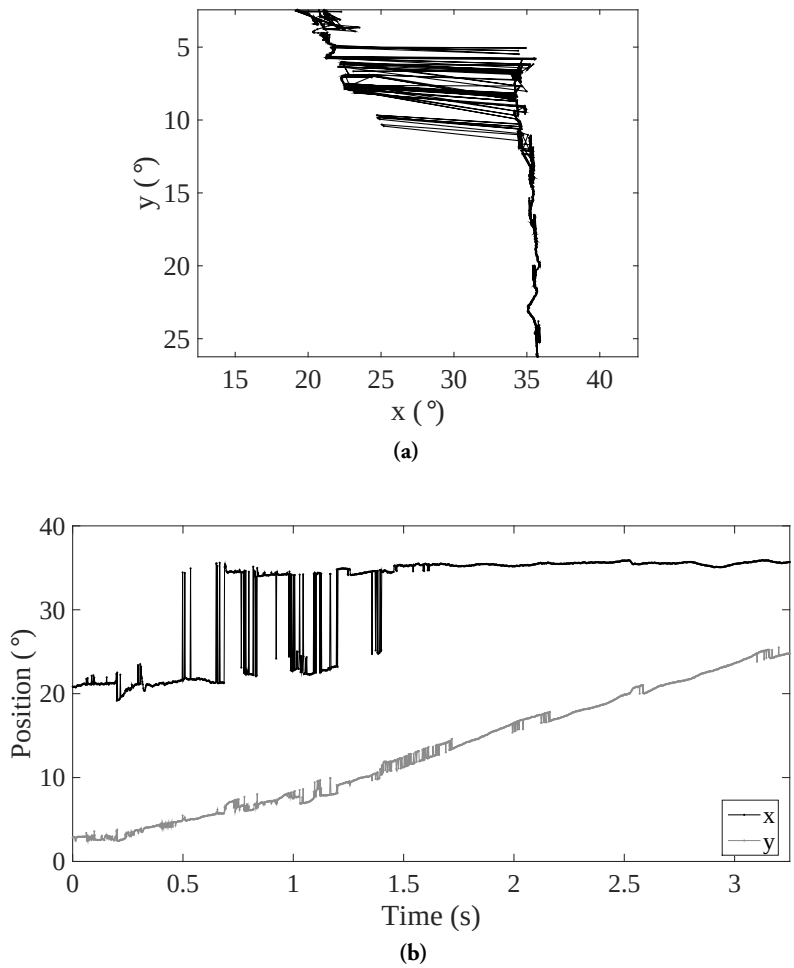


Figure 4.4: Example of the appearance of noise and artifacts in a recorded eye-tracking signal. (a) In the xy-plane, and in (b) over time.

4.2 Event detection algorithms

As mentioned above, the purpose of an event detection algorithm is to segment the eye-tracking signal into different types of events. Depending on whether a stationary or mobile eye-tracker was used to record the signals, the subsequent analysis may differ. Four different approaches are considered in this thesis, and are listed below:

- The first approach, shown in Fig. 4.5, is event detection for stationary eye-tracking setups which record the eye-in-space signal. The eye-in-space signal is either recorded using the tower mounted system where the head of the participant is fixated or the remote eye-tracker where the camera has a stationary position.

- The second approach, shown in Fig. 4.6a, may be used for event detection in mobile eye-tracking signals. Most types of mobile eye-trackers record eye movements in relation to movements of the head and body, i.e., the eye-in-head signal. In this approach, the eye-in-head signal is used directly in the event detection algorithm. Since the eye-in-head signal is influenced by head and body movements, these movements limit the number of event types that can be separated in the eye-in-head signal. Most often the event detection algorithm separates between saccades [39] and the intervals between the saccades [40], which may include fixations, smooth pursuit movements, VOR, and OKN. Since the eye-in-head signal may differ from the eye-in-space signal, not all algorithms used in the first approach are applicable.

- In the third approach, shown in Fig. 4.6b, the eye-in-head signal recorded with the mobile eye-tracker is combined with a positioning system, e.g., a motion capture system. Another option is that the integrated scene camera of the mobile eye-tracker is utilized for compensation of the head and body movements. By combining the eye-in-head signal with a system that estimates the position and orientation of the user, the eye-in-space signal is estimated. For the estimated eye-in-space signal, an event detection algorithm, similar to the ones used in the first approach can be applied.

- The fourth approach, shown in Fig. 4.6c, may be used for mobile eye-tracking signals. In this approach, the eye-in-head signal is mapped onto the image from the scene camera. The signal is segmented either by manually annotating whether the frame belongs to a fixation or not, or an automatic process where objects in the scene video are detected and related to the gaze. In this approach only fixations are considered.

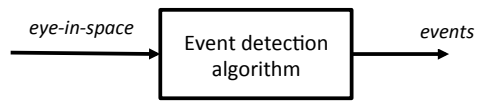


Figure 4.5: Event detection in eye-tracking signals recorded with a stationary setup.

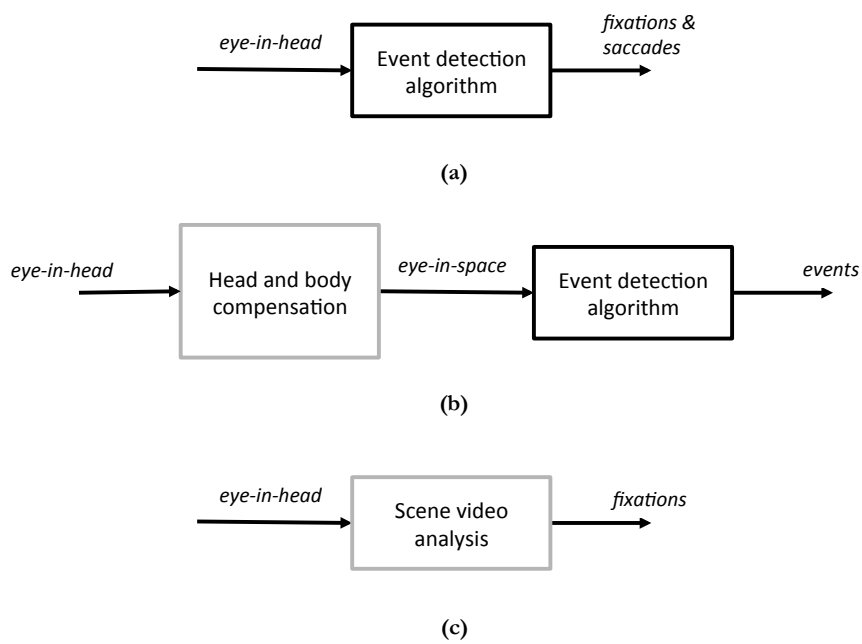


Figure 4.6: Three types of approaches to perform event detection in mobile VOG-systems. A grey block indicates that the method is described in Chapter 5.

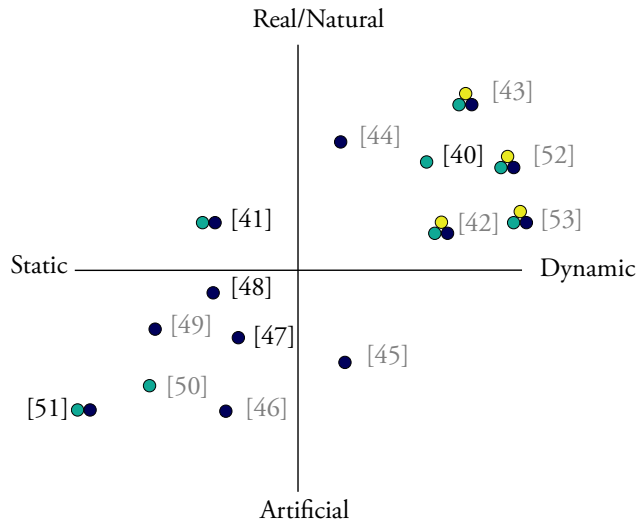


Figure 4.7: Number of event detection algorithms in the end of 2010. Each group of dot(s) indicates an algorithm and the color of the dot(s) the types of events that are detected by that algorithm. Dark blue dot indicates saccades, turquoise dot indicates fixations, and yellow dot indicates smooth pursuit movements. Algorithms that are systematically evaluated are shown in black.

In this section, the event detection algorithms used in approaches 1–3 are described, and in Chapter 5, methods for compensation of head and body movements in the third approach and methods for analyzing the eye-tracking data using the scene camera in the fourth approach, are presented.

Historically, event detection algorithms have been divided into two categories: dispersion-based and velocity-based algorithms [51]. In recent years, several event detection algorithm have been developed that are a combination of both categories or additional features have been included, which makes this type of categorization less attractive. Therefore, in the following section, the algorithms are divided based on whether they are developed for static stimuli or dynamic stimuli. This categorization focuses on the functionality of the algorithms rather than their internal structure. In Figs. 4.7 – 4.8, the number of algorithms published until the end of year 2010 and end of year 2015 respectively, are shown. The four sections in the figures show if the stimuli were static-artificial, static-natural, dynamic-artificial, or dynamic-natural. In Figs. 4.7 – 4.8, the algorithms that are systematically evaluated, in terms of either using manual annotations, simulations, or a comparison to existing algorithms, are emphasized in black. Note, that even though the total number of systematically evaluated algorithms has increased, only two new algorithms for dynamic-natural stimuli have been published since 2010 and only one

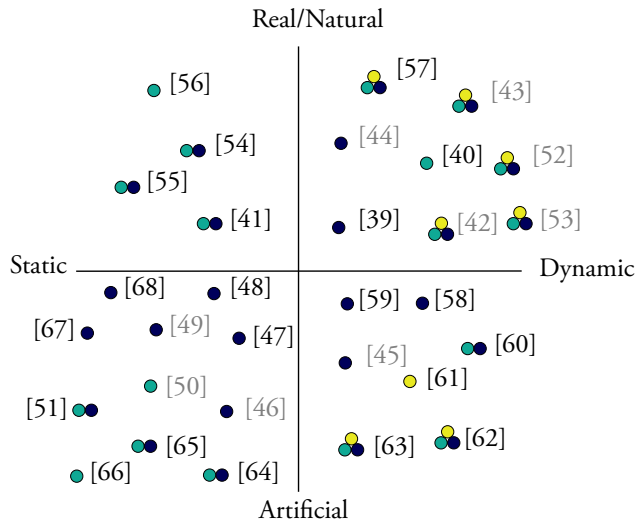


Figure 4.8: Number of event detection algorithms in the end of 2015. Each group of dot(s) indicates an algorithm and the color of the dot(s) the types of events that are detected by that algorithm. Dark blue dot indicates saccades, turquoise dot indicates fixations, and yellow dot indicates smooth pursuit movements. Algorithms that are systematically evaluated are shown in black.

of them considers smooth pursuit movements.

4.2.1 Static stimuli

The very first automatic event detection algorithms were based on a few thresholds. One of the first is the dispersion based algorithm [50], later referred to as the I-DT algorithm [51]. The I-DT algorithm has two parameters: a dispersion threshold and the length of a time window in which the dispersion is calculated. The length of the time window is often set to the minimum duration of a fixation, which is around 100-200 ms [51, 50]. The dispersion is calculated within the window and if the dispersion is below the dispersion threshold, the window is extended one sample to the right until the dispersion threshold is exceeded. Then all, but the last sample are labelled as a fixation and a new window with the initial length starts with the last unlabelled sample. The dispersion is again calculated, and if the dispersion exceeds the dispersion threshold the first sample in the window is labelled as a saccade. The window is moved one sample and the procedure continues until all samples are labelled. Several different metrics have been used for calculation of the dispersion, see [69] for an overview. The performance of the I-DT algorithm has recently been evaluated in [70].

The original algorithm that has inspired many subsequent algorithms uses a velocity threshold, and is referred to as the I-VT algorithm [51]. Since a high velocity is typical for saccades whereas a low velocity is typical for fixations, the I-VT algorithm has mainly been used for separation between saccades and fixations. The velocity is calculated as the difference between samples, i.e., the sample-to-sample velocity, and all samples that have velocities higher than the threshold are classified as saccade samples and all samples that have velocities lower than the threshold are classified as being fixation samples [51]. The velocity threshold has in the literature varied from $5°$/s up to $300°$/s [48] depending on the sampling frequency, which eye-tracker that has been used, and if a filter was used when the velocity was calculated.

The I-VT algorithm was further developed by using the mean velocity over five samples instead of using the sample-to-sample velocity, which made the velocity signal smoothed [46]. This algorithm calculated the velocity signal over a complete trial and by estimating the standard deviation of the velocity signal using a median estimator, the velocity threshold was calculated as a multiple of the estimated standard deviation of the velocity signal. The calculated velocity thresholds for the x- and y- directions were inserted into the formula of an ellipse and all samples outside the ellipse in the xy-plane were classified as saccades and all samples inside the ellipse were classified as fixations [46]. This algorithm was developed to better adapt to different levels of noise between different participants and trials without the need for manual adjustments of the velocity threshold. Originally, the algorithm was proposed for microsaccade detection, but has also been used for normal saccades. A further development of the algorithm in [46] was performed in [41], where the algorithm is also velocity based. The velocity signal was calculated using a Savitzky-Golay filter [71], which is a polynomial filter that smoothens the velocity signal more than the moving average filter used in [46]. The algorithm is iterative, and in the first step, the algorithm finds the peaks of the saccades. For each peak, a local velocity threshold is calculated based on the mean and standard deviation of the noise in the previous inter-saccadic interval. The onset and offset of the saccade is found by iteratively and sample-wise evaluating the velocity against a local velocity threshold. In the next step, PSO are detected if the velocity exceed two separate velocity thresholds during a constant time window placed immediately after the detected saccade [41]. This algorithm was developed for adaptation of the thresholds to different noise levels and is the first algorithm to detect PSO, (referred to as glissades in [41]), as a separate event.

In [47], the I-VT algorithm was extended to detect saccades by using a constant false alarm rate procedure. The thresholds were continuously updated based on the observed data, which made the detection algorithm less sensitive to noise.

Another method which was developed to minimize the number of settings was proposed in [60]. The algorithm combines the position signal, the velocity signal,

and the acceleration signal and uses k-means clustering to separate saccades from fixations. The algorithm is iterative and both local and global clustering is performed. The number of clusters is automatically calculated from the data, but the user has to set the thresholds for the minimum duration of fixations and for the minimum duration of saccades.

The continuous Wavelet transform has been used to detect blinks, saccades, and fixations in EOG data [55], also referred to as CWT-SD. The algorithm was developed to identify everyday tasks from the eye movement pattern, e.g., reading a text, scrolling a web page, and writing a text. By calculating the continuous wavelet coefficients for the recorded eye-tracking signal, the saccades were detected by comparing one of the coefficients to a threshold. The fixations were detected using a dispersion threshold and the blinks using another threshold based on the wavelet coefficients.

The Wavelet transform has also been used to detect microsaccades [68]. During fixational eye movements, the microsaccades can be viewed as singularities in the recorded data. By using the continuous wavelet transform these singularities can be found and by investigating the signal surrounding the singularity using principle component analysis, the microsaccades can be characterized. The method can also be used for normal saccades without adjusting the parameters [68].

One of the first stochastic algorithms that has been applied to eye-tracking signals is the Hidden Markov Model, referred to as I-HMM [51]. The method is based on two states where the first is for fixations and the second is for saccades. The method is based on that fixations have low velocities and that saccades have high velocities. The model uses one probability to stay in the same state and one probability to make a transition to the other state. The parameters of the I-HMM are derived from training data, e.g., where manually annotated data are used, before classification is performed.

A stochastic algorithm that uses Bayesian statistics to discriminate between saccades, fixations, and blinks is proposed in [65]. The algorithm was developed for real-time analysis of EOG-data. The algorithm uses two features: The normalized derivate of consecutive samples and the difference between the maximum and the minimum of the derivative of the vertical eye-tracking signal. The intrinsic parameters of the algorithm are set during an unsupervised training procedure in the beginning of the data sequence using the Expectation-Maximization algorithm for Gaussian mixture models [65]. The thresholds for maximum and minimum durations of blinks and saccades still have to be predefined.

Another approach where three serially connected artificial neural networks were used for detection of saccades, micro-saccades, fixations, and blinks was proposed in [64]. Within two windows, one for the position signal and one for the velocity signal, seven features were calculated for each window which together formed a feature vector that was fed into the first neural network. First, blinks where sepa-

rated from the other types of movements, i.e., saccades and fixations. In the second stage of the network, the fixations where separated from the saccades and in the last stage the saccades where distinguished from microsaccades. In the evaluation of this algorithm, a very large database was used with 1392 participants.

In [66], the I-DT algorithm was extended to C-DT, and instead of using the dispersion measure to distinguish between fixations and saccades, the variances in the x- and y-position signals were evaluated. This algorithm is based on the assumption that the variances in x and y are of similar magnitude and independent of each other during a fixation. This assumption is evaluated by calculating the covariance and the statistical F-test for equal variance [66].

A completely nonparametric method was proposed in [56]. The method uses gap statistics [72] in order to calculate both the velocity threshold to separate non-saccadic data from saccades and the duration threshold for fixations.

Most of the algorithms that have been developed for event detection in eye-tracking signals have been monocular, i.e., they use only the signal from one eye. One exception is the BIT-algorithm, Binocular-Individual Threshold [54], which is a velocity-based algorithm that separates fixations from saccades by using the minimum determinant covariance estimator and control chart procedures. It is a parameter-free algorithm that explores the relationships between the left and right eye and between the horizontal and vertical directions [54].

Another approach to make use of the binocular signals is to calculate the average between the signals from the two eyes, i.e., the average of the two x-signals and the average of the two y-signals, separately [73]. By calculating the average between the two eye-tracking signals, the noise level may be reduced.

In [46], and later in [74], the binocular nature of microsaccades was explicitly used by only considering saccades that are, with a maximum time lag, detected in both signals.

4.2.2 Dynamic stimuli

A common denominator for the algorithms that are presented in this subsection is that they can be used for eye-tracking signals recorded during dynamic stimuli. Furthermore, the algorithms can be categorized into algorithms that identify fixations and saccades in the presence of smooth pursuit movements and those that also detect smooth pursuit movements.

Saccades and fixations in presence of smooth pursuit

In the early 90s, Sauter *et. al.* [45] proposed an algorithm for detection of saccades in signals that also contained smooth pursuit movements. The algorithm used a Kalman filter that compared the predicted and the calculated velocities [45]. The idea was that during smooth pursuit movements, the velocity is predictable, and

during saccades, it is not, and by calculating the difference between the predicted and the present velocities, the saccades could be detected.

Also in [58], an algorithm was developed to accurately detect saccades in data that contain smooth pursuit movements. The algorithm contained three steps. First, the eye-tracking signal was median filtered in order to remove the velocity component of the smooth pursuit movement signal, and in the second step, a template of a saccade was moved across the velocity compensated signal for cross-correlation calculation between the template and the signal. The samples with high correlation to the template were marked as candidate saccades and the others as candidate fixations. In the third step, the samples marked as candidate saccades were grouped together and the duration of the candidate saccade and time between two candidate saccades were investigated in order for the samples to be considered as actual saccades.

In [59], a particle filter was used to detect saccades with varying amplitudes. A particle filter is a Bayesian state estimator that is a powerful tool to describe a non-linear and non-Gaussian system [59]. The algorithm suppresses the velocity component that is related to the smooth pursuit movement and was therefore able to detect blinks, micro-saccades, and saccades performed during both smooth pursuit movements and free viewing of static images.

The I-VT algorithm has been extended in various ways for different purposes, either to only detect fixations or to only detect saccades or to detect both of them. For eye-tracking signals recorded with a mobile eye-tracker, when the user can move head and body freely, the I-VT algorithm was extended to also include a threshold for the minimum fixation duration [40]. The algorithm was used to detect fixations, which in their work included smooth pursuit movements, OKN, and VOR.

In order to robustly detect saccades in signals recorded using a mobile eye-tracker while walking, an algorithm was proposed in [39]. Their algorithm used a combination of the position, velocity, and acceleration signals in order to detect the saccades.

An algorithm solely based on the acceleration signal was proposed in [49], and later modified to be a real-time and adaptive algorithm for separating saccades from other nonsaccadic components [44]. The algorithm is based on that the distribution of the acceleration signal of non-saccadic movements have zero mean and that saccadic components belong to a distribution with non-zero mean. The acceleration threshold is based on this assumption and is updated for every sample with a window of 200 ms duration. When the acceleration signal exceeds the threshold, the onset of the saccade is found and in order to determine the offset, both the position signal and the acceleration signal are used [44].

Smooth pursuit movements

The Kalman filter proposed in [45], was later extended in [75] to detect saccades, fixations, and smooth pursuit movements. A Chi-square test was used to classify the samples that are above the threshold as saccades and the samples that are below the threshold as fixations. In a following step, the intervals detected as neither saccades nor fixations were considered to be smooth pursuit movements.

In the original implementation of the I-VT algorithm, only saccades and fixations were detected. Extensions have been proposed where also smooth pursuit movements are included in the I-VT algorithm [43, 63, 42]. A common feature of these extensions is that the saccades are detected using a velocity threshold and that an additional threshold is used for separation between fixations and smooth pursuit movements. One such implementation was made in [43] where a velocity threshold was combined with principle component analysis for separation between saccades and smooth pursuit movements. The algorithm was developed for data where both humans and monkeys watched dynamic stimuli. In [63], the I-VT algorithm was extended for detection of smooth pursuit movements using an additional velocity threshold, (I-VVT), a dispersion threshold [42], (I-VDT), and by analysis of the movement pattern [52], (I-VMP). In a comparison between the three algorithms, the I-VDT algorithm showed the most robust results for detection of smooth pursuit movements [63].

In order to detect smooth pursuit movements, a set of signal measures were introduced [76], and later extended to include, e.g., velocity, variance of the position signal, range, and slope [61]. These measures were calculated in several consecutive windows and a k-nearest neighbor classifier was used to detect smooth pursuit movements [61]. An upper and a lower threshold for each measure were found by evaluating the signal measures for a manually annotated part of the database.

Bayesian detection theory has also been used for detection of saccades, fixations, and smooth pursuit movements in a real-time algorithm proposed in [62]. The algorithm uses the velocity to separate between saccades and fixations and the total positional movement within a time window for detection of smooth pursuit movements. The thresholds are calculated automatically during a training period using the Expectation and Maximization-algorithm, similar to [65].

In order to detect saccades, fixations, smooth pursuit movements, and VORs, an extended version of the I-HMM algorithm was proposed in [53]. A two dimensional Hidden Markov Model was used, which included both the velocity of the eye-tracking signal and the velocity of the recorded head movements. The algorithm was developed to be used for mobile eye-tracking data, when also the head movements are available.

An algorithm that is based on the same idea as the I-HMM is the algorithm in [57]. It uses a Bayesian mixture model to separate between saccades and fixations and a principle component based algorithm to later separate between fixations

and smooth pursuit movements. The algorithm has been used for low-speed data recorded during driving sessions [57].

4.3 Performance evaluation

There are a number of conceptually different approaches that have been used for performance evaluation of event detection algorithms. In this section, the most commonly used methods are presented, and their respective advantages and drawbacks are discussed.

4.3.1 Properties of detected events

One of the most commonly used methods for performance evaluation of event detection algorithms is to calculate a set of established properties of the detected events, e.g., duration, amplitude, and peak velocity [41, 60, 77]. This method is useful when manual annotations of the signals are not available and when the stimuli is not artificial, i.e., when texts, images, or video clips are used. By comparing the distributions of the calculated properties to well-established values from the literature, the overall performance of the detection algorithm can be investigated. In [77], the plotted distribution of the fixation durations showed that the applied algorithm did not give satisfying results. The main drawback of plotting distributions is that each detected event is not compared to the corresponding true one. And, even when the distribution looks as expected, the accuracy of the detections on a sample-to-sample level is not evaluated.

4.3.2 Stimuli-based

When artificial stimuli is used, e.g., a dot is moving across the screen, the coordinates and the time when the dot is shown are known. If the participant is instructed to follow the movement of the dot, the detected event can be compared to the presented stimuli in order to evaluate the performance of the event detection algorithm. This type of performance evaluation assumes that the participant follows the given instructions, but even if this is true, not only the performance of the algorithm is evaluated but also the participant's ability to actually follow the movements of the dot. In Fig. 4.9, the eye-tracking signal is plotted together with the corresponding position of the stimuli. Note that the number of fixations in the eye-tracking signal is not equal to the number of plateaus in the stimuli signal. Using this evaluation strategy, only the types of eye movements that are triggered by stimuli can be evaluated, i.e., saccades, fixations and smooth pursuit movements. Other types of events in the eye-tracking signal, e.g., blinks, PSO, catch-up saccades and microsaccades, cannot be evaluated. Since it takes some time for the human visual system to react

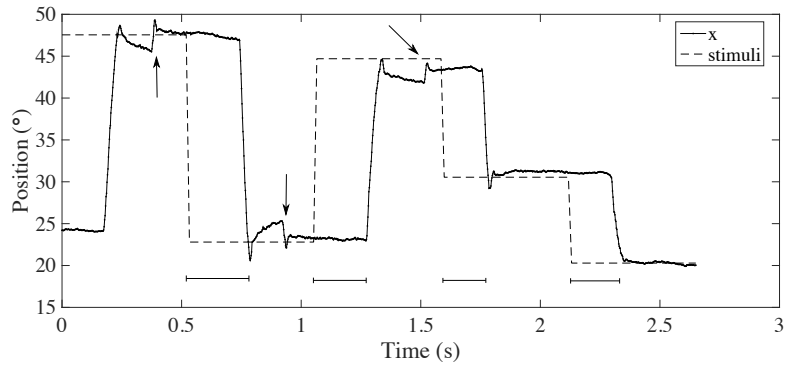


Figure 4.9: Eye-tracking signal compared to the corresponding stimuli. The number of fixations and saccades are not the same as in the stimuli. The arrows show examples of corrective saccades that are not initiated by the stimuli, and the bars at the bottom show the latency between stimuli and the initiation of the saccades. For readability only the x-coordinate is shown.

on visual stimulation, e.g., the latency to launch a saccade or latency before initiating smooth tracking of an object during smooth pursuit movements, a time window after or around the time when the presented stimuli is shown is often investigated. If the corresponding event appears within this time window, it is considered to be correct [65]. Due to this time window, and depending on the length of it, the onsets and offsets are difficult to evaluate accurately. Despite this, performance measures such as sensitivity, specificity, accuracy, and recall are typically calculated in order to evaluate the performance of algorithms.

In order to improve the above described method of using the stimuli, a set of scores was proposed for evaluation of saccades and fixations [48]. Seven scores were calculated in order to evaluate the performances of the compared algorithms. Four of the scores were well-known parameters of the detected events: average number of fixations, average fixation duration, average number of saccades, and average saccade amplitude. In addition, three scores reflecting the user's ability to follow the stimuli were calculated: the ratio between the number of samples detected as fixations and the number of samples when fixation points are shown in the stimuli, the distance between the center of the fixation point in the stimuli and the corresponding center point of the detected fixation samples, and finally, the sum of total detected amplitudes of the saccades is compared to the sum of the total distance the dot in the stimuli has moved. The seven scores were compared to their respective ideal score. The closer to the ideal score, the better the performance of the detection algorithm. In [63], the scores were extended to work also when smooth pursuit movements are present in the eye-tracking signal. Three additional scores were proposed for smooth pursuit movements: the average difference in distance between the smooth

pursuit movement and the corresponding moving dot in the stimuli, the difference in speed between the detected smooth pursuit movement and the corresponding moving dot in the stimuli, and the ratio between the total length of the detected smooth pursuit movements and the total length of the smooth pursuit stimuli. In addition, a score was introduced to reflect the amount of detected smooth pursuit samples that were incorrectly detected as fixations. This score was calculated as the ratio between the number of samples detected as smooth pursuit movement when the stimuli were not moving and the total number of fixation samples in the stimuli.

4.3.3 Manual annotations

In order to evaluate the performance of an event detection algorithm more accurately, manual annotations may be an alternative. Many of the most recently developed algorithms have been evaluated using manual annotations [56, 58, 64, 65, 62, 57]. Although the use of manual annotations is an accurate method for evaluation of an event detection algorithm, it is also time consuming, subjective, and cumbersome to perform. The reasons for using manual annotations may be that the participants do not fully follow the stimuli [62], events are not trigged by the stimuli [65, 64], or the stimuli is not artificial, i.e., texts, images, or video clips, are used. It is common to use more than one expert to perform manual annotations of the same data set [40, 56, 57]. There are several ways to deal with more than one expert. Either the results based on each of them are presented [56], or if a detection made by the algorithm agrees with one of them it is counted as correct [40], or all samples were the experts do not agree are removed in order to only evaluate the samples where all experts are in agreement [57]. Another option is to calculate the inter-rater agreement on parts of the data, and if it is sufficiently high, only one of the experts is used for the complete dataset. Having more than one expert may become complicated if the agreement between them is low. On the other hand, removing samples where the experts do not agree may lead to that the most complicated patterns of events, which may also be the most interesting parts of the data to evaluate, are removed.

4.3.4 Simulations of eye-tracking signals

In many signal processing areas it is common to perform simulations of signals and evaluate the performance of algorithms based on these simulations. However, it is a challenge to construct simulated eye-tracking signals with authentic structure, variation, disturbance patterns, and measurement system dependent properties which make them useful for performance evaluation purposes. Some attempts have been made: In order to evaluate an algorithm for the detection of microsaccades, data were simulated by averaging 100 microsaccade from recorded eye-tracking data. In-

tervals between the microsaccades were simulated with white Gaussian noise sent into an AR-process [67]. Others have used recorded eye-tracking signals and added Gaussian noise with different variances to the signals in order to evaluate the algorithm's sensitivity to different levels of noise [59, 56]. In [47], the author argues that simulated signals are better for performance evaluation of algorithms than subjective and time consuming manual annotations, and therefore that algorithm is only evaluated on simulated signals.

4.4 Available databases and algorithms

In order to be able to compare results between eye-tracking laboratories, there are a number of research groups that share their recorded data. For a complete overview of eye-tracking databases and the corresponding image and video stimuli, see [78, 79]. Although there is a large number of available databases, there are only few datasets that are recorded for the purpose to evaluate event detection algorithms. One such dataset is described in [62], where both the eye-tracking signals and the annotations of events are provided.

There are also a number of researchers who share their algorithms, either directly on their own webpages or as supplementary material on the journal's webpage [48, 63, 67, 59, 65, 56, 62, 80, 43, 66, 60, 41, 54]. In 2012, the Eye Movement Researchers' Association (EMRA) was founded with the aim to be a platform for sharing, e.g., open-access tools and datasets, see `www.eye-movements.org`. A summary of the available algorithms and databases for the algorithms presented in this chapter is given in Table 4.1.

Table 4.1: Summary of available databases and algorithms that are described in this chapter.

Component	Year	Algorithm	Database	Eye movements
Berg *et. al* [43]	2009	✓	✓	Saccade, Fixation, Smooth pursuit
Dorr *et. al* [42]	2010	✓	✓	Saccade, Fixation, Smooth pursuit
Nyström *et. al* [41]	2010	✓		Saccade, Fixation,
Komogortsev *et. al* [48]	2010	✓		Saccade, Fixation
van der Lans *et. al* [54]	2011	✓		Saccade, Fixation
Veneri *et. al* [66]	2011	✓		Fixation
Mould *et. al* [56]	2012	✓		Fixation
Komogortsev *et. al* [63]	2013	✓		Saccade, Fixation Smooth pursuit
Otero-Millan *et. al* [67]	2014	✓		Microsaccade
König *et. al* [60]	2014	✓		Saccade,Fixation
Daye *et. al* [59]	2014	✓		Saccade
Toivanen *et. al* [60]	2015	✓		Saccade, Fixation
Santini *et. al* [62]	2016	✓	✓	Saccade, Fixation Smooth pursuit

Chapter 5
Analysis of Mobile Eye-Tracking Data

When performing eye-tracking in real world situations using a mobile eye-tracker, tracking of only the eyes is sometimes not enough in order to be able to adequately analyze the recorded data. As described in Section 3.1.3, the signals are most often recorded in relation to the coordinate system of the head. Three different approaches for analyzing mobile eye-tracking data have been used: In the first approach, the event detection algorithm is applied to the recorded signal directly, and disregarding the fact that there may be significant influence of head and body movements in the signal, see Fig.4.6a. In the second approach, the recorded signal is converted to a world coordinate system through compensation of head and body movements before an event detection algorithm similar to the ones described in Chapter 4 is used, see Fig. 4.6b. Finally, in the third approach, the recorded eye-tracking data are analyzed via detected objects in the scene video, see Fig. 4.6c. In this chapter the two latter approaches are described in detail.

5.1 Systems to track head- and body movements

One approach to analyze eye-tracking signals when the user moves freely is to express the recorded signal in a world coordinate system by performing compensation of head and body movements. In this section, the most common strategies that have been proposed for compensation of head and body movements in mobile eye-tracking data are described.

5.1.1 Magnetic field tracking system

A magnetic tracking system consists of a fixed transmitter and one or several receivers mounted on the object or person to be tracked. The transmitter emits a

pulsed magnetic field, and if the receivers are in the area of the magnetic field, the three dimensional position and rotations (6DOF) of the receiver(s) are measured [81]. A magnetic field tracker has been used in combination with an eye-tracker to study VOR movements [81], and to study the coordination of head, hand, and eye movements in natural tasks [82]. In both studies, a receiver was placed on top of the head of the test person in order to record head movements at the same time as a mobile eye-tracker recorded eye movements. The accuracies of the measurements of positions and rotations are high from the magnetic tracking system. The main drawback, however, is that the strength of the magnetic field rapidly decreases with distance from the transmitter, which makes the tracking range of the system limited to relatively small volumes [83].

Another type of magnetic field tracker was used in [84], where one search coil was used to track movements of the eye and another coil was used to track the head in order to study the latency and gain of the VOR. Measurements using coils are known to have a low level of noise and of being accurate, but a disadvantage is that the user must be in the center of the magnetic field and that the coils can only measure rotations and not larger positional changes of the head [81].

5.1.2 Optical tracking system

An optical motion tracking system is in many aspect similar to a magnetic field tracking system. The system consists of a static transmitter that sends beams of laser, that scans a volume. The receiver, an IR-sensor, is placed on the object that is tracked which registers the laser beams. Optical systems often give accurate measurements with high spatial resolution, but they suffer from line-of-sight problems, which occur when the sensor is occluded [83]. The optical system LaserBird has been used in combination with an eye-tracker in a driving simulator [57].

5.1.3 Computer vision based tracking

Since the majority of mobile eye-trackers today are equipped with a scene camera, the use of computer vision techniques is the most commonly used strategy to estimate head and body movements. These methods can be divided into two categories, outside-in and inside-out [85]. Outside-in methods use external cameras, e.g., a motion capture system, to track the position of the eye-tracker or the position of the test person, while inside-out methods use the scene camera of the eye-tracker. In general outside-in methods, e.g., a motion capture system, are more expensive than the inside-out methods. Both types of methods are used to estimate the POR, in three dimensions.

Motion capture systems and external cameras have in different ways been combined with an eye-tracker to estimate the POR in three dimensions [86, 87, 88, 89,

90]. A real-time gaze tracking system that consisted of a mobile eye-tracker and a motion capture system was proposed in [87] for estimation of gaze when users were looking at large displays and allowing free movements of the head and body. A method for calibration of the eye-tracker together with the motion capture system was proposed in [86] for estimation of gaze when the user performs tasks with free head and body movements.

In order to automatically analyze gaze in 3D on objects that are grasped, 3D gaze coordinates were estimated by placing markers on objects in the environment, and on the hand of the user. The markers were tracked in the motion capture system and the movements of the eyes were tracked using an eye-tracker. The object that had the shortest Euclidean distance to the gaze point was assumed to be looked at [88]. The advantage of this approach is that it can be used also for objects that are moving.

A geometry-based method to measure gaze orientation in space was proposed in [89]. The position of the head-in-space signal was measured with the motion capture system, and the camera of a mobile eye-tracker directed towards the eye was used to film the pupil. The mapping between the pupil position in that image and the world coordinate system was performed using a non-linear constrained optimization method during a calibration procedure [89]. The method also corrected for positional changes of the helmet, which the eye-tracker is attached to.

An external camera can also be used to capture information about the environment and use the information to build a 3D map of it [90]. The computer vision technique SLAM, Simultaneous Localization And Mapping, was used to localize the test person within this map and at the same time estimate the position and orientation of the body and head of the test person. The 3D-map can then be used for visualization of the gaze in three dimensions.

With the goal to be able to estimate the gaze orientation and gaze coordinate in space without any external equipment but the scene camera of the mobile eye-tracker and possibly an additional attached camera, several computer vision methods have been proposed [53, 91, 92, 85, 93]. One of the early methods used an omnidirectional vision sensor to capture a larger view of the environment that was in front of the user [53]. A video camera pointing upwards was placed on the user's head. By combining this video camera with a mirror, a circular image of the environment that surrounded the user was captured. Based on the captured images of the omnidirectional vision sensor, the method was able to estimate the rotational head movements but had difficulties to estimate translational head movements.

A completely different approach was proposed in [85], were markers referred to as fiducial augmented reality markers were placed in the environment of the recording. Every marker is unique in order to be able to differentiate between them. The markers were detected in the image of the scene camera and their rotation and transformation were calculated. In addition, the experimental setup was geometrically

3D modeled, which approximated the positions of the objects of interest together with the markers. By combining the markers from the 3D model with the markers detected in the scene camera, the position of the objects in the scene could be calculated. The data from the eye-tracker were mapped into the 3D model and the gaze could be analyzed in relation to the objects [85].

A method that uses only the integrated scene camera of the mobile eye-tracker was proposed in [91], for estimation of the 3D POR as well as the position and orientation of the user's head. This approach matched objects between key frames and a triangulation technique was used to estimate the position and orientation of the camera, i.e., the orientation of the head.

In [92], the authors succeeded to compensate for the ego-motion of the user and analyze the eye movements without estimating the 3D POR. The ego-motion was compensated for by estimating the motion of the scene camera between two frames. The previous frame was used as a matched filter for the next frame, and by calculating the cross-correlation between the two frames using phase correlation [94], the ego-motion was estimated. One limitation of the method was that it assumed that the world was static, i.e., that there were no moving objects in the surroundings of the person being eye-tracked [92].

In order to estimate the 3D POR and at the same time model the surroundings without any extra equipment, a computer vision technique known as SLAM, was proposed in [93]. The method uses feature points extracted from the scene camera of the mobile eye-tracker to build a 3D map of the surroundings. At the same time, the method also localizes the user in the reconstructed map and estimates the position of the user's head. By using triangulation of the extracted feature points the head position and direction can be estimated, similar as used in [91]. By continuously building a model of the 3D environment that the user is moving in, this 3D map can later be used to visualize the gaze in three dimensions.

5.1.4 Intertial sensor system

An inertial sensor, e.g., an inertial measurement unit (IMU), typically consists of a gyroscope, an accelerometer, and a magnetometer. IMUs have been used to estimate the orientation of the head during interaction with computer screens and head mounted displays [95, 83, 96]. The reasons to why inertial sensor systems generally are not that popular, are due to that gyroscopes suffer from drift when used during longer periods of time, accelerometers are stable over time but are less precise, and magnetometers are very noisy and very sensitive to magnetic and electrical interference [95]. Recently, methods have evolved that combine the signals from the gyroscope, accelerometer, and the magnetometer, into a drift free estimation of the orientation over longer time periods [95, 97]. For an overview of these methods, see [97].

Figure 5.1: An example of a frame from the scene video of a mobile eye-tracker. The gaze is indicated with a red circle and the AOI is indicated with a red square.

In combination with eye-tracking, an accelerometer was used to estimate the direction of the driver's head and relate it to the gaze direction when driving a truck [98]. To estimate translations, and also positions, of the user using only an IMU is a difficult task. In [99], a monocular camera was combined with an IMU to estimate indoor position.

5.2 Analyzing mobile eye-tracking data through the scene video

Another approach for analysis of mobile eye-tracking data is to map the eye-tracking data directly onto the recorded scene video and either manually or automatically detect objects and areas of interest. The most common approach for analysis of mobile eye-tracking data is to manually annotate the gaze data mapped onto the scene video. The annotation is most often performed by defining one or several area-of-interests, AOIs, and by calculating the time in each AOI, referred to as the dwell time. Examples of available open-source solutions are ELAN [100] and DynAOI [101]. Commercial eye-tracking companies, e.g., Tobii and SMI, have their own analysis software customized for their data. An example of a frame captured by a scene camera of a mobile eye-tracker where both the gaze and an AOI are marked, is shown in Fig. 5.1.

In order to automatize the analysis process of mobile eye-tracking data, several methods that use object recognition techniques have been proposed [102, 103, 104,

105, 106]. In [103], the software JVideoGazer was proposed, where the user selects objects that should be tracked in the scene video, by "roping" the objects with a lasso. The software automatically tracks the objects throughout the scene video and in a final step the positions of the tracked objects are compared to the coordinates of the gaze. This semi-automated process was 9 times faster to process the eye-tracking data compared to manually annotate the same set of eye-tracking data. An extension of the JVideoGazer, the GazeVideoAnalyser, was proposed in [106]. The GazeVideoAnalyser uses the same manual object selection procedure, but in the extended version, several object recognition techniques are implemented in order to cope with variations in the environment during the recording process.

Another similar approach was proposed in [104], where a trained object recognition algorithm was used to track static and dynamic objects in the scene video. A region of interest around the gaze coordinate in each frame was analyzed, and compared to the images of the objects in the training database. The training database was built by letting an additional user walk around and look at interesting objects [104]. The approach was later enhanced by letting an expert look at one of the scene videos and click on objects that should be used in the training database [105]. The region-of-interest was defined and key points were calculated both for the region-of-interest and for the objects in the training database. The key points of the region-of-interest and the objects in the training database were compared and a score was calculated of how well the images overlapped. If the score exceeded a predefined threshold, they were considered as a match. If the gaze was close to an object over several frames, these samples were grouped together into a fixation. A threshold for the minimum fixation duration was applied as well as a smoothing filter that decreased the number of short false positives [105]. Special care was also taken to faces and human bodies, which were separately identified in order to detect when the user looked at persons or faces.

Chapter 6
Summary of the Included Papers

Summary of the main contributions

The four papers in this thesis address different aspects of how to perform robust event detection in different types of eye-tracking data. In total, the content of the four papers constitutes a complete framework for how to perform event detection in eye-tracking data recorded during dynamic stimuli and for how to evaluate the performance of such event detection algorithms. Three complete algorithms are proposed:

- By combining the first two papers, I and II, a complete event detection algorithm for eye-tracking signals recorded with a tower mounted eye-tracker with a high sampling frequency is proposed which is able to detect the three most common types of eye movements, i.e., saccades, fixations, and smooth pursuit movements. In addition, PSO are detected which gives the user a choice of where to include them, or to exclude them completely. The novelty of this algorithm is that it can divide the data into these four types of events also when videos are used as stimuli and that its performance is validated to manual annotations. This means that properties of fixations and smooth pursuit movements can be analyzed separately, and that measures that previously have been used for fixations recorded during static stimuli, e.g., durations and number of fixations, now also can be applied when watching video stimuli.

- By combining Paper I and Paper III, an even better algorithm with the same purpose as above is achieved which in addition utilizes binocular signals in order to further improve the detection performance. The requirement that both eyes need to move in a synchronized manner during smooth pursuit movements reduces the number of false smooth pursuit detections when viewing

Table 6.1: Summary of the content in each paper.

Component	**I**	**II**	**III**	**IV**
Tower eye-tracker	✓	✓	✓	
Mobile eye-tracker				✓
Monocular	✓	✓		✓
Binocular			✓	
Fixation		✓	✓	✓
Saccade	✓			✓
Smooth pursuit		✓	✓	✓
PSO	✓			
Objects (for evaluation)			✓	
Objects (event detection)				✓
Manual annotations	✓	✓		
Dynamic stimuli	✓	✓	✓	✓
Static stimuli	✓	✓	✓	✓

static stimuli.

- In the last part, paper IV, a complete algorithm for detection of the three most common types of eye movements in data recorded using a mobile eye-tracker with low sampling rate is presented. The algorithm is evaluated for mobile eye-tracking data recorded for participants that are free to move their head, but not the body. The method includes an IMU-based head movement compensation stage and utilizes both the eye-tracking signals as well as information about video objects extracted from the scene camera when segmenting the signal into events. The results show that the proposed method performs better than the compared ones.

In addition to these contributions, paper II contains an extensive performance evaluation, comparing not only algorithms but also five different performance evaluation strategies. Paper III proposes a completely new strategy for performance evaluation which is based on automatic detection of moving objects in the video stimuli to which the detected events are compared. In Table 6.1, a summary of the contents of the four papers is presented.

Table 6.2: Cohen's kappa for the proposed algorithm and the algorithm in [41]. (Paper I)

	Image	**Video**	**Moving dot**
Proposed algorithm	0.814	0.822	0.756
Algorithm in [41]	0.512	0.398	0.232

Paper I: Detection of Saccades and Postsaccadic Oscillations in the Presence of Smooth Pursuit

A novel algorithm for detection of saccades and PSO is designed, implemented, and evaluated. The algorithm detects these two types of eye movements regardless of whether the stimuli are static or dynamic. A database with eye-tracking signals when users viewed both static and dynamic stimuli was recorded using a high-speed tower mounted eye-tracker. The stimuli contain images, text, video clips, and dots moving in different directions and with different speeds. The first step in the proposed algorithm is the preprocessing stage, where blinks and disturbances originating from the recording are removed. In the second step, the acceleration signal is derived from the eye-tracking data and the saccades are detected. The acceleration signal is used since it is easier to discriminate between smooth pursuit movements and saccades in the acceleration signal compared to when the velocity signal is used. For detailed detection of the onset and offset of each saccade, three specialized criteria based on directional information in the position signal are used. In the third step, the PSO are detected. The detection is performed by modeling the position signal directly after each saccade using an all-pole model. The estimated parameters of the model determine whether the requirements for being a postsaccadic oscillation are satisfied.

The proposed method was evaluated by comparing the results of the algorithm to manual annotations and to the detection results of an adaptive velocity based algorithm [41]. Cohen's kappa, which measures the inter-observer agreement, was calculated between the results of the algorithm and the annotations. The results show that the detected events are in good agreement with the annotations and that the proposed algorithm outperforms the algorithm in [41]. The values for Cohen's kappa are shown in Table 6.2.

The paper discusses PSO and their appearance in the data. The detailed mechanisms behind PSO are still an unsolved problem, but PSO remain to appear in the data and need to be accounted for in a systematic way. The algorithm in Paper I allows the user to decide whether PSO should be classified as belonging to the saccades, as belonging to the fixations, as being their own event, or if they should simply be removed from the recorded data. This choice is crucial for important

measures such as duration of fixations and amplitudes of saccades.

Paper II: Detection of Fixations and Smooth Pursuit Movements in High-Speed Eye-Tracking Data

A novel algorithm for separation of fixations and smooth pursuit movements in high-speed eye-tracking data is proposed. The algorithm uses the inter-saccadic intervals between saccades and PSO, e.g., resulting from the algorithm in Paper I. In contrast to most of the previously proposed algorithms for detection of smooth pursuit movements, the proposed algorithm calculates characteristics of the signal at several spatial scales. In the first stage, characteristics on a sample-to-sample level are determined and the signal is segmented based on the local uniformity in the direction. In the second stage, four parameters are calculated for each segment. The four parameters are: dispersion, consistency in direction, positional displacement, and the range of the signal in the segment. Based on the four parameters, the segments are classified into three categories. In the final stage, fixations and smooth pursuit movements are discriminated by merging the categorized segments based on their properties. The advantage of using several spatial scales instead of only the sample-to-sample level is that the global characteristics of the signal can be taken into account. The algorithm is evaluated by computing five different performance measures that capture both general and specific aspects of the segmentation into fixations and smooth pursuit movements. The performance measures are: event properties, distribution of different types of events, sensitivity and specificity analysis, Cohen's kappa analysis, and scores evaluation. In this work, the detections of the proposed algorithm are compared to the detections of a velocity and dispersion based algorithm (I-VDT), to the detections of an algorithm based on principle component analysis, (I-PCA), and to annotations by two experts. The resulting Cohen's kappa are shown in Table 6.3, with Expert 1 as reference, and in Table 6.4, with Expert 2 as reference. The results show that the proposed algorithm outperforms the I-VDT algorithm and the I-PCA algorithm, but the inter-rater agreement between the two experts is even higher.

The paper discusses possibilities and challenges when separating fixations and smooth pursuit movements in high-speed eye-tracking data. The proposed algorithm makes it possible to separately investigate fixation and smooth pursuit properties. Together with the algorithm proposed in Paper I, it constitutes a complete event detector for eye-tracking signals recorded during dynamic stimuli. In addition, Paper II extensively discusses and compares different strategies for performance evaluation of event detection algorithms.

Table 6.3: Cohen's kappa when Expert 1 is compared to the proposed algorithm, the I-VDT, and Expert 2. (Paper II)

	Image	**Video**	**Moving dot**
Proposed algorithm	0.620	0.671	0.446
I-VDT	0.524	0.180	0.098
I-PCA	0.475	0.113	0.083
Expert 2	0.806	0.784	0.573

Table 6.4: Cohen's kappa when Expert 2 is compared to the proposed algorithm, the I-VDT, and Expert 1. (Paper II)

	Image	**Video**	**Moving dot**
Proposed algorithm	0.667	0.530	0.412
I-VDT	0.537	0.127	0.050
I-PCA	0.501	0.074	0.052
Expert 1	0.834	0.779	0.550

Paper III: Smooth Pursuit Detection in Binocular Eye-Tracking Data with Automatic Video-Based Performance Evaluation

Today, an increasing number of researchers record binocular eye-tracking signals from participants viewing moving stimuli. Since the majority of event detection algorithms are developed for monocular eye-tracking signals and often do not consider smooth pursuit movements, the additional information from using both eyes are not exploited in current event detection algorithms. The purpose of this study was to develop an event detection algorithm that uses binocular eye-tracking signals for improved detection of fixations and smooth pursuit movements. In this paper, the algorithm presented in Paper I is used for the detection of inter-saccadic intervals in binocular eye-tracking signals recorded during image-, moving dot-, and video- stimuli. The video stimuli contained clips with both stationary and moving cameras. The inter-saccadic intervals from both eyes are separately verified to have high enough signal quality by calculating and evaluating the high-frequency content in each interval. All intervals that pass the quality test, are included in a directional clustering procedure where samples with similar direction are clustered

Table 6.5: Results for the eye-tracking signals recorded with image stimuli for the test database (development database). (Paper III)

Algorithm	% **smooth pursuit**	% **correct smooth pursuit**	% **incorrect smooth pursuit**	% **fixation**
Proposed (Bin)	1.7 (4.1)	0.0 (0.0)	1.7 (4.1)	98.3 (95.9)
Proposed (Mono R)	6.9 (7.5)	0.0 (0.0)	6.9 (7.5)	93.1 (92.5)
Proposed (Mono L)	8.8 (7.7)	0.0 (0.0)	8.8 (7.7)	91.2 (92.3)
Algorithm in Paper II	4.5 (5.7)	0.0 (0.0)	4.5 (5.7)	95.5 (94.3)

together. In order to evaluate the temporal aspect of the directional clustering, a set of binary filters are applied to the resulting clustered samples. The binary filters are designed to either emphasize properties of fixations or properties of smooth pursuit movements. The output signals from the binary filters are added together and fixations and smooth pursuit movements are detected based on the sign of the summed signal.

The advantages of this method compared to the one in Paper II are that it is designed to imitate how we visually inspect the data and that it also is able to discriminate vergence from smooth pursuit movements if binocular data are available.

In order to evaluate the proposed algorithm, a novel evaluation strategy based on automatically detected objects in the stimuli is developed. A model, referred to as the video-gaze model, is proposed where intervals where the gaze is moving close to and in similar direction as the automatically detected moving objects are labelled as video-gaze movements. In the evaluation, the video-gaze movements are compared to the smooth pursuit movements detected by the proposed algorithm. The results of the evaluation are shown in Tables 6.5 – 6.6 for the proposed algorithm for both binocular and monocular mode. For comparison, the corresponding results of the algorithm in Paper II are also shown. The results show that it is advantageous to use binocular information to decrease the false detections of smooth pursuit movements in image stimuli without impairing the algorithm's ability to detect smooth pursuit movements in moving dot and video stimuli.

The novel evaluation strategy presented in this paper is the first automatic evaluation strategy for eye-tracking signals recorded with real video clips where the positions of the objects are not known or predefined beforehand by the experimenter. The advantages of using an automatic evaluation procedure is that time consuming and subjective manual annotations can be avoided. In addition, a larger number of signals can be used in the evaluation of the algorithm's performance. The drawback is that the evaluation is less detailed than, e.g., sensitivity and specificity analysis based on manual annotations.

Table 6.6: Results for the eye-tracking signals recorded with moving dot stimuli for the test database (development database). (Paper III)

Algorithm	% **smooth pursuit**	% **correct smooth pursuit**	% **incorrect smooth pursuit**	% **fixation**
Proposed (Bin)	83.1 (77.6)	80.4 (74.7)	2.7 (2.8)	16.9 (22.4)
Proposed (Mono R)	85.4 (79.9)	82.2 (76.7)	3.2 (3.2)	14.6 (20.1)
Proposed (Mono L)	85.1 (80.5)	82.0 (77.0)	3.1 (3.5)	14.9 (19.5)
Algorithm in Paper II	80.6 (73.4)	77.9 (70.8)	2.6 (2.6)	19.4 (26.6)

Paper IV: Head Movement Compensation and Multi-Modal Event Detection for Mobile Eye-Trackers

The aim of this study was to develop a multi-modal method for the detection of saccades, fixations, and smooth pursuit movements in eye-tracking data recorded using a mobile eye-tracker. The method includes compensation of head movements using an IMU and an event detection algorithm based on eye-tracking signals combined with information extracted from the scene video. Eye-tracking signals were recorded using a mobile eye-tracker and the orientation of the head was recorded using an IMU mounted on top of the mobile eye-tracker when participants were seated on a chair in front of a large screen. Participants were allowed to move their heads freely, but not change position in the room. Four experimental tasks were performed, which consisted of only eye movements (EM), only head movements (HM), and two parts with a combination of eye and head movements. The last two parts included artificial stimuli (EHM) and natural static stimuli (NS). The first step of the proposed method deals with compensation of head movements in the eye-tracking signal using the recorded orientation of the head. Objects in the scene camera were detected and their coordinates were head movement compensated in order to be expressed in the same coordinate system as the head movement compensated eye-tracking signals. In this multi-modal event detection algorithm, the saccades were detected using a combination of the velocity signal, the acceleration signal, the slope, and the amplitude of the eye-tracking signal. The eye-tracking signal of each inter-saccadic interval was compared to the trajectories of the detected objects of the scene camera image, and if any object moved in a similar direction and speed as the eye-tracking signal in the inter-saccadic interval, that object was selected. The selected object was used in the subsequent fixation and smooth pursuit detection. First the eye-tracking signal and the selected object trajectory were clustered based on their respective sample-to-sample directions. The binary filters presented in Paper III were optimized for the low sampling rate of the mobile eye-tracking signals, and applied to both coordinates of the eye-tracking signal and to

those of the selected object. The outputs when applying the filters to both types of signals were combined and the sign of the combined signal was used in order to segment the signal into fixations and smooth pursuit movements. Since smooth pursuit movements cannot be performed without a moving object, the presence of a selected object was utilized in the event detection algorithm in order to support the detection of smooth pursuit movements that move similar to the selected object. Correspondingly, smooth pursuit movements that were detected when no moving object was present were disqualified. In order to evaluate the performance of the IMU-based head movement compensation method, standard deviations of the inter-saccadic intervals for the compensated eye-tracking signal were compared to the standard deviations of the corresponding uncompensated signal intervals and to the standard deviations resulting from an alternative video-based head movement compensation strategy, see Table 6.7. The results show that all three experimental parts benefitted from the head movement compensation, and that the results of the IMU and the video-based methods were comparable for this type of data.

The event detection algorithm was evaluated by comparing the detection results to those of the built-in algorithm in the mobile eye-tracker and to the detections by the I-VDT algorithm. The average balanced accuracies of the three algorithms are shown in Table 6.8, together with the results of the proposed algorithm when disregarding information about moving objects. The results in Table 6.8 show that the proposed algorithm performs considerably better than the compared algorithms and that it is beneficial to include information in the event detection algorithm about moving objects detected in the scene video. In summary, Paper IV shows that by compensating for head movements using an IMU and by using information extracted from the scene video in the event detection algorithm, the performance of the event detector could be improved.

This paper discusses the challenges in event detection in eye-tracking signals recorded using a mobile eye-tracker with dynamic stimuli. The proposed method strives to use information from additional sources, such as the scene camera and the IMU, in order to improve the event detector's performance. A limitation of the proposed method is that it only allows head movements but not translational changes of the body position.

Table 6.7: Standard deviations of the gaze positions in inter-saccade intervals for three parts of the experiment. Uncompensated data are compared to compensated data both using an IMU data and using head movements extracted from the scene video.

	Not compensated		**Compensated**			
			IMU		**Video**	
	σ_{Ex} (°)	σ_{Ey} (°)	σ_{Gx} (°)	σ_{Gy} (°)	σ_{GVx} (°)	σ_{GVy} (°)
EM	0.16 (0.16)	0.18 (0.19)	0.09 (0.09)	0.12 (0.14)	0.10 (0.09)	0.13 (0.14)
EHM	0.81 (0.84)	0.69 (0.71)	0.14 (0.14)	0.17 (0.18)	0.18 (0.18)	0.20 (0.21)
HM	8.99 (8.73)	7.49 (6.54)	3.31 (3.11)	3.33 (2.93)	3.43 (2.96)	3.43 (2.70)

Table 6.8: Results of the average balanced accuracy for the proposed multi-modal algorithm, the proposed algorithm without selection of objects, the I-VDT-algorithm, and the built-in-algorithm of the mobile eye-tracker.

Algorithm	**Average balanced accuracy**
Proposed multi-modal	0.90
Proposed (no objects)	0.88
I-VDT	0.85
Built-in-algorithm	0.75

References

[1] E. Widmaier, H. Raff, and K. Strang, *Vander's Human Physiology - The Mechanisms of Body Function*. Mc Graw Hill, 2006.

[2] L. S. Liebovitch, "Why the eye is round," in *The Biology of the Eye*, vol. 10, pp. 1 – 19, Elsevier, 2005.

[3] N. Ehlers and J. Hjortdal, "The cornea: Epithelium and stroma," in *The Biology of the Eye*, vol. 10, pp. 83 – 111, Elsevier, 2005.

[4] M. la Cour and B. Ehinger, "The retina," in *The Biology of the Eye*, vol. 10, pp. 195 – 252, Elsevier, 2005.

[5] J. Enderle and J. Bronzino, *Introduction to biomedical engineering*. Academic Press, 2011.

[6] S. Martinez-Conde, S. Macknik, and D. Hubel, "The role of fixational eye movements in visual perception," *Nature Reviews Neuroscience*, vol. 5, no. 3, pp. 229–240, 2004.

[7] R. Leigh and D. Zee, *The Neurology of Eye Movements*. Oxford University Press, 2006.

[8] M. Rucci and J. D. Victor, "The unsteady eye: an information-processing stage, not a bug," *Trends in neurosciences*, vol. 38, no. 4, pp. 195–206, 2015.

[9] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2011.

[10] A. T. Bahill, M. R. Clark, and L. Stark, "The main sequence, a tool for studying human eye movements," *Mathematical Biosciences*, vol. 24, no. 3, pp. 191–204, 1975.

[11] C. Ludwig and I. Gilchrist, "Measuring saccade curvature: A curve-fitting approach," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 618–624, 2002.

[12] E. Kowler, "Eye movements: The past 25 years," *Vision Research*, vol. 51, pp. 1457–1483, 2011.

[13] H. Wyatt and J. Pola, "Smooth pursuit eye movements under open-loop and closed-loop conditions," *Vision Research*, vol. 23, no. 10, pp. 1121–1131, 1983.

[14] C. Meyer, A. Lasker, and D. Robinson, "The upper limit of human smooth pursuit.," *Vision Research*, vol. 25, no. 4, pp. 561–563, 1985.

[15] D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 478 –500, 2010.

[16] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imaging*, vol. 8, no. 5, pp. 357–377, 2002.

[17] D. H. Yoo and M. J. Chung, "A novel non-intrusive eye gaze estimation using cross-ratio under large head motion," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 25–51, 2005.

[18] R. S. Hessels, R. Andersson, I. T. Hooge, M. Nyström, and C. Kemner, "Consequences of eye color, positioning, and head movement for eye-tracking data quality in infant research," *Infancy*, vol. 20, no. 6, pp. 601–633, 2015.

[19] N. Holmberg, H. Sandberg, and K. Holmqvist, "Advert saliency distracts children's visual attention during task-oriented internet use," *Frontiers in psychology*, vol. 5, 2014.

[20] K. Gidlöf, A. Wallin, R. Dewhurst, and K. Holmqvist, "Gaze behavior during decision making in a natural environment," *Journal of Eye movement research*, vol. 6, no. 1, pp. 1–14, 2013.

[21] M. Land and P. McLeod, "From eye movements to actions: how batsmen hit the ball," *Nature neuroscience*, vol. 3, no. 12, pp. 1340–1345, 2000.

[22] L. R. Young and D. Sheena, "Survey of eye movement recording methods," *Behavior research methods & instrumentation*, vol. 7, no. 5, pp. 397–429, 1975.

[23] A. Duchowski, *Eye tracking methodology: Theory and practice*, vol. 373. Springer Science & Business Media, 2007.

[24] D. Robinson, "A method of measuring eye movement using a scleral search coil in a magnetic field," *IEEE Transactions on Bio-medical Electronics*, vol. 10, no. 4, pp. 137–145, 1963.

[25] J. Van der Geest and M. Frens, "Recording eye movements with video-oculography and scleral search coils: a direct comparison of two methods," *Journal of neuroscience methods*, vol. 114, no. 2, pp. 185–195, 2002.

[26] H. D. Crane and C. Steele, "Accurate three-dimensional eyetracker," *Applied optics*, vol. 17, no. 5, pp. 691–705, 1978.

[27] H. D. Crane and C. M. Steele, "Generation-v dual-purkinje-image eye-tracker," *Applied Optics*, vol. 24, no. 4, pp. 527–537, 1985.

[28] H. Deubel and B. Bridgeman, "Fourth purkinje image signals reveal eye-lens deviations and retinal image distortions during saccades," *Vision Research*, vol. 35, no. 4, pp. 529–538, 1995.

[29] K. Rayner, "Eye movements in reading and information processing: 20 years of research.," *Psychological bulletin*, vol. 124, no. 3, p. 372, 1998.

[30] A. T. Bahill, M. R. Clark, and L. Stark, "Glissades-eye movements generated by mismatched components of the saccadic motorneuronal control signal," *Mathematical biosciences*, vol. 26, pp. 303–318, 1975.

[31] Z. Kapoula, D. Robinson, and T. Hain, "Motion of the eye immediately after a saccade," *Experimental Brain Research*, vol. 61, no. 2, pp. 386–394, 1986.

[32] S. Barnaby Hutton, "Lens mobility influences post-saccadic ringing in video-based eye tracking," in *Book of abstracts of the 17th European Conference on Eye movement*, p. 251, Journal of Eye movement research, 2013.

[33] M. Nyström, I. Hooge, and K. Holmqvist, "Post-saccadic oscillations in eye movement data recorded with pupil-based eye trackers reflect motion of the pupil inside the iris," *Vision Research*, vol. 92, pp. 59–66, 2013.

[34] I. Hooge, M. Nyström, T. Cornelissen, and K. Holmqvist, "The art of braking: Post saccadic oscillations in the eye tracker signal decrease with increasing saccade size," *Vision Research*, vol. 112, pp. 55–67, 2015.

[35] D. M. Stampe, "Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems," *Behavior Research Methods, Instruments, & Computers*, vol. 25, no. 2, pp. 137–142, 1993.

[36] S. Chartier and P. Renaud, "Eye-tracker data filtering using pulse coupled neural network," in *17th IASTED International Conference on Modelling and Simulation*, 2006.

[37] M. Toivanen, "An advanced kalman filter for gaze tracking signal," *Biomedical Signal Processing and Control*, vol. 25, pp. 150–158, 2016.

[38] O. Špakov, "Comparison of eye movement filters used in hci," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 281–284, ACM, 2012.

[39] S. Stuart, B. Galna, S. Lord, L. Rochester, and A. Godfrey, "Quantifying saccades while walking: Validity of a novel velocity-based algorithm for mobile eye tracking," in *Proceedings of 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5739–5742, IEEE, 2014.

[40] S. M. Munn, L. Stefano, and J. B. Pelz, "Fixation-identification in dynamic scenes: Comparing an automated algorithm to manual coding," in *Proceedings of the 5th symposium on Applied perception in graphics and visualization*, pp. 33–42, ACM, 2008.

[41] M. Nyström and K. Holmqvist, "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data," *Behavior Research Methods*, vol. 42, no. 1, pp. 188–204, 2010.

[42] M. Dorr, T. Martinetz, K. R. Gegenfurtner, and E. Barth, "Variability of eye movements when viewing dynamic natural scenes," *Journal of vision*, vol. 10, no. 10, pp. 1–17, 2010.

[43] D. J. Berg, S. E. Boehnke, R. A. Marino, D. P. Munoz, and L. Itti, "Free viewing of dynamic stimuli by humans and monkeys," *Journal of Vision*, vol. 9, no. 5, pp. 1–15, 2009.

[44] F. Behrens, M. MacKeben, and W. Schröder-Preikschat, "An improved algorithm for automatic detection of saccades in eye movement data and for calculating saccade parameters," *Behavior Research Methods*, vol. 42, no. 3, pp. 701–708, 2010.

[45] D. Sauter, B. Martin, N. Di Renzo, and C. Vomscheid, "Analysis of eye tracking movements using innovations generated by a kalman filter," *Medical and Biological Engineering and Computing*, vol. 29, no. 1, pp. 63–69, 1991.

[46] R. Engbert and R. Kliegl, "Microsaccades uncover the orientation of covert attention," *Vision Research*, vol. 43, no. 9, pp. 1035–1045, 2003.

[47] P.-H. Niemenlehto, "Constant false alarm rate detection of saccadic eye movements in electro-oculography," *Computer Methods and Programs in Biomedicine*, vol. 96, no. 2, pp. 158–171, 2009.

[48] O. Komogortsev, D. Gobert, S. Jayarathna, D. Hyong Koh, and S. Gowda, "Standardization of automated analyses of oculomotor fixation and saccadic behaviors," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2635–2645, 2010.

[49] F. Behrens and L.-R. Weiss, "An algorithm separating saccadic from nonsaccadic eye movements automatically by use of the acceleration signal," *Vision Research*, vol. 32, no. 5, pp. 889–893, 1992.

[50] H. Widdel, "Operational problems in analysing eye movements," in *A. 13. Gale & F. Johnson (Eds.), Theoretical and Applied Aspects of Eye Movement Research*, (New York), pp. 21–29, Elsevier, 1984.

[51] D. Salvucci and J. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, (New York), pp. 71–78, ACM, 2000.

[52] J. S. A. Lopez, *Off-the-shelf Gaze Interaction*. PhD thesis, PhD thesis, 2009.

[53] C. A. Rothkopf and J. B. Pelz, "Head movement estimation for wearable eye tracker," in *Proceedings of the 2004 symposium on Eye tracking research & applications*, pp. 123–130, ACM, 2004.

[54] R. van der Lans, M. Wedel, and R. Pieters, "Defining eye-fixation sequence across individuals and tasks: the binocular-individual threshold (bit) algorithm," *Behavior Research Methods*, vol. 43, no. 1, pp. 239–257, 2011.

[55] A. Bulling, J. Ward, H. Gellersen, G. Tröster, *et al.*, "Eye movement analysis for activity recognition using electrooculography," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 741–753, 2011.

[56] M. S. Mould, D. H. Foster, K. Amano, and J. P. Oakley, "A simple nonparametric method for classifying eye fixations," *Vision Research*, vol. 57, pp. 18–25, 2012.

[57] E. Kasneci, G. Kasneci, T. Kübler, and W. Rosenstiel, "Online recognition of fixations, saccades, and smooth pursuits for automated analysis of traffic hazard perception," in *Artificial Neural Networks*, vol. 4 of *Springer Series*

in Bio-/Neuroinformatics, pp. 411–434, Springer International Publishing, 2015.

[58] D. B. Liston, A. E. Krukowski, and L. S. Stone, "Saccade detection during smooth tracking," *Displays*, vol. 34, no. 2, pp. 171–176, 2013.

[59] P. M. Daye and L. M. Optican, "Saccade detection using a particle filter," *Journal of neuroscience methods*, vol. 235, pp. 157–168, 2014.

[60] S. D. König and E. A. Buffalo, "A nonparametric method for detecting fixations and saccades using cluster analysis: Removing the need for arbitrary thresholds," *Journal of neuroscience methods*, vol. 227, pp. 121–131, 2014.

[61] M. Vidal, A. Bulling, and H. Gellersen, "Detection of smooth pursuits using eye movement shape features," in *Proceedings of the symposium on eye tracking research and applications*, pp. 177–180, ACM, 2012.

[62] T. Santini, W. Fuhl, T. Kübler, and E. Kasneci, "Bayesian identification of fixations, saccades, and smooth pursuits," in *Proceedings of the 2016 symposium on Eye tracking research & applications*, ACM, 2016. Forthcomming.

[63] O. Komogortsev and A. Karpov, "Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades," *Behavior Research Methods*, vol. 45, no. 1, pp. 203–215, 2013.

[64] A. I. Korda, P. A. Asvestas, G. K. Matsopoulos, E. M. Ventouras, and N. P. Smyrnis, "Automatic identification of oculomotor behavior using pattern recognition techniques," *Computers in biology and medicine*, vol. 60, pp. 151–162, 2015.

[65] M. Toivanen, K. Pettersson, and K. Lukander, "A probabilistic real-time algorithm for detecting blinks, saccades, and fixations from eog data," *Journal of Eye Movement Research*, vol. 8, no. 2, 2015.

[66] G. Veneri, P. Piu, F. Rosini, P. Federighi, A. Federico, and A. Rufa, "Automatic eye fixations identification based on analysis of variance and covariance," *Pattern Recognition Letters*, vol. 32, no. 13, pp. 1588–1593, 2011.

[67] J. Otero-Millan, J. L. A. Castro, S. L. Macknik, and S. Martinez-Conde, "Unsupervised clustering method to detect microsaccades," *Journal of vision*, vol. 14, no. 2, p. 18, 2014.

[68] M. Bettenbühl, C. Paladini, K. Mergenthaler, R. Kliegl, R. Engbert, and M. Holschneider, "Microsaccade characterization using the continuous wavelet transform and principal component analysis," *Journal of Eye Movement Research*, vol. 4, no. 5, 2011.

[69] P. Blignaut, "Fixation identification: The optimum threshold for a dispersion algorithm," *Attention, Perception, & Psychophysics*, vol. 71, no. 4, pp. 881–895, 2009.

[70] K. Harezlak and P. Kasprowski, "Evaluating quality of dispersion based fixation detection algorithm," in *Information Sciences and Systems 2014*, pp. 97–104, Springer International Publishing, 2014.

[71] A. Savitzky and M. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[72] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[73] A. Duchowski, E. Medlin, N. Cournia, H. Murphy, A. Gramopadhye, S. Nair, J. Vorah, and B. Melloy, "3-d eye movement analysis," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 573–591, 2002.

[74] R. Engbert and K. Mergenthaler, "Microsaccades are triggered by low retinal image slip," *Proceedings of the National Academy of Sciences*, vol. 103, no. 18, pp. 7192–7197, 2006.

[75] O. V. Komogortsev and J. I. Khan, "Kalman filtering in the design of eye-gaze-guided computer interfaces," in *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, pp. 679–689, Springer, 2007.

[76] M. Vidal, A. Bulling, and H. Gellersen, "Analysing eog signal features for the discrimination of eye movements with wearable devices," in *Proceedings of the 1st international workshop on pervasive eye tracking & mobile eye-based interaction*, pp. 15–20, ACM, 2011.

[77] S. V. Wass, T. J. Smith, and M. H. Johnson, "Parsing eye-tracking data of variable quality to provide accurate fixation duration estimates in infants and adults," *Behavior Research Methods*, vol. 45, no. 1, pp. 229–250, 2013.

[78] S. Winkler and S. Ramanathan, "Overview of eye tracking datasets.," in *Proceedings of International Workshop on Quality of Multimedia Experience*, pp. 212–217, 2013.

[79] K. Kurzhals, C. F. Bopp, J. Bässler, F. Ebinger, and D. Weiskopf, "Benchmark data for evaluating visualization and analysis techniques for eye tracking for video stimuli," in *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pp. 54–60, ACM, 2014.

[80] I. R. S. de Urabain, M. H. Johnson, and T. J. Smith, "Grafix: A semiautomatic approach for parsing low-and high-quality eye-tracking data," *Behavior Research Methods*, vol. 47, no. 1, pp. 53–72, 2014.

[81] R. S. Allison, M. Eizenman, and B. S. Cheung, "Combined head and eye tracking system for dynamic testing of the vestibular system," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 11, pp. 1073–1082, 1996.

[82] J. Pelz, M. Hayhoe, and R. Loeber, "The coordination of eye, head, and hand movements in a natural task," *Experimental Brain Research*, vol. 139, no. 3, pp. 266–277, 2001.

[83] E. M. Foxlin, M. Harrington, and Y. Altshuler, "Miniature six-dof inertial system for tracking hmds," in *Aerospace/Defense Sensing and Controls*, pp. 214–228, International Society for Optics and Photonics, 1998.

[84] H. Collewijn and J. B. Smeets, "Early components of the human vestibuloocular response to head rotation: latency and gain," *Journal of Neurophysiology*, vol. 84, no. 1, pp. 376–389, 2000.

[85] T. Pfeiffer and P. Renner, "Eyesee3d: A low-cost approach for analyzing mobile 3d eye tracking data using computer vision and augmented reality technology," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 369–376, ACM, 2014.

[86] R. Ronsse, O. White, and P. Lefevre, "Computation of gaze orientation under unrestrained head movements," *Journal of neuroscience methods*, vol. 159, no. 1, pp. 158–169, 2007.

[87] S. Herholz, L. L. Chuang, T. Tanner, H. H. Bülthoff, and R. W. Fleming, "Libgaze: Real-time gaze-tracking of freely moving observers for wall-sized displays," in *13th International Fall Workshop on Vision, Modeling, and Visualization (VMV 2008)*, pp. 101–110, IOS Press, 2008.

[88] K. Essig, D. Dornbusch, D. Prinzhorn, H. Ritter, J. Maycock, and T. Schack, "Automatic analysis of 3d gaze coordinates on scene objects using data from eye-tracking and motion-capture systems," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 37–44, ACM, 2012.

[89] B. Cesqui, R. van de Langenberg, F. Lacquaniti, and A. d'Avella, "A novel method for measuring gaze orientation in space in unrestrained head conditions," *Journal of vision*, vol. 13, no. 8, p. 28, 2013.

[90] L. Paletta, K. Santner, G. Fritz, H. Mayer, and J. Schrammel, "3d attention: measurement of visual saliency using eye tracking glasses," in *CHI'13*

Extended Abstracts on Human Factors in Computing Systems, pp. 199–204, ACM, 2013.

[91] S. M. Munn and J. B. Pelz, "3d point-of-regard, position and head orientation from a portable monocular video-based eye tracker," in *Proceedings of the 2008 symposium on Eye tracking research & applications*, pp. 181–188, ACM, 2008.

[92] T. Kinsman, K. Evans, G. Sweeney, T. Keane, and J. Pelz, "Ego-motion compensation improves fixation detection in wearable eye tracking," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 221–224, ACM, 2012.

[93] K. Takemura, K. Takahashi, J. Takamatsu, and T. Ogasawara, "Estimating 3-d point-of-regard in a real environment using a head-mounted eye-tracking system," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 4, pp. 531–536, 2014.

[94] B. S. Reddy and B. N. Chatterji, "An fft-based technique for translation, rotation, and scale-invariant image registration," *IEEE transactions on image processing*, vol. 5, no. 8, pp. 1266–1271, 1996.

[95] N. Sim, C. Gavriel, W. W. Abbott, and A. Faisal, "The head mouse?head gaze estimation" in-the-wild" with low-cost inertial sensors for bmi use," in *Proceedings of 6th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 735–738, IEEE, 2013.

[96] K. Satoh, S. Uchiyama, and H. Yamamoto, "A head tracking method using bird's-eye view camera and gyroscope," in *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 202–211, IEEE Computer Society, 2004.

[97] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pp. 1–7, IEEE, 2011.

[98] C. Ahlstrom, T. Victor, C. Wege, and E. Steinmetz, "Processing of eye/head-tracking data in large-scale naturalistic driving data sets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 553–564, 2012.

[99] Y. Zhang, J. Tan, Z. Zeng, W. Liang, and Y. Xia, "Monocular camera and imu integration for indoor position estimation," in *Proceedings of 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1198–1201, IEEE, 2014.

[100] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes, "Elan: a professional framework for multimodality research," in *Proceedings of LREC*, vol. 2006, p. 5th, 2006.

[101] F. Papenmeier and M. Huff, "Dynaoi: A tool for matching eye-movement data with dynamic areas of interest in animations and movies," *Behavior research methods*, vol. 42, no. 1, pp. 179–187, 2010.

[102] D. F. Pontillo, T. B. Kinsman, and J. B. Pelz, "Semanticode: using content similarity and database-driven matching to code wearable eyetracker gaze data," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pp. 267–270, ACM, 2010.

[103] K. Essig, N. Sand, T. Schack, J. Künsemöller, M. Weigelt, and H. Ritter, "Fully-automatic annotation of scene videos: Establish eye tracking effectively in various industrial applications," in *Proceedings of SICE Annual Conference 2010*, pp. 3304–3307, 2010.

[104] G. Brône, B. Oben, and T. Goedemé, "Towards a more effective method for analyzing mobile eye-tracking data: integrating gaze data with object recognition algorithms," in *Proceedings of the 1st international workshop on pervasive eye tracking & mobile eye-based interaction*, pp. 53–56, ACM, 2011.

[105] S. De Beugher, G. Brône, and T. Goedemé, "Automatic analysis of in-the-wild mobile eye-tracking experiments using object, face and person detection," in *Proceedings of the international conference on computer vision theory and applications (VISIGRAPP 2014)*, vol. 1, pp. 625–633, 2014.

[106] K. Essig, D. Abashidze, M. Prasad, and T. Schack, "Gazevideoanalyser: A modular software approach towards automatic annotation of gaze videos," in *Proceedings of the second international workshop on solutions for automatic Gaze-data analysis 2015 (SAGA 2015)*, 2015.

Part II

Included Papers

Paper I

Detection of Saccades and Postsaccadic Oscillations in the Presence of Smooth Pursuit

Abstract

A novel algorithm for detection of saccades and postsaccadic oscillations in the presence of smooth pursuit movements is proposed. The method combines saccade detection in the acceleration domain with specialized on- and offset criteria for saccades and postsaccadic oscillations. The performance of the algorithm is evaluated by comparing the detection results to those of an existing velocity based adaptive algorithm and a manually annotated database. The results show that there is a good agreement between the events detected by the proposed algorithm and those in the annotated database with Cohen's kappa around 0.8 for both a development and a test database. In conclusion, the proposed algorithm accurately detects saccades and postsaccadic oscillations as well as intervals of disturbances.

Linnéa Larsson, Marcus Nyström, and Martin Stridh,
"Detection of Saccades and Postsaccadic Oscillations in the Presence of Smooth Pursuit,"
in *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 9, pp. 2484–2493, 2013.

1 Introduction

Measurement of eye movements is important for basic research in visual attention, perception, and cognition, as well as for clinical applications investigating the functionality of the brain or to diagnose physiological disorders, such as Alzheimer's [1], HIV-1 infected patients with eye movement dysfunction [2], and schizophrenia [3]. The interest in eye-tracking is also increasing in applied fields with strong commercial interests, e.g., web page navigation, online shopping, and interaction with computers. An important new development in the field is that eye movement studies are starting to use more realistic dynamic scenes as stimuli, e.g., short videos, compared to earlier when mainly static images were used. Since the tools for analyzing eye movement signals are mainly developed for static images, the use of dynamic scenes requires a new set of algorithms for segmentation of recorded signals into eye movement events. In order to be able to draw correct conclusions about the underlying processes in the brain or to be able to control a computer, reliable algorithms for detection and classification of eye movements are crucial.

When studying eye movements, mainly three movements are identified: The slow period when the eye is more or less still and visual information is taken in is referred to as a *fixation*, which is characterized by low positional dispersion, low velocity, and a duration of about 200 – 300 ms [4]. When the eye is shifting from one position to another, the movement is referred to as a *saccade*, which is a very rapid movement with typical velocities ranging from 30°/s to 500°/s and durations ranging from 30 ms to 80 ms [4]. Very little visual information is gathered during saccades [5]. These two eye movements are the most common ones when observing static objects. When the observed objects are moving, e.g., when watching a dynamic scene, other eye movements may occur that are related to the movement in the scene. One such eye movement is the *smooth pursuit*, which occurs when the eye has a moving target to follow [5]. The velocity of a smooth pursuit movement depends on the speed of the moving target, but is typically below 30°/s [4], even though the velocity can be as high as 100°/s [6]. However, for targets with velocities higher than 30°/s, the eye movements typically consist of both saccades and smooth pursuit movements. In order for the eye to be able to accurately follow the target, catch-up saccades are required since the pursuit gain falls below one [5].

In addition to these three eye movements, we investigate in this paper the oscillatory behavior that may occur at the end of a saccade. In [7], three different types of movements in connection to the saccade were categorized: dynamic overshoot, which is a fast movement with velocities of 10 – 100°/s, glissadic overshoot which is a slow drifting movement with velocities of 2 – 20°/s, and static overshoot, which is a corrective saccade that starts 200 ms after the primary saccade. In this paper, we are interested in detecting all types of high-velocity transients that may occur at the end of the saccade, e.g., overshoot/undershoot, oscillatory behavior and immediate changes in direction compared to the preceding saccade. All of these types of

movements are referred here to as *postsaccadic oscillations* (PSO).

Algorithms for detection of eye movements can broadly be divided into two groups: dispersion based and velocity/acceleration based algorithms. Algorithms based on dispersion are mainly used for signals with a lower sampling frequency (<200 Hz), while the velocity/acceleration based methods are used for signals with a higher sampling frequency (>200 Hz) [4]. In recent years, algorithms used for detection of eye movements have developed from solely using a preset threshold of dispersion or velocity/acceleration [8], towards adaptive algorithms where the thresholds are estimated from the signals [9, 10, 11, 12]. By using an adaptive threshold, individual differences between participants and trials can be taken into account, and the algorithm becomes less dependent on the user's ability to correctly set the thresholds. The majority of the adaptive algorithms referred to above are velocity-based and were developed for detection of eye movements when the stimulus is static and the signals thus do not contain any smooth pursuit movements. However, in signals where the stimulus is dynamic, it is important to also consider the smooth pursuit movements in the detection algorithm, since their presence may otherwise render the detection of the other eye movements difficult. The major problem is that the velocity of a fast smooth pursuit movement overlaps with the velocity range of a slow saccade [13, 11], making it difficult to set a velocity threshold for discrimination between these two types of eye movements [14].

In order to reliably detect saccades in signals where smooth pursuit movements are present, the acceleration signal has been used since the acceleration of saccades is higher than that of smooth pursuit movements [11]. A real-time algorithm where the acceleration signal is employed for detection of saccades is found in the commercial EyeLink algorithm. In order for a saccade to be detected, a combination of thresholds for changes in position, velocity, and acceleration has to be satisfied [15]. Another real-time algorithm that uses the acceleration signal is the adaptive algorithm proposed in [11]. In order for the algorithm to detect the beginning of the saccade, the acceleration of each sample is compared to a threshold computed from the preceding 200 samples. For the determination of the end of the saccade, a combination of the acceleration threshold and the end of the monotonicity in the position signal for the saccade is employed [11].

Today, there are very few algorithms that include the detection of PSO. A velocity based adaptive algorithm for detection of PSO in signals recorded for participants viewing static stimuli was proposed in [9]. As pointed out by the authors, it is important for an algorithm to be consistent to whether the PSO are included in the saccades, in the fixations, or are marked as a separate type of eye movement. This choice is crucial for calculation of durations of fixations and amplitudes of saccades. While the origin of PSO in pupil-based eye-trackers remains unclear, they have been reported to occur in 48% and 59% of the saccades for participants performing reading and scene perception, respectively [9]. Postsaccadic oscillations

have been reported in data recorded with Dual Purkinje eye-trackers (DPIs), but were not found in simultaneous recordings with scleral search coils [16]. This suggests that PSO in data recorded by a DPI result from motion of the lens relative to the eyeball rather than motion of the eye in its orbit. To our knowledge, the occurrence and the properties of PSO when viewing a dynamic scene have not been investigated.

In order to objectively evaluate the performance of a detection algorithm for eye movements, the actual movement of the eye needs to be available. Strategies to estimate the actual movement of the eye includes to let an expert annotate the signals [4], generate simulated signals [17], or use the stimuli as a reference to the actual eye movement [18]. When evaluating the performance of a detection algorithm for PSO, the only option is to manually annotate the signals, since the occurrence of PSO seems to be involuntary and idiosyncratic [19].

The purpose of this paper is threefold: First, we propose a robust algorithm for the detection of saccades and PSO in signals recorded when viewing static as well as dynamic scenes, where the intrinsic structure is motivated by underlying physiological properties of the saccades and PSO. Second, a mathematical model for describing the properties of PSO is proposed. Third, a framework for evaluation and comparison of different detection algorithms is proposed, including a Graphical User Interface (GUI) for presentation of the outputs of detection algorithms and for annotation of signals. The paper is outlined as follows: The proposed algorithm and the evaluation procedure are presented in Sec. 2. A description of the database with eye movement signals is given in Sec. 3. The results are presented in Sec. 4, and finally, the algorithm and its potential are discussed in Sec. 5.

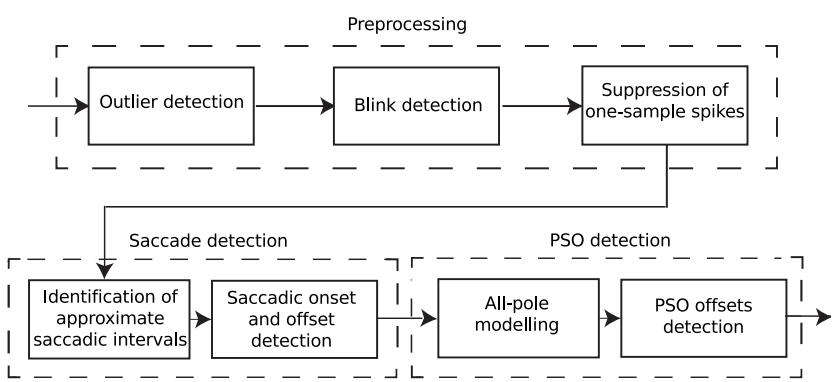


Figure 1: The overall structure of the algorithm.

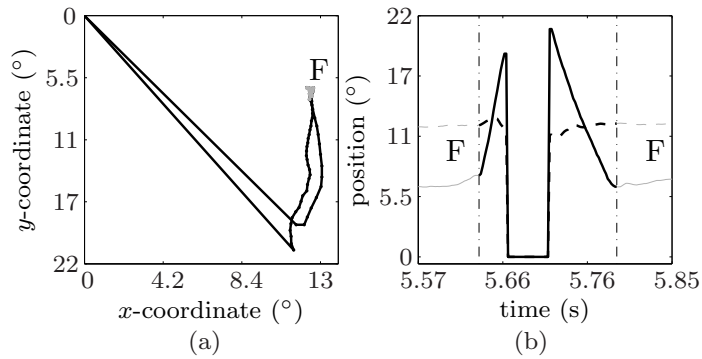


Figure 2: An example of blink detection. In (a), the vertical saccade-like movement in the y-coordinate in the beginning and in the end of the blink. The black line is showing the blink and the grey line the fixation (F) before and after the blink. In (b), the detection of a blink in the $x-$coordinate (dashed) and $y-$coordinate (solid) over time. Onset and offset of the blink is marked with dashed-dotted vertical lines, and the black line indicates the blink and the grey line the fixation (F) before and after the blink.

2 Methods

The proposed method comprises three different stages: preprocessing, saccade detection, and PSO detection. An overview of the method is shown in Fig. 1. In the first stage, disturbances originating from the recording process are removed. In the second stage, the saccades are detected using criteria reflecting their physiological characteristics, and finally, in the third stage, the detection of PSO is performed.

2.1 Preprocessing

In the preprocessing stage, three different types of disturbances are excluded from the dataset: screen outliers, blinks, and one-sample spikes. All samples corresponding to positions outside a margin of $1.5°$ added to the geometry of the stimulus screen are marked as disturbances. During blinks, when the eyelid is closed, the eye-tracker cannot detect the pupil and therefore the eye-tracker used in this study sets the x- and y- coordinates to $(0, 0)$. By detecting these zeros in the position signal, the blinks are detected. However, in the beginning and end of a blink, the eyelid is not completely closed and the pupil can therefore partly be detected. In the position signal, vertical saccade-like movements therefore appear at the start and end of the blink, see Fig. 2a. In order to remove as much as possible of the erroneous coordinates caused by the blink, the on- and offsets of the blink are defined as the first local minimum in the y- coordinate, before and after the detected zeros,

see the vertical dash-dotted lines in Fig. 2b. In this work, the blink duration was limited to 700 ms. Notice that also during other disturbances than blinks when the eye-tracker for some reason cannot detect the pupil, the x- and y- coordinates are set to $(0, 0)$. These disturbances are treated in the same way as blinks and all such samples are marked as disturbances. A common type of disturbance in video-based eye-tracking is that the corneal reflection is not correctly detected in the image of the eye which will result in a rapid positional change of one sample in an unexpected direction and back again. This type of artifact is referred to as a one-sample spike [20]. In order to remove one-sample spikes, a median filter of length 3 can be used [20]. However, in order to remove one-sample spikes but avoid suppressing PSO as well as small variations during fixations, an amplitude and a velocity criteria, of which both need to be satisfied, are used to activate the filter. The amplitude threshold, a_{min}, requires a minimum amplitude of the removed one-sample spike. The value of the parameter a_{min} is given in Table 3, which lists the settings of all intrinsic parameters in the proposed algorithm. The velocity criteria is based on that, since PSO always occurs directly after saccades, the sample-to-sample velocities before PSO are larger than those during PSO. Therefore, the sample-to-sample velocities before a one-sample spike must be lower than those during a one-sample spike. An example of the suppression of one-sample spikes in a signal from the database used in this paper is shown in Fig. 3.

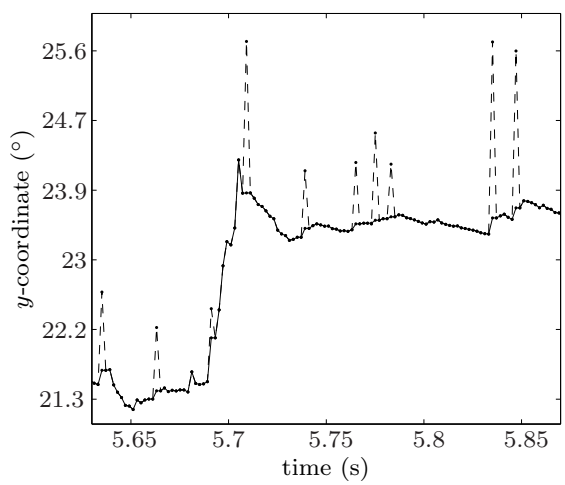


Figure 3: Example of removal of one-sample spikes, where the dashed line is the unfiltered signal and the solid line is the signal after the one-sample spike suppression.

2.2 Saccade detection

The first type of eye movement to be detected after the preprocessing stage is the saccade. The detection of saccades is divided into two steps which are shown in the lower left block of Fig. 1.

Identification of approximate saccadic intervals

Since saccades, in contrast to both fixations and smooth pursuit movements, are fast eye movements with high acceleration, they are detected in the acceleration domain. The angular velocities, v_x and v_y, are computed by filtering the position signals, $x(n)$ and $y(n)$, using the filter,

$$h(n) = \frac{1}{20}\left[\begin{array}{ccccccccc} -1 & -1 & -1 & -1 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$$

which is a modified version of the difference and smoothening filter proposed in [10] for signals sampled at 250 Hz. In order for the filter to operate at the same period of time and to give similar effect on signals sampled at 500 Hz, the length of the filter was doubled and the filter coefficients were scaled. The angular accelerations, a_x and a_y, are calculated by applying the same filter to the velocities. In the acceleration signal, *approximate saccadic intervals*, which are segments that include both saccades and possible PSO, are detected. The method used for delineation of such intervals is based on the method for detection of microsaccades proposed in [10]. Since the nature of saccades and microsaccades are similar in many aspects, (cf. [21]), the method can be used also for saccade detection. For each of the x- and y- components, the individual acceleration threshold is based on the standard deviations, σ_x and σ_y, of the acceleration distributions. The thresholds are defined as, $\eta_x = \lambda\sigma_x$ and $\eta_y = \lambda\sigma_y$, where λ is a constant that decides how many standard deviations that separates saccades and possible PSO from the rest of the eye movements. Two index vectors indicating the approximate saccadic intervals in the x- and y- components, $I_x(n)$ and $I_y(n)$, are created such that

$$I_x(n) = \begin{cases} 1, & |a_x(n)| > \eta_x \quad \forall n \\ 0, & otherwise \end{cases}$$

$$I_y(n) = \begin{cases} 1, & |a_y(n)| > \eta_y \quad \forall n \\ 0, & otherwise \end{cases}$$

where ones indicate saccades or possible PSO and zeros reflect other types of eye movements or disturbances. The index vectors $I_x(n)$ and $I_y(n)$ are merged into one index vector.

$$I'(n) = \begin{cases} 1, & I_x(n) = 1 \quad | \quad I_y(n) = 1 \\ 0, & otherwise \end{cases}$$

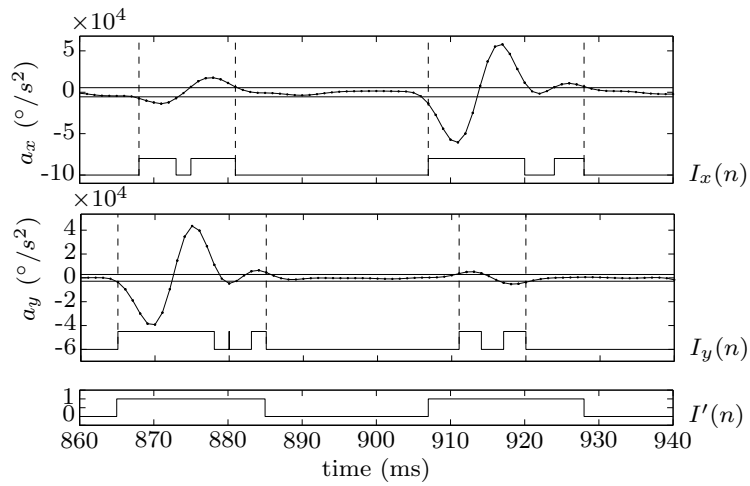


Figure 4: Example of two consecutive approximate saccadic intervals for the x- and y- components of the acceleration signal, respectively. Below each component, the index vectors $I_x(n)$ and $I_y(n)$, are shown. The bottom panel shows the index vector, $I'(n)$, for the two final approximate saccadic intervals.

Every group of consecutive ones comprises an approximate saccadic interval. Since two saccades cannot appear closer than a certain time, t_{min}, which is the time corresponding to the minimum duration of a fixation, two approximate saccadic intervals that occur closer than t_{min} are merged and the zeros between are converted to ones. Finally, an approximate saccadic interval must have a duration larger than T ms in order to be valid. The x- and y- components of the acceleration signal for two approximate saccadic intervals are together with the index vectors $I_x(n)$ and $I_y(n)$, shown in Fig. 4.

Saccadic onset and offset detection

In the second part of the saccade detection, the exact onsets and offsets of the saccades are identified. For each approximate saccadic interval, i, the detection starts by determining the sample of maximum velocity, k_i. From sample k_i, the search for the exact onset and offset is performed using three criteria $(a) - (c)$. These criteria are evaluated from sample k_i in the forward direction for the offset search, and in the backward direction for the onset search. In both directions, each criteria is compared to a predefined threshold. The exact onset and offset are set when at least one of the criteria is satisfied for a sufficient number of consecutive samples. The criteria are:

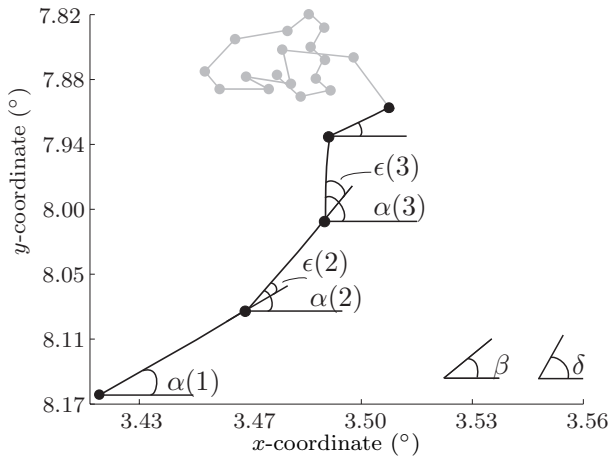


Figure 5: An illustration of the calculation of the directions, $\alpha(n)$ and change in directions, $\epsilon(n)$, n = 1,2,3, for the transition from a saccade (black) to a fixation (grey). In the lower right corner the thresholds, β for the change in sample-to-sample direction and, δ, for the deviation from the main direction, are shown.

(a) Deviation from the main direction. Although a saccade often is slightly curved, an inherent physical property of saccades is that they do not deviate much from their main direction. In order to use this ballistic behavior to find the on- and offsets of the saccade, the sample-to-sample direction, $\alpha(n)$, is employed. The sample-to-sample direction, $\alpha(n)$, is the angle to the *x*-axis of the vector between consecutive samples and is defined as

$$\alpha(n) = \arctan\left(\frac{d_y(n)}{d_x(n)}\right) \tag{1}$$

where $d_x(n)$ and $d_y(n)$ are the x- and y- components of the sample-to-sample velocity, respectively, i.e., $d_x(n) = x(n+1) - x(n)$ and $d_y(n) = y(n+1) - y(n)$. An illustration of the calculation of $\alpha(n)$ is shown in Fig. 5. The main direction, γ, is calculated as the average sample-to-sample direction of three consecutive samples centered around sample k_i, i.e., $\gamma = \frac{1}{3}(\alpha(k_i - 1) + \alpha(k_i) + \alpha(k_i + 1))$. In each direction, starting from sample k_i, the saccade begins/ends if $|\alpha(n) - \gamma| > \delta$ for $K = t_K \cdot F_s$ consecutive samples, where t_K is the threshold for the maximum duration of deviation from the main direction. Of the K samples exceeding the threshold, the onset/offset is set to the sample closest to k_i, see Fig. 6 for an example.

(b) Inconsistent sample-to-sample direction. Due to the ballistic behavior of a saccade, it cannot abruptly change its direction from one sample to the next.

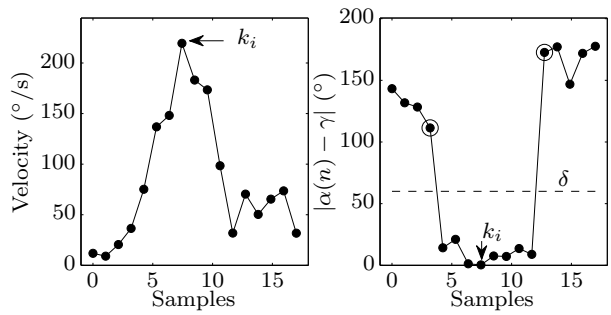


Figure 6: An example of the principle for the on- and offset detection for criteria *a)*. Left: the velocity where k_i is marked. Right: deviation from the main direction, where the onset and offset are marked with (o), which in each direction is the sample closest to the k_i of the consecutive samples that have exceed the threshold δ.

The change in sample-to-sample direction, $\epsilon(n)$, is calculated between consecutive samples in each approximate saccadic interval, and is defined as $\epsilon(n) = \alpha(n) - \alpha(n-1)$, see Fig. 5. If the change in direction is larger than β,

$$|\epsilon(n)| > \beta \tag{2}$$

for $N = t_N \cdot F_s$ consecutive samples, where t_N is the threshold for the maximum duration of inconsistent sample-to-sample direction. The onset/offset is set to the sample most distant from k_i of the N samples that exceed the threshold.

(c) Distance between directional changes. The distance between significant directional changes is measured as the Euclidean distance between samples satisfying $|\epsilon(n)| > \beta$. Thus, the number of such distances is lower than the number of samples. This third criterion exploits the fact that these distances are decaying when moving away from the center of the saccade and into the fixation. The onset and offset are reached when the eye is moving shorter distances before changing direction compared to the corresponding average distance in the intersaccadic intervals, ν. In detail, when M consecutive distances are shorter than ν, the onset/offset is set to the position of the outermost such distance counted from sample k_i. The average distance between directional changes in the intersaccadic intervals, ν, is individually set for each recording and is calculated in a similar way as the distance between directional changes in the approximate saccadic intervals. However, the intersaccadic intervals may not only contain fixations, but also smooth pursuit movements. Therefore, a piecewise linear model of the signal is subtracted before the calculation

Table 1: An overview of the properties and criteria for detection of saccades and PSO.

Eye movement	**Physiological property**	**Mathematical property**	**Criteria in the algorithm**
saccade	fast movement	acceleration	acceleration threshold
		velocity	velocity threshold
saccade	ballistic	uniform direction	onset/offset criteria:
			(a) deviation from the main direction
			(b) inconsistent sample-to-sample direction
			(c) distance between directional changes
PSO	instability	decreasing	pole model and placement
		oscillation	of the poles

of the average distance between directional changes. A block length of 100 ms was deemed sufficient for representation of the smooth pursuit movements while not interfering significantly to the calculation of ν. The residual positions $x'(n)$ and $y'(n)$ are used to calculate the residual direction $\alpha'(n)$ as

$$\alpha'(n) = \arctan\left(\frac{d'_y(n)}{d'_x(n)}\right) \tag{3}$$

where $d'_x(n)$ and $d'_y(n)$ are the x- and y- components of the residual velocity, respectively, i.e., $d'_x(n) = x'(n+1) - x'(n)$ and $d'_y(n) = y'(n+1) - y'(n)$. Next, the positions where the change in residual direction $\epsilon'(n)$, is larger than β is found,

$$|\epsilon'(n)| > \beta \tag{4}$$

where $\epsilon'(n) = \alpha'(n) - \alpha'(n-1)$, and the Euclidean distances between these positions are calculated. In order to robustly estimate ν, the value of the 90th percentile of these Euclidean distances is chosen.

In addition to criteria (b) and (c), the sample-to-sample velocity needs to be lower than 20% of the peak velocity for the current saccade to begin/end. The physiological motivations for the criteria used for saccade delineation are summarized in Table 1.

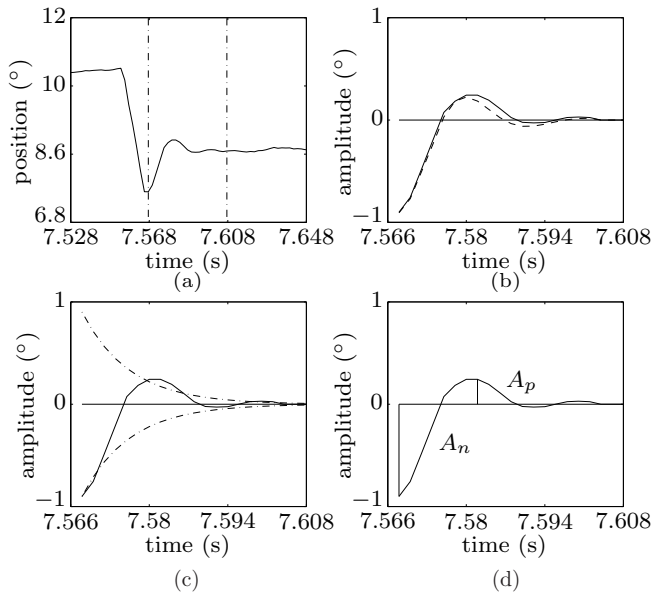


Figure 7: An example of the principle for detection of PSO. In (a) the interval used for detection of PSO in the $x-$coordinate is shown. In (b) the interval is zoomed in and the signal $g(n)$ (solid) and its corresponding impulse response $\hat{g}_p(n)$ (dashed), with normalized RMSE = 0.07, are shown. In (c) the signal and its corresponding decaying component $f(n)$ and $-f(n)$ (dashed-dotted), are shown. In (d) the amplitude A_p and A_n are shown together with $g(n)$.

2.3 PSO detection

In the third part of the algorithm, the PSO are detected. Postsaccadic oscillations can physiologically be described as instabilities or oscillatory movements that may occur at the end of a saccade. The durations of PSO are in previous research shown to be between 10 to 35 ms [9]. In order to mathematically describe this property, an all-pole model is employed. The model, $g(n)$, $n = 0, 1,..., L-1$, where $L = t_L \cdot 10^{-3} F_s$ and F_s is the sampling frequency of the signal, is applied directly after the saccade offset. The x- and y- components of the interval are modeled separately, see Fig. 7a for an example of such interval. In order for the all-pole model to be meaningful, it is required that the signal is decaying. Therefore, in order to not include the beginning of a possible smooth pursuit movement in the signal to be modeled, a variable interval length L, corresponding to t_L ms, is considered. Initially t_L is set to 40 ms. If the oscillation is not ended before $t_L = 40$ ms, determined by different signs of the slopes fitted to the samples within 8 ms before and after the interval end, the initial interval length is extended to t_L = 60 ms.

Starting from t_L, the interval is shortened in order to only include the entirety of the PSO by performing the following steps: A first order polynomial is fitted from the end of $g(n)$ and to the left. In detail, one polynomial is fitted to $g(L-3)$ and $g(L-1)$; the slope of the polynomial is denoted the reference slope. Another first order polynomial is fitted to $g(L-4)$ and $g(L-3)$; this slope is denoted the test slope. The test slope is compared to the reference slope, and if the difference between the two slopes is less than θ the reference slope is extended one sample to the left, and the test slope is moved one sample to the left. The slopes are compared until the difference between them is larger than θ, indicating the beginning of a linear behavior in the end of $g(n)$. The entire reference slope interval is replaced by a constant level, corresponding to the level in the beginning of the replaced interval. In addition, in order for $g(n)$ to end at zero, $g(L-1)$ is subtracted from $g(n)$, as illustrated in Fig. 7b. The following all-pole system function is employed,

$$G_p(z) = \frac{b_0}{1 + \sum_{k=1}^{p} a_p(k) z^{-p}} \tag{5}$$

where p is the order of the model, $a_p(k)$ are the coefficients of the model, and b_0 is a scaling factor. In order to estimate the coefficients $a_p(k)$ and the scaling factor b_0, Prony's method is used [22]. The correlation function $r_g(k)$ of $g(n)$ is calculated as:

$$r_g(k) = \sum_{n=0}^{L-1} g(n) g(n-k) \tag{6}$$

By solving the following normal equations, the coefficients $a_p(k)$, are calculated:

$$\sum_{l=1}^{p} a_p(l) r_g(k-l) = -r_g(k), k = 1, ..., p \tag{7}$$

In order to determine which order p of the model that best fits $g(n)$, the coefficients of the model are calculated for $p = 1, 2, 3, 4$. For each p, the root mean square error (RMSE) between $g(n)$ and the impulse response of the model, $\hat{g}_p(n)$, is calculated and normalized with the maximum absolute amplitude of the signal in the interval. The normalized RMSE for each order p is compared to the normalized RMSE for $p = 1$. If a higher order improves the normalized RMSE for $p = 1$ by 5% or more and the normalized RMSE is lower than 0.15, that order is used. If the normalized RMSE is larger than or equal to 0.15, the signal is shifted in time and the start of the modeling interval is iteratively moved one sample forward until the normalized RMSE is lower than 0.15. If none of the normalized RMSE satisfies this requirement, the model with the lowest normalized RMSE is employed. The poles of the polynomial for the selected order are calculated and for each pole the distance to the origin, r, indicates how quickly the signal decays to zero. A value of

r close to zero indicates a fast decay while a value close to one indicates a slow decay. In order to summarize the distances from the poles to origin for models with $p > 1$, the maximum value, r_{max}, is used. In order for the signal to be identified as PSO, $\hat{g}_p(n)$ must have $r_{max} < r_{th}$ and a maximum absolute amplitude $A > A_{min}$. The offsets of the PSO are determined by using the function $f(n) = Ar_{max}^n$, which describes the decaying component of the modeled signal, see Fig. 7c. The difference between $\hat{g}_p(n)$ and $f(n)$ is calculated as

$$u(n) = \begin{cases} \hat{g}_p(n) - f(n), & \text{if } \sum \hat{g}_p(n) < 0 \\ \hat{g}_p(n) - (-f(n)), & \text{if } \sum \hat{g}_p(n) > 0 \end{cases}$$

The offsets of the PSO are defined as the first sample where $u(n) < \xi$ during $R = t_R \cdot F_s$ consecutive samples. In order to not detect very slow movements, the following ratio between the amplitude and the duration of the PSO are calculated,

$$s = \frac{A_n + A_p}{t_g} \tag{8}$$

where A_n is the maximum absolute amplitude of the negative part of $g(n)$, A_p is the maximum amplitude of the positive part of $g(n)$, and t_g is the duration of the detected PSO, see Fig. 7d. All PSO that have $s < \frac{0.4 \cdot 10^3}{2F_s}$ are discarded. If PSO are detected in both the x- and y- components, the offset is set to the index that corresponds to the latest offset.

2.4 Performance evaluation

In order to evaluate the performance of the algorithm in terms of sensitivity and specificity, (cf. [23]), a manually annotated database is used as a reference. The sensitivity describes the algorithm's ability to correctly classify each type of eye movement and a value close to one is desired. For each type of eye movement i, where i = {S = Saccade, PSO = Postsaccadic oscillations, D = Disturbances, SF = Smooth pursuit/Fixation}, the sensitivity$_i$ is calculated as

$$\text{sensitivity}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{9}$$

where *true positives*, TP$_i$, is the number of correctly classified samples for eye movement type i, and the *false negatives*, FN$_i$, is the number of samples that should have been classified as eye movement type i, but have incorrectly been classified as another type of eye movement.

The specificity$_i$ describes the algorithm's ability to only find the samples of eye movement type i and a value close to one is desired. For each type of eye movement

i, the specificity$_i$ was calculated as

$$\text{specificity}_i = \frac{\text{TN}_i}{\text{TN}_i + \text{FP}_i} \tag{10}$$

where *true negatives*, TN_i, is the number of samples that the algorithm correctly classified as another type of eye movement than i. The *false positives*, FP_i, is the number of samples that the algorithm falsely classified as eye movement type i.

The comparison between the manual annotation and the detections of the algorithm is summarized by Cohen's kappa [24], κ,

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \kappa \in [0, 1] \tag{11}$$

where P_o is the observed proportion of agreement between the output of the algorithm and the manual annotation, and P_e is the proportion of agreement expected by chance between the output of the algorithm and the manual annotation. For a detailed description on the calculation of P_o and P_e, see [25]. A value of κ close to one is desired, and indicates that there is an overall good agreement between the algorithm and the manual annotation.

The proposed algorithm is also compared to the adaptive velocity based algorithm described in [9]. This algorithm was chosen because it is one of few algorithms that is able to detect PSO and is freely available. In addition, it outperformed two of the most commonly used algorithms today: the identification by velocity threshold (I-VT) algorithm and the identification by dispersion threshold (I-DT) algorithm [9]. In order to be able to compare different algorithms which may detect different numbers and lengths of each type of eye movement, the entire performance evaluation is based on the classification of each sample.

3 Experiment and database

The eye-tracking signals used in this paper were collected in an experiment where 33 participants, students and personnel from Lund University, took part. The mean age of the participants was 31.2 $\pm$ 9.9 (M $\pm$ SD) years. In the experiment, two computers were used, one for showing the stimuli and one for controlling the eye-tracker. Stimuli were presented using Matlab R2009b and Psychophysics toolbox (version 3.0.8, Revision 1591), on a Samsung Syncmaster 931c TFT LCD 19 inch (380x300 mm) monitor, with a screen refresh rate of 60 Hz and a resolution of 1024x768 pixels. The computer controlling the eye-tracker was running iView X (version 2.4.19). The signals were recorded binocularly with the iView X Hi-Speed 1250 eye-tracker from SensoMotoric Instruments (Berlin, Germany), at a sampling frequency of 500 Hz. The viewing distance from the eye-tracker to the screen was

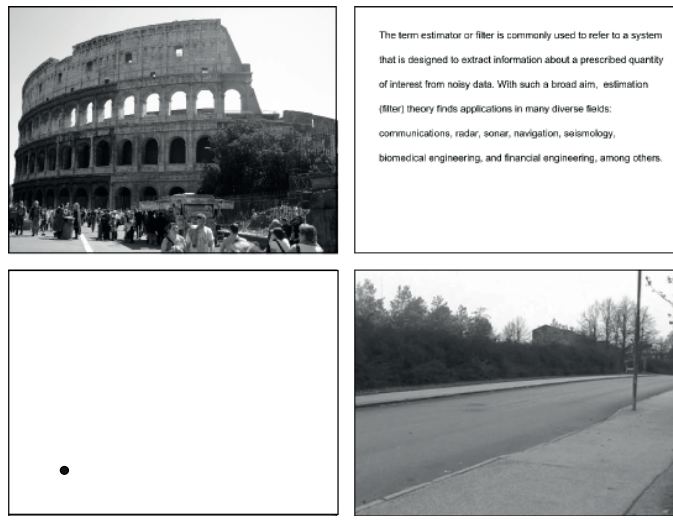


Figure 8: An example of the different types of stimuli used in the experiment. Upper left: Image, upper right: Text, lower left: moving dot and lower right: a frame of a video clip.

670 mm. At the start of the experiment, a calibration procedure was performed for each participant. The calibration procedure contained a nine-target binocular calibration in iViewX followed by four targets used to validate the accuracy of the calibration. The average accuracy across participants and validation targets was 0.41° and 0.41° for the x- and y- components, respectively.

The experiment contained five blocks with different types of stimuli: images, texts, moving dots, short video clips, and a scrolling text. Examples of the stimuli are shown in Fig. 8, and a summary of the content in the different blocks is shown in Table 2. Each block contained a number of trials with the same type of stimuli. Both the block order and the internal trial order within the block were randomized for each participant. Before the next trial started, the participants were instructed to fixate at a centrally located cross. Before a new block started, the participants were given detailed written instructions on the screen about the next task. These instructions were: to freely view an image for 10 s, to read a text at your own speed, and to follow moving dots and moving objects for video clips.

The database used in this paper is a subset of the complete database recorded during the described experiment. From now on, in this paper, the database refers to the signals recorded from participants viewing images, moving dots, and video clips. These trials are marked with bold font in Table 2. Of the 33 participants that were included in the experiment, 31 were used; two participants were excluded

Table 2: A summary of the experiment. In this paper the stimuli marked in bold font are used.

Blocks	**Content**	**Annotated/Total**
Image	**photographs with nature motives**	14/155
Text	texts	-
Moving dot	**one black dot on white background** in 8 directions (0, $\pm\frac{\pi}{4}$,$\pm\boldsymbol{\frac{\pi}{2}}$,$\pm\frac{3\pi}{4}$,$\pi$) 4 speeds (**5**, 10, **20**, 30°/s)	11/62
	sinusoid	-
	blinking static dots	-
Video clip	**real-world videos with moving objects,** e.g, road with traffic and a roller coaster	9/186
Scrolling text	vertical scrolling text	-

due to saving problems during the recording process. Only signals recorded from the right eye were used in the analysis. The database was divided into two parts by splitting the participants into two equally large groups; one part for development and one part for testing of the algorithm.

A subset of the signals from both the development part and the test part of the database was annotated manually by a domain expert (author MN). In order to facilitate the annotation process, a GUI was developed in Matlab, showing the x- and y- coordinates over time, the velocity over time, the vertical diameter of the pupil over time, the coordinates in the xy- space, and a zoomed in version of the last manually annotated event in the xy- space. These representations were judged sufficient for the expert to reliably detect the eye movements. The expert classified each sample into six different types of events: fixations, saccades, PSO, smooth pursuit movements, blinks, and undefined events. An undefined event is when a sample does not conform to any of the other eye movements. The manual annotation was performed without knowing beforehand which type of stimulus that was used. In order to ensure that a representative set of signals were used both for development and for evaluation of the algorithm, the manual annotation set was chosen with respect to the quality of the signals; half of the selected signals for each type of stimuli had a lower quality and the other half had a higher quality. In order to determine the quality of the signals, the percentage of data loss and the average distance between directional changes in the intersaccadic intervals, ν, were computed. The term data loss contained both the amount of blinks and the amount of samples with unreasonable high velocities and accelerations. The blink detection

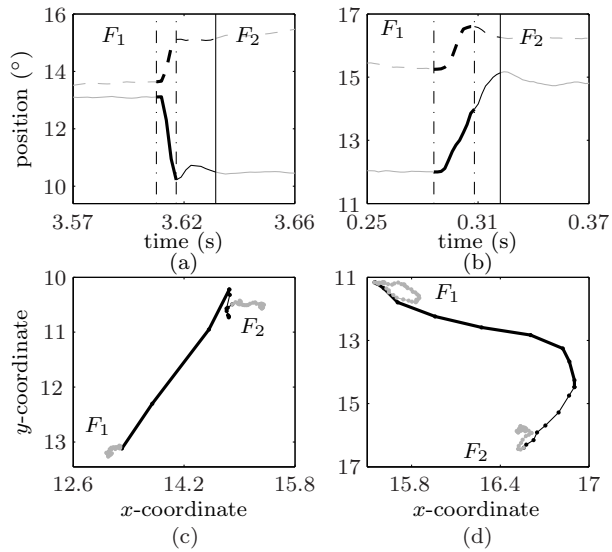


Figure 9: Two examples of the detection of saccades and PSO, where (a) – (b) are showing the x- and y- components over time and (c) – (d) are showing the xy-domain for the two examples. The thicker black line marks the saccades, the thiner black line the PSO, the vertical dash-dotted line the on- and offsets of the saccades, the solid vertical black line the offsets of the PSO and the grey line corresponds to the fixations, F_1 and F_2, before and after the saccade in each case.

described in the preprocessing part was used. A signal was judged to have a high quality when both the amount of data loss and ν were lower than respective median of the two measures and the opposite was true for signals exhibiting lower qualities.

4 Results

The settings of all intrinsic algorithm parameters, given in Table 3, were used in the entire results section. All intrinsic parameters were adjusted using only the development part of the database. Two examples of the detection of saccades and PSO for two different types of PSO recorded during image stimuli are shown in Fig. 9, in both the time and the xy- domain.

Table 3: The settings for the intrinsic parameters in the proposed algorithm for all types of stimuli. Note that the $^\circ$ notation in the cases of δ and β refers to an angle in the image plane on the stimulus screen while it in all other cases represent degrees of visual angle.

Parameter	**Value**	**Description**
Preprocessing		
a_{min}	0.3°	Min. amplitude for a one-sample spike
Saccade detection		
t_{min}	20 ms	Min. time between two saccades
T	6 ms	Min. duration of a saccade
λ	6	No. standard deviations for η_x and η_y
δ	60°	Max. allowable deviation from the main direction
t_K	6 ms	Max. duration of deviation from the main direction
β	40°	Largest allowable change in intra-saccadic direction
t_N	8 ms	Max. duration of inconsistent sample-to-sample direction
M	2	No. distances below ν
θ	1.7	Min. difference between ref. slope and test slope
PSO detection		
r_{th}	$0.89^{500/F_s}$	Max. distance from origin to a pole
t_L	40 ms	Initial length of the interval for PSO modeling
A_{min}	0.2°	Min. amplitude for PSO
ξ	0.08°	Min. value of the difference between the decaying component and the modeled signal
t_R	6 ms	Max. duration of ξ

4.1 Evaluation of the algorithm

In order to evaluate the performance of the proposed algorithm, the eye movements detected by the algorithm are compared to the manually annotated eye movements and to those detected by the velocity based adaptive algorithm described in [9]. Six properties of the detected saccades and PSO are presented in Tables 4 – 6, for the annotated part of the database. For comparison, the corresponding values for the entire development part of the database are shown in brackets. The durations of the saccades detected by the proposed algorithm are in agreement with the durations of those detected by the expert, while the saccades detected by the algorithm in [9] are in general longer, 40 – 50 ms compared to 23 – 29 ms for the expert. The issue with longer durations of the detected saccades is mentioned in [9]. The durations of the PSO detected by the proposed algorithm are in general slightly longer than those detected by the expert and the durations of the PSO detected by the algorithm in [9] are longer than those detected by both the expert and the proposed algorithm. Around 85% of all saccades detected by the expert have PSO for image and video stimuli. For moving dot stimuli, the expert detects 65% saccades with PSO. The proposed algorithm detects close to equally many PSO as the expert. The algorithm in [9] detects a lower number of saccades and in addition also fewer PSO for all types of stimuli compared to the expert.

The sensitivities and specificities for the detection of saccades, PSO, disturbances, and periods of smooth pursuit movements and/or fixations are shown in Table 7. Disturbances include everything that is not detected as eye movements, i.e., blinks, screen outliers, and one-sample spikes. The eye movements included in the category of smooth pursuit movements and/or fixations correspond to samples that the algorithm does not count as saccades, PSO or disturbances. The performance of the saccade detection for the two algorithms are equally good with specificity above 0.93 and sensitivity above 0.80, for all stimuli. However, it should be noted in Ta-

Table 4: Mean values for the properties of the detected saccades and PSO for the Expert, the proposed algorithm, and the algorithm described in [9], for images. In brackets, the corresponding value for the entire development part of the database is shown.

Measure	Expert	Prop. Alg.		Alg. in [9]	
Saccade duration (ms)	28.7	28.1	(29)	50	(49.1)
Saccade peak velocity (°/s)	404	394	(316)	383	(327)
PSO duration (ms)	20.4	24.6	(25.8)	25.8	(23.7)
% of saccades with PSO	84.5	83.9	(77.4)	69.2	(62.7)
Number detected saccades	283	286	(2310)	266	(2073)
Number detected PSO	239	240	(1789)	184	(1300)

Table 5: Mean values for the properties of the detected saccades and PSO for the Expert, the proposed algorithm, and the algorithm described in [9], for video stimuli. In brackets, the corresponding value for the entire development part of the database is shown.

Measure	Expert	Prop. Alg.		Alg. in [9]	
Saccade duration (ms)	23.9	26.6	(26.4)	40.2	(43.1)
Saccade peak velocity (°/s)	335	316	(281)	321	(299)
PSO duration (ms)	21.2	23.4	(24.9)	24	(23.7)
% of saccades with PSO	85.7	78.7	(74.2)	82.9	(61.1)
Number detected saccades	84	89	(7231)	76	(6016)
Number detected PSO	72	70	(5362)	63	(3675)

Table 6: Mean values for the properties of the detected saccades and PSO for the Expert, the proposed algorithm, and the algorithm described in [9], for moving dot stimuli. In brackets, the corresponding value for the entire development part of the database is shown.

Measure	Expert	Prop. Alg.		Alg. in [9]	
Saccade duration (ms)	23.5	25	(28.2)	43.4	(43.5)
Saccade peak velocity (°/s)	265	163	(200)	174	(206)
PSO duration (ms)	14.7	20.6	(23.8)	24	(22.8)
% of saccades with PSO	62.1	54.5	(66)	41.7	(39.1)
Number detected saccades	29	33	(106)	24	(87)
Number detected PSO	18	18	(70)	10	(34)

bles 4 – 6, that the durations of the saccades differ between the two algorithms and the similarities in sensitivities and specificities are due to that the proposed algorithm detects a larger and more correct number of saccades with shorter durations in contrast to the algorithm in [9] that detects a lower number of saccades with longer durations.

For the detection of PSO, there is a larger difference between the two compared algorithms, where the proposed algorithm outperforms the algorithm in [9]. The values of the sensitivity are 0.73 – 0.76 for the proposed algorithm compared to 0.14 – 0.37 for the algorithm in [9]. Both algorithms have equally high specificities, with values in the range of 0.96 – 0.99. The lower level of the sensitivity for the two algorithms indicates that there are too few samples that are detected as PSO compared to the annotation. Since the two algorithms have different strategies for detection of disturbances, the sensitivities and specificities for the two algorithms differ. The proposed algorithm detects the disturbances with high specificity, 0.99, and a slightly lower sensitivity, 0.67 – 0.89, than the algorithm in [9], with speci-

Table 7: Sensitivity and specificity for the proposed algorithm (Prop.) and the algorithm in [9].

	Image		**Video**		**Moving dot**	
	Prop.	Alg. [9]	Prop.	Alg. [9]	Prop.	Alg. [9]
Sensitivity_S	0.838	0.907	0.893	0.89	0.815	0.801
Specificity_S	0.984	0.926	0.983	0.963	0.982	0.968
Sensitivity_{PSO}	0.758	0.244	0.753	0.369	0.727	0.144
Specificity_{PSO}	0.973	0.957	0.986	0.973	0.989	0.987
Sensitivity_D	0.882	0.965	0.892	1	0.667	1
Specificity_D	0.992	0.846	0.999	0.809	0.999	0.665
Sensitivity_{SF}	0.955	0.739	0.973	0.751	0.978	0.614
Specificity_{SF}	0.885	0.978	0.895	0.971	0.828	0.949

Table 8: Cohens kappa for the proposed algorithm and the algorithm in [9] for the development part of the database.

	Image	**Video**	**Moving dot**
Proposed algorithm	0.814	0.822	0.756
Algorithm in [9]	0.512	0.398	0.232

Table 9: Cohens kappa for the proposed algorithm and the algorithm in [9] for the test part of the database.

	Image	**Video**	**Moving dot**
Proposed algorithm	0.745	0.804	0.736
Algorithm in [9]	0.484	0.336	0.288

ficity 0.67 – 0.85 and sensitivity 0.97 – 1. The algorithm in [9] puts in general more uncertain samples in the disturbance category. These different strategies for detection of disturbances also affect which samples that become marked as smooth pursuit movements and/or fixations.

In order to summarize the general performance of the two detection algorithms for all types of eye movements, Cohen's kappa, κ, which measures the inter-observer agreement, is computed as described in Section 2.4. The κ for the two algorithms summarized for all types of stimuli are shown in Table 8. In general, κ for the proposed algorithm is considerably larger than κ for the algorithm in [9]. In order to validate the results from the development part of the database, κ was also computed for the test part of the database, see Table 9. The results for the test part of the

Table 10: Percentage (%) of use for each criterion in the saccade on- and offset detection.

	Image	**Video**	**Moving dot**
Saccade onset			
(a) Deviation from the main direction	75.3	76.3	69.9
(b) Inconsistent sample-to-sample direction	21.3	19	24.3
(c) Distance between directional changes	3.25	4.78	5.65
Saccade offset			
(a) Deviation from the main direction	63.4	67.2	66
(b) Inconsistent sample-to-sample direction	23.8	16.7	15.1
(c) Distance between directional changes	12.8	16.2	18.9

database are comparable with the results for the development part of the database.

4.2 Evaluation of parameter settings

The parameter settings shown in Table 3 are chosen according to known physiological limitations of eye movements, visual inspection of detection results using the development part of the database, and previous literature. Table 10 shows how often the criteria *(a)* – *(c)* are employed in the detection of the saccadic on- and offsets. As shown in Table 10, all the suggested criteria are used for the detection of both on- and offsets of the saccades. The criterion deviation from the main direction is the most commonly used criterion, (63 – 76%), for the detection of both on- and offsets. The least used criterion is the distance between directional changes.

5 Discussion

An algorithm for detection of saccades and PSO in eye-tracking signals was proposed. The proposed algorithm has been tested on signals recorded during both static and dynamic stimuli, where the latter contained smooth pursuit movements. Its performance was evaluated in comparison to manually annotated eye movements and an adaptive velocity based algorithm, described in [9].

The performance of the saccade detection was in terms of sensitivity and specificity generally similar to the algorithm in [9]. However, the durations and the number of saccades differed between the two algorithms. The lower number of saccades using the algorithm in [9] can be explained by the presence of smooth pursuit movements in the signal, which increases the velocity threshold such that small saccades are missed. The reason for the longer durations for the algorithm in [9] is that

it solely uses a velocity threshold, and searches for a local minima in the velocity before and after the peak, while the proposed algorithm uses a combination of several criteria, for the detection of the on- and offsets of the saccades.

The appearance of PSO is highly dependent on the type of eye-tracker that is used [16], and PSO have therefore been treated unsystematically, or not at all by most algorithms. Explicitly detecting and modeling of PSO, as is described in this work, leaves the user with several options: The PSO can be: 1) included in the saccades, 2) included in the subsequent eye movements, 3) classified as its own type of eye movement, 4) discarded because of its unknown perceptual and cognitive consequences, or 5) substituted by a simplified first order all-pole model in order to suppress the PSO. Different options may be suitable for different types of research and are all supported by the proposed method.

There was a large difference in the performance of the detection of PSO between the two compared algorithms. The two algorithms use two completely different strategies for the detection of the PSO, where the proposed algorithm uses an all-pole model, while the algorithm in [9] uses an adaptive velocity threshold. By using a model, the proposed algorithm was, in addition to perform more accurate offsets detection of the PSO, also able to identify different types of PSO that was not possible when only using the velocity signal. According to the properties of the detected PSO in Tables 4 – 6, there appears to be no difference in duration between PSO measured during image and video stimuli in this database.

The quality of the signals used for evaluation of the proposed algorithm has been measured. Signals with both higher and lower quality have been used both for the development and testing of the algorithm. It has, however, not been evaluated if there is a difference in performance for signals with different level of quality.

The algorithm was also tested for the same database downsampled to 250 Hz with a similar/slightly degraded performance (Cohen's kappa = 0.78, 0.76 and 0.70 for image, video, and moving dot stimuli) and for text reading data (ten cases of in total 395 s) sampled at 1250 Hz where the performance was in good agreement with the performance when running the same data downsampled to 500 Hz (Cohen's kappa = 0.80).

The proposed algorithm is intended for offline use. Since the velocity filter is not causal and the proposed algorithm performs segmentation of the signals before the final classification, it is not suitable for real-time applications.

The eye movements detected by the proposed algorithm is compared to manual annotation performed by an expert. By using the annotated eye movements, the performance of the algorithm was quantitatively evaluated. Since the annotation is performed on a sample to sample basis, the exact performance of the detection algorithm can be evaluated in contrast to previously used evaluation methods where, e.g., statistics of amplitudes, durations or number of detected events have been calculated and evaluated [12, 9]. By using a combination of the measures and values

in Tables 4 – 8, the performance of the algorithm can be evaluated at different scales, from the exact on- and offsets of individual events to Cohen's kappa which summarizes all sensitivities and specificities over the entire annotated part of the database.

6 Conclusions

In this work, an algorithm for event detection and eye movement classification based on eye-tracking signals is proposed. In summary, the proposed algorithm provides more accurate on- and offsets estimation of the saccades, outperforms a previously suggested estimation method for PSO, and in addition allows modeling of the PSO. Furthermore, a methodological framework for objective testing of event detection algorithms applied to eye-tracking signals have been developed, including a large partly annotated, database recorded during both static and dynamic stimuli, a graphical user interface for easy annotation and viewing of eye movement events, and an evaluation procedure which summarizes the overall performance into one Cohen's kappa value.

Acknowledgment

This work was supported by the Strategic Research Project eSSENCE, funded by the Swedish Research Council. Data were recorded in the Lund University Humanities Laboratory.

References

[1] T. Crawford, S. Higham, T. Renvoize, J. Patel, M. Dale, A. Suriya, and S. Tetley, "Inhibitory control of saccadic eye movements and cognitive impairment in alzheimer's disease," *Biological Psychiatry*, vol. 57, pp. 1052–1060, 2005.

[2] J. Sweeney, B. Brew, J. Keilp, J. Sidtis, and R. Price, "Pursuit eye movement dysfunction in hiv-1 seropositive individuals," *Journal of Psychiatry & Neuroscience*, vol. 16, no. 5, pp. 247–252, 1991.

[3] K.-M. Flechtner, B. Steinacher, R. Sauer, and A. Mackert, "Smooth pursuit eye movements in schizophrenia and affective disorder," *Psychological Medicine*, vol. 27, pp. 1411–1419, 1997.

[4] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2011.

[5] R. Leigh and D. Zee, *The Neurology of Eye Movements*. Oxford University Press, 2006.

[6] C. Meyer, A. Lasker, and D. Robinson, "The upper limit of human smooth pursuit.," *Vision Research*, vol. 25, no. 4, pp. 561–563, 1985.

[7] A. T. Bahill, M. R. Clark, and L. Stark, "Glissades-eye movements generated by mismatched components of the saccadic motorneuronal control signal," *Mathematical biosciences*, vol. 26, pp. 303–318, 1975.

[8] D. Salvucci and J. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, (New York), pp. 71–78, ACM, 2000.

[9] M. Nyström and K. Holmqvist, "An adaptive algorithm for fixation, saccade, and glissade detection in eyetracking data," *Behavior Research Methods*, vol. 42, no. 1, pp. 188–204, 2010.

[10] R. Engbert and R. Kliegl, "Microsaccades uncover the orientation of covert attention," *Vision Research*, vol. 43, no. 9, pp. 1035–1045, 2003.

[11] F. Behrens, M. MacKeben, and W. Schröder-Preikschat, "An improved algorithm for automatic detection of saccades in eye movement data and for calculating saccade parameters," *Behavior Research Methods*, vol. 42, no. 3, pp. 701–708, 2010.

[12] R. van der Lans, M. Wedel, and R. Pieters, "Defining eye-fixation sequence across individuals and tasks: the binocular-individual threshold (bit) algorithm," *Behavior Research Methods*, vol. 43, no. 1, pp. 239–257, 2011.

[13] P. Mital, T. Smith, R. Hill, and J. Henderson, "Clustering of gaze during dynamic scene viewing is predicted by motion," *Cognitive computation*, vol. 3, no. 1, pp. 5–24, 2010.

[14] M. Dorr, T. Martinetz, K. R. Gegenfurtner, and E. Barth, "Variability of eye movements when viewing dynamic natural scenes," *Journal of vision*, vol. 10, no. 10, pp. 1–17, 2010.

[15] *EyeLink user manual. Version 1.3.0*, 2007.

[16] H. Deubel and B. Bridgeman, "Fourth purkinje image signals reveal eye-lens deviations and retinal image distortions during saccades," *Vision Research*, vol. 35, no. 4, pp. 529–538, 1995.

[17] M. Bettenbühl, C. Paladini, M. Holschneider, K. Mergenthaler, R. Kliegl, and R. Engbert, "Microsaccade characterization using the continuous wavelet transform and principal component analysis," *Journal of eye movement research*, vol. 3, no. 5, pp. 1–14, 2010.

[18] O. Komogortsev, D. Gobert, S. Jayarathna, D. Hyong Koh, and S. Gowda, "Standardization of automated analyses of oculomotor fixation and saccadic behaviors," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2635–2645, 2010.

[19] Z. Kapoula, D. Robinson, and T. Hain, "Motion of the eye immediately after a saccade," *Experimental Brain Research*, vol. 61, no. 2, pp. 386–394, 1986.

[20] D. Stampe, "Heuristic filtering and reliable calibration methods for video-based pupil tracking systems," *Behavior research methods instruments and computers*, vol. 25, no. 2, pp. 137–142, 1993.

[21] J. Otero-Millan, X. Troncoso, S. Macknik, I. Serrano-Pedraza, and S. Martinez-Conde, "Saccades and microsaccades during visual fixation, exploration, and search: Foundations for a common saccadic generator," *Journal of vison*, vol. 8, no. 14, pp. 1–18, 2008.

[22] M. Hayes, *Statistical digital signal processing and modeling*. Wiley, 1996.

[23] L. Sörnmo and P. Laguna, *Bioelectrical signal processing in cardiac and neurological applications*. Elsevier, 2005.

[24] C. Jean, "Assessing agreement on classification tasks: The kappa statistic.," *Computational Linguistics*, vol. 22, no. 2, pp. 249–254, 1996.

[25] K. Berry and P. Mielke, "A generalization of cohen's kappa agreement measure to interval measurement and multiple raters," *Educational and psychological measurement*, vol. 48, pp. 921–933, 1988.

Paper II

Detection of Fixations and Smooth Pursuit Movements in High-Speed Eye-Tracking Data

Abstract

A novel algorithm for the detection of fixations and smooth pursuit movements in high-speed eye-tracking data is proposed, which uses a three-stage procedure to divide the intersaccadic intervals into a sequence of fixation and smooth pursuit events. The first stage performs a preliminary segmentation while the latter two stages evaluate the characteristics of each such segment and reorganize the preliminary segments into fixations and smooth pursuit events. Five different performance measures are calculated to investigate different aspects of the algorithm's behavior. The algorithm is compared to the current state-of-the-art (I–VDT and the algorithm in Berg et al., 2009), as well as to annotations by two experts. The proposed algorithm performs considerably better (average Cohen's kappa 0.42) than the I–VDT algorithm (average Cohen's kappa 0.20) and the algorithm in Berg et al. (2009) (average Cohen's kappa 0.16), when compared to the experts' annotations.

Linnéa Larsson, Marcus Nyström, Richard Andersson, and Martin Stridh,
"Detection of Fixations and Smooth Pursuit Movements in High-Speed Eye-Tracking Data,"
in *Biomedical Signal Processing and Control*, vol. 18, pp. 145–152, April 2015.

1 Introduction

Measurement of eye movements is an important tool in basic research in, e.g., visual attention, perception, cognition, and medicine. In studies of visual attention and perception, eye movements are used to investigate, e.g., how the focus of our attention is chosen depending on the content of an image [1], how objects are identified [2], and how decisions are made [3]. In medicine, eye tracking is employed in studies investigating the functionality of the brain, e.g., in patients with schizophrenia [4].

Until recently, the majority of eye-tracking studies have used static stimuli, e.g., images and texts. The two most common types of eye movements when viewing static stimuli are *fixations* and *saccades*. Fixations are periods when the eye is more or less still, while saccades are fast movements between the fixations that take the eyes from one object of interest to the next. Currently, the interest in dynamic stimuli is growing and it is becoming increasingly common to conduct studies where video clips are used as stimuli [5]. The type of eye movement called *smooth pursuit* occurs when the eyes are following a moving object [6]. Traditionally, algorithms have been developed for signals recorded during static stimuli, i.e., developed to detect fixations and saccades. When smooth pursuit movements are not considered by an algorithm, they will be spread into the other types of detected eye movements and make the interpretation of these difficult. Smooth pursuit movements may for instance be erroneously classified as very long fixations interspersed with very short saccades [7].

Many of the measures that earlier have been used to investigate eye movements during image viewing are based on the detection of fixations and their properties, e.g., fixation duration and number of fixations [8]. When dynamic stimuli are used, these fixation measures are still of interest. However, in order to be able to investigate and draw well-founded conclusions from fixations in data where smooth pursuit movements are present, a robust algorithm for separation of fixations and smooth pursuit movements is needed.

Since the signal characteristics of fixations and smooth pursuit movements are overlapping [9], classification of fixations in the presence of smooth pursuit movements is a difficult task [5, 10]. The task is also different depending on whether the algorithm is intended for analysis of data recorded with a high or low sampling frequency, and for real-time or offline processing. Classification of data with different sampling frequencies require different event detection methods, mainly due to differences in the level of high frequency noise.

In [10], three algorithms for detection of fixations, saccades, and smooth pursuit movements were evaluated: a velocity based algorithm with two velocity thresholds (I–VVT), a velocity and movement pattern based algorithm (I–VMP), and a velocity and dispersion based algorithm (I–VDT). All algorithms were evaluated with data recorded using the EyeLink 1000 from SR Research. The stimuli consisted of

dots moving with different speeds and different directions. The results showed that the most successful method was the I–VDT, which used a combination of velocity and dispersion thresholds.

Another algorithm, proposed in [11], employed principal component analysis in combination with a velocity threshold to distinguish between saccades, fixations, and smooth pursuit movements. The algorithm was used to analyze saccades in humans and monkeys watching short video clips, but the performance of the algorithm was not evaluated in detail. In the following, the algorithm proposed in [11] is referred to as I–PCA.

A completely different method, intended for real-time detection of smooth pursuit movements using a low-speed mobile eye-tracker was proposed in [12]. The method used a set of features and a k-nearest neighbor classifier in order to distinguish between smooth pursuit movements and the remaining parts of the data. The performance of the algorithm was evaluated using data recorded with stimuli where a dot was moving over the screen in different speeds and different directions. The results showed that a combination of features that capture temporal aspects of smooth pursuit movements was a successful detection method.

In this work, the focus is on offline processing of fixations and smooth pursuit movements in data recorded using a high-speed eye-tracker. The paper consists of two main parts: First, an algorithm for classification of fixations and smooth pursuit movements is developed for eye-tracking data when dynamic stimuli are used, and secondly, a detailed evaluation is performed, where the performance of the algorithm is evaluated from different aspects.

2 Methods

A schematic overview of the proposed algorithm for detection of fixations and smooth pursuit movements is shown in Fig. 1. The algorithm is applied to the *intersaccadic intervals*, i.e., the intervals between the detected saccades, PSO, and blinks, and comprises three stages where the first stage performs a preliminary segmentation while the latter two evaluate the characteristics of each such segment and reorganize the preliminary segments into fixations and smooth pursuit events. In this paper, the intersaccadic intervals are identified using the algorithm in [13].

2.1 Preprocessing

In order to avoid any influence of adjacent saccades or PSO, the intersaccadic intervals are preprocessed. Since neither fixations nor smooth pursuit movements physiologically can have a velocity higher than 100°/s [14], the sample-to-sample velocities of the intervals are computed and all samples in the beginning and/or end of each interval exceeding this threshold are removed.

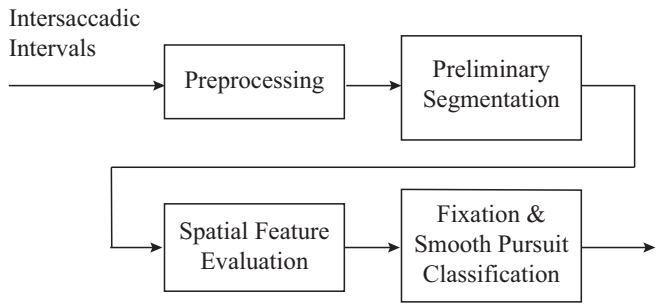


Figure 1: Overview of the proposed algorithm.

2.2 Preliminary segmentation

Each intersaccadic interval is divided into windows, w_{i}, of size t_{w} ms, with an overlap of t_{o} ms. For all pairs of x- and y-coordinates contained in the window, the sample-to-sample direction, $\alpha(n)$, is computed as the angle of the line between two consecutive pairs of x- and y-coordinates to the x-axis. In order to investigate whether the sample-to-sample directions in each window are consistent a Rayleigh test is performed [15]. The sample-to-sample direction, $\alpha(n)$, is transformed into Cartesian coordinates $r_i(n)$, for $n = 1, 2, ..., N-1$, where N is the number of samples in w_i.

$$r_i(n) = \begin{pmatrix} \sin(\alpha(n)) \\ \cos(\alpha(n)) \end{pmatrix} \tag{1}$$

The mean vector, $\bar{r}_i$, is calculated as

$$\bar{r}_i = \frac{1}{N} \sum_{n=1}^{N} r_i(n) \tag{2}$$

The Reyleigh test uses the resultant vector $R_i = \|\bar{r}_i\|$ to determine whether the sample-to-sample directions in the window are uniformly distributed or not. An approximation of the p-value under H_0 is computed using

$$P_i = \exp[\sqrt{1 + 4N + 4(N^2 - (R_i \cdot N)^2)} - (1 + 2N)] \tag{3}$$

The null and alternative hypotheses of the test, H_0 and H_{A}, respectively, are:

H_0: The samples in the window are distributed uniformly around the unit circle.

H_A: The samples in the window are not distributed uniformly around the unit circle.

The p-value of the test, P_i, is computed for each window i. Since there is an overlap between the windows, each sample may belong to more than one window. The mean value of P_j, for all windows j which sample k belongs to is computed as,

$$P_{\text{mean}}(k) = \frac{1}{K}\sum_{j=1}^{K} P_j \tag{4}$$

where K is the number of windows each sample belongs to, $k = 1, 2, ..., M$, and M is the number of samples in the intersaccadic interval. All consecutive samples in the interval satisfying either $P_{\text{mean}}(k) \geq \eta_P$ or $P_{\text{mean}}(k) < \eta_P$ are grouped together into preliminary segments sharing similar properties in terms of directionality. These preliminary segments are further analyzed in the next step.

2.3 Evaluation of spatial features in the position signal

For all preliminary segments that have a duration longer than t_{min}, four parameters, p_D, p_{CD}, p_{PD}, and p_R, are calculated. These four parameters describe the dispersion (D), the consistency in the direction (CD), the positional displacement (PD), and the range (R) of the segment, which all are parameters that are typical for a smooth pursuit movement. In order to measure the dispersion, Principle Component Analysis (PCA) is employed. The first principle component determines the direction in which the data have their largest variance and the second principle component is chosen orthogonal to the first one. The principle components, pc_1 and pc_2, are computed by removing the respective mean from the preliminary x- and y- segments and estimating the covariance matrix, $\hat{C}$, between these. The zero mean data are projected onto the principle components, d_{pc_1} and d_{pc_2} respectively, and the lengths of the corresponding vectors are calculated, [11]. An illustration of d_{pc_1} and d_{pc_2}, is shown in Fig. 2a. The first parameter, p_D, determines the relationship between the lengths of the first and the second principle components, d_{pc_1} and d_{pc_2}.

$$p_D = \frac{d_{\text{pc}_2}}{d_{\text{pc}_1}} \tag{5}$$

The parameter, p_D, measures if a preliminary segment is more dispersed in one direction than in the other, i.e., a value of p_D close to one means that the segment is equally spread in both directions.

The second parameter, p_{CD}, measures if the segment has a consistent direction or not. It is determined by computing the Euclidean distance (ED) between the

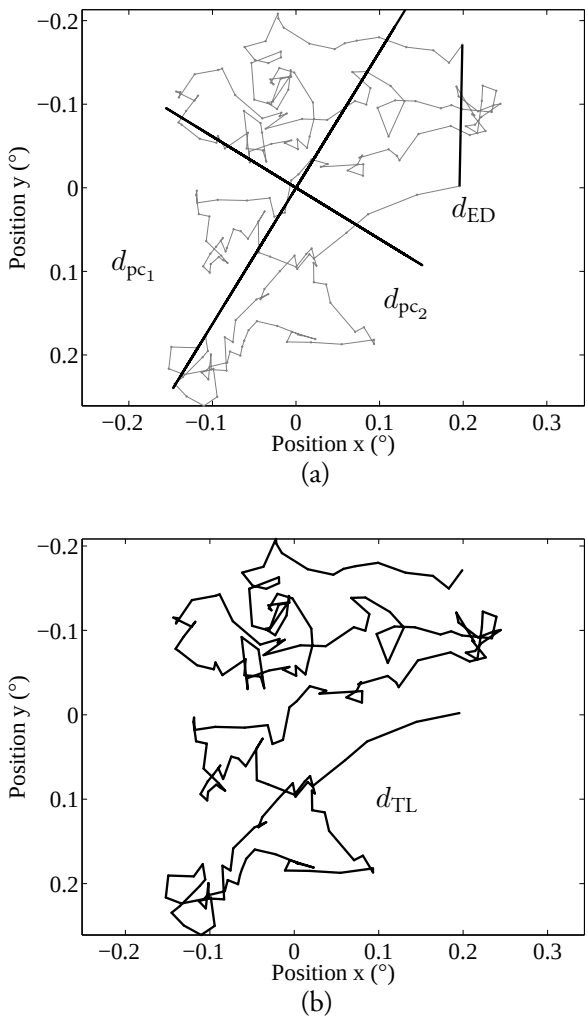


Figure 2: (a) Illustration of d_{pc_1}, d_{pc_2}, and d_{ED}, in parameters p_{D} and p_{CD}. (b) Illustration of d_{TL} which is measured in parameter p_{PD}.

starting and ending positions of the interval, d_{ED}, and comparing it to d_{pc_1}. An example of d_{ED} is shown in Fig. 2a.

$$p_{\text{CD}} = \frac{d_{\text{ED}}}{d_{\text{pc}_1}} \tag{6}$$

Hence, a value of p_{CD} close to one corresponds to that the data in the preliminary segment are starting and ending in the largest direction of the data. The third parameter, p_{PD}, measures the relationship between d_{ED} and the trajectory length (TL) of the segment, d_{TL}.

$$p_{\text{PD}} = \frac{d_{\text{ED}}}{d_{\text{TL}}} \tag{7}$$

A straight line will have p_{PD} equal to one, see Fig. 2b for an illustration of d_{TL}.

The fourth parameter, p_{R}, measures the absolute spatial range of the segment, and is computed as

$$p_{\text{R}} = \sqrt{(\max x - \min x)^2 + (\max y - \min y)^2} \tag{8}$$

where x and y are the x- and y-coordinates in the segment. The four parameters are calculated for each preliminary segment, and are compared to individual thresholds resulting in one criterion for each parameter.

1. Dispersion: $p_{\text{D}} < \eta_{\text{D}}$
2. Consistent direction: $p_{\text{CD}} > \eta_{\text{CD}}$
3. Positional displacement: $p_{\text{PD}} > \eta_{\text{PD}}$
4. Spatial range: $p_{\text{R}} > \eta_{\text{maxFix}}$

2.4 Classification of fixations and smooth pursuit movements

The segments are divided into three categories, depending on how many criteria that are satisfied. All segments where none of the criteria are satisfied are classified as fixations. Likewise, all segments with all criteria satisfied are classified as smooth pursuit movements. Finally, all segments where 1-3 criteria are satisfied are placed in a third category containing uncertain segments. The segments in this category have properties that may characterize both fixations and smooth pursuit movements. Consecutive segments belonging to the same category are grouped together.

The categorization of other segments in the same intersaccadic interval may provide information of whether the uncertain segment is part of a larger fixational interval or a larger smooth pursuit interval. The following strategy is used: First, each uncertain segment is evaluated through criterion 3, which is the criterion that

evaluates the most typical feature of a smooth pursuit movement compared to a fixation. If criterion 3 is satisfied, the uncertain segment is most similar to a smooth pursuit movement and the spatial range, p_{R}, is recalculated by adding the spatial ranges of other smooth pursuit segments in the intersaccadic interval that has a mean direction that does not differ more than ϕ to the mean direction of the uncertain segment. If the merged segment has a $p_{\mathrm{R}} > \eta_{\mathrm{minSmp}}$, the segment is classified as a smooth pursuit and otherwise as a fixation. If, on the other hand, the segment is most similar to a fixation, i.e., criterion 3 is not satisfied, criterion 4 decides whether the segment is classified as a fixation; if criterion 4 is not satisfied, the segment is classified as a fixation and vice versa.

2.5 Performance evaluation

The performance of the proposed algorithm is evaluated using the following five methods:

1. *Event properties.* The total number of fixations and smooth pursuit movements are calculated as well as the mean duration for each of the two types of events.

2. *Proportion of events for different types of stimuli.* The percentage of each type of event is calculated for image-, and moving dot stimuli. The expected result for image stimuli is to have close to 100% detected fixations and close to 0% detected smooth pursuit movements. For moving dot stimuli the expected result is to have an as large amount of detected smooth pursuit movements as possible.

3. *Sensitivity and specificity analysis.* The sensitivity describes the ability of the algorithm to detect a certain type of event. The specificity is a complementary measure that describes the ability of the algorithm to correctly detect each type of event (c.f. [16]). When calculating the sensitivity and specificity, manual annotations are used as the "gold standard". The annotations of each intersaccadic interval are compared to the detections of the algorithm. Since the on- and offsets of the saccades may differ between the algorithm and the annotations, the data is classified into four groups: Fixations (Fix), Smooth pursuit movements (Smp), Disturbances (Dist), and Others, where Disturbances includes all samples that are detected as blinks or removed outliers and Others contains samples from adjacent saccades and PSO.

4. *Cohen's kappa analysis.* In order to evaluate the overall agreement between the manual annotations and the detections of the algorithm, Cohen's kappa is used. A detailed description of the calculations of sensitivity, specificity, and Cohen's kappa can be found in [13].

5. *Scores evaluation.* In [17] and later in [10], scores were proposed as an evaluation method for saccades, fixations, and smooth pursuit movements. Since the work in this paper focuses on the separation between fixations and smooth pursuit movements, only a set of the proposed scores in [10] are computed. The following scores are used:

   - **PQnS** – The ratio between the sum of the durations of all the detected smooth pursuit movements and the sum of the durations of moving dots in the stimuli. PQnS is compared to its corresponding ideal value, $\mathrm{PQnS}_{\mathrm{ideal}}$, which is calculated as the total duration of moving dots in the stimuli where the duration of the first fixation and the duration of the first corrective saccade are removed.
   - **PQlS$_{\mathrm{P}}$** – Determines the mean distance between the moving dot stimuli and the samples detected as smooth pursuit movements. $\mathrm{PQlS}_{\mathrm{P}}$ is compared to its ideal value which is $0°$.
   - **PQlS$_{\mathrm{V}}$** – Determines the mean difference between the velocities of the detected smooth pursuit and of the corresponding stimuli. $\mathrm{PQlS}_{\mathrm{V}}$ is compared to its ideal value which is $0°/\mathrm{s}$.

   A detailed description and background to all scores can be found in [17, 10].

3 Experiment and database

The eye-tracking signals used in this paper were collected during an experiment described in [13], where a Hi-Speed 1250 eye-tracker from SensoMotoric Instruments (Teltow, Germany) was used. The eye-tracking signals were recorded binocularly, with a sampling frequency of 500 Hz. In this paper, the signals from the right eye were used. The experiment was designed specifically for the evaluation of event detection algorithms when smooth pursuit movements are present. The experiment includes static images and short video clips as well as dots moving in different directions and speeds. The database was split into two parts: one development database and one test database. A subset of each database was manually annotated by two experts. In total for all stimuli, 33 trials were annotated by Expert 1 and 58 trials by Expert 2.

4 Results

All results presented in this section were generated using the settings shown in Table 1, which were chosen to maximize both the sensitivity and specificity of the algorithm with respect to the manually annotated development database. The detected

Table 1: Parameter settings for the proposed algorithm.

Parameter	Value	Description
t_{w}	22 ms	window size
t_{o}	6 ms	overlap of the windows
η_{P}	0.01	significance level for the Rayleigh test
η_{D}	0.45	threshold for p_{D}
η_{CD}	0.5	threshold for p_{CD}
η_{PD}	0.2	threshold for p_{PD}
η_{maxFix}	1.9°	threshold for max spatial range for a fixation
η_{minSmp}	1.7°	threshold for min spatial range for a smooth pursuit movement
ϕ	$\frac{\pi}{4}$	max difference in mean direction for a smooth pursuit movement
t_{min}	40 ms	minimum fixation duration

fixations and smooth pursuit movements are compared to those detected by the I–VDT algorithm proposed in [10] and the I–PCA algorithm proposed in [11]. The I–VDT algorithm is used with the parameter settings proposed in [10], i.e., using a velocity threshold $\mathrm{T_V} = 75^\circ/\mathrm{s}$, a temporal window $\mathrm{T_W} = 150$ ms, and a dispersion threshold $\mathrm{T_D} = 1.9^\circ$. The I–PCA algorithm, which is part of the iLab C++ Neuromorphic Vision Toolkit, was downloaded from http://iLab.usc.edu/toolkit and used with default settings. The preprocessing, where disturbances and blinks are removed, is the same for the three algorithms, see [13] for a description.

4.1 Event properties

The average properties of the detected fixations and smooth pursuit movements are shown in Tables 2 – 4, for images, video, and moving dot stimuli, respectively. The results are summarized below:

- **Images** – For the development database the mean fixation durations are similar between the two experts and the three algorithms, with values ranging from 217 ms to 241 ms. The mean durations for the detected smooth pursuit movements are, however, less similar across the algorithms and the experts. In general, I–VDT detects the most and the shortest smooth pursuit movements with a mean duration of 48.7 ms. Expert 1 detects the fewest number of smooth pursuit movements with a mean duration of 361 ms. Except for the I–PCA, the algorithms detect a larger number of smooth pursuit movements than the experts. For the test database, the result has a similar pattern

as for the development database. The largest difference is that Expert 1 does not detect any smooth pursuit movements.

- **Videos** – In the development database, the differences between experts and algorithms are larger than for images. The mean fixation duration is slightly larger for the proposed algorithm (218 ms) and considerably larger for I–VDT (360 ms) and I–PCA (298 ms), compared to the two experts with 206 ms and 179 ms, respectively. The I–VDT algorithm detects the largest number of smooth pursuit movements (66) and the I–PCA algorithm the fewest (22). For the test database, the agreement between the experts on the number of detected smooth pursuit movements is lower than for the development database.

- **Moving dots** – For the development database, the largest difference in the results is for the number of detected fixations, which ranges from 5 for Expert 1 to 37 for the I–VDT algorithm. The agreement between the proposed algorithm and the two experts is high for the number of detected smooth pursuit movements, 21 compared to 27 and 24, respectively. However, between the two experts, there is a large disagreement on the number of detected fixations and their durations, where Expert 1 detects fewer but longer fixations compared to Expert 2, with 5 and 17 fixations, respectively. For the test database, I–VDT and I–PCA have the shortest mean durations of smooth pursuit movements, 93.3 ms and 127 ms, respectively, compared to the proposed algorithm with a mean duration of 345 ms.

Table 2: Event properties for detected fixations and smooth pursuit movements, for image stimuli. A = Proposed algorithm, B = I–VDT, C = I–PCA, D = Expert 1, and E = Expert 2.

	Development database				
Measure	A	B	C	D	E
Mean fixation duration (ms)	217	241	224	217	214
Mean smooth pursuit duration (ms)	191	48.7	114	361	283
Number detected fixations	278	250	260	304	298
Number detected smooth pursuits	26	177	10	3	8
	Test database				
Measure	A	B	C	D	E
Mean fixation duration (ms)	317	372	345	350	346
Mean smooth pursuit duration (ms)	350	38.4	80	0	310
Number detected fixations	96	87	92	99	93
Number detected smooth pursuits	9	82	3	0	9

Table 3: Event properties for detected fixations and smooth pursuit movements, for video stimuli. A = Proposed algorithm, B = I–VDT, C = I–PCA, D = Expert 1, and E = Expert 2.

	Development database				
Measure	A	B	C	D	E
Mean fixation duration (ms)	218	360	298	206	179
Mean smooth pursuit duration (ms)	542	90.8	138	509	477
Number detected fixations	67	72	85	56	55
Number detected smooth pursuits	30	66	22	39	46
	Test database				
Measure	A	B	C	D	E
Mean fixation duration (ms)	406	616	463	509	338
Mean smooth pursuit duration (ms)	759	147	173	583	484
Number detected fixations	25	29	37	26	26
Number detected smooth pursuits	13	19	12	12	24

Table 4: Event properties for detected fixations and smooth pursuit movements, for moving dot stimuli. A = Proposed algorithm, B = I–VDT, C = I–PCA, D = Expert 1, and E = Expert 2.

	Development database				
Measure	A	B	C	D	E
Mean fixation duration (ms)	191	266	297	256	157
Mean smooth pursuit duration (ms)	417	54.5	104	388	384
Number detected fixations	15	37	32	5	17
Number detected smooth pursuits	21	40	14	27	24
	Test database				
Measure	A	B	C	D	E
Mean fixation duration (ms)	187	240	259	142	203
Mean smooth pursuit duration (ms)	345	93.3	127	328	344
Number detected fixations	8	23	20	4	2
Number detected smooth pursuits	16	18	8	20	20

Table 5: Percentage of fixations and smooth pursuit movements in the intersaccadic intervals, for image and moving dot stimuli. A = Proposed algorithm, B = I–VDT, C = I–PCA.

	Development database					
	Image			**Moving dot**		
Measure	A	B	C	A	B	C
% Fixations	91.2	86.9	98.1	21.4	52.8	86.6
% Smooth pursuits	8.79	13.1	1.94	78.6	47.2	13.4
	Test database					
	Image			**Moving dot**		
Measure	A	B	C	A	B	C
% Fixations	93.2	88.1	99.2	16.9	47.9	83.5
% Smooth pursuits	6.81	11.9	0.76	83.1	52.1	16.5

4.2 Proportion of events for different types of stimuli

The percentages of fixations and smooth pursuit movements in the intersaccadic intervals are calculated for image and moving dot stimuli, see Table 5. The values for the proposed algorithm are based on the intersaccadic intervals resulting from the algorithm in [13], the values for I–VDT and I–PCA are resulting from the intersaccadic intervals from the saccade detection of each algorithm. The percentages are calculated for the complete development database and the complete test database. The results are summarized below:

- **Images** – The expected result for images is to have close to 100% detected fixations and 0% detected smooth pursuit movements. The proposed algorithm detects 91.2% fixations, I–VDT 86.9%, and I–PCA 98.1%. The corresponding numbers for detected smooth pursuit movements are 8.79%, 13.1%, and 1.94% for the development database. For the test database, the results are very similar to the results of the development database.

- **Moving dots** – The expected result is to have as large amount of detected smooth pursuit movements as possible and as few detected fixations as possible. The proposed algorithm detects 78.6% smooth pursuit movements and 21.4% fixations. The I–VDT algorithm detects 47.2% smooth pursuit movements and 52.8% fixations for the development database. The I–PCA algorithm detects the largest amount of fixations 86.6% and only 13.4% of smooth pursuit movements. For the test database, the results are very similar to the development database, with a slight increase in the percentages of smooth pursuit movements for all three algorithms.

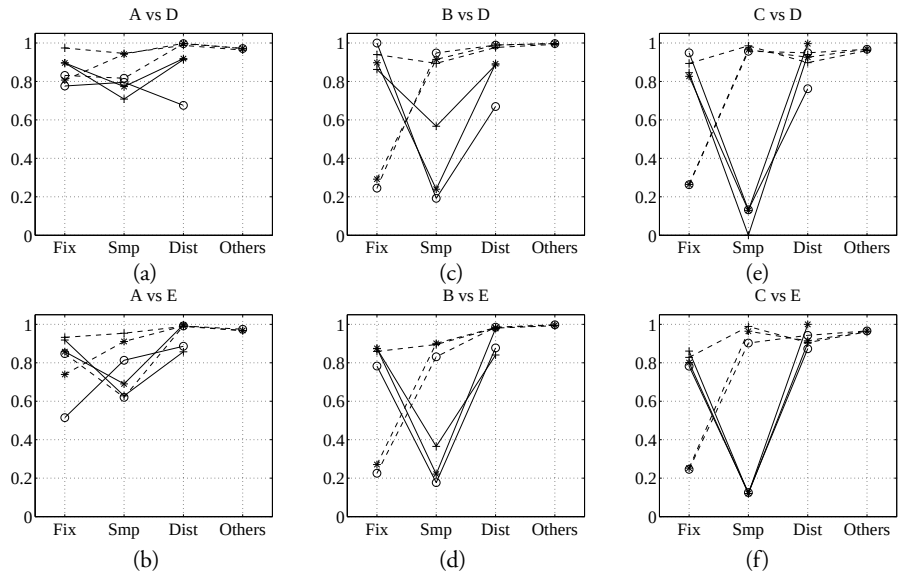


Figure 3: Sensitivity (solid) and specificity (dashed), for images (+), video (*), and moving dot (o). (a) Proposed algorithm (A) with Expert 1 (D) as reference. (b) Proposed algorithm (A) with Expert 2 (E) as reference. (c) I–VDT algorithm (B) with Expert 1 (D) as reference. (d) I–VDT algorithm (B) with Expert 2 (E) as reference. (e) I–PCA algorithm (C) with Expert 1 (D) as reference. (f) I–PCA algorithm (C) with Expert 2 (E) as reference.

4.3 Sensitivity and specificity analysis

Using both experts as references, the sensitivities and specificities of the detected fixations and smooth pursuit movements for the proposed algorithm, the I–VDT algorithm, and the I–PCA algorithm, respectively, are shown in Fig. 3. In the ideal case, both the sensitivity and specificity should be as close to one as possible. Below is a summary for the different types of events for the development database:

- **Fixations** – In general, the sensitivity for fixations is high, with values around 0.8 – 0.9, for the algorithms and with both experts as reference. For the specificity, there is a larger difference between the algorithms, where the proposed algorithm has values around 0.8 – 0.9 for all stimuli, while I–VDT and I–PCA have values ranging from 0.3 for video and moving dot stimuli, to 0.9 for image stimuli.

- **Smooth pursuit movements** – The sensitivity for the proposed algorithm is generally in the same range as for fixations, i.e., between 0.6 – 0.8 for all types

of stimuli and compared to both experts. For the I–VDT algorithm, the sensitivity is between 0.2 – 0.6 for all types of stimuli. For the I–PCA algorithm, the sensitivity is between 0 – 0.2 for all types of stimuli, which is much lower than for fixations. The specificity for smooth pursuit movements is high for all algorithms, for all types of stimuli, and compared to both experts.

- **Disturbances and Others** – The sensitivity and specificity for Disturbances are high for all algorithms, independent of stimuli and expert, with values around 0.9. Since the proposed algorithm and I-VDT have the same preprocessing procedure the results for the two algorithms are almost identical. For I–PCA, which besides the preprocessing part from [13], has its own very strict disturbances detection, the result is slightly different from the other two algorithms. The specificities for the event type Others are high for both algorithms, which means that the expert and the algorithms are in agreement about the transitions between saccades/PSO and other events.

4.4 Cohen's kappa analysis

In order to be able to measure the overall agreement between the experts and the algorithms, Cohen's kappa, κ, is calculated between each of the two experts and each of the three algorithms, see Tables 6 – 7. For the development database, Cohen's kappa for the proposed algorithm is larger than Cohen's kappa for I–VDT and I–PCA, for all types of stimuli. However, the agreement between the experts is even larger. For the test database, Cohen's kappa for the proposed algorithm is larger for video and moving dot stimuli, but lower than the other two algorithms for image stimuli. Also, Cohen's kappa between the experts is much lower for image stimuli than for other stimuli types.

Table 6: Cohen's kappa between Expert 1 and the proposed algorithm, I–VDT, I–PCA, and Expert 2.

	Development database			**Test database**		
	Image	**Video**	**Moving dot**	**Image**	**Video**	**Moving dot**
Proposed algorithm	0.620	0.671	0.446	0.0685	0.383	0.423
I–VDT	0.524	0.180	0.098	0.091	0.378	0.0522
I–PCA	0.475	0.113	0.0827	0.12	0.24	0.0242
Expert 2	0.806	0.784	0.573	0.113	0.402	0.816

Table 7: Cohen's kappa between Expert 2 and the proposed algorithm, I–VDT, I–PCA, and Expert 1.

	Development database			Test database		
	Image	**Video**	**Moving dot**	**Image**	**Video**	**Moving dot**
Proposed algorithm	0.667	0.530	0.412	0.0595	0.401	0.309
I–VDT	0.537	0.127	0.050	0.105	0.172	0.0362
I–PCA	0.501	0.0744	0.0524	0.066	0.152	0.0257
Expert 1	0.834	0.779	0.550	0.116	0.395	0.687

4.5 Scores evaluation

The performance of the proposed algorithm is also evaluated by calculating scores for smooth pursuit movements, as proposed in [10]. Since scores can only be used when the coordinates of the stimuli are known, they are calculated for 17 trials containing moving dot stimuli. The scores were computed for the proposed algorithm, the I–VDT algorithm, the I–PCA algorithm, and Expert 2, and their values over the 17 trials are shown in Table 8, (these trials were not annotated by Expert 1).

- **PQnS** –All the algorithms and Expert 2 have a value of $\mathrm{PQnS}_{\mathrm{ideal}}$ close to 90%. The PQnS value of Expert 2 is closest to its corresponding ideal value, 76.8% compared to ideal the value 90.9%. Between the algorithms, the proposed algorithm is closer to its corresponding ideal value with 62.7% and ideal value 88.9%, compared to I–VDT with 24.7% and ideal value 86%, and I–PCA with 13.1% and ideal value 84.9%.

- **PQlS**$_{\mathrm{P}}$ – The values for $\mathrm{PQlS}_{\mathrm{P}}$, which describes the mean distance between the smooth pursuit samples and the stimuli, are around $2.1 - 2.9°$ for the proposed algorithm, I–VDT, I–PCA, and Expert 2. The corresponding ideal value is $0°$.

- **PQlS**$_{\mathrm{V}}$ – The mean differences between the velocity of the stimuli and that of the eye are ranging from $12.1°$/s for the proposed algorithm to $36.6°$/s for I–PCA. The corresponding ideal value is $0°$/s.

Cohen's kappa is also calculated separately between Expert 2 and the proposed algorithm, I–VDT, and I–PCA, respectively, for the 17 trials used in the calculation of the scores. The results are shown in Table 8; the proposed algorithm has a Cohen's kappa of 0.31 and I–VDT and I–PCA have 0.07. These results are in the same range as the values for Cohen's kappa for moving dot stimuli in Table 7.

Table 8: Values of the scores and Cohen's kappa between Expert 2 and the proposed algorithm, I–VDT, and I–PCA. A = Proposed algorithm, B = I–VDT, C = I–PCA, and E = Expert 2.

Measure	A	B	C	E
$PQnS_{ideal}$ (%)	88.9	86.0	84.9	90.9
PQnS (%)	62.7	24.7	13.1	76.8
$PQlS_P$ (°)	2.83	2.98	2.1	2.9
$PQlS_V$ (°/s)	12.1	17.9	36.6	13.4
Cohen's kappa	0.31	0.07	0.07	1

5 Discussion

An algorithm for discriminating between fixations and smooth pursuit movements was developed. In order to perform the discrimination, the algorithm uses four features of the position signal. The algorithm was evaluated using signals recorded during both static and dynamic stimuli presentation, and was compared to the I–VDT algorithm [10] and the I–PCA algorithm [11], as well as to annotations performed by experts. In general, regardless of stimuli, the proposed algorithm detected longer but fewer smooth pursuit movements than the I–VDT algorithm. One reason for this behavior may be that the I–VDT algorithm used only one feature of the signal, the dispersion, in order to detect the smooth pursuit movement, and when the dispersion exceeded the threshold several times, the signal became more segmented. In comparison to the I–PCA, which uses several features to detect the smooth pursuit movements, the proposed algorithm detects longer and a larger amount of smooth pursuit movements.

The percentages of fixations and smooth pursuit movements were calculated for two types of stimuli – images and moving dots. In theory, it is expected to have close to 100% detected fixations and close to 0% detected smooth pursuit movements for images and the opposite for moving dots. For images, the results were 91.2% fixations and 8.8% smooth pursuit movements for the proposed algorithm. A part of the samples that was detected as smooth pursuit movements during image stimuli may be due to vergence. Since the proposed algorithm uses data from one eye only, it cannot distinguish such movements from smooth pursuit movements. It should be noted that also the experts detected 1 – 2% smooth pursuit movements in image stimuli, calculated for the manually annotated part of the development database.

The settings for the I–VDT algorithm were chosen as suggested in [10]. By using these settings on our database, the I–VDT algorithm is tuned to detect fixations well, which can be seen from the values of Cohen's kappa in Table 6, (0.52, 0.18, 0.10), for image, video, and moving dot stimuli respectively. In order to

make the algorithm less sensitive to fixations, Cohen's kappa was calculated also for $T_D = 1.1°$, (0.33, 0.22, 0.28). By lowering the dispersion threshold, the I–VDT algorithm detects a larger number of smooth pursuit movements and shorter durations of fixations in the video and the moving dot stimuli, and gives a larger Cohen's kappa for these types of stimuli. Even though Cohen's kappa becomes more evenly distributed over the three types of stimuli, it is not in the ranges of the proposed algorithm that has a larger Cohen's kappa for all types of stimuli.

For the I–PCA algorithm the default settings of the algorithm were chosen. The algorithm is clearly tuned and developed to be able to detect saccades and fixations of a predefined size, shape, and duration, which is shown by the low percentage-values in Table 5 for image stimuli. The results in Table 5 also show the difficult trade-off between accurately detecting few smooth pursuit movements in image stimuli and at the same time a large amount of smooth pursuit movements in moving dot stimuli.

When comparing Cohen's kappa of the algorithms to Cohen's kappa between the experts, the agreement between the experts is higher. However, Cohen's kappa between the experts still has a large variation between the different types of stimuli, with values from 0.55 for moving dot stimuli to 0.83 for image stimuli, which indicates that the separation of fixations and smooth pursuit movements is a difficult task even for experts. One explanation to why Cohen's kappa is a bit lower than expected between experts and between experts and algorithms, may be due to that the data are unbalanced; for image stimuli, the majority of samples are fixations with very few smooth pursuit movements while the opposite is true for moving dot stimuli. When Cohen's kappa is calculated for databases that have an unbalanced distribution between the types of events, small differences in the two compared detections lead to a substantial decrease in Cohen's kappa, even though the detections are correct most of the time. An example is given in Fig. 4 where Cohen's kappa is 0 between the expert and the proposed algorithm since the expert does not classify any sample as a fixation. An unbalanced database in combination with a low number of trials are the reasons for the much lower values of Cohen's kappa in Tables 6 – 7 for images stimuli in the test database, both for algorithms and experts.

An important question is whether the annotations represent a "gold standard". The information that the algorithm and the expert is using in order to make the decision may differ a lot and may potentially render the comparison unfair. The experts can often guess which types of stimuli that have been used. This may partly explain why the two experts have a larger agreement between themselves than between experts and algorithms. The fact that the two experts sometimes differ makes it even harder to decide which one to trust or use as the "gold standard". In [18], three manual coders were used and the correlation between the coders ranged between 0.58 – 0.85. This is comparable to the Cohen's kappa between experts reported in this paper.

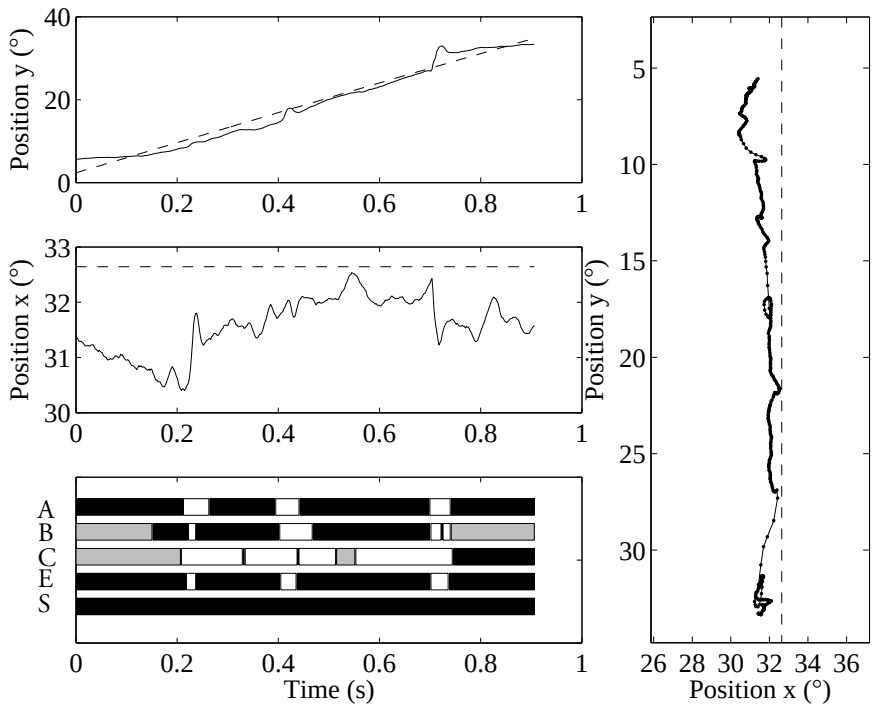


Figure 4: Example of a trial with a moving dot, where the detections for the proposed algorithm are in agreement with the expert most of the time, but Cohen's kappa is 0. The lower panel shows the detection results for the proposed algorithm (A), I–VDT (B), I–PCA (C), Expert 2 (E), and what the stimuli was (S). Black color represents smooth pursuit movements, grey samples are fixations and white are all other types of eye movements.

The performance of the proposed algorithm was evaluated using five different methods, each with advantages and drawbacks. In order to provide an overview of the detected events, their properties, and the proportion of events for different types of stimuli, method 1, (event properties) and 2, (proportion of events for different types of stimuli), are satisfactory methods. However, these methods do not reveal whether the events were correctly detected or not. By using method 3, (Sensitivity and specificity analysis), and method 4, (Cohen's kappa analysis), the accuracy of the classification is taken into account. The drawback with these methods is that there is a need for a "gold standard", to which the results of the algorithm can be compared. In this paper, manual annotations from two experts were used. When using method 5, (Scores), there is no need for time consuming annotations since the stimuli are used as references. However, this strategy cannot be used for all types of stimuli, e.g., not for images, text stimuli or video stimuli. In addition, not all types of events can be evaluated, e.g., PSO, since they are not driven by the stimulus. To summarize, when comparing and evaluating algorithms, the prerequisite in terms of stimuli and types of events to be detected will control which type of evaluation method that should be used. All methods are complementary and no single method will show the complete performance of the evaluated algorithm.

So far, discrimination between fixations and smooth pursuit movements has mainly been used in human-computer interaction using low speed eye-trackers, e.g., to stabilize the cursor during gaze control of a computer screen [19], and in interaction with information screens [20]. Having the possibility to separate between the two event types also for high-speed eye-trackers is paving the way for studies where the properties of the two types of events can be investigated and compared. Two examples of such applications are to measure the difference in smooth pursuit characteristics between experts and novices when watching dynamic stimuli [21], and the amount of smooth pursuit when viewing natural stimuli as a diagnostic tool for neural disorders [22].

6 Conclusions

Discrimination between fixations and smooth pursuit movements is a difficult task since many of the signal characteristics of the two event types are similar. In this work, an algorithm for the discrimination between fixations and smooth pursuit movements in high-speed eye-tracking data is developed and compared with two existing algorithms and to annotations from two experts. A rigorous performance evaluation strategy was employed to capture different aspects of the algorithm's behavior. The proposed algorithm outperforms two current state-of-the-art algorithms for detection of fixations and smooth pursuit movements, regardless of the stimuli and evaluation method. However, the agreement to annotations is not as high as the inter-rater agreement between the experts.

Acknowledgment

This work was supported by the Strategic Research Project eSSENCE, funded by the Swedish Research Council. Data were recorded in the Lund University Humanities Laboratory.

References

[1] B. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *Journal of Vision*, vol. 7, no. 14, pp. 1 – 17, 2007.

[2] J. Henderson and A. Hollingworth, "High-level scene perception," *Annual Review of Psychology*, vol. 50, pp. 243 – 271, 1999.

[3] K. Gidlöf, A. Wallin, R. Dewhurst, and K. Holmqvist, "Gaze behavior during decision making in a natural environment," *Journal of Eye movement research*, vol. 6, no. 1, pp. 1–14, 2013.

[4] K.-M. Flechtner, B. Steinacher, R. Sauer, and A. Mackert, "Smooth pursuit eye movements in schizophrenia and affective disorder," *Psychological Medicine*, vol. 27, pp. 1411–1419, 1997.

[5] M. Dorr, T. Martinetz, K. R. Gegenfurtner, and E. Barth, "Variability of eye movements when viewing dynamic natural scenes," *Journal of vision*, vol. 10, no. 10, pp. 1–17, 2010.

[6] E. Kowler, "Eye movements: The past 25 years," *Vision Research*, vol. 51, pp. 1457–1483, 2011.

[7] P. Mital, T. Smith, R. Hill, and J. Henderson, "Clustering of gaze during dynamic scene viewing is predicted by motion," *Cognitive computation*, vol. 3, no. 1, pp. 5–24, 2010.

[8] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2011.

[9] M. Vidal, A. Bulling, and H. Gellersen, "Analysing eog signal features for the discrimination of eye movements with wearable devices," in *Proceedings of the 1st international workshop on pervasive eye tracking & mobile eye-based interaction*, pp. 15–20, ACM, 2011.

[10] O. Komogortsev and A. Karpov, "Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades," *Behavior Research Methods*, vol. 45, no. 1, pp. 203–215, 2013.

[11] D. J. Berg, S. E. Boehnke, R. A. Marino, D. P. Munoz, and L. Itti, "Free viewing of dynamic stimuli by humans and monkeys," *Journal of Vision*, vol. 9, no. 5, pp. 1–15, 2009.

[12] M. Vidal, A. Bulling, and H. Gellersen, "Detection of smooth pursuits using eye movement shape features," in *Proceedings of the symposium on eye tracking research and applications*, pp. 177–180, ACM, 2012.

[13] L. Larsson, M. Nyström, and M. Stridh, "Detection of saccades and post-saccadic oscillations in the presence of smooth pursuit," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 9, pp. 2484–2493, 2013.

[14] C. Meyer, A. Lasker, and D. Robinson, "The upper limit of human smooth pursuit.," *Vision Research*, vol. 25, no. 4, pp. 561–563, 1985.

[15] P. Berens, "Circstat: A matlab toolbox for circular statistics," *J. Stat. Softw.*, vol. 31, no. 10, pp. 1–21, 2009.

[16] L. Sörnmo and P. Laguna, *Bioelectrical signal processing in cardiac and neurological applications*. Elsevier, 2005.

[17] O. Komogortsev, D. Gobert, S. Jayarathna, D. Hyong Koh, and S. Gowda, "Standardization of automated analyses of oculomotor fixation and saccadic behaviors," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2635–2645, 2010.

[18] S. M. Munn, L. Stefano, and J. B. Pelz, "Fixation-identification in dynamic scenes: Comparing an automated algorithm to manual coding," in *Proceedings of the 5th symposium on Applied perception in graphics and visualization*, pp. 33–42, ACM, 2008.

[19] J. S. A. Lopez, *Off-the-shelf Gaze Interaction*. PhD thesis, PhD thesis, 2009.

[20] M. Vidal, A. Bulling, and H. Gellersen, "Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 439–448, ACM, 2013.

[21] H. Jarodzka, K. Scheiter, P. Gerjets, and T. van Gog, "In the eyes of the beholder: How experts and novices interpret dynamic stimuli," *Learning and Instruction*, vol. 20, no. 2, pp. 146–154, 2010.

[22] M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Wearable eye tracking for mental health monitoring," *Computer Communications*, vol. 35, no. 11, pp. 1306–1311, 2012.

Paper III

Smooth Pursuit Detection in Binocular Eye-Tracking Data with Automatic Video-Based Performance Evaluation

Abstract

Objective: An increasing number of researchers record binocular eye-tracking signals from participants viewing moving stimuli, but the majority of event detection algorithms are, however, monocular and do not consider smooth pursuit movements. The purposes of the present study are to develop an algorithm that discriminates between fixations and smooth pursuit movements in binocular eye-tracking signals and to evaluate its performance using an automated video-based strategy. *Methods:* The proposed algorithm uses a clustering approach that takes both spatial and temporal aspects of the binocular eye-tracking signal into account, and is evaluated using a novel video-based evaluation strategy based on automatically detected moving objects in the video stimuli. *Results:* The binocular algorithm detects 98% fixations in image stimuli compared to 95% when only one eye is used, while for video stimuli, both the binocular and monocular algorithms detect around 40% smooth pursuit movements. *Conclusion:* The present paper shows that using binocular information for discrimination of fixations and smooth pursuit movements is advantageous in static stimuli, without impairing the algorithms ability to detect smooth pursuit movements in video and moving dot stimuli. *Significance:* By using an automated evaluation strategy, time consuming manual annotations are avoided and a larger amount of data can be used in the evaluation process.

Based on:
Linnéa Larsson, Marcus Nyström, Håkan Ardö, Kalle Åström and Martin Stridh, "Smooth Pursuit Detection in Binocular Eye-Tracking Data with Automatic Video-Based Performance Evaluation,"
Submitted for publication.

1 Introduction

Eye-tracking is an important research tool which measures the movements of the eyes. It is an established technique to investigate the comprehension and understanding of, e.g., a text [1] or an image [2]. Research using eye-tracking also includes clinical applications, e.g., examination of eye movement dysfunctions in patients with schizophrenia [3], dyslexia [4], and the human vestibular system [5]. The most common movements of the eye are *fixations* and *saccades*. A fixation is when the eye is more or less still and visual information is taken in. A saccade is instead a fast eye movement that redirects the eye from one position to the next. When the eye follows a moving target, the eye movement is called a *smooth pursuit*. In order to see a moving object clearly during smooth pursuit, the object must be aligned with the direction of gaze. When the object is not perfectly followed by the eye, small corrective saccades are used to re-align the direction of the gaze to that of the moving object. A smooth pursuit is divided into two stages: open-loop and closed-loop [6]. The open-loop stage is the initial stage when the smooth pursuit is initiated by a movement of an object. The second closed-loop stage is a feedback system where the velocity of the eye is controlled in order to keep the eye on the moving object. The upper limit for the velocity of a smooth pursuit movement is 100°/s [7]. No lower limit for smooth pursuit velocity seems to exist, and the pursuit system can operate in the same velocity range as fixational eye movements [8].

A majority of eye-tracking studies are performed using monocular recordings [9], i.e., only one eye is recorded. The popularity of monocular recordings is partly due to the common belief that the two eyes are performing the same movements at the same time, which is not always the case [9, 10, 11]. In addition, monocular eye-trackers are cheaper than binocular ones, which contributes to its popularity [9]. Studies where binocular aspects are important are most often clinical where the binocular coordination and control are investigated, e.g, for children with dyslexia [4] and for patients with cerebellar dysfunction [12].

Since a majority of studies are based on monocular recordings, event detection algorithms, which classify the eye-tracking signal into different types of eye movements, are typically also developed for monocular data. An exception is the Binocular-Individual Threshold (BIT) algorithm which uses both eyes and is developed to adapt its internal settings to each specific task and participant [10]. The BIT-algorithm is a velocity based algorithm that uses minimum determinant covariance estimates and control chart procedures in order to detect fixations, saccades, and blinks. Other algorithms have indirectly used binocular information, e.g., by averaging the data from the two eyes in order to reduce the level of noise [13]. Another strategy is used in [14], where the detection algorithm separately analyzes the data from the two eyes and at a later stage combines the two series of events into one. The algorithm was proposed for the detection of microsaccades which occurred simultaneously in both eyes. None of the algorithms that uses binocular

information, [10, 13, 14], detects smooth pursuit movements.

To accurately differentiate between smooth pursuit movements and fixations in eye-tracking data recorded during dynamic scene viewing is still a major challenge. Inclusion of binocular information may improve classification robustness and make it easier to distinguish between smooth pursuit movements and vergence movements where the eyes move in opposite directions. Therefore, in this paper, we address the above issues by proposing a new event detection algorithm to discriminate between fixations and smooth pursuit movements which takes both spatial and temporal aspects of binocular eye-tracking signals into account.

Another novelty of this paper is a video-based performance evaluation strategy that is based on automatic detection of moving objects in the video stimuli. Evaluation of event detection algorithms has earlier mainly been performed through *manual annotations* [15], by using *simulated eye-tracking data* [16], and by recording eye-tracking data for *artificial stimuli* such as moving dots [17, 18]. None of these methods is completely satisfactory or practical for dynamic scene viewing. Manual annotations are time consuming, and suffer from subjectivity and often large inter-rater variability. To build simulation models that mimic the complexity and individuality of eye-tracking signals is difficult to achieve for all types of eye movements. When artificial stimuli is used not only the algorithm's performance is evaluated, but also the viewer's ability to follow the presented stimuli. An evaluation method based on artificial stimuli is proposed in [17, 18]. The speed and position of the moving dots are compared to the corresponding characteristics of the eye-tracking signals, and a set of scores are calculated. The method is, however, limited to stimuli were the coordinates of the moving dots are known. To evaluate the performance of a smooth pursuit detection algorithm when eye-tracking data are recorded for complex videos, automatic tracking of the trajectories of the moving objects is needed. Knowledge about the trajectories of all moving objects opens up for new possibilities to evaluate the performance of event detectors. The proposed video-based evaluation strategy relates the eye-tracking signal to the trajectories of the moving objects and compares them to the smooth pursuit movements detected by the proposed algorithm. The main assumption of the video-based evaluation strategy is that a detected smooth pursuit movement is correct only when the eye-tracking signal is aligned with a moving object in terms of position, velocity, and direction.

Compared to manual annotation, the video-based evaluation strategy does not provide as detailed information about the detected events, but has three important advantages: it is more objective, it is faster, and it relates the eye-tracking signal to the content of the video. This is practical for future studies where longer sequences of dynamic stimuli may be used in combination with a larger number of participants. It is, however, important to note that, in the current work, the video data are used only for evaluation purposes, and not as part of the event detector as is

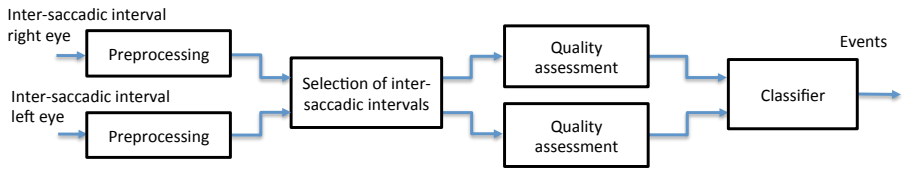


Figure 1: Overview of the structure for the proposed binocular event detection algorithm.

proposed in [19].

The paper is structured as follows: The proposed algorithm and the video-based evaluation strategy are described in Section 2, and in Section 3, a description of the eye-tracking recording procedure and the database is given. In Section 4, the results are presented, and, finally, in Section 5, the results are discussed.

2 Methods

The following section is divided into two parts, which describe the proposed binocular event detection algorithm and the video-based evaluation strategy, respectively.

2.1 Binocular event detection algorithm

The proposed algorithm contains four stages; an overview is shown in Fig. 1. The algorithm is applied to inter-saccadic intervals derived, e.g., from [20], which are intervals between detected saccades/postsaccadic oscillations (PSO) and blinks. In the first stage, the inter-saccadic intervals are preprocessed and in the second stage the inter-saccadic intervals from the two eyes are compared and intervals that occur in both eyes simultaneously are selected. The third stage contains quality assessment of the selected inter-saccadic intervals, and in the final stage, the samples in the intervals are classified into fixations and smooth pursuit movements based on the directionality of the data from the two eyes.

Preprocessing

The main objective of the preprocessing stage is to remove samples that do not belong to fixations or smooth pursuit movements. Since smooth pursuit movements can not move faster than 100°/s [7], all samples in the beginning and/or end of the inter-saccadic interval with corresponding velocities that exceed 100°/s are removed and assumed to belong to adjacent saccades or PSO.

Selection of inter-saccadic intervals

In this stage, the inter-saccadic intervals that occur in both eyes simultaneously are determined. If a saccade is detected in both eyes, the onset and offset of the saccade with the longest duration determines the end of the previous inter-saccadic interval and the beginning of the next. Saccades that are not detected in both eyes are classified as "true" monocular saccades or noise. The main difference between a true saccade and noise is measured in the velocity signal, where the noise is several order of magnitudes larger and most often contains spikes. In order to determine whether the detected saccades contain spikes, the sample-to-sample velocity, $v^e(n)$, for each eye separately, is compared to $v^e_m(n)$, which is $v^e(n)$ filtered with a median filter of length 3, where $e = \{\text{L} - \text{left eye, R} - \text{right eye}\}$. The residual signal, $r_e(n) = v^e(n) - v^e_m(n)$, is calculated to contain the spike, where $v^e(n) = \sqrt{(v^e_x(n))^2 + (v^e_y(n))^2}$ is the sample-to-sample velocity and $v^e_m(n) = \sqrt{(v^e_{mx}(n))^2 + (v^e_{my}(n))^2}$ the median filtered velocity. The spike index, SI_e, is calculated as the ratio between the residual signal and the original $v^e(n)$ and is for each eye calculated as

$$SI_e = \frac{\sum_{n=1}^{M} |r_e(n)|}{\sum_{n=1}^{M} |v^e(n)|} \tag{1}$$

where M is the number of samples in the saccade. If $SI_e > \eta_{SI}$ for both eyes, the detected saccade is classified as noise and the two previous inter-saccadic intervals which were split by the saccade, are merged into one. A saccade is considered to be correctly classified if $SI_e \leq \eta_{SI}$ for both eyes. Moreover, if $SI_e \leq \eta_{SI}$ for one of the eyes, the amplitude of that saccade is determined, and if the amplitude is larger than η_{SA} the saccade is considered to be correctly classified.

Quality assessment

The quality of the eye-tracking signal may vary, especially in terms of undesired high frequency noise. By calculating the high frequency content of the recorded signal, an estimate of the amount of high frequency noise is obtained. For each eye separately, a non-overlapping sliding window of 50 ms is applied to the inter-saccadic intervals. Within each window a differential filter of length 2 is applied. A high frequency noise content index, $I_{hf}(i)$, representing the energy of the differential signal for each window i, is calculated as

$$I^{ex}_{hf}(i) = \frac{1}{N-1} \sum_{n=2}^{N} |x_e(n) - x_e(n-1)|^2 \tag{2}$$

$$I^{ey}_{hf}(i) = \frac{1}{N-1} \sum_{n=2}^{N} |y_e(n) - y_e(n-1)|^2 \tag{3}$$

where $x_e(n)$ and $y_e(n)$ represent the respective coordinates of the eye-tracking signals for each eye e. The number of samples in the sliding window is denoted N, and $i = 1, ..., M$, where M is the number of windows in the inter-saccadic interval. For each inter-saccadic interval and for each eye separately, the maximum values of the high frequency noise content indices are calculated.

$$I^R_{hf_{max}} = \max[I^{Rx}_{hf}(i), I^{Ry}_{hf}(i)] \quad \forall i \tag{4}$$

$$I^L_{hf_{max}} = \max[I^{Lx}_{hf}(i), I^{Ly}_{hf}(i)] \quad \forall i \tag{5}$$

The maximum high frequency content, $I^R_{hf_{max}}$, and $I^L_{hf_{max}}$, are mapped on to the generalized logistic function S,

$$S(a) = A + \frac{K - A}{(1 + Qe^{-B(a-P)})^{1/\nu}} \tag{6}$$

where $A = 0$, $K = 1$, $Q = 0.001$, $P = 3000$, $B = 0.001$, and $\nu = Q$. The parameters of the function $S(a)$ are determined by visual inspection of the complete database to best separate high frequency noise from data with good quality. The range of the generalized logistic function, $S(a)$, is from 0 to 1, where 0 indicates a low level of high frequency noise content and 1 indicates a high level of high frequency noise content. Therefore, all inter-saccadic intervals where $S(I^R_{hf_{max}}) < \eta_S$ OR $S(I^L_{hf_{max}}) < \eta_S$, are considered to have high enough quality in order to be further classified into events.

Classifier

For each inter-saccadic interval that occurs in both eyes simultaneously and for which the signal has $S(a) < \eta_S$, a classifier is applied. The classifier consists of the following steps: directional clustering, binary filters, and classification.

Directional clustering For each consecutive pair of x- and y- coordinates, the sample-to-sample direction, $\alpha(n)$, is calculated. It is defined as the angle between the line connecting consecutive pairs of x- and y- coordinates and the x- axis. The sample-to-sample directions are mapped on to the unit circle and are clustered using the iterative minimum-squared-error clustering algorithm [21]. The procedure of the clustering algorithm is described below. First, the threshold for the maximum angular span of a cluster is initialized to $\gamma_{\max}$, which is the maximum size of the sector for one cluster. Each cluster, i, is described by its angular span, γ_i, and its mean direction, m_i. In the initial iteration, all $\alpha(n)$ are placed into cluster $i = 1$, and the mean, m_1, and the angular span, γ_1, are calculated.

Assuming that the number of clusters is L, each following iteration starts by determining which cluster, j, that has the maximum angular span. If $\gamma_j > \gamma_{\max}$,

Algorithm 1: The directional clustering algorithm.

Initial iteration:

- Compute $\alpha(n)$ for all samples in the inter-saccadic interval.
- Place all $\alpha(n)$ into cluster 1, calculate m_1 and γ_1.
- Initialize γ_{max}.

while $\max[\gamma_i] > \gamma_{\text{max}}$ **do**
  Divide the largest cluster into two new clusters.
  Remove the affiliation of the samples.
  foreach $\alpha(n)$ **do**
    Calculate the angles β_i between $\alpha(n)$ and the mean directions, m_i.
    Assign $\alpha(n)$ to cluster i which has the minimum β_i.
  end
  Compute the angular span γ_i for each cluster.
end

cluster j is split into two clusters, cluster j and cluster $L+1$. The mean direction, m_{L+1}, is initialized to the sample of $\alpha(n)$ of those belonging to cluster j which has the largest angle $\beta_j(n)$ to m_j. The mean direction, m_j, is recalculated as the mean value of the remaining directions that belong to cluster j. The mean values of the clusters are saved and the affiliations of the samples are removed.

By randomly selecting one $\alpha(k)$ at the time and by measuring the angles $\beta_i(k)$ between $\alpha(k)$ and the mean directions of each cluster, $\alpha(k)$ is assigned to the cluster with the closest mean direction, where k ranges from 1 to the sample length of the inter-saccadic interval. Each time an $\alpha(k)$ is assigned to a cluster, the mean direction and the angular span of that cluster is updated. When all $\alpha(n)$ are reassigned to a cluster, the maximum angular span, γ_j, is again calculated and compared to γ_{max}. The procedure continues until all clusters have an angular span that is smaller than γ_{max}. When the clustering process has converged, all samples are assigned to a cluster. A short pseudo code of the algorithm is shown in Algorithm 1.

Binary filters In the next step, four different types of binary filters are used to discriminate between fixations and smooth pursuit movements. In this paper, a binary filter refers to a filter which has a length and a criteria. If the samples in the filter satisfy the criterion, the output for the central sample of the filter is 1 or -1, depending on the purpose of the filter. If the samples do not satisfy the criterion, the output is 0. The purpose of having different types of binary filters is that each filter emphasizes either typical properties of fixations or typical properties

of smooth pursuit movements. The filters are applied with different lengths and criteria to the clustered signal. The four types of filters are: Transition, Directional consistency, Total distance, and Synchronization. A Transition filter counts the number of transitions between the clusters within the filter length. A transition occurs when two consecutive samples belong to clusters that differ between their mean directions, m_i, with an angle larger than α_T. Transitions between clusters are more frequent in samples belonging to fixations than in samples belonging to smooth pursuit movements. Therefore, a large transition rate more likely represents a fixation.

The Directional consistency filter counts the number of samples that are in the same cluster or in a neighboring cluster maximally α_T away. A large number of samples in the same cluster represents samples that are heading in the same direction, which is a typical feature of a smooth pursuit movement.

The Total distance filter determines the distance, d_S, that the samples in the filter have moved in total. In order to determine the distance that the samples, $x(n)$ and $y(n)$ in each cluster, i, has moved, the distance d_i is calculated as

$$d_i = \sum_{n=1}^{M} \sqrt{(x(n+1) - x(n))^2 + (y(n+1) - y(n))^2} \tag{7}$$

where M is the number of samples that are covered by the filter. Each cluster i has mean direction, m_i, which the corresponding d_i is mapped to in order to calculate the total distance. The distance d_S represents the actual movement of the samples in the filter.

$$d_S = \sum_{i=1}^{N} \sqrt{(d_i \cos m_i)^2 + (d_i \sin m_i)^2} \tag{8}$$

where N is the number of clusters. The total distance d_S is compared to the criterion of the filter. A small distance is representative for a fixation and a longer distance is representative for a smooth pursuit movement.

Finally, the Synchronization filter measures the synchronization between the eye-tracking signal from the two eyes, and is therefore only active if signals from both eyes are present. The filter counts the number of samples where the sample-to-sample direction $\alpha(n)$ from the two eyes, are in the same or a neighboring cluster, maximally α_T away, at the same time.

The filters and their lengths and criteria are shown in Tables 1–2. When the criterion of a filter is fulfilled, the central sample receives the output -1 for filters emphasizing fixations (Table 1) and 1 for filters emphasizing smooth pursuit movements (Table 2), resulting in K binary responses, $r_l(n)$, $l = 1, 2, ..., K$, for each inter-saccadic interval. If only the signal from one eye has passed the quality assessment, $K = (7 + 9) = 16$, i.e., filters F1-F7 and S1-S9 are used. While when the

Table 1: Settings for binary filters, F1-F9, which emphasize fixations. The filter length for the filter Total distance is described as the percentage of the current inter-saccadic interval.

Number	Type of filter	Length	Criterion
F1	Transition	100 ms	$> 60\%$
F2		120 ms	$> 50\%$
F3		200 ms	$> 65\%$
F4	Directional consistency	60 ms	$< 50\%$
F5	Total distance	100%	$< 0.68°$
F6		75%	$< 0.55°$
F7		50%	$< 0.55°$
F8	Synchronization	50 ms	$< 30\%$
F9		80 ms	$< 35\%$

Table 2: Settings for binary filters, S1-S9, which emphasize smooth pursuit movements. The filter length for the filter Total distance is described as the percentage of the current inter-saccadic interval.

Number	Type of filter	Length	Criterion
S1	Transition	40 ms	$< 20\%$
S2		110 ms	$< 20\%$
S3		150 ms	$< 25\%$
S4	Directional consistency	50 ms	$> 90\%$
S5		130 ms	$> 90\%$
S6		180 ms	$> 90\%$
S7	Total distance	100%	$> 1.00°$
S8		75%	$> 0.68°$
S9		50%	$> 0.51°$

signals from both eyes have passed the quality assessment, $K = 2 \cdot (7 + 9) + 2 = 34$, i.e., filters F1-F7 and S1-S9 are used for both eyes separately and filter F8-F9 for the combination of the two eyes. Finally, the responses are added together to one summation signal $s(n)$.

$$s(n) = \sum_{l=1}^{K} r_l(n) \tag{9}$$

Classification Based on the summation signal $s(n)$, the samples are classified into fixations and smooth pursuit movements. In general, when $s(n) \geq 0$, sample n is classified as a smooth pursuit movement and when $s(n) < 0$, sample n is classified as a fixation. In order to prevent the samples in the inter-saccadic interval to be divided into small segments of smooth pursuit movements and fixations, the dominant type of eye movement of the inter-saccadic interval is estimated. The estimation is based on the sign of the mean value of $s(n)$, and is used to filter out non-matching candidate fixations or smooth pursuit movements that are shorter than, t_{minFix} or t_{minSmp}, respectively. Correspondingly, the dominant event is fixation, i.e., the sign of the mean of $s(n) < 0$, smooth pursuit movements shorter than t_{minSmp} are converted to fixations. If the dominant event is smooth pursuit, i.e., the sign of the mean of $s(n) \geq 0$, shorter fixations than t_{minFix} are converted to smooth pursuit movements.

2.2 Video-based performance evaluation strategy

The performance of the proposed algorithm is evaluated by a video-based evaluation strategy, which comprises three parts. First, the positions of objects in the stimuli are detected. In the second part, a model is proposed where the coordinates of the eye-tracking signal are related to the movements of the detected objects. Finally, performance measures are calculated by comparing the intervals where the eye-tracking signal is moving close to and in alignment with a moving object, to the intervals where the proposed algorithm detects smooth pursuit movements. These three parts are in the following described in detail.

Figure 2: A frame from the video stimuli.

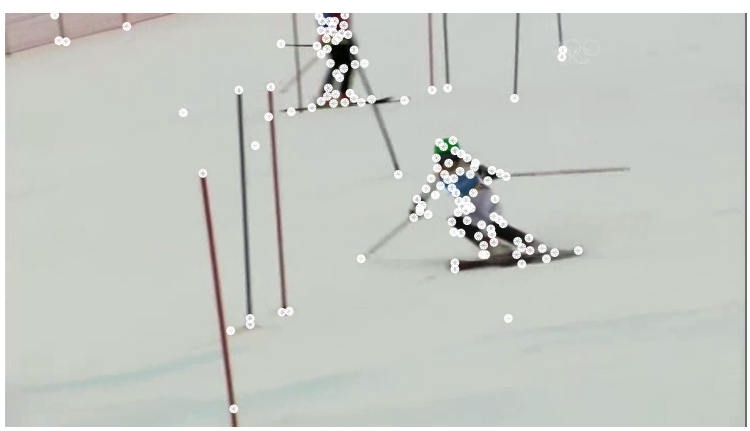


Figure 3: A frame where detected feature points are marked.

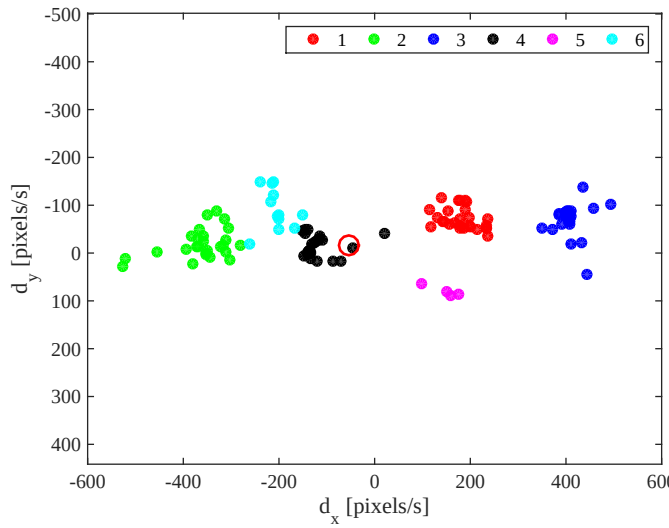


Figure 4: Overview of the clustering for the sample-to-sample velocities of the tracks into 6 clusters. The red circle marks the velocity of the eye.

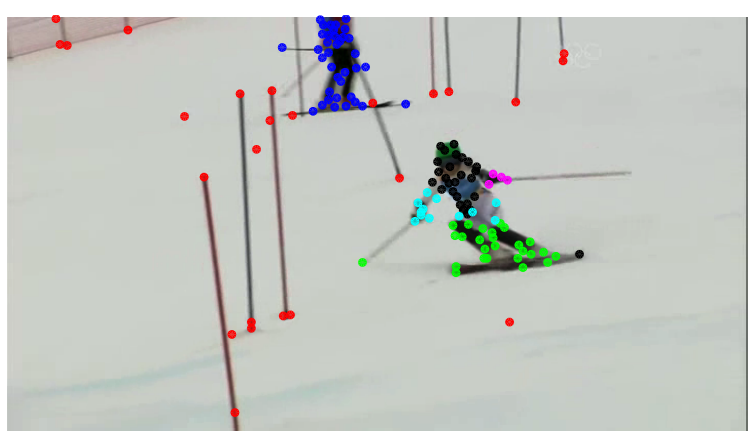


Figure 5: A frame where detected feature points are marked with respect to which cluster they belong to.

Automatic detection of objects in video

For video stimuli, the positions of the moving objects need to be detected and tracked for each frame. In order to determine the trajectories of moving objects and possibly also of the background of the video, feature points are extracted [22], see Fig. 2 for an example of a frame and Fig. 3 for extracted feature points. The extracted feature points are then connected between frames into tracks [23]. The velocity, v, and the direction δ of the tracks between two consecutive frames are calculated as

$$d_{xp}(n) = \frac{x_p(n+1) - x_p(n)}{\Delta t} \tag{10}$$

$$d_{yp}(n) = \frac{y_p(n+1) - y_p(n)}{\Delta t} \tag{11}$$

$$v_p(n) = \sqrt{d_{xp}(n)^2 + d_{yp}(n)^2} \tag{12}$$

$$\delta_p(n) = \arctan \frac{d_{yp}(n)}{d_{xp}(n)} \tag{13}$$

where $d_{xp}(n)$ and $d_{yp}(n)$ are the sample-to-sample velocities in the x- and y- directions, for track p and in frame n, and Δt is the time between two frames in the video.

Tracks that move in similar directions and speeds are grouped together into clusters using the k-means method as shown in Fig. 4. Since the number of objects in each frame is unknown, the number of clusters $k = 1, 2, ..., 6$ are tested using the Calinski-Harabasz criterion in order to find the optimal number of clusters, (see the Matlab 2014b statistic and machine learning toolbox). The tracks belonging to one cluster forms a detected object. Figure 5 shows one frame with the clustered tracks marked according to the clusters in Fig. 4.

Video-gaze model

A video-gaze model is introduced to indicate whether the positions of the eye-tracking signals are moving close to and in alignment with a detected video object. The model consists of four requirements that need to be satisfied for both eyes:

1. The detected object is classified as moving.

2. The eye-tracking signal has moved.

3. The velocity and the direction of the eye-tracking signal match with the velocity and direction of a detected object.

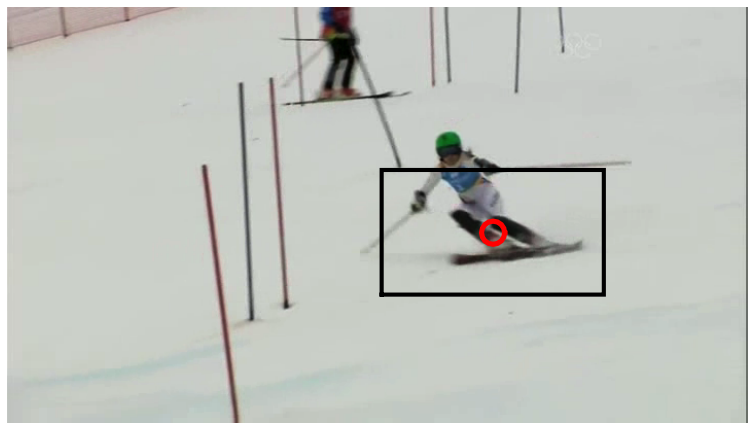


Figure 6: A frame with eye-tracking data (red circle) together with the region of interest (black square).

4. The position of the eye-tracking signal is close to the area covered by the detected object.

The detected objects in the video are classified as moving if $v_p(n) > 10$ pixels/s (requirement 1). The second requirement is satisfied if $v_e > 15$ pixels/s, where v_e is the mean velocity of the eye-tracking signal calculated for a window with length 100 ms. For the third requirement, the cluster that is closest to the velocity of the eye-tracking signal in the velocity domain is identified, see Fig. 4. The feature points that belong to that cluster are mapped back to the frame and the positions of the feature points are compared to the positions of the eye-tracking signal in the frame. A rectangular region of interest is centered around the most recent eye coordinate. The height and width of the region of interest is 30% of the resolution of the frame. The gaze coordinates together with the region of interest are shown in Fig. 6. When the four requirements are fulfilled, a Video-Gaze Movement (VGM) is indicated. Intervals indicated as a VGM are likely to contain smooth pursuit movements, but may also contain small proportions of other types of eye movements. If a VGM is not indicated, it corresponds to that a smooth pursuit movement cannot have been performed.

Performance evaluation measures

In order to evaluate the performance of the proposed algorithm the following parameters are calculated: percentage of smooth pursuit movements, percentage of fixations, percentage of correct smooth pursuit movements, percentage of incorrect smooth pursuit movements, and a balanced performance measure. The parameters are calculated for the inter-saccadic intervals during which the proposed binocular

algorithm uses both eyes. The percentage of smooth pursuit movements, P_{SP}, is calculated as

$$P_{SP} = \frac{N_{SP}}{N_{ISI}} \tag{14}$$

where N_{SP} is the total number of samples detected as smooth pursuit movements and N_{ISI} is the total number of samples of the inter-saccadic intervals. The percentage of fixations, P_F, is calculated as

$$P_F = \frac{N_F}{N_{ISI}} \tag{15}$$

where N_F is the total number of samples detected as fixations. The percentage of correct smooth pursuit movements, P_C, is calculated as

$$P_C = \frac{N_C}{N_{ISI}} \tag{16}$$

where N_C are the total number of samples detected as smooth pursuit movements and indicated as VGM by the video-gaze model. The percentage of incorrect smooth pursuit movements, P_{IC}, is calculated as

$$P_{IC} = \frac{N_{IC}}{N_{ISI}} \tag{17}$$

where N_{IC} are the total number of samples detected as smooth pursuit movements and not indicated as VGM by the video-gaze model.

The balanced performance measure, B, is calculated as the mean of P_F for image stimuli and P_C for moving dot stimuli. The balanced performance measure indicates the algorithm's ability to detect few falsely detected smooth pursuit movements in image stimuli and at the same time a high rate of correctly detected smooth pursuit movements in moving dot stimuli. A value close to 100 is desired.

3 Experiment and database

The eye-tracking signals used for the evaluation of the proposed algorithm were recorded during two separate experiments. In both experiments a Hi-speed eye-tracker from SMI (SensoMotoric Instrument, Teltow, Germany), with a sampling frequency of 500 Hz, was used. A detailed description of the first experiment is given in [20]. In this work, the eye-tracking signals recorded with images and moving dot stimuli in [20], were used. A subset of the signals was manually annotated.

The second experiment had 21 participants (four female), with a mean age 32.9 ($SD = 7$) years. Binocular eye movements were recorded at 500 Hz with the Hi-speed 1250 system and iView X (v. 2.8.26) from SensoMotoric instruments (SMI).

Table 3: Proportion of moving objects in video clips with static camera (1-7) and moving camera (8-14).

Video Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**
% moving content	91.6	99.6	99.7	99.5	90.1	77.1	98.3
Video Number	**8**	**9**	**10**	**11**	**12**	**13**	**14**
% moving content	93.9	97.1	80.3	99.7	72.8	99.8	95.7

Table 4: Settings for intrinsic parameters for the proposed binocular detection algorithm.

Parameter	**Value**	**Description**
η_{SA}	$0.75°$	Minimum saccade amplitude
η_{SI}	30%	Maximum proportion of residual signal
η_S	0.8	Regulation of amount of high frequency content
γ_{max}	$\frac{\pi}{5}$	Maximum size of each cluster
α_T	$\frac{\pi}{2}$	Maximum angle between neighboring clusters
t_{minFix}	50 ms	Minimum duration of a fixation
t_{minSmp}	60 ms	Minimum duration of a smooth pursuit

Stimuli were presented with Experiment Center v. 3.5.101 on an Asus VG248QE screen (53.2 x 30.0 cm) with a resolution of 1920 x 1080 pixels and refresh rate of 144 Hz.

The participants were seated and asked to place their head in the eye tracker. The head was supported by a chin- and forehead rest. The viewing distance to the screen was 70 cm. A 13 point calibration was performed followed by a four point validation of the calibration. The average accuracy reported by Experiment Center was $0.25°$ and $0.38°$, for the horizontal and the vertical directions, respectively.

In this experiment, short video clips were used as stimuli. The stimuli were both material from the benchmark data described in [24] and video clips downloaded from `http://pi4.informatik.uni-mannheim.de/~kiess/test_sequences/download/`. The video clips contained both static and moving camera/background. All video clips contained objects that moved most of the time, see Table 3. The participants were instructed to follow the moving objects as closely as possible.

The recorded signals were divided into a development database and a test database. The development database was used during the development and implementation of the algorithm, while the test database was used only for evaluation.

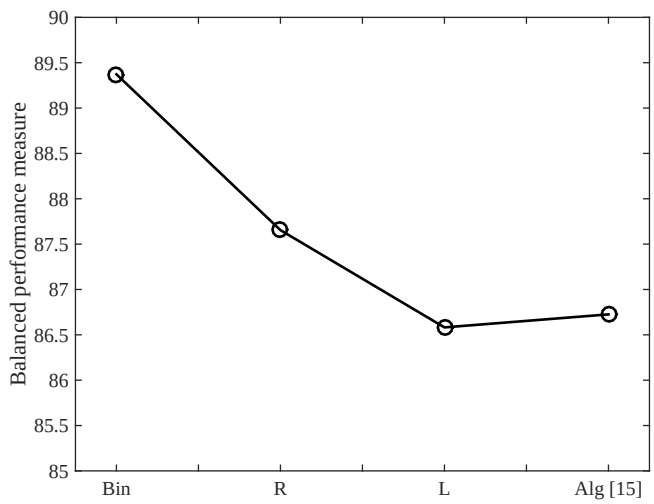


Figure 7: The balanced performance measure, mean of the percentage of correctly detected smooth pursuit movements in moving dot and percentage of fixations in images, for the proposed binocular algorithm (Bin), the proposed monocular right (R), the proposed monocular left (L), and the algorithm in [15].

4 Results

The parameters of the proposed algorithm are found in Table 4. The parameters were adjusted based on the development part of the database. The inter-saccadic intervals were generated using the algorithm in [20].

4.1 Binocular event detection algorithm

The performance of the proposed algorithm is evaluated by calculating the percentages of detected fixations and smooth pursuit movements in image-, video-, and moving dot-stimuli, respectively. In addition, the percentages of correct and incorrect smooth pursuit movements according to the video-gaze model are calculated. Three versions of the proposed algorithm (binocular, monocular right eye, and monocular left eye) are compared to the algorithm in [15], which is a state-of-the-art algorithm proven to outperform earlier smooth pursuit detection algorithms, (cf. [15]). The results of the four compared algorithms are shown in Tables 5–8. For image stimuli, the ideal results are 0% detected smooth pursuit movements and 100% detected fixations in the inter-saccadic intervals. The binocular version of the proposed algorithm outperforms the other algorithms with 1.7% detected smooth pursuit movements and 98.3% detected fixations, compared to values around $5-9\%$ and $91-95\%$, detected smooth pursuit and detected fixations, respectively for the three other algorithms. Since there are no moving objects in image stimuli, all detected smooth pursuit movements are incorrect.

For moving dot stimuli, there is a dot moving 100% of the time. In Fig. 8a, the percentage of time marked as VGM by the video-gaze model is shown for four different speeds of the moving dot. Between $80-90\%$ VGM are indicated which can be compared to the results in Table 6, where the proposed algorithm, both the monocular and binocular versions, detects a larger amount of smooth pursuit movements than the algorithm in [15], $83-85\%$ compared to 81%. The percentage of correct smooth pursuit movements for the proposed binocular algorithm is 80%. See Fig. 8a for a comparison between different speeds of the moving target. The balanced performance, calculated as the mean of the percentage of correctly detected smooth pursuit movements for moving dot stimuli and the percentage of correctly detected fixations for image stimuli, is shown in Fig. 7. In summary, the binocular version of the proposed algorithm detects a large amount of smooth pursuit movements for moving dots stimuli and at the same time decreases the percentage of false smooth pursuit detections for image stimuli.

For video stimuli, the maximal percentages of detected smooth pursuit movements depends on the percentage of time that moving objects are present in the video stimuli. For the video clips used in this study, the percentage of time with moving objects vary between $72-100\%$, where the calculation also includes moving

Table 5: Results for the eye-tracking signals recorded with image stimuli for the test database (development database).

Algorithm	% **smooth pursuit**	% **correct smooth pursuit**	% **incorrect smooth pursuit**	% **fixation**
Proposed (Bin)	1.7 (4.1)	0.0 (0.0)	1.7 (4.1)	98.3 (95.9)
Proposed (Mono R)	6.9 (7.5)	0.0 (0.0)	6.9 (7.5)	93.1 (92.5)
Proposed (Mono L)	8.8 (7.7)	0.0 (0.0)	8.8 (7.7)	91.2 (92.3)
Algorithm in [15]	4.5 (5.7)	0.0 (0.0)	4.5 (5.7)	95.5 (94.3)

Table 6: Results for the eye-tracking signals recorded with moving dot stimuli for the test database (development database).

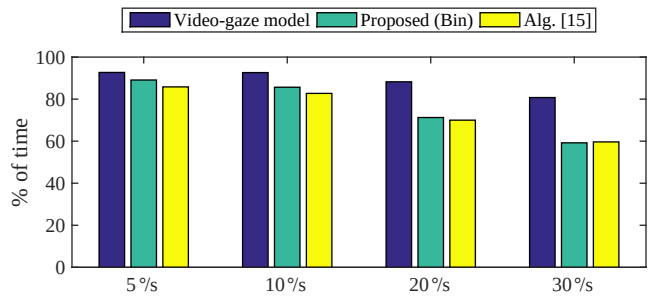
Algorithm	% **smooth pursuit**	% **correct smooth pursuit**	% **incorrect smooth pursuit**	% **fixation**
Proposed (Bin)	83.1 (77.6)	80.4 (74.7)	2.7 (2.8)	16.9 (22.4)
Proposed (Mono R)	85.4 (79.9)	82.2 (76.7)	3.2 (3.2)	14.6 (20.1)
Proposed (Mono L)	85.1 (80.5)	82.0 (77.0)	3.1 (3.5)	14.9 (19.5)
Algorithm in [15]	80.6 (73.4)	77.9 (70.8)	2.6 (2.6)	19.4 (26.6)

Table 7: Results for the eye-tracking signals recorded with video stimuli with static camera for the test database (development database).

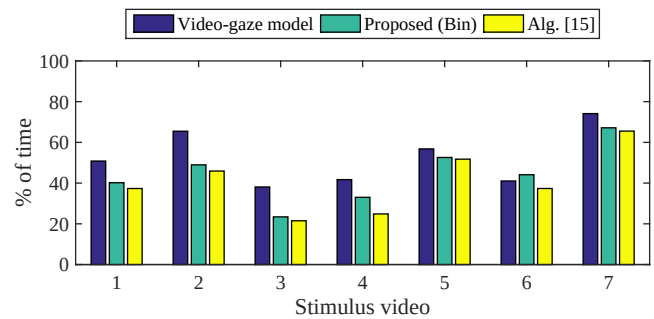
Algorithm	% **smooth pursuit**	% **correct smooth pursuit**	% **incorrect smooth pursuit**	% **fixation**
Proposed (Bin)	47.7 (58.9)	39.2 (48.2)	8.5 (10.6)	52.3 (41.1)
Proposed (Mono R)	52.0 (62.8)	41.5 (49.9)	10.4 (12.9)	48.0 (37.2)
Proposed (Mono L)	51.3 (62.5)	41.4 (50.1)	9.9 (12.5)	48.7 (37.5)
Algorithm in [15]	45.8 (55.7)	37.6 (45.8)	8.2 (9.9)	54.2 (44.3)

Table 8: Results from the signals recorded with video stimuli with moving camera for the test database (development database).

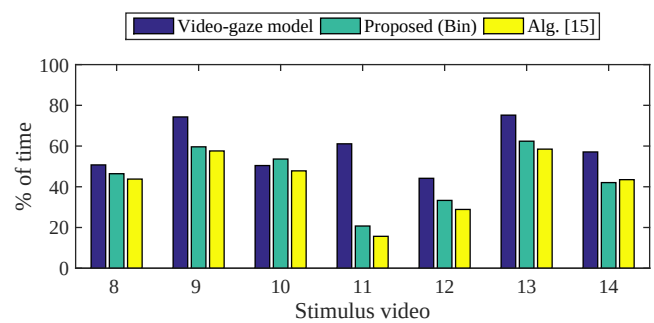
Algorithm	% **smooth pursuit**	% **correct smooth pursuit**	% **incorrect smooth pursuit**	% **fixation**
Proposed (Bin)	42.5 (52.6)	32.3 (41.6)	10.2 (11.0)	57.5 (47.4)
Proposed (Mono R)	47.9 (57.0)	34.8 (43.7)	13.0 (13.3)	52.1 (43.0)
Proposed (Mono L)	49.1 (58.4)	35.6 (44.4)	13.6 (14.0)	50.9 (41.6)
Algorithm in [15]	39.0 (48.2)	30.3 (38.8)	8.7 (9.4)	61.0 (51.8)



(a) Moving dot stimuli.



(b) Video clips with static camera.



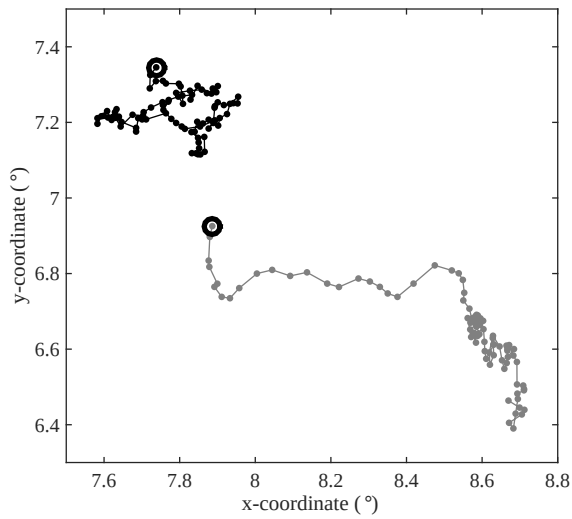
(c) Video clips with moving camera.

Figure 8: Percentage of time indicated as VGM by the video-gaze model and percentage of time in smooth pursuit movements for the proposed algorithm and the algorithm in [15].

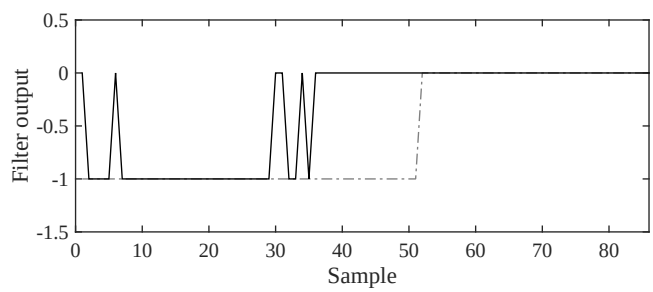
background/moving camera, see Table 3. In order for an object to be considered as moving, the sample-to-sample velocity, $v_i(n)$, must be larger than 10 pixels/s. The percentages of time that the video-gaze model has marked as VGM together with the percentages of detected smooth pursuit movements by the binocular version of the proposed algorithm, and the algorithm in [15], are shown in Figs. 8b– 8c. Fig. 8b shows the percentages for video clips with static stimuli and Fig. 8c shows the percentages for video clips with a moving camera or a moving background. For a majority of the video clips, independent of whether the camera is moving or not, the percentage of time indicated as VGM by the video-gaze model is larger compared to that of the event detection algorithms. In general, the proposed binocular algorithm detects higher percentages of smooth pursuit movements than the algorithm in [15], see also Tables 7–8. For video stimuli, the percentage of correct smooth pursuit movements, i.e., the amount of smooth pursuit detections which are in agreement with the video-gaze model, are around 40% for static camera and around 35% for moving camera. For detection of smooth pursuit movements in videos with moving camera, the proportion of incorrectly detected smooth pursuit movements is larger than for other types of stimuli. The monocular versions of the proposed algorithm detect the largest amount of smooth pursuit movements, but at the cost of more incorrectly detected smooth pursuit movements.

4.2 Synchronization between the two eyes

An advantage of using binocular information in the event detection algorithm is that the temporal alignment between the positions of the two eye-tracking signals can be measured and compared. An example of an inter-saccadic interval recorded during image viewing is shown in Fig. 9a. This example shows how different the eye-tracking signals acquired from the left and the right eyes may be over shorter periods of time. In Fig. 9b, the outputs from the two synchronization filters used in the proposed binocular algorithm are shown. The filters measure if the positions of the eye-tracking signals from the two eyes are temporally aligned. An output of -1 means that the signals are unsynchronized, while an output of 0 means that signals are synchronized. The two output signals show that the eye-tracking signals are not synchronized until after the first $30-40$ samples. In the proposed binocular algorithm, this feature is used to promote fixations and prevent that vergence-like movements, similar to the case shown in Fig. 9a, may be confused with smooth pursuit movements.



(a) Binocular eye-tracking data of an inter-saccadic interval recorded during image stimuli. The right eye is shown in grey and the left eye in black. The black circles show the start of the inter-saccadic interval. In the beginning of the interval the signals from the two eyes move differently, and are deemed as moving unsynchronized.



(b) Output signals from the two synchronization filters. The solid line (-) is the output from the 50 ms long filter and the dash-dotted line (.-) is the output from the 80 ms long filter, both filters are described in Table 1.

Figure 9: An example of binocular eye-tracking data (a) and the corresponding output signals from the synchronization filters (b).

4.3 Quality assessment

In the first part of the proposed algorithm quality assessment of the inter-saccadic intervals was performed. Due to poor quality, 4.6%, 9.7%, 4.6%, and 2.3%, of the data were rejected, i.e., data were deemed to have poor quality in both eyes, for images-, moving dot-, static camera, and moving camera stimuli, respectively.

5 Discussion

Experiments where binocular eye-tracking signals are recorded from participants viewing dynamic stimuli have become increasingly common. Few event detection algorithms, however, take into account that such signals contain smooth pursuit movements. Moreover, information from both eyes are rarely used to make robust decisions about when a specific event occurs. We propose a binocular event detection algorithm based on directional clustering of the eye-tracking signal using both spatial and temporal filters. The proposed algorithm was developed with the idea that signals from two eyes contain more information about the performed type of eye movement than a signal from only one of the eyes. Tables 5 and 6, together with Fig. 7, show that the binocular version of the proposed algorithm provides the best balance between the percentage of correctly detected smooth pursuit movements and percentage of correctly detected fixations. The monocular versions of the proposed algorithm detect more correct smooth pursuit movements but also much more incorrectly detected smooth pursuit movements. Since the binocular version of the proposed algorithm requires that the two eyes are synchronized during smooth pursuit movements, its performance is more robust with fewer incorrectly detected smooth pursuit movements. This is especially true for the image stimuli as seen in Table 5, where the binocular version of the proposed algorithm detects 1.7% smooth pursuit movements. These detections are smooth pursuit like movements that could for instance be due to post-saccadic drift that occurs in both eyes at the same time or due to changes in pupil size causing drift in the eye-tracking signal [25].

It should also be pointed out that movements were taking place $70 - 100\%$ of the time in the videos (see Table 3), and that the participants were asked to follow the moving objects to the largest extent possible. The results of Fig. 8 verify that the instruction was followed; the video-gaze model indicates VGM $60 - 80\%$ of the time. In Fig. 8, some cases show a large gap between the VGM indicated by the video-gaze model and the percentage smooth pursuit movements detected by the algorithms solely based on the eye-tracking signals. One reason is that the videos, e.g., number 11, contain slow moving objects, where the corresponding eye movements move so slowly that they are not always considered as smooth pursuit movements by the algorithms.

The proposed algorithm is evaluated using a novel video-based evaluation strategy, which uses information automatically extracted from the stimuli videos. The logic behind the strategy is that smooth pursuit is only possible when there is a moving object to follow, (see [26]); if the eye-tracking signal is aligned with a moving object in terms of speed, direction and position, it is therefore very likely that the participant is pursuing the object. This type of automatic evaluation strategy may not be as accurate as manual annotations, but gives a general and objective picture of the performance of the evaluated algorithm. The main advantage of using automatic evaluation compared to manual annotations is that significantly larger amounts of data can be used in the evaluation process. Manual annotations are very time consuming and not a practical solution for large data sets. With the proposed video-based evaluation strategy, longer videos can be used and a larger number of participants can be included when the performance of a new algorithm is evaluated.

Despite its clear logic, the proposed video-based evaluation strategy is new, and lacks objective validation. To address this issue the VGM indicated by the video-gaze model are compared to smooth pursuit movements annotated manually in data recorded during moving dot stimuli. In this comparison, the sensitivity and the specificity were determined to 0.87 and 0.58, respectively. The rather low value of specificity indicates that the video-gaze model cannot distinguish between fixations and smooth pursuit movements when the eye-tracking signal is close to a slowly moving object.

In this work, the video-gaze model is used to confirm or reject smooth pursuit detections for the purpose of performance evaluation. It would be possible to instead incorporate such information in the event detector in order to provide even better fixation and smooth pursuit discrimination. Such strategy may be particularly well-suited for mobile eye-trackers where a scene camera is used to record the scene which the user is looking at. This is, however, outside the scope of this paper.

In the present paper, the eye-tracking signals are recorded for participants with normal or corrected-to-normal vision with good binocular coordination. The proposed binocular algorithm has therefore not been tested on participants with poor binocular control. Future studies will show if the requirements of synchronization will work also for this group of participants.

6 Conclusions

An algorithm for the detection of fixations and smooth pursuit movements using binocular eye-tracking data is proposed. Using binocular information is most advantageous in image stimuli, where vergence or drift-like movements otherwise may be confused with smooth pursuit movements. The proposed binocular algorithm detects a larger amount of smooth pursuit movements in moving dot stimuli than previous algorithm, without increasing the percentage of falsely smooth pursuit de-

tections for image stimuli. The proposed algorithm is evaluated using a novel video-based evaluation strategy based on automatically detected moving objects in video stimuli. Compared to manual annotation of data, this makes it practically feasible to evaluate larger amounts of data.

Acknowledgment

This work was supported by the Strategic Research Project eSSENCE, funded by the Swedish Research Council. Data were recorded in the Lund University Humanities Laboratory.

References

[1] K. Rayner, K. Chace, T. Slattery, and J. Ashby, "Eye movements as reflections of comprehension processes in reading," *Scientific Studies of Reading*, vol. 10, no. 3, pp. 241–255, 2009.

[2] K. Rayner, "Eye movements and attention in reading, scene perception, and visual search," *The Quarterly Journal of Experimental Psychology*, vol. 62, no. 8, pp. 1457–1506, 2009.

[3] K.-M. Flechtner, B. Steinacher, R. Sauer, and A. Mackert, "Smooth pursuit eye movements in schizophrenia and affective disorder," *Psychological Medicine*, vol. 27, pp. 1411–1419, 1997.

[4] G. F. Eden, J. F. Stein, H. M. Wood, and F. B. Wood, "Differences in eye movements and reading problems in dyslexic and normal children," *Vision Research*, vol. 34, no. 10, pp. 1345–1358, 1994.

[5] R. S. Allison, M. Eizenman, and B. S. Cheung, "Combined head and eye tracking system for dynamic testing of the vestibular system," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 11, pp. 1073–1082, 1996.

[6] R. Leigh and D. Zee, *The Neurology of Eye Movements*. Oxford University Press, 2006.

[7] C. Meyer, A. Lasker, and D. Robinson, "The upper limit of human smooth pursuit.," *Vision Research*, vol. 25, no. 4, pp. 561–563, 1985.

[8] A. J. Martins, E. Kowler, and C. Palmer, "Smooth pursuit of small-amplitude sinusoidal motion," *Journal of the Optical Society of America A*, vol. 2, no. 2, pp. 234–242, 1985.

[9] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2011.

[10] R. van der Lans, M. Wedel, and R. Pieters, "Defining eye-fixation sequence across individuals and tasks: the binocular-individual threshold (bit) algorithm," *Behavior Research Methods*, vol. 43, no. 1, pp. 239–257, 2011.

[11] J. A. Kirkby, L. A. D. Webster, H. I. Blythe, and S. P. Liversedge, "Binocular coordination during reading and non-reading tasks," *Psychological Bulletin*, vol. 134, no. 5, pp. 742–763, 2008.

[12] M. Versino, O. Hurko, and D. S. Zee, "Disorders of binocular control of eye movements in patients with cerebellar dysfunction," *Brain*, vol. 119, no. 6, pp. 1933–1950, 1996.

[13] A. Duchowski, E. Medlin, N. Cournia, H. Murphy, A. Gramopadhye, S. Nair, J. Vorah, and B. Melloy, "3-d eye movement analysis," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 573–591, 2002.

[14] R. Engbert and R. Kliegl, "Microsaccades uncover the orientation of covert attention," *Vision Research*, vol. 43, no. 9, pp. 1035–1045, 2003.

[15] L. Larsson, M. Nyström, R. Andersson, and M. Stridh, "Detection of fixations and smooth pursuit movements in high-speed eye-tracking data," *Biomedical Signal Processing and Control*, vol. 18, pp. 145–152, 2015.

[16] J. Otero-Millan, J. L. A. Castro, S. L. Macknik, and S. Martinez-Conde, "Unsupervised clustering method to detect microsaccades," *Journal of vision*, vol. 14, no. 2, p. 18, 2014.

[17] O. Komogortsev, D. Gobert, S. Jayarathna, D. Hyong Koh, and S. Gowda, "Standardization of automated analyses of oculomotor fixation and saccadic behaviors," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2635–2645, 2010.

[18] O. Komogortsev and A. Karpov, "Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades," *Behavior Research Methods*, vol. 45, no. 1, pp. 203–215, 2013.

[19] K. Essig, N. Sand, T. Schack, J. Künsemöller, M. Weigelt, and H. Ritter, "Fully-automatic annotation of scene videos: Establish eye tracking effectively in various industrial applications," in *Proceedings of SICE Annual Conference 2010*, pp. 3304–3307, 2010.

[20] L. Larsson, M. Nyström, and M. Stridh, "Detection of saccades and post-saccadic oscillations in the presence of smooth pursuit," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 9, pp. 2484–2493, 2013.

[21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley-Interscience, 2001.

[22] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, IEEE, 1994.

[23] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, pp. 1–10, 2001.

[24] K. Kurzhals, C. F. Bopp, J. Bässler, F. Ebinger, and D. Weiskopf, "Benchmark data for evaluating visualization and analysis techniques for eye tracking for video stimuli," in *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pp. 54–60, ACM, 2014.

[25] J. Drewes, W. Zhu, Y. Hu, and X. Hu, "Smaller is better: Drift in gaze measurements due to pupil dynamics," *PLoS ONE*, vol. 9, no. 10, p. e111197, 2014.

[26] M. J. Steinbach, "Pursuing the perceptual rather than the retinal stimulus," *Vision Research*, vol. 16, no. 12, pp. 1371–1376, 1976.

Paper IV

Head Movement Compensation and Multi-Modal Event Detection for Mobile Eye-Trackers

Abstract

The complexity of analyzing eye-tracking signals increases as eye-trackers become more mobile. The signals from a mobile eye-tracker are recorded in relation to the head coordinate system and when the head and body move, the recorded eye-tracking signal is influenced by these movements which render the subsequent event detection difficult. The purpose of the present paper is to develop a method that performs robust event detection in signals recorded using a mobile eye-tracker. The proposed method performs compensation of head movements recorded using an inertial measurement unit (IMU) and employs a multi-modal event detection algorithm. The event detection algorithm is based on the head compensated eye-tracking signal combined with information about detected video objects extracted from the scene camera of the mobile eye-tracker. The proposed method for head compensation decreases the standard deviation during intervals of fixations from 8° to 3.3° for eye-tracking signals recorded during large head movements. The multi-modal event detection algorithm outperforms both the I-VDT and the built-in-algorithm of the mobile eye-tracker with an average balanced accuracy, calculated over all types of eye movements, of 0.90, compared to 0.85 and 0.75, respectively for the compared algorithms.

Based on:
Linnéa Larsson, Andrea Schwaller, Marcus Nyström, and Martin Stridh, "Head Movement Compensation and Multi-Modal Event Detection for Mobile Eye-Trackers,"
Submitted for publication.

1 Introduction

In recent years the popularity of mobile eye-trackers has increased drastically. This is partly due to that electronics have been made smaller and thereby mobile eye-trackers have become lighter; from a camera mounted on top of a bicycle helmet to a neat pair of glasses. Compared to stationary eye-tracking performed in the laboratory in front of a computer screen, mobile eye-tracking allows the recording of eye movements in natural environments. As a result, new areas are developing, e.g., decision making in the supermarket [1], package design [2], and sport activities [3]. However, the increase in mobility when recording in natural situations comes with increased difficulties in the analysis and interpretation of the recorded signals. Due to movements of the head, the body, and the environment during the recording, the complexity of the analysis of the eye-tracking signal is significantly increased compared to the analysis of eye-tracking signals recorded using a stationary setup. Consequently, algorithms developed for stationary eye-tracking, when the stimuli are presented on a computer screen, are not applicable without first removing parts of the eye-tracking signal that are caused by movements of the body and/or the head. In addition to the three most common types of eye movements, fixations, saccades, and smooth pursuit movements, vestibular ocular reflexes (VOR) occur when eye movements are recorded using a mobile eye-tracker where the position of the head is not fixed. Vestibular ocular reflexes are movements that the eye performs as a compensatory movement in the opposite direction of, e.g., a rotation of the head [4].

Typically, when recording movements of the eye using a video based eye-tracker, the *eye-in-head motion* is recorded, i.e., the movement of the eye with respect to the head. When the head is still, the eye-in-head motion is equivalent to the *eye-in-space motion*, i.e., the movement of the eye with respect to the world. The eye-in-space motion is preferable since it describes the gaze direction in space. In mobile eye-tracking, the eye-in-head positions are usually given in the coordinate system of the scene camera, which is changing in relation to how the scene camera is moving with the head and the body. The eye-in-space motion can be achieved by compensating for the movements of the head and the body. One strategy is to use information from the scene camera to compensate for head and body movements [5, 6]. In [5], an additional camera was used to film the environment and compensate for motion, while in [6] the ego-motion was calculated from the scene video of the eye-tracker. There are, however, situations where the scene video does not provide information accurate enough to estimate the ego-motion, e.g., caused by blurring of the image during fast motion [7], or when there is not enough texture in the image to detect changes [6]. In order to overcome these problems, a motion capture system has been used to track the head and body movements in combination with an eye-tracker [8, 9, 10, 11]. The purpose was to represent the eye-tracking signal in the same world coordinate system as the objects in the surroundings. The problem with

these systems is that they are limited to the range of the motion capture system.

In this paper, we propose a method for compensation of head movements in the recorded eye-tracking data where head movements are recorded using an inertial measurement unit, IMU. An IMU consists of an accelerometer, a gyroscope, and a magnetometer, which is used to estimate its orientation. The advantages of using an IMU to record head movements are that the recordings can be performed without limitations of the recording area and that it is independent of the quality of the images from the scene camera. As a first step towards analyzing eye-tracking signals recorded with a mobile eye-tracker for any type of head and body movements, the proposed method is developed and evaluated on eye-tracking signals recorded using a mobile eye-tracker from participants watching stimuli on a big screen and where no positional changes of the entire body are allowed.

The main purpose of performing head movement compensation in mobile eye-tracking is to be able to perform robust and reliable event detection, i.e, to discriminate between the different types of eye movements in the recorded data. Due to the lack of reliable event detection algorithms for mobile eye-trackers, researchers often have to perform the event detection manually by inspecting the scene video together with the eye-tracking data frame-by-frame [1, 12].

A second novelty in this work is that we propose an event detection algorithm that is based on the head compensated eye-tracking signals combined with information about moving objects extracted from the scene video. Earlier approaches for detection of smooth pursuit movements in mobile eye-tracking signals have used k-means clustering [13], and a Bayesian mixture model [14]. None of these methods has used additional information extracted from the scene video in order to support the classification of smooth pursuit movements. Information about objects extracted from the scene video has previously been used for fixation detection [12] and for scan path comparison [15].

The present paper is outlined as follows: The proposed method for head movement compensation and multi-modal event detection is presented in Section 2. The experimental setup and the data recording procedure are described in Section 3. The results of the work are presented in Section 4, and finally, in Section 5, the proposed method and results are discussed.

2 Methods

The proposed method comprises five parts, see subsections 2.1–2.5. In the first part, the method for tracking of head movements is described and in the second part, head movement compensation in the eye-tracking signal is performed, i.e., the eye-in-space signal is estimated. The third part describes the approach for tracking of moving objects in the scene video, and in the fourth part, the trajectories of these detected objects are head movement compensated, i.e., the object-in-space signal

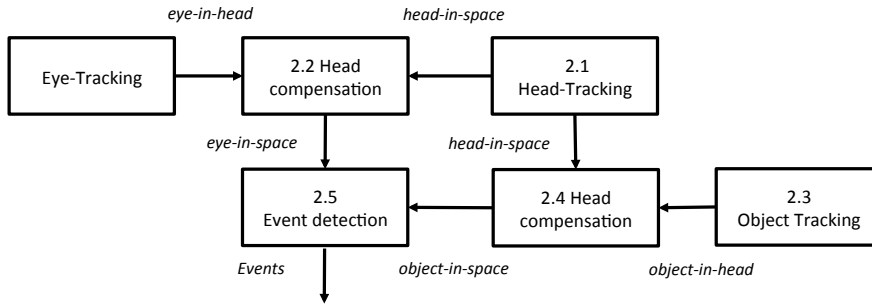


Figure 1: Overview of the proposed method.

is estimated. The fifth part describes the multi-modal event detection algorithm that combines the estimated eye-in-space signal and the estimated object-in-space signal for classification of saccades, fixations, and smooth pursuit movements. An overview of the proposed method is shown in Fig. 1.

2.1 Head Tracking

When recording eye movements using a mobile eye-tracker, the eye-in-head signal is recorded. For a distant target, i.e., a target further away than 1 m [16], the eye-in-space signal, $s_G(n)$, is the sum of the head-in-space signal, $s_H(n)$ and the eye-in-head signal, $s_E(n)$ [4], i.e.,

$$s_G(n) = s_H(n) + s_E(n) \tag{1}$$

where subscript G, H, and E, denote eye-in-space (G), head-in-space (H), and eye-in-head (E) signals. For a near target, the relationship between the eye-in-space signal, the eye-in-head signal, and the head-in-space signal is more complicated. Due to the short distance to the target, the separation between the two eyes as well as the fact that the head and the eyes have different axes of rotation cannot be neglected [4]. In this paper, (1) is used under the assumption that all stimuli are viewed at a far distance and without any translational movements of the body.

In the experimental setup, described in Section 3 and illustrated in Fig. 5, there are two coordinate systems: the world coordinate system and the coordinate system of the scene camera. The origin of the world coordinate system is assumed to be at the center of the participant's head. The stimulus screen is located at a distance d from the participant in a plane that is perpendicular to the Y-axis of the world coordinate system. The signals recorded by the IMU are expressed in Euler angles, ψ, θ, and ϕ, which corresponds to rotations around the Z-, Y-, and X- axes of the

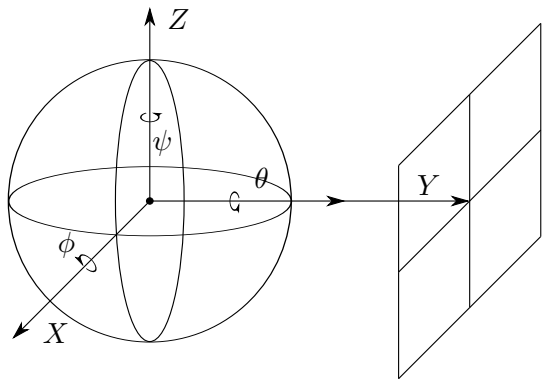


Figure 2: Overview of the world coordinate system, which has its origin at the center of the participant's head. The relation to the stimuli screen is also indicated.

world coordinate system, respectively, see Fig. 2. These rotations can be express as rotations of a heading vector, $\mathbf{v}_H(0) = (0, d, 0)$, between the origin and the plane which coincides with the Y-axis and has length d. Any rotation of the heading vector is expressed by

$$\mathbf{v}_H = \mathbf{R}\mathbf{v}_H(0) \tag{2}$$

where $\mathbf{R}$ is a rotation matrix, consisting of the rotation matrices around each axis.

$$\mathbf{R} = \mathbf{R}_Z(\psi)\mathbf{R}_Y(\theta)\mathbf{R}_X(\phi) \tag{3}$$

$$\mathbf{R}_Z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\mathbf{R}_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

The coordinate system of the scene camera is expressed in pixels and is moving with the head of the participant around the same origin as the world coordinate system. The coordinate system of the scene camera is connected to the world coordinate system through the vector $\mathbf{v}_H$ which points to the center of the scene camera image.

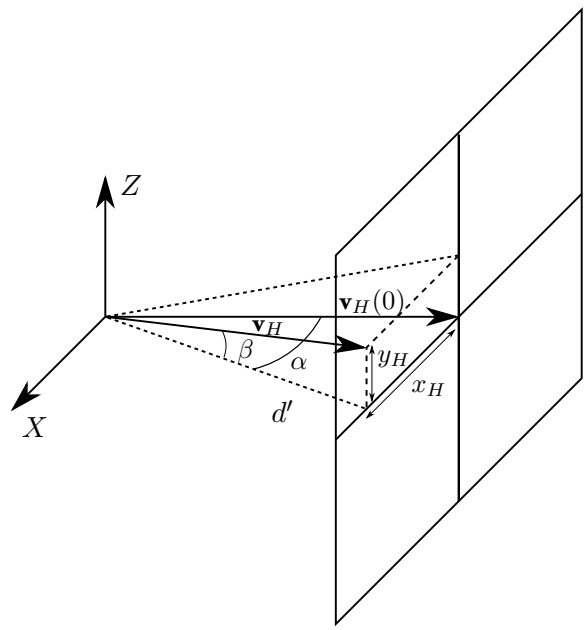


Figure 3: Mapping of the heading vector $\mathbf{v}_H$ to the 2-dimensional plane of the stimuli screen.

In order to be able to perform head movement compensation in the eye-tracking signal that is expressed in the coordinate system of the scene camera, the orientation of the head must be expressed as a coordinate, (x_H, d, y_H), on the screen in the world coordinate system. This is done by mapping $\mathbf{v}_H$ to the 2-dimensional plane of the stimulus screen, see Fig. 3. The mapping is performed separately for x_H and y_H.

A head movement corresponding to the angle α in the XY-plane of the world coordinate system is mapped to the 2-dimensional plane of the stimuli screen and corresponds to a movement x_H, as is shown in Fig. 4. It is described by:

$$\tan \alpha = \frac{x_H}{d} \tag{4}$$

Since the resolution, (x_{max}, y_{max}), of the scene camera, and the maximum angular view of the camera, $(\alpha_{max}, \beta_{max})$, are known from the specifications of the camera, a relationship between these and the distance to the screen can be written as.

$$\tan(\alpha_{max}/2) = \frac{x_{max}/2}{d} \tag{5}$$

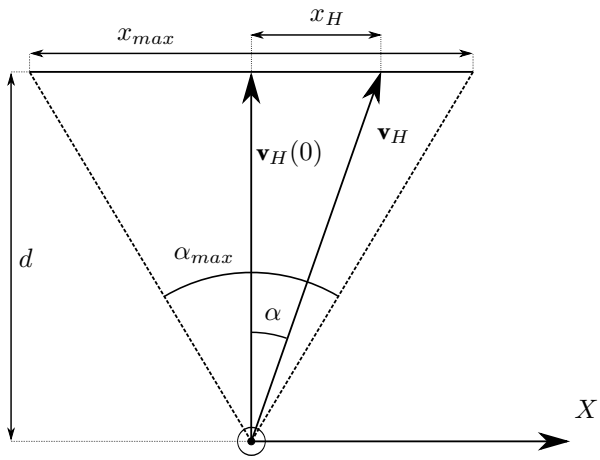


Figure 4: Horizontal mapping of the heading vector $\mathbf{v}_H$ to the 2-dimensional plane of the stimuli screen.

Combining (4) and (5) gives the following expression for x_H

$$x_H = \frac{x_{max}/2}{\tan(\alpha_{max}/2)} \tan(\alpha). \tag{6}$$

Similarly to the horizontal case shown in Fig. 4, a head rotation corresponding to the angle β, in Fig. 3, is mapped to the 2-dimensional plane of the stimuli screen and corresponds to a movement y_H. Note that for the angle β, the distance between the origin and the 2-dimensional plane is d'. The expression for y_H is,

$$\tan\beta = \frac{y_H}{d'}, \qquad \tan(\beta_{max}/2) = \frac{y_{max}/2}{d}, \tag{7}$$

where $d' = d/\cos(\alpha)$, which yields

$$y_H = \frac{y_{max}/2}{\tan(\beta_{max}/2)\cos(\alpha)} \tan(\beta). \tag{8}$$

So far, the relationship between (α, β) and the coordinates (x_H, y_H) has been calculated. In order to perform the mapping from the Euler angles, (ψ, θ, ϕ), reported by the IMU, to the coordinates (x_H, y_H), a relationship between (ψ, θ, ϕ) and (α, β) is needed. This relationship is investigated by expressing $\mathbf{v}_H$ in spherical coordinates.

$$\mathbf{v}_H = \begin{bmatrix} r\sin(\frac{\pi}{2} - \beta)\cos(\frac{\pi}{2} - \alpha) \\ r\sin(\frac{\pi}{2} - \beta)\sin(\frac{\pi}{2} - \alpha) \\ r\cos(\frac{\pi}{2} - \beta) \end{bmatrix}$$

where r is the radius of the sphere. The expression for $\mathbf{v}_H$ in cartesian coordinates is given by

$$\mathbf{v}_H = \mathbf{R}\mathbf{v}_H(0) = \begin{bmatrix} d(\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi) \\ d(\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi) \\ d\cos\theta\sin\phi \end{bmatrix}$$

By combining the two expressions for $\mathbf{v}_H$, the relationships between the angles (α, β) and (ψ, θ, ϕ) are found to be:

$$\alpha = \frac{\pi}{2} - \arctan\left(\frac{\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi}{\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi}\right) \tag{9}$$

$$\beta = \frac{\pi}{2} - \arccos(\cos\theta\sin\phi) \tag{10}$$

By combining (6) and (8) with (9) and (10), the rotations recorded by the IMU are mapped to the 2-dimensional plane of the stimuli screen.

2.2 Compensation of head movements in eye-tracking signals

In order to compensate for head movements in the eye-tracking signal, i.e., estimate the eye-in-space signal, (1) is used for the x- and y-coordinates separately.

$$x_G(n) = x_H(n) + x_E(n) \tag{11}$$

$$y_G(n) = y_H(n) + y_E(n) \tag{12}$$

2.3 Detection of objects in the scene video

In this work, two important types of objects are detected in the scene video of the mobile eye-tracker. First, since the major part of the stimuli in this study consists of black dots moving in different patterns, a simple black and white image analysis algorithm was implemented to detect the black dots in each frame. Secondly, the corners of the projected stimulus screen were also detected, in order to also be able to calculate head movements from the scene video. Each detected black dot, i, is referred to as detected object i, with corresponding coordinates, $(x^i_{obj}(n), y^i_{obj}(n))$, expressed in the coordinate system of the scene camera. In this work, maximally two dots were present at the same time.

2.4 Compensation of head movements for objects detected in the scene video

The coordinates of the detected objects $(x^i_{obj}(n), y^i_{obj}(n))$ are head movement compensated in the same way as the eye-tracking signals, i.e., the object-in-space coordinates, $(x^i_{objS}(n), y^i_{objS}(n))$, are estimated using the following equations for each object i.

$$x^i_{objS}(n) = x_H(n) + x^i_{obj}(n) \tag{13}$$

$$y^i_{objS}(n) = y_H(n) + y^i_{obj}(n) \tag{14}$$

2.5 Multi-modal event detection

The proposed method for detection of saccades, fixations, and smooth pursuit movements, is performed by combining the eye-in-space signals and the object-in-space signals as described below:

Saccade detection and noise detection

This part of the event detector is divided into two steps, where the first step detects saccade candidates in the velocity domain, and in the second step, the acceleration, amplitude, and slope of the saccade candidates are examined in order to distinguish saccades from noise. Saccade candidates are detected by first calculating the sample-to-sample velocities, $v_x(n)$ and $v_y(n)$,

$$v_x(n) = \frac{x_G(n+1) - x_G(n)}{\Delta t} \tag{15}$$

$$v_y(n) = \frac{y_G(n+1) - y_G(n)}{\Delta t} \tag{16}$$

where $x_G(n)$ and $y_G(n)$ are the coordinates of the eye-in-space signal, and Δt is the time between two samples. The sample-to-sample velocities are fed into the algorithm proposed in [17] which estimates velocity thresholds as multiples of the standard deviations of the sample-to-sample velocities in the x- and y-directions, separately. Consecutive samples which have larger velocities than the selected threshold are grouped into saccade candidates. The acceleration, $a(n)$, of the samples belonging to the saccade candidates are calculated as

$$a(n) = \frac{\sqrt{(v_x(n+1) - v_x(n))^2 + (v_y(n+1) - v_y(n))^2}}{\Delta t} \tag{17}$$

The saccade candidates with a peak acceleration larger than η_a, i.e., $\max(a(n)) > \eta_a$, are classified as noise. For the saccade candidates that are not considered as noise, the slopes, k_x and k_y, of the saccade candidates are computed for $x_G(n)$ and $y_G(n)$, separately. In order to calculate the slope, a first order polynomial is fit in the least-square sense between the start and end points of the saccade candidate. The maximum value of $k = \max(k_x, k_y)$ is selected as the slope of the saccade candidate. In order for the saccade candidate to be considered as a saccade, the amplitude A must be larger than η_{SA}. The amplitude is for each saccade candidate calculated as:

$$A = \sqrt{(\max \mathbf{x} - \min \mathbf{x})^2 + (\max \mathbf{y} - \min \mathbf{y})^2} \tag{18}$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors of $x_G(n)$ and $y_G(n)$, respectively, within the saccade candidate. Saccade candidates satisfying both $A > \eta_{SA}$ and $k \geq \eta_k$ are considered as true saccades. The remaining saccade candidates are left with no label. Due to their low amplitudes, they are considered to be too small to be noise.

Discontinuity detection

The stimuli contain dots that abruptly move between different locations on the screen which causes discontinuities in the object-in-space signal. These discontinuities are detected in the object-in-space signal using the algorithm described above for the detection of saccades in the eye-in-space signals.

Fixation and smooth pursuit detection

Since fixations and especially smooth pursuit movements are strongly correlated to the movements of objects in the stimuli, the presence of moving objects in the scene video is utilized for discrimination between fixations and smooth pursuit movements. The fixation and smooth pursuit detection includes the following steps: Directional clustering, Object selection, Binary filters, and Classification.

Directional clustering The directional clustering is based on the method presented in [18]. A brief description is given below: In the inter-saccadic intervals, i.e., intervals between the detected saccades, the sample-to-sample directions, $\alpha(n)$, are calculated for both (x_G, y_G) and (x^i_{objS}, y^i_{objS}). The sample-to-sample directions, $\alpha_G(n)$ and $\alpha^i_{objS}(n)$, respectively, are the angles between the line formed by two consecutive pairs of x- and y- coordinates and the x-axis. The sample-to-sample directions $\alpha_G(n)$ and $\alpha^i_{objS}(n)$ are divided into clusters based on their directions. The clustering is based on the iterative minimum-squared error clustering algorithm [19]. Each sample is assigned to a cluster, and based on the samples of $\alpha(n)$ in each cluster, c, a mean direction, m_c, is calculated.

Object selection In order for detected video objects to support the separation between fixations and smooth pursuit movements, the proximity between an object and the eye-tracking signal is evaluated in terms of position and trajectory, see criteria below. Only objects that move similarly to the corresponding eye-tracking signal and are spatially close, are used. For this purpose, the total mean direction, m_T^{eye}, and the total achieved distance, d_T^{eye}, of the eye-tracking signal, are compared to the total mean directions, $m_T^{objS_i}$, and the total achieved distances, $d_T^{objS_i}$, of the detected objects that are present in the inter-saccadic interval. The total mean direction and the total achieved distance of the inter-saccadic interval for the eye-tracking signal and the objects are calculated as follows: In each inter-saccadic interval, the Euclidean distance, $ED^{eye}(n)$, between consecutive samples in the eye-tracking signal is calculated as:

$$ED^{eye}(n) = \sqrt{d_{x_G}^2(n) + d_{y_G}^2(n)} \tag{19}$$

where $d_{x_G}(n) = x_G(n+1) - x_G(n)$ and $d_{y_G}(n) = y_G(n+1) - y_G(n)$. Based on the directional clustering, each sample and its corresponding $ED^{eye}(n)$ is labeled with a cluster number $c = 1, 2, ..., K$, where K is the number of clusters. All $ED^{eye}(n)$ that belong to the same cluster c, $ED_c^{eye}(l)$, are summed together into d_c^{eye}, which corresponds to the total achieved distance by the samples in cluster c. Thus,

$$d_c^{eye} = \sum_{l=1}^{M_c} ED_c^{eye}(l) \tag{20}$$

where M_c is the number of samples in cluster c. Similarly, for each detected video object i in each inter-saccadic interval, the corresponding Euclidean distance, $ED^{objS_i}(n)$, is calculated as.

$$ED^{objS_i}(n) = \sqrt{d_{x_{objS_i}}^2(n) + d_{y_{objS_i}}^2(n)} \tag{21}$$

where $d_{x_{objS_i}}(n) = x_{objS_i}(n+1) - x_{objS_i}(n)$ and $d_{y_{objS_i}}(n) = y_{objS_i}(n+1) - y_{objS_i}(n)$. Based on the directional clustering, each sample and its corresponding $ED^{objS_i}(n)$ is labeled with a cluster number $c = 1, 2, ..., K$, where K is the number of clusters. All $ED^{objS_i}(n)$ that belong to the same cluster c, $ED_c^{objS_i}(l)$, are summed together,

$$d_c^{objS_i} = \sum_{l=1}^{N_c} ED_c^{objS_i}(l) \tag{22}$$

where N_c is the number of samples in each cluster c. In order to calculate the total achieved distance, d_T^{eye}, each d_c^{eye} is mapped on to the cluster's corresponding mean

direction m_c,

$$d_T^{eye} = \sum_{c=1}^{K} \sqrt{(d_c^{eye} \cos m_c)^2 + (d_c^{eye} \sin m_c)^2} \tag{23}$$

where K is the number of clusters. For each object i, the total achieved distance, $d_T^{objS_i}$, is calculated as.

$$d_T^{objS_i} = \sum_{c=1}^{K} \sqrt{(d_c^{objS_i} \cos m_c)^2 + (d_c^{objS_i} \sin m_c)^2} \tag{24}$$

The total mean direction for the eye-tracking signal, m_T^{eye}, of the inter-saccadic interval, is calculated as

$$x_m^{eye} = \frac{1}{K} \sum_{c=1}^{K} d_c^{eye} \cos m_c \tag{25}$$

$$y_m^{eye} = \frac{1}{K} \sum_{c=1}^{K} d_c^{eye} \sin m_c \tag{26}$$

$$m_T^{eye} = \arctan \frac{y_m^{eye}}{x_m^{eye}}, \tag{27}$$

and the total mean direction for each object, $m_T^{objS_i}$, of the inter-saccadic interval is calculated as

$$x_m^{objS_i} = \frac{1}{K} \sum_{c=1}^{K} d_c^{objS_i} \cos m_c \tag{28}$$

$$y_m^{objS_i} = \frac{1}{K} \sum_{c=1}^{K} d_c^{objS_i} \sin m_c \tag{29}$$

$$m_T^{objS_i} = \arctan \frac{y_m^{objS_i}}{x_m^{objS_i}}. \tag{30}$$

In order for a detected object in the current inter-saccadic interval to be used in the algorithm, it must satisfy the following criteria.

1. Directional criterion: $|m_T^{objS_i} - m_T^{eye}| < \alpha_T$
2. Total distance criterion: $|d_T^{objS_i} - d_T^{eye}| < \eta_T$
3. Spatial criterion: $d_P < \eta_P$

where $d_P = \sqrt{(\bar{x}_G - \bar{x}^i_{objS})^2 + (\bar{y}_G - \bar{y}^i_{objS})^2}$ with $\bar{x}_G$ and $\bar{y}_G$ calculated as the means of $x_G(n)$ and $y_G(n)$, respectively, in the inter-saccadic interval, and $\bar{x}^i_{objS}$ and $\bar{y}^i_{objS}$ calculated as the means of $x^i_{objS}(n)$ and $y^i_{objS}(n)$, respectively, in the inter-saccadic interval. If the three criteria are satisfied for several objects, the object that has the smallest difference in direction is selected. The selected object is used for fixation and smooth pursuit movement detection, by supporting decisions of smooth pursuit movements when the selected object is moving similarly as the eye-tracking signal and by disqualifying smooth pursuit movements when the selected object is not moving.

Binary filters Next, the results of the directional clustering for the eye-tracking signal and the selected video object in each inter-saccadic interval, respectively, are applied to a set of binary filters. These are designed to either emphasize fixations, *fixation filters*, or smooth pursuit movements, *smooth pursuit filters*. A binary filter is a filter with a length and a criterion. If the criterion is fulfilled for fixation filters the output is -1, and if the criterion is fulfilled for smooth pursuit filters, the output is 1. For both types of filters, if the criterion is not fulfilled the output is 0. In this work, three types of binary filters are used: Total distance, Transition, and Synchronization to the selected object. The lengths and criteria for the respective fixation and smooth pursuit filters are listed in Tables 1–2.

The Total distance filter is a filter that, based on the directional clustering, calculates the total achieved distance that the samples within the filter length have moved. For eye-tracking signals, (20) and (23) are used to calculate the total achieved distance of the movement of the samples within the filter length. If an object was selected, (22) and (24) are used to calculate the total achieved distance for that object within the filter length. A long total achieved distance corresponds to that multiple samples are heading in a similar direction which is typical for a smooth pursuit movement. A short total achieved distance corresponds to that samples have moved in different directions which is typical for a fixation. The length of the Total distance filter is adapted to the length of the respective inter-saccadic interval. In Tables 1–2, the length of the filter is given as a percentage of the respective inter-saccadic interval. The Total distance filter aims to reflect the overall structure of the inter-saccadic interval in terms of distance and direction, and is applied to both the eye-tracking signal and to the trajectory of the selected object.

The Transition filter calculates the number of transitions between clusters for consecutive samples within the filter length. A transition happens when the mean directions of two clusters differ with more than α_T. A high transition rate is typical for fixations, while a low transition rate is typical for smooth pursuit movements. The Transition filter is applied to the directional clustering of both the eye-tracking signal and to the trajectory of the selected object in each inter-saccadic interval. The length of the Transition filter is variable and uses the same strategy as the I-VDT

Table 1: Settings for binary filters which emphasize fixations.

Number	Type of filter	Length	Criterion
F1	Total distance	75%	$< 1.5^\circ$
F2	Transition	300 ms	$\geq 30\%$
F3		250 ms	$\geq 30\%$

Table 2: Settings for binary filters which emphasize smooth pursuit.

Number	Type of filter	Length	Criterion
S1	Total distance	75%	$\geq 1.5^\circ$
S2		75%	$\geq 2^\circ$
S3	Transition	400 ms	$< 20\%$
S4	Synchronization	200 ms	$\geq 65\%$
S5		100 ms	$\geq 65\%$

algorithm [20]. A description for the smooth pursuit filter follows: For each inter-saccadic interval, an initial filter length is used. For this filter length, the criterion of the filter is tested. The length of the filter is extended one sample at a time until the criterion for the smooth pursuit filter is fulfilled. Then, all output samples within the filter length, except the last one, are set to 0. A new window with the initial window length is initialized starting at the last sample of previous window. If the samples in the new window fulfill the smooth pursuit criteria, the first output sample in the window is set to 1, and a new window with the initial length is again initialized. Similarly, for the fixation filter, the initial filter length is initialized and the criterion for the fixation filter is tested. The window is extended one sample at the time until the criterion of the fixation filter is not fulfilled. Then, the output samples in the window, except for the last sample, are set to -1 and a new window is initialized with the initial length. If the samples in the new window do not fulfill the criterion of the fixation filter, the first sample of the filter is set to 0, and a new window is initialized. In Tables 1–2, the initial lengths of the filters are given. The Transition filter is developed to reflect structural differences in the inter-saccadic interval on a lower temporal level than the Total distance filter.

The Synchronization filter measures the similarity between the movements of the eye-tracking signal and the trajectory of the selected video object. The two signals are considered to move similarly if the eye-tracking signal and the selected object belong to the same cluster at the same time or belong to two clusters separated by at most α_T. The lengths of the Synchronization filters are shown in Table 2. The Synchronization filters are used as sliding multi-input filters that move one sample with each new calculation.

The outputs of the different filters for the eye-tracking signal, $r^l_{eye}(n)$, are summed together into one summation signal, $s_{eye}(n)$.

$$s_{eye}(n) = \sum_{l=1}^{L} r^l_{eye}(n) \tag{31}$$

where L is the number of filters. When there is no selected object, $L = 3 + 3 = 6$, i.e., fixation filters F1–F3 and smooth pursuit filters S1–S3 are used. When there is a selected object, $L = 3+5 = 8$, i.e., fixation filters F1–F3 and smooth pursuit filters S1–S5 are used. In addition, when there is a selected object, filter outputs, $r^l_{obj}(n)$, are calculated. Here, the fixation filters F1–F3 and smooth pursuit filters S1–S3 are used to determine whether the selected object is moving. The filter outputs, $r^l_{obj}(n)$, are summed together into the summation signal, $s_{obj}(n)$.

$$s_{obj}(n) = \sum_{l=1}^{L} r^l_{obj}(n) \tag{32}$$

where $L = 6$, is the number of responses.

Classification The initial classification of fixations and smooth pursuit movements is based on the sign of $s_{eye}(n)$. When $s_{eye}(n) \geq 0$, the sample is classified as a smooth pursuit candidate, and otherwise the sample is classified as a fixation candidate. In order for a smooth pursuit candidate to be classified as a smooth pursuit movement, the selected object need to be moving, i.e., $s_{obj}(n) \geq 0$. In order to compare $s_{eye}(n)$ to $s_{obj}(n)$, two binary signals are determined.

$$s_{beye}(n) = \begin{cases} 1 & \text{if } s_{eye}(n) \geq 0 \\ 0 & \text{if } s_{eye}(n) < 0 \end{cases}$$

$$s_{bobj}(n) = \begin{cases} 1 & \text{if } s_{obj}(n) \geq 0 \\ 0 & \text{if } s_{obj}(n) < 0 \end{cases}$$

The difference between the two binary signals is calculated as,

$$b(n) = s_{bobj}(n) - s_{beye}(n), \tag{33}$$

which describes the agreement between the movement of the object and the classification of the eye-tracking signal into fixation and smooth pursuit candidates. Since a smooth pursuit movement cannot be performed without a moving object, all samples classified as smooth pursuit candidates when the object is not moving, i.e., $b(n) = -1$, are disqualified and are instead marked as a disturbance.

In order to prevent the samples in the inter-saccadic interval to be divided into small segments of smooth pursuit movements and fixations, the dominant type of eye movement of the inter-saccadic interval is estimated. The estimation is based on the sign of the mean value of $s_{eye}(n)$, and is used to filter out candidate fixations or smooth pursuit movements in minority that are shorter than, t_{minFix} or t_{minSmp}, respectively [18]. When the dominant event is a fixation, i.e., the sign of the mean of $s_{eye}(n) < 0$, smooth pursuit candidates shorter than t_{minSmp} are converted into fixation candidates. Similarly, if the dominant event is a smooth pursuit, i.e., the sign of the mean of $s_{eye}(n) \geq 0$, shorter fixation candidates than t_{minFix} are converted into smooth pursuit candidates. After this step, smooth pursuit candidates are classified as smooth pursuit movements and fixation candidates as fixations.

2.6 Performance evaluation

Compensation of head movements in eye-tracking signals

In order to evaluate the performance of the head movement compensation, the standard deviation during intervals when stationary targets are shown is calculated. The evaluation is performed for three combinations of eye and head movements; only eye movements, (EM), eye- and head movements, (EHM), and head movements only, (HM), see Section 3 for a description of the experimental setup. The standard deviations, σ_{Gx} and σ_{Gy}, for the x- and y- coordinates are calculated as:

$$\sigma_{Gx} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_G(n) - \bar{x}_G)^2} \tag{34}$$

$$\sigma_{Gy} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_G(n) - \bar{y}_G)^2} \tag{35}$$

where N is the total number of samples in intervals with stationary targets. For comparison, the standard deviations, σ_{Ex} and σ_{Ey}, of the eye-in-head signals, $x_E(n)$ and $y_E(n)$, are calculated similarly as is done in (34) and (35). As an alternative approach, the eye-in-head signal is also compensated with head movements extracted from the scene video. The corresponding standard deviations, σ_{GVy} and σ_{GVy}, are again calculated similarly as in (34) and (35).

Event detection

The performance of the proposed event detection algorithm is evaluated by comparing the detections of the algorithm to the presented stimuli, i.e., for the movement

patterns described in Section 3; patterns I-IV corresponds to fixations and V-IX corresponds to smooth pursuit movements. The sensitivity and specificity for each type of eye movement are calculated. For saccades, a sample is considered to be correct if it occurs within a 500 ms interval after the stimuli switched position. Samples classified as saccades outside the 500 ms window are considered to be incorrect. A sample classified as a fixation is considered to be correct if it occurs when a stationary target is shown from the end of the 500 ms window for saccades until the target switches position. Otherwise, it is considered to be incorrect. A sample classified as a smooth pursuit is considered to be correctly classified if it occurs when the target is moving across the screen. The sensitivity for eye movement type q, SEN_q, where q = {S = Saccade, F = Fixation, and P = Smooth pursuit}, is calculated as

$$SEN_q = \frac{\mathrm{TP}_q}{\mathrm{TP}_q + \mathrm{FN}_q} \tag{36}$$

where *true positives*, TP_q, is the number of correctly classified samples for eye movement type q, and the *false negatives*, FN_q, is the number of samples that the algorithm falsely classified as another type of eye movement than type q. A value close to 1 is desired.

The specificity, $SPEC_q$, describes the algorithm's ability to only find the samples of eye movement type q and a value close to 1 is desired. For each type of eye movement q, the $SPEC_q$ was calculated as

$$SPEC_q = \frac{\mathrm{TN}_q}{\mathrm{TN}_q + \mathrm{FP}_q} \tag{37}$$

where *true negatives*, TN_q, is the number of samples that the algorithm correctly classified as another type of eye movement than q. The *false positives*, FP_q, is the number of samples that the algorithm falsely classified as eye movement type q.

In order to evaluate the algorithm's ability to be both sensitive and specific, the balanced accuracy, B_q, for eye movement type q is calculated. The balanced accuracy is calculated as the average of the sensitivity and specificity,

$$B_q = \frac{SEN_q + SPEC_q}{2} \tag{38}$$

and a value close to 1 desired.

Settings of the binary filters

In order to evaluate the settings of the binary filters, the output from each filter is compared to the stimuli, and the balanced accuracy, B_F and B_P, for fixations and smooth pursuit movements, respectively, are calculated using (38). Several window lengths and criteria thresholds were tested and the corresponding ROC-curves were evaluated. The settings with the highest B_F and B_P, according to the ROC-curves, were chosen for the fixation filters and the smooth pursuit filters, respectively.

3 Experiment and database

3.1 Participants and apparatus

The signals used in this paper were recorded during an experiment where 21 participants were included, (4 females, mean age 32.9 ± 7 years). The participants were seated on a chair facing a big white screen with dimensions 1.80 x 1.46 m. The chair was placed 2.6 m from the screen. A projector (Sanyo Pro xtraX Multiverse Projector), placed on a shelf behind the participant, was used to project the stimuli onto the screen. An illustration of the setup is shown in Fig. 5. The stimuli were presented on the screen using PsychoPy (version 1.80.03, [21]). The eye-tracking signals were recorded with a sampling frequency of 60 Hz using the mobile eye-tracker ETG 2.0 from SMI (SensoMotoric Instruments, Berlin, Germany) connected to a laptop running iViewETG, (v. 2.2.2). In order to synchronize the stimulus with the eye-tracking signals, the laptop received triggers from the stimulus computer via the parallel port. Head movements were recorded with a sampling frequency of 512 Hz using an Inertial Measurement Unit (IMU), from x-io Technology [22], which was mounted above the eye-tracking glasses with a headband. On the IMU, an AHRS algorithm is implemented [23], which is a fusion algorithm used to prevent the recorded signals from drift. In this experiment, the IMU works as a standalone data logger. Before the experiment started, the clock on the IMU was synchronized to the clock on the computer presenting the stimuli.

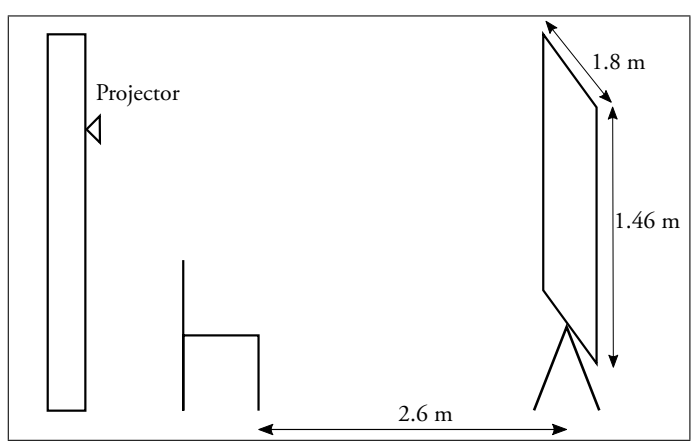
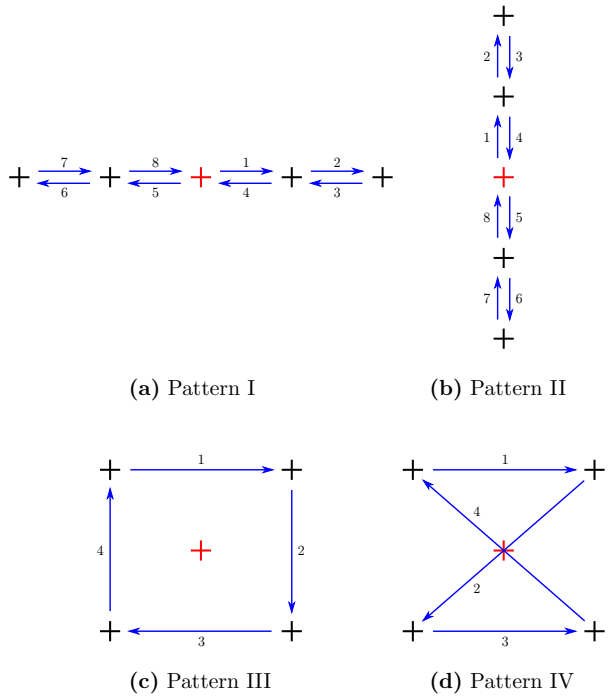


Figure 5: An illustration of the setup during the recording.



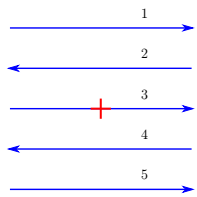
(a) Pattern I **(b)** Pattern II

(c) Pattern III **(d)** Pattern IV

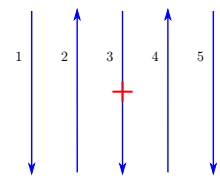
Figure 6: Eye- and head-movement patterns I - IV, which contain stimuli for fixational eye movements. The + indicates where the black dot stopped for a few seconds before instantaneously moving to the next position as indicated by the arrow.

3.2 Design and stimuli material

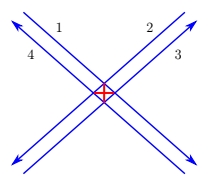
The experiment consisted of calibration, synchronization, followed by four experimental parts. The four experimental parts were as mention above: only eye movements (EM), combined eye- and head movements (EHM), only head movements (HM), and a natural task that requires both eye and head movements. The three first experimental parts were based on a set of 10 movement patterns of a black dot, with a diameter of 1°, presented on a white background. The 10 movement patterns are shown in Figs. 6–8. The stimuli of the fourth part were mainly photographs of products on shelves in the supermarket.



(a) Pattern V

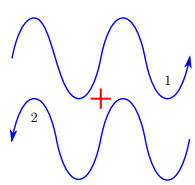


(b) Pattern VI

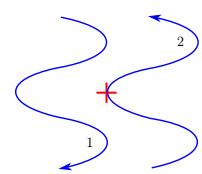


(c) Pattern VII

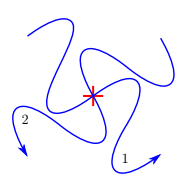
Figure 7: Eye- and head-movement patterns V - VII, which contain stimuli for smooth pursuit movements. The + indicates the center of the screen.



(a) Pattern VIII



(b) Pattern IX



(c) Pattern X

Figure 8: Eye- and head-movement patterns VIII - X, which contain stimuli for smooth pursuit movements. The + indicates the center of the screen.

3.3 Procedure

In the beginning of the experiment, the participant was informed about the aim of the study and the procedure of the experiment. The IMU was mounted on the participant's head and the recording of the head movements started. Thereafter, the eye-tracking glasses were put on the participant and adjusted to the correct position. Before the recording of the eye-tracking signals started, a 3-point calibration was performed. The calibration was followed by a synchronization session, which consisted of VOR-movements where the participant was asked to fixate the eyes on a blue dot in the middle of the screen while moving the head according to a green dot moving back and forth, first horizontally and then vertically. Since this movement gives a response in both the eye-tracking signal and the IMU signal, and the VOR-latency is around 10 ms, it was used as a control signal for the synchronization between the IMU- and the eye-tracking signals. In the EM part, the participants were asked to keep the head as still as possible and only move the eyes according to the movements of a black dot. The black dot moved according to movement patterns I - X. In the EHM part, the participants were asked to move the head and eyes simultaneously according to the 10 movement patterns of the dot. In the HM part, the participants were asked to fixate the eyes on a blue cross in the middle of the screen, while moving the head according to the movements of a green dot. The dots were moving according to movement patterns I - IV. In order to give the participants a chance to become familiar with the 10 movement patterns of the black dot, a round of practice was performed before the start of each part. Finally, in the fourth experimental part, the participants were asked to move the head and eyes freely while performing several different tasks. Examples of tasks are counting objects, looking freely at an image, and make a decision of what product to buy from a photograph of a shelf in a supermarket.

3.4 Database

The 21 recordings were divided into two subsets, a development database with 11 recordings and a test database with 10 recordings. The development part of the database was used during the development of both the head compensation method and the event detection algorithm. Settings were only adjusted based on the data in the development database. The test database was used for the final calculation of results only.

4 Results

The results presented in this section were generated using the settings in Tables 1-4.

Table 3: Settings for the head compensation.

Parameter	Value	Description
d	2.6 m	Distance to the screen
x_{max}	1280 px	Resolution of the scene camera x
y_{max}	960 px	Resolution of the scene camera y
α_{max}	60°	Maximum angular view of the scene camera in x [24]
β_{max}	46°	Maximum angular view of the scene camera in y [24]

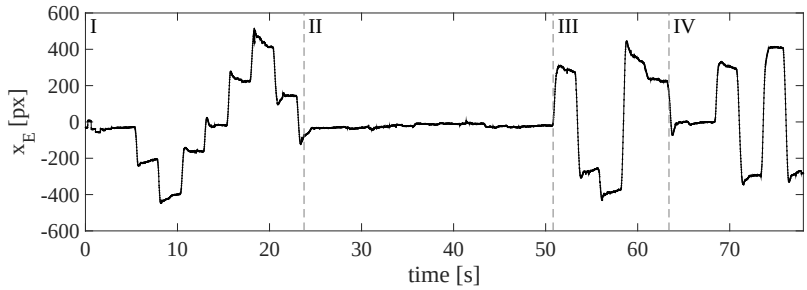
Table 4: Settings for parameters of the proposed event detection algorithm.

Parameter	Value	Description
η_a	$36000°/s^2$	Minimum acceleration threshold for noise
η_{SA}	0.75°	Minimum saccade amplitude
η_k	0.2	Minimum slope of a saccade
α_T	$\frac{\pi}{2}$	Maximum deviation between eye-tracking signal selected object
η_T	7.5°	Maximum difference in total distance between eye-tracking signal and selected object
η_P	5°	Maximum spatial distance between eye-tracking signal and selected object
t_{minFix}	100 ms	Minimum duration of a fixation
t_{minSmp}	100 ms	Minimum duration of a smooth pursuit

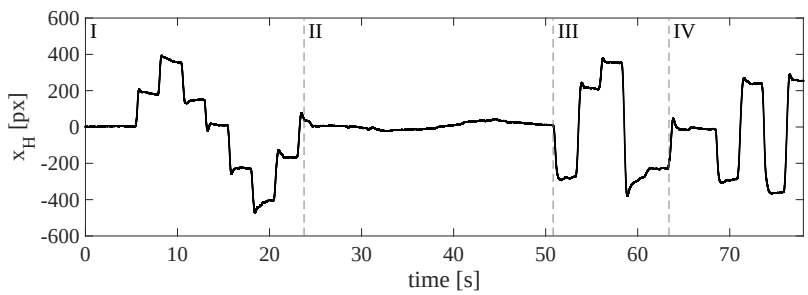
4.1 Head movement compensation

Two representative examples of recorded signals during HM and EHM, respectively, are shown in Figs. 9 – 10. During HM, the participant was instructed to keep the gaze stable in the middle of the screen while moving the head according to movement patters I - IV. The recorded eye-tracking signal and the corresponding IMU signal are shown in Fig. 9a-b, respectively. Since the mobile eye-tracker records the eye-in-head signal, the eye-tracking signal moves in the opposite direction compared to the signal recorded with the IMU. In Fig. 9c, the estimated eye-in-space signal is shown. In this example, if the user was able to keep the gaze in the middle of the screen and if the head movement compensation succeeded, the eye-in-space signal should be close to zero. During EHM, the participant was asked to move the head and eyes according to the stimuli. In Fig. 10, the recorded signals are shown during moving patterns III - V. In the recorded eye-tracking signal in Fig. 10a, it is difficult to see what types of eye movements that have been performed, e.g., the smooth pursuit movements related to movement pattern V are not visible at all. The large spikes at, e.g., $t = 55$, 70, and 80 s, result from a combination of saccades and rapid head movements. The corresponding head movement signal recorded with the IMU converted to pixels, is shown in Fig. 10b. The compensated signal, the estimated eye-in-space signal, is shown in Fig. 10c. In the compensated signal, the saccades and fixations related to movement patterns III - IV, and the smooth pursuit movements related to movement pattern V, are clearly visible.

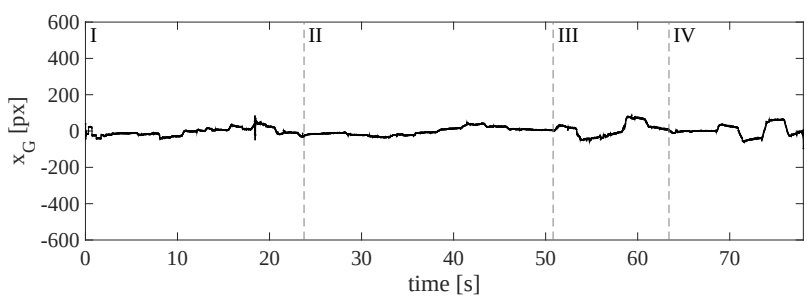
In order to evaluate the performance of the estimated eye-in-space signal, the standard deviation was calculated for movement patterns I - IV, which do not include any moving stimuli. The inter-saccadic intervals found by the proposed algorithm were used in the calculation. The mean values of the standard deviation for the test and development databases, for each of the EM, EHM, and HM parts, are shown in Table 5. For all three parts, the standard deviations were reduced as an effect of the head compensation. Even for EM when the participant was asked to keep the head as still as possible, the compensation decreased the standard deviation from around 0.16° to 0.09°. During EHM and HM when the head was moving, the head movement compensation had the largest effect. For HM the standard deviation decreased from around 8° to 3° for the x- direction and from 7° to 3° for the y- direction. There was only a very small difference in standard deviations between IMU-based head movement compensation and compensation based on head movements extracted from the scene video for this setup.



(a) Eye-in-head signal.

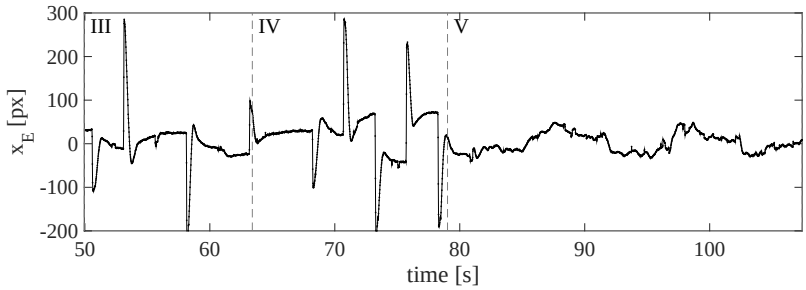


(b) Head-in-space signal.

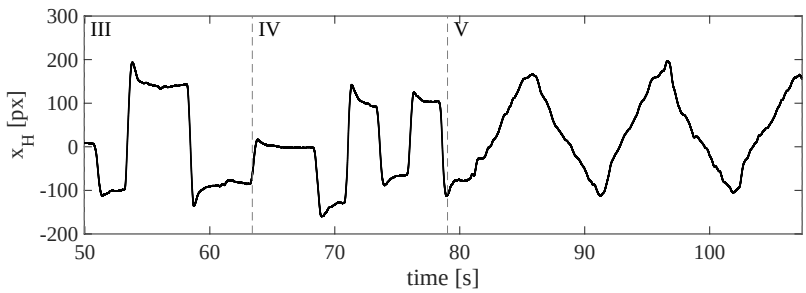


(c) Eye-in-space signal.

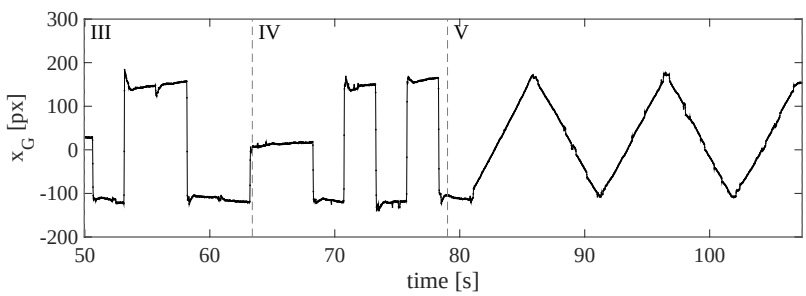
Figure 9: Examples of recorded signals during experimental part HM, (a) recorded eye-tracking signal, (b) recorded IMU-signal, and (c) head movement compensated signal. For readability, only the x component is plotted.



(a) Eye-in-head signal.



(b) Head-in-space signal.



(c) Eye-in-space signal.

Figure 10: Examples of recorded signals during experimental part EHM, (a) recorded eye-tracking signal, (b) recorded IMU-signal, and (c) head movement compensated signal. For readability, only the x component is plotted.

Table 5: Standard deviations of the positions in inter-saccadic intervals for three parts of the experiment. Uncompensated data are compared to compensated data both using an IMU and using head movements extracted from the scene video.

	Not compensated		**Compensated**			
			IMU		**Video**	
	σ_{Ex} (°)	σ_{Ey} (°)	σ_{Gx} (°)	σ_{Gy} (°)	σ_{GVx} (°)	σ_{GVy} (°)
EM	0.16 (0.16)	0.18 (0.19)	0.09 (0.09)	0.12 (0.14)	0.10 (0.09)	0.13 (0.14)
EHM	0.81 (0.84)	0.69 (0.71)	0.14 (0.14)	0.17 (0.18)	0.18 (0.18)	0.20 (0.21)
HM	8.99 (8.73)	7.49 (6.54)	3.31 (3.11)	3.33 (2.93)	3.43 (2.96)	3.43 (2.70)

4.2 Event detection

Performance evaluation

The proposed event detection algorithm for detection of saccades, fixations, and smooth pursuit movements was evaluated by comparing the detections to the stimuli and by calculating the sensitivity and specificity for each type of movement. For comparison, the sensitivity and specificity are also calculated for the built-in-algorithm in BeGaze from SMI for uncompensated data, referred to as the built-in-algorithm, and for head compensated data for the I-VDT algorithm in [20], referred to as I–VDT. The results are found in Tables 6–8. In general, when the stimuli do not contain all types of eye movements, the sensitivity and specificity cannot always be calculated. This is in Tables 6–8 indicated with a '-'. For saccades, the balanced accuracies are equally high between the proposed algorithm and the I-VDT, with 0.980 and 0.978, respectively. The corresponding balanced accuracy for the built-in-algorithm was 0.932. For EM, EHM, and HM, the results for saccades are similar. For the proposed algorithm, the balanced accuracies for fixations for all three parts are between $0.79-0.91$, compared to $0.70-0.85$ for I-VDT, and $0.53-0.91$ for the built-in-algorithm. Since the built-in-algorithm algorithm does not classify smooth pursuit movements, the specificities for fixations for parts EM and EHM are very low. For smooth pursuit movements, the balanced accuracies for the proposed algorithm are between $0.82-0.90$, compared to $0.74-0.85$ for the I-VDT and $0.50-1.00$ for the built-in-algorithm. It should be pointed out that for the HM part, which according to the stimuli contains only fixations, the built-in-algorithm has the largest sensitivity for fixations and the largest specificity for smooth pursuit movements since it can detect neither any correct nor any false smooth pursuit movements.

Table 6: Sensitivity, specificity, and balanced accuracy for saccades (S), fixations (F), and smooth pursuit movements (P), for the proposed algorithm, the I-VDT algorithm in [20], and the bulit-in-algorithm for the test database (development database). Bold font marks the best performing algorithm for that type of eye movement. When sensitivity or specificity can not be calculated the column is marked with (-).

	EM		
	Proposed	**I-VDT**	**Built-in-alg.**
SEN_S	0.980 (0.954)	0.986 (0.967)	0.995 (0.998)
$SPEC_S$	0.979 (0.985)	0.971 (0.976)	0.868 (0.881)
B_S	**0.980** (0.969)	0.978 (0.972)	0.932 (0.939)
SEN_F	0.989 (0.991)	0.916 (0.917)	0.933 (0.943)
$SPEC_F$	0.837 (0.869)	0.799 (0.785)	0.207 (0.190)
B_F	**0.913** (0.930)	0.858 (0.851)	0.570 (0.567)
SEN_P	0.807 (0.846)	0.758 (0.750)	0.000 (0.000)
$SPEC_P$	0.998 (0.995)	0.934 (0.931)	1.000 (1.000)
B_P	**0.902** (0.921)	0.846 (0.841)	0.500 (0.500)

Table 7: Sensitivity, specificity, and balanced accuracy for saccades (S), fixations (F), and smooth pursuit movements (P), for the proposed algorithm, the I-VDT algorithm in [20], and the bulit-in-algorithm for the test database (development database). Bold font marks the best performing algorithm for that type of eye movement. When sensitivity or specificity can not be calculated the column is marked with (-).

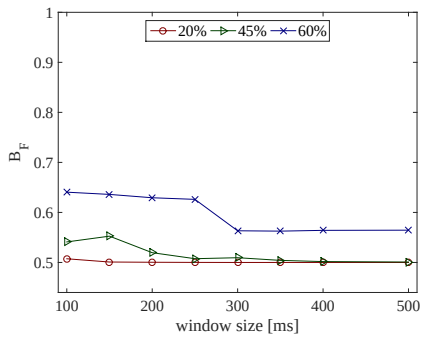
	EHM		
	Proposed	**I-VDT**	**Built-in-alg.**
SEN_S	0.978 (0.966)	0.987 (0.972)	0.985 (0.989)
$SPEC_S$	0.972 (0.982)	0.957 (0.973)	0.865 (0.887)
B_S	**0.975** (0.974)	0.971 (0.973)	0.925 (0.938)
SEN_F	0.971 (0.970)	0.772 (0.828)	0.891 (0.922)
$SPEC_F$	0.769 (0.912)	0.785 (0.813)	0.185 (0.164)
B_F	**0.870** (0.941)	0.778 (0.820)	0.538 (0.543)
SEN_P	0.729 (0.889)	0.730 (0.778)	0.000 (0.000)
$SPEC_P$	0.987 (0.983)	0.814 (0.854)	0.999 (1.000)
B_P	**0.858** (0.936)	0.772 (0.816)	0.500 (0.500)

Table 8: Sensitivity, specificity, and balanced accuracy for saccades (S), fixations (F), and smooth pursuit movements (P), for the proposed algorithm, the I-VDT algorithm in [20], and the bulit-in-algorithm for the test database (development database). Bold font marks the best performing algorithm for that type of eye movement. When sensitivity or specificity can not be calculated the column is marked with (-).

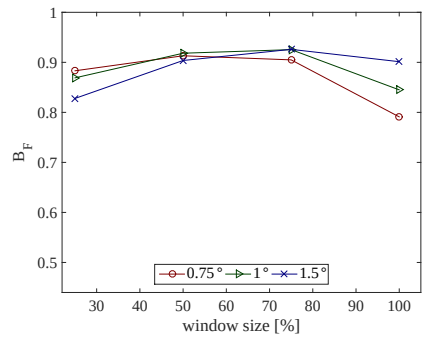
	HM		
	Proposed	**I-VDT**	**Built-in-alg.**
SEN_S	- (-)	- (-)	- (-)
$SPEC_S$	0.973 (0.981)	0.964 (0.970)	0.911 (0.921)
B_S	**0.973** (0.981)	0.964 (0.970)	0.911 (0.921)
SEN_F	0.792 (0.733)	0.702 (0.742)	0.911 (0.921)
$SPEC_F$	- (-)	- (-)	- (-)
B_F	0.792 (0.733)	0.702 (0.742)	**0.911** (0.921)
SEN_P	- (-)	- (-)	- (-)
$SPEC_P$	0.819 (0.752)	0.738 (0.772)	1.000 (1.000)
B_P	0.819 (0.752)	0.738 (0.772)	**1.000** (1.000)

Evaluation of the settings of the binary filters

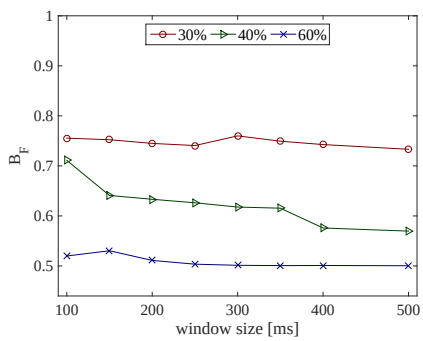
Three types of binary filters for binocular high speed eye-tracking signals recorded at 500 Hz were proposed in [18]. The three filters are: Directional consistency, Total distance, and Transition. In [18], the thresholds for each type of filter were adjusted to either emphasize fixations or emphasize smooth pursuit movements, referred to as fixation filters and smooth pursuit filters, respectively. In order to evaluate how suitable the filters are to detect fixations and smooth pursuit movement in low speed data recorded with a sampling frequency of 60 Hz, the filters were tested separately. The balanced accuracy, B_q, was calculated for a range of lengths and criteria thresholds for the three pairs of filters. The most important results are shown in Figs. 11a-c and 11d-f, for fixation and smooth pursuit filters, respectively. For fixation filters, the Total distance filter performed the best with a balanced accuracy around 0.9, and the Direction consistency filter the worst with a maximum balanced accuracy of 0.65. For smooth pursuit filters, the Total distance filter performed the best with a balanced accuracy of 0.9, and the two other filters performed equally good with balanced accuracies around 0.8. The results in Fig. 11 were used for guidance when choosing the settings of the binary filters used in the proposed algorithm, i.e., the settings of the filters with the best balanced accuracy was chosen.



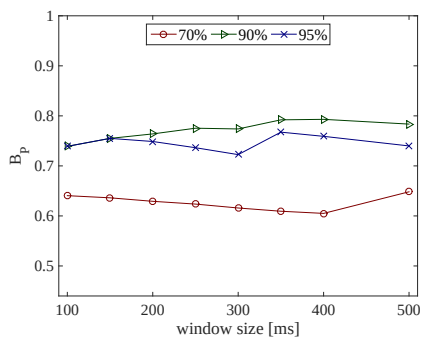
(a) Fixation filter Directional consistency



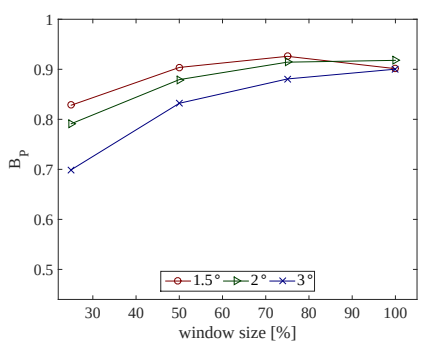
(b) Fixation filter Total distance.



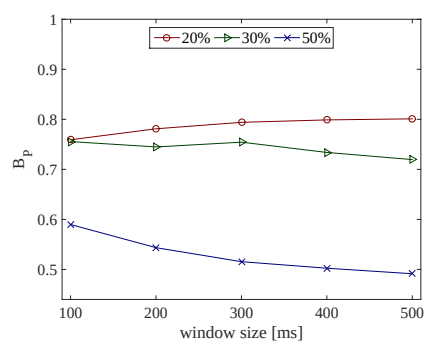
(c) Fixation filter Transition.



(d) Smooth pursuit filter Directional consistency.



(e) Smooth pursuit filter Total distance.



(f) Smooth pursuit filter Transition.

Figure 11: Balanced accuracies for fixations and smooth pursuit filters that were proposed in [18], with a range of criteria thresholds and window lengths.

(a) Participant 1 **(b)** Participant 2

Figure 12: Examples of the estimated eye-in-space signals for two participants mapped on a reconstruction of the shown image.

Video input in event detection

In order to evaluate the algorithms' overall performances, the average balanced accuracies for all types of eye movements are calculated. For the proposed algorithm, the average is 0.9 compared to 0.85 and 0.75 for the I-VDT and the built-in-algorithm, respectively. The proposed event detector includes moving objects extracted from the scene video. In order to evaluate the advantage of this inclusion of moving objects, the average balanced accuracies for all types of eye movements is calculated when the proposed algorithm does not include moving objects. The average balanced accuracies for the proposed algorithm without objects included is 0.88, which can be compared to 0.85 and 0.75 for I-VDT and the built-in-algorithm, respectively, and shows that the proposed algorithm even without objects performs better than earlier algorithms. It also shows that it is beneficial to use information about objects extracted from the stimuli.

4.3 Overall results

In order to illustrate the importance and advantage of head movement compensation in the eye-tracking signal, Fig. 12 shows the scan-path of the estimated eye-in-space signals from two participants deciding which juice they would like to buy from a shelf in the supermarket. During the task, the two participants were allowed to move their heads freely, and without the eye-in-space estimation it is not possible to map the scan-path onto a 2D-image of the stimuli. Based on the eye-in-space signal, the patterns of several participants can be compared.

5 Discussion

In this paper, a method for compensation of head movements in mobile eye-tracking signals and a multi-modal event detection algorithm for low speed eye-trackers are proposed. The method for head movement compensation estimates the eye-in-space signal by compensating for the head movements recorded using a head mounted IMU in the eye-in-head signal recorded using a mobile eye-tracker. The proposed event detection algorithm uses the estimated eye-in-space signal together with information about the detected objects in the scene video to classify saccades, fixations, and smooth pursuit movements. Both temporal and spatial aspects of the eye-tracking signal and of the detected objects are utilized in the proposed detection algorithm.

The results in Table 5 show that the proposed method for head movement compensation reduces the standard deviation of the position for all three experimental parts. The part that benefitted the most from the compensation was HM.

The proposed multi-modal event detection algorithm outperformed the two compared algorithms with an overall average balanced accuracy of 0.90, compared to 0.85 and 0.75, for the I-VDT and the built-in-algorithm from SMI, respectively. The proposed algorithm was also evaluated without using the presence of moving objects, resulting in an average balanced accuracy of 0.88, which shows that it is beneficial to include moving objects in the event detection algorithm.

In the evaluation, shown in Table 5, the proposed IMU-based method for head movement compensation is evaluated and compared to a method based on the scene video. The difference in standard deviation is small, suggesting that the two methods in this study are equally good for head movement compensation. Both types of head movement estimates can be applied to the proposed head compensation algorithm. It should, however, be pointed out that the movements in the present study were very simple to process with a bright screen on a dark background. Generally, the advantage of using an IMU is that the signal is available all the time and is independent of the resolution, quality, and content of the video. On the other hand, using the scene camera for compensation is advantageous since the detected movements are in the same coordinate system as the eye-tracking signals, and no extra equipment is needed. The largest drawback of using the scene camera is that it may be very difficult to estimate the movements in more complex situations where the head, the body, and the surrounding environment may be moving.

Compensation of head movements can be performed in the position domain or in the velocity domain. Earlier proposed methods that involve compensation in the position domain use a motion capture system [8, 9, 10, 11], an optical system [14], or a magnetic field tracker for compensation [25]. With the goal to perform event detection of eye-tracking signals from recordings outside the laboratory, these systems are not applicable. When video-based compensation is performed, as in [6], it is performed in the velocity domain, i.e., the relative motion is used. If the goal

is to perform reliable event detection that involves smooth pursuit movement, the velocity signal is proven to not contain sufficient information to separate fixations from smooth pursuit movements [20]. In that context, performing head movement compensation in the position domain, as done in this work, is advantageous.

The main drawback of using an IMU is that it in certain environments may be effected by magnetic fields, which causes drift in the sensor. In this study, an AHRS-algorithm that combines the signals from a gyro, an accelerometer, and a magnetometer, is included in the IMU to compensate for the drift [22]. The present study was performed in the laboratory without any known sources of magnetic disturbance. Whether magnetic disturbances in outdoor environment may effect the components of the IMU is not investigated in this study.

Head movement compensation in eye-tracking signals requires good synchronization between the IMU and the eye-tracking signals. In the present study, the signals were synchronized using signals recorded during VOR, as described in Section 3. This type of synchronization removes the natural lag between eye and head movements. If the VOR-latency is of interest to study, the synchronization must be handled with an external software or an eye-tracker that has an integrated IMU where synchronization between the different recording systems is handled internally, e.g., the Tobii Pro Glasses 2 [26], may be used.

The method for head movement compensation of eye-tracking signals is evaluated by computing the standard deviation in intervals with stationary targets. This evaluation strategy does not only evaluate the method but also the participant's ability to look at the presented stimuli and follow the given instructions. As an example, the HM part is very difficult to perform without sufficient training. Therefore, the standard deviation for this part, as is shown in Table 5, is large both between participants and compared to the corresponding values of EM and EHM.

The proposed multi-modal event detection algorithm is evaluated by comparing the detections of the algorithm to the presented stimuli, i.e., fixations are expected when the target is stationary and smooth pursuit movements are expected when the target is moving. This evaluation strategy gives a general picture of the performance, but there are some movements that cannot be evaluated using this method, e.g., blinks, noise, catch-up saccades, and fixations when smooth pursuit stimuli are presented. Both catch-up saccades and fixations that are performed when smooth pursuit stimuli are presented decreases the specificity for saccades and fixations, respectively, and decreases the sensitivity for smooth pursuit movements, even though the events were correctly detected.

In order to connect the eye-in-space signal with the scene in front of the user, the scene needs to be in the same coordinate system. It can be achieved by taking photographs, e.g., as is shown in Fig. 12 or by building a 3D-model of the scene [27]. For real world situations, where it may be impossible to reconstruct the scene, head movement compensation may be used mainly for reliable event detection purposes

and not for visualization.

In the proposed event detection algorithm, the eye-in-space signal is used to classify saccades, fixations, and smooth pursuit movements. The VORs are not separately classified as an own event. As proposed in [5], the recorded head signal can be included in the event detection algorithm to also classify VORs. This is, however, outside the scope of this paper.

The present study is the first step towards reliable event detection in eye-tracking signals recorded in natural situations. The two major limitations of the present study are that the body position of the participant is fixed and that the stimuli is artificial and presented on a screen. In order to be able to compensate for head and body movements when walking around freely, the proposed method must be combined with a system that tracks the body position.

6 Conclusions

An event detector which includes head movement compensation is proposed. The head movement compensation decreased the standard deviation of the position of the eye-tracking signal when stationary targets were fixated. The proposed IMU-method was compared to a video-based method and the results show that the two methods in this study are provide comparable results for head movement compensation. The proposed multi-modal event detector outperforms the I-VDT and the built-in-algorithm.

Acknowledgment

This work was supported by the Strategic Research Project eSSENCE, funded by the Swedish Research Council. Data were recorded in the Lund University Humanities Laboratory.

References

[1] K. Gidlöf, A. Wallin, R. Dewhurst, and K. Holmqvist, "Gaze behavior during decision making in a natural environment," *Journal of Eye movement research*, vol. 6, no. 1, pp. 1–14, 2013.

[2] J. Clement, "Visual influence on in-store buying decisions: an eye-tracking experiment on the visual influence of package design," *Journal of marketing management*, vol. 23, no. 9–10, pp. 917– 928, 2007.

[3] M. Land and P. McLeod, "From eye movements to actions: how batsmen hit the ball," *Nature neuroscience*, vol. 3, no. 12, pp. 1340–1345, 2000.

[4] R. Leigh and D. Zee, *The Neurology of Eye Movements*. Oxford University Press, 2006.

[5] C. A. Rothkopf and J. B. Pelz, "Head movement estimation for wearable eye tracker," in *Proceedings of the 2004 symposium on Eye tracking research & applications*, pp. 123–130, ACM, 2004.

[6] T. Kinsman, K. Evans, G. Sweeney, T. Keane, and J. Pelz, "Ego-motion compensation improves fixation detection in wearable eye tracking," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 221–224, ACM, 2012.

[7] Y. Zhang, J. Tan, Z. Zeng, W. Liang, and Y. Xia, "Monocular camera and imu integration for indoor position estimation," in *Proceedings of 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1198–1201, IEEE, 2014.

[8] R. Ronsse, O. White, and P. Lefevre, "Computation of gaze orientation under unrestrained head movements," *Journal of neuroscience methods*, vol. 159, no. 1, pp. 158–169, 2007.

[9] S. Herholz, L. L. Chuang, T. Tanner, H. H. Bülthoff, and R. W. Fleming, "Libgaze: Real-time gaze-tracking of freely moving observers for wall-sized displays," in *13th International Fall Workshop on Vision, Modeling, and Visualization (VMV 2008)*, pp. 101–110, IOS Press, 2008.

[10] K. Essig, D. Dornbusch, D. Prinzhorn, H. Ritter, J. Maycock, and T. Schack, "Automatic analysis of 3d gaze coordinates on scene objects using data from eye-tracking and motion-capture systems," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 37–44, ACM, 2012.

[11] B. Cesqui, R. van de Langenberg, F. Lacquaniti, and A. d'Avella, "A novel method for measuring gaze orientation in space in unrestrained head conditions," *Journal of vision*, vol. 13, no. 8, p. 28, 2013.

[12] K. Essig, N. Sand, T. Schack, J. Künsemöller, M. Weigelt, and H. Ritter, "Fully-automatic annotation of scene videos: Establish eye tracking effectively in various industrial applications," in *Proceedings of SICE Annual Conference 2010*, pp. 3304–3307, 2010.

[13] M. Vidal, A. Bulling, and H. Gellersen, "Detection of smooth pursuits using eye movement shape features," in *Proceedings of the symposium on eye tracking research and applications*, pp. 177–180, ACM, 2012.

[14] E. Kasneci, G. Kasneci, T. Kübler, and W. Rosenstiel, "Online recognition of fixations, saccades, and smooth pursuits for automated analysis of traffic hazard perception," in *Artificial Neural Networks*, vol. 4 of *Springer Series in Bio-/Neuroinformatics*, pp. 411–434, Springer International Publishing, 2015.

[15] T. Kübler, D. Bukenberger, J. Ungewiss, A. Wörner, C. Rothe, U. Schiefer, W. Rosenstiel, and E. Kasneci, "Towards automated comparison of eye-tracking recordings in dynamic scenes," in *2014 5th European Workshop on Visual Information Processing (EUVIP)*, pp. 1–6, 2014.

[16] S. T. Moore, E. Hirasaki, B. Cohen, and T. Raphan, "Effect of viewing distance on the generation of vertical eye movements during locomotion," *Experimental brain research*, vol. 129, no. 3, pp. 347 – 361, 1999.

[17] R. Engbert and R. Kliegl, "Microsaccades uncover the orientation of covert attention," *Vision Research*, vol. 43, no. 9, pp. 1035–1045, 2003.

[18] L. Larsson, M. Nyström, H. Ardö, K. Åström, and M. Stridh, "Smooth pursuit detection in binocular eye-tracking data with automatic video-based performance evaluation." Under Review, 2016.

[19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley-Interscience, 2001.

[20] O. Komogortsev and A. Karpov, "Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades," *Behavior Research Methods*, vol. 45, no. 1, pp. 203–215, 2013.

[21] J. W. Peirce, "Psychopy – psychophysics software in python," *Journal of Neuroscience Methods*, vol. 162, no. 1–2, pp. 8 – 13, 2007.

[22] x ioTechnologies, "x-IMU User Manual 5.2, x-io Technologies, November 2013." `http://www.x-io.co.uk/downloads/`.

[23] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pp. 1–7, IEEE, 2011.

[24] *iViewETG User Guide. SensoMotoric Instruments. Version 2.0*, 2013.

[25] R. S. Allison, M. Eizenman, and B. S. Cheung, "Combined head and eye tracking system for dynamic testing of the vestibular system," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 11, pp. 1073–1082, 1996.

[26] *Tobii Pro Glasses 2, Product Description. Version 1.0.8*, 2015.

[27] T. Pfeiffer, "Measuring and visualizing attention in space with 3d attention volumes," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, (New York, NY, USA), pp. 29–36, ACM, 2012.