VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GAZE GESTURES IN INTERACTION WITH PROBLEM-SOLVING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. HANA VRZÁKOVÁ

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

POHLEDOVÁ GESTA PŘI ŘEŠENÍ ÚLOH

GAZE GESTURES IN INTERACTION WITH PROBLEM-SOLVING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. HANA VRZÁKOVÁ
AUTHOR

VEDOUCÍ PRÁCE Ing. MICHAL HRADIŠ
SUPERVISOR

BRNO 2011

Abstrakt

Tato práce se zabývá analýzou pohybů očí jakožto charakteristiky lidských úmyslů. Během hraní hry 8Puzzle byly extrahovány pohyby očí a rozděleny na základě stisku tlačítka, které ve hře symbolizovalo hráčův úmysl pohnout herní kostičkou. Takto rozdělené sekvence představují reflexivní chování oka, tzv. pohledové gesto, které představují zdroj příznaků. Příznaky extrahované z pohybů očí pak popisují pohledová gesta spojená jak s úmysly a tak bez úmyslů. Nově bylo do analýzy zahrnuto také pozorování změn v zorničce jakožto zdroj informací, který by mohl pomoci v rozlišení úmyslných pohledů od pohledů bez záměru. Tento úkol zahrnuje binární klasifikaci, která byla realizována pomocí navrženého predikčního modelu s využitím SVM a RBF jádra. Tato práce se také zaměřuje na studium vlivu normalizace na celkové výsledky. Vyhodnocení modelu bylo realizování pomocí křivky pod grafem. Z výsledků bylo dobře patrné, že datová sada příznaků založená na fixacích a sakádách lépe rozlišila úmyslné sekvence od neúmyslných, zatímco úspěšnost sad příznaků postavených na odezvách pupily se pohybovala na hranici náhodného klasifikátoru. Dosažené výsledky vyzývají k dalšímu studiu lidských úmyslů pomocí pohybů očí, přestože klasifikace v reálném čase na základě takto navržených příznaků by prozatím nebyla 100% spolehlivá.

Abstract

This thesis is focused on employing analysis of eye movements as description of human intents. Eye movements a pupil dilation during problem solving game 8Puzzle were extracted and classified in parallel with performed button presses, which expressed user's desires to move A selected puzzle tile. The extracted sequences represented involuntary behaviour of the eye, so-called gaze gesture, and also a source for feature vectors. Extracted eye movement features expressed intentional and non-intentional gaze gestures. In addition, pupil dilations were employed as source of information distinguishing between desired and unwanted interaction with problem. This machine learning task consisted of binary classification using a predictive model based on support vector machines with an RBF kernel. Effect of normalization type was also examined to reveal how various approaches influence overall classification performance. The results were measured using Area under the Curve. Findings revealed significantly better performance in classification of features based on fixation and saccades while performance of pupillary responses were at the level of random classifier. However, the findings encourage for further studies of the relationship between intentional and non-intentional eye movements, even though their use in real time classification still would not be 100% reliable.

Klíčová slova

eye tracking, pohledová gesta, úmysl, fixace, sakády, odezvy zorniček, normalizace, efekt Midasova doteku

Keywords

eye tracking, gaze gestures, intent, fixation, saccades pupillary responses, normalization, Midas touch effect

Citace

Hana Vrzáková: Gaze Gestures in Interaction with Problem-Solving, diplomová práce, Brno, FIT VUT v Brně, 2011

Gaze Gestures in Interaction with Problem-Solving

Prohlášení

I declare that I worked on this thesis on my own and under supervision of Ing. Michal Hradiš.

. . . .
Hana Vrzáková
July 31, 2011

Poděkování

I would like to thank to Ing. Michal Hradiš and Ing. Roman Bednarik, PhD. for supervision of the thesis, as well as MSc. Tersia Gowases for her help.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Development of computer controlling devices does not spread so fast since mouse and keyboard have been accepted as a standard. Research in controllers is commonly related to game industry because a new device has a power to make a game play more natural and authentic. Use of eye tracking technology in human-computer interaction represents an alternative of controlling the computer without any physical interaction and therefore extends possibilities of application design and comes closer to users needs.

Possibilities of eye tracking are tightly related to hardware development of eye trackers. Eye tracker as a device consists of LED diodes emitting infrared light and video camera, which captures reflections of the invisible light from the eye. The Reflections are determined and thanks to interpolation between known calibration points, the human gaze as a target point in transformed to screen coordinates.

At the beginning, research has focused mainly on hardware acceleration and videocamera performance, as a result two types of eye trackers have been developed. The first type tracks the user's gaze from small distance and the video camera is part of a head-mounted device. More comfortable approach is table-mounted systems [16]. Since the current gaze tracking devices provide good precision of tracking and, current development is focused on accessibility to ordinary users, applications and tools' usage stay as a main domain of current studies.

Eye tracking as the technology of recording and evaluating the human gaze is employed in two branches. A passive use of eye trackers plays a significant role in psychological and marketing field Thanks to analytical tools that facilitate an interpretation of the human gaze as a reflection of thoughts. In this way, research of commercials or feasibility of application interface design is investigated [16]. The second approach of the eye tracker utilization concentrates on human gaze as an input stimulus for the computer application.

The eye tracker as an alternative to mouse and keyboard has been primarily studied for its possible contribution to disabled people. Hence, one type of tools utilizes eye tracker functionality so that handicapped users are able to control the computer by their gaze in similar way as healthy persons do by hands. Thus, eye-typing and eye-drawing applications have been the most commonly implemented interfaces since they belong to basic performed activities with the computer [3, 10].

Eye tracking can also be exploited in computer games as another type of controlling device, which is able to support or replace standard game controllers [2, 35]. An example of such approach has been implemented in the extension of World of Warcraft game interface where gaze gestures have supported orientation in 3D space. A favourable impact of such a controller has been seen in the natural coordination via the gaze, although the difficulties in fluency of motions have arisen [22].

Another study has tested efficiency and differences of the eye-tracking interface in a problem solving game. Via several gaze-based and gaze-added controllers, participant's concentration and level of cognitive load has been observed. Results suggest that thanks to the use of eye tracking instead of the mouse, users' strategy and planning has improved. The probable reason for that is that users have spent more time thinking about a logical challenge than clicking and observing the game interface actions [6]. Thereby, the eye tracker does not need to be perceived only as a platform for handicapped patients but it can positively influence human decision-making.

Mass use of human gaze as an input is limited by severe challenges, which are characteristic for eye tracking applications. The first well known constrain is related to eye tracker accuracy and seemingly chaotic spatial dispersion of human gaze. Thereby, application design needs to consider this fact and adjust controls of the graphical user interface, as a solution of the problem [16]. The second major issue deals with the Midas touch effect, which is defined as interface over-reacting to the human gaze [30]. Such interaction evokes feeling of anxiety, which is caused by a loss of gaze freedom. That way, users are stressed by unintended events and reactions that are fired unexpectedly and require users attention and response.

An example of such a need can be seen in the study where a movement controller of a wheelchair based on eye tracking has been implemented as an aid for disabled people. Difficulties started when the eye tracker could not recognize a difference between the user's intent to move and regular investigation of surroundings [47]. Because of this flaw, the wheelchair would react to the gaze stimulus even tough it was not a user's purpose. Such controller behaviour can lead to severe consequences, especially when the user is suddenly attracted by some outer stimulus.

1.1 Goals of Research

A general goal of this work is focused on overcoming the Midas touch effect and proposing more natural interactive interface. Research interests are centered in an analysis of eye movements and processing of changes in pupil size as a signal. In this work, I presume that a pattern of the eye behaviour can be found and serves as a marker of user's intents. A possibility of intent detection can prevent the Midas touch effect since the computer could recognize which gaze is meant as a command for the computer and which is not. Such a prediction would improve the accuracy of any interactive method since unwanted reactions are common source of interaction errors.

1.2 Research Question and Method

The main question of this thesis concerns the enhancement of human computer interaction (HCI) through an analysis of eye movements. From this point of view, the gaze gesture refers to eye behaviour described by an eye movement pattern that occurs as a inference of performing desired tasks in the game. This work focuses on the possibility that this kind of pattern, characterizing human intents, can be recognized in eye movements even though they are unique for each human being.

A method proposed in this work is based on extracting gaze features that distinguish intentions from non-intentions during problem solving and expressing them as a computational model. In contrast to previous research, which engaged the analysis of fixation and saccades as representatives of eye movements, this work aims to utilize also pupil dilations. This approach has been chosen since changes in the pupil size are tightly connected to higher cognitive load. Therefore, meaningful features can reveal in pupil dilations and represent another important descriptor of human intents.

On the basis of selected feature sets, a predictive model is designed and evaluated to describe how well distinctive the chosen features are and moreover, whether they are suitable for a real time prediction in eye tracking applications. A side investigation focuses on how chosen normalization of features influences performance because During literature review, several possibilities of normalization have been suggested.

1.3 Thesis Structure

The thesis is organized in a following way. The introduction presents previous research done in the field of eye tracking and challenges that provide a motivation and challenges for this thesis. A research goal and related research question and method are also introduced in this chapter.

In Chapter 2 a biological background of eye movements is described as a necessary explanation for further principles. A concept of fixations, saccades and pupillary dilation are explained here.

Afterwards, Chapter 3 introduces a principle and methods of eye tracking, as blinking, dwell time and gaze gestures, from the interactive point of view. Example applications using gaze as an input are also presented here since they introduce various directions of HCI based on eye tracking. At the end of this chapter the Midas touch effect together with the accuracy are mentioned as limitations of eye tracking and also as a motivation for further research.

Chapter 4 is focused on intentional gaze gestures as characteristic behaviour of human gaze. An analysis of tracked data is presented here together with a structure of recorded data and choice of eye-movement metrics. A special emphasis is given to pupillary dilations extraction, preprocessing and a features description. A design of the predictive model, based on SVM, is described here as a tool for further evaluation.

In Chapter 5, a method of the experiment is introduced together with used materials and apertures. Subsequently, a way of evaluation and results related to the experiment are presented here and discussed as a possible ground for the real time prediction.

A conclusion contains an over-all evaluation of the work and suggestions for future steps in research.

Chapter 2

Principles of Human Visual System Relevant to Eye Tracking

Eye tracking systematically detects a position of an eye and according to position changes an eye tracking application reacts in a defined way. A human visual system (HVS) is set out since it presents the needed background explaining eye movements. A taxonomy of eye movements introduces required knowledge base of eye movement types that are used in eye tracking methods.

2.1 Human Visual System

The human visual system is a complex system consisted of eyes, a retina, extraoculor muscles and signals. Through signals, input information is lead to the brain via visual pathways, and after that, to exact regions of the brain. These regions are responsible for analysis and further reactions to the input stimuli [16]. The eye as a output device responds according to types of the stimuli, thus "I can see it in your eyes" is more than true proverb in a field of eye tracking and therefore eyes becomes a real "mirror of the soul".

Figure 2.1 shows the process of entering a light beam into the eye. The incident beam is bent on a cornea and eye lens and after that it is lead through sclera to fovea, which is the most important part of the retina. By employing eye muscles, the beam is finally displayed as an image on the retina.

The retina is one of inner eye layers that consist of rods and cones, the cells that are

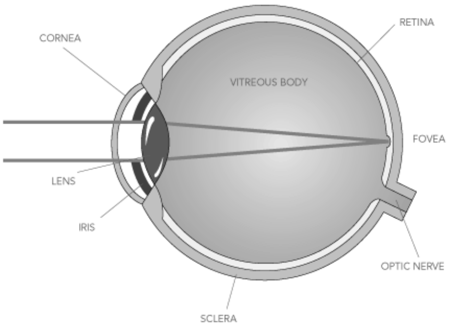


Figure 2.1: Entering the light beam into the eye. Taken from [11]

.

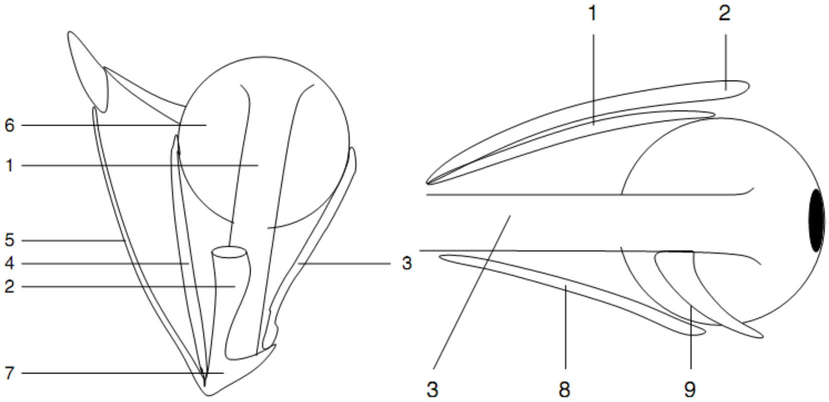


Figure 2.2: Extrinsic muscles of the eye. Left (view from above): 1, superior rectus; 2, levator palbebrae superioris; 3, lateral rectus; 4, medial rectus; 5, superior oblique; 6, reflected tendon of the superior oblique; 7, annulus of Zinn. Right (lateral view): 8, inferior rectus; 9, inferior oblique. Taken from [16].

sensitive to dim achromatic light and to brighter chromatic light, respectively. On the retina two important spots are placed - a fovea and blind spot. Vision through the blind spot is not possible. On the other hand, the fovea provides sharp or foveal vision which is defined as a perception of 2° around this spot. Parafoveal vision represents a space on retina between the fovea and the blind spot. The importance of retina is in an ability of the cells to convert incoming light to neural signals that delivered information further to the brain [16].

As the image is successfully placed on the retina, rods and cones transform the light beam into electrochemical signals and via an optic nerve they send them into the visual cortex, where they are processed into visual perception.

2.2 Taxonomy of Eye Movements

The eye is controlled by three pairs of extraocular muscles, as is showed in Figure 2.2, that grant six basic directions of gazing and ability to focus to a desired object. During gazing, eyes are not still and eye movements are not continuous, moreover they seem to be fast and almost chaotic. An explanation is found in the retina because if an image displayed on the retina stays fixated one position, objects will fade away after 1 – 3 seconds. Making the object projection steady on the retina is performed by neural control system and by extraocular muscles that serve as a feedback circuite. Five different eye movements – saccades, smooth pursuits, fixations, vergence and vestibular and psychological nystagmus are mainly employed in this task. In the field of eye tracking, saccades and fixations are usually objects of concern [16].

Saccades

Saccadic movements serve to voluntary and reflexive changing of gaze direction according to a new point of interest. Duration of saccades usually oscillates between 10 and 100ms, while no image is rendered and vision stays blind. According to Carpenter's research saccades can be described as ballistic and stereotyped movements [9].

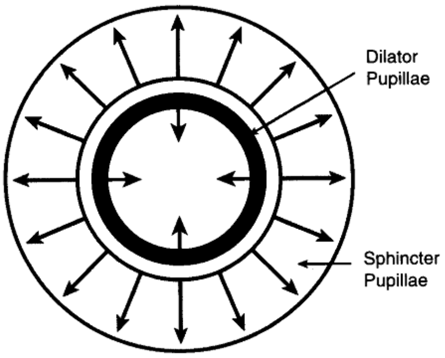


Figure 2.3: Iris muscles employed in pupillary contractions and dilations. Taken from [5].

The stereotyped eye movements are described by patterns that are periodically repeated. The second feature about ballistic behaviour postulates that a target location of each saccadic movement is precompiled during 200ms before the movement and when this computation is done the saccade will not change [16]. In eye tracking, saccades are commonly observed as markers of voluntary changes in attention.

Fixations

The second type of eye movement used in eye tracking applications is an eye fixation. Generally, fixations stabilize the retina on the focused object. An average fixation duration is between 150 – 600 ms and the eye spent about 90% by fixationing [14]. Fixational eye movements are composed by three other movements – tremor, drift and microsaccades, thanks to which an image on the retina will not disappear. A detection of fixations reminds a noisy signal which appears in 5° of visual angle around the focused object. Such behaviour creates constrains to application design which needs to consider it by scaling graphical user interface controls. On the other hand, fixations, used as an input direction, are employed in the interaction methods like dwell time or gaze gestures which are built on evaluation of fixation position and duration [16]. More details are given in Section 3.1. In behavioural research, fixations are investigated as pointers to participant higher attention.

Pupillary movements

Among commonly observed eye movements, changes in a pupil diameter, so called pupillary movements, are also employed. Pupillary movements are caused by two opposite muscles in iris, illustrated in Figure 2.3. A group of sphincter muscles reduces a pupil size while a group of dilator muscles creates its increase. Contractions of iris muscles are controlled by brain activity mainly according to outer conditions. For example, amount of incoming light is regulated by a pupillary reflex, which represents a response to the level of surrounding luminance. Another type of pupil reaction, called near reflex or accommodation response, automatically adjusts a depth of field by changing curvatures of lens [5]. Besides these reflexes, there is another type of pupillary movements related to the inner state of human mind.

Pupillary reflex dilations, also called psychosensory reflexes [33], are explained as pupil reaction to outer sensory stimulus, like a touch, visual or audio stimuli, as well as internal mental load which consists of emotions, attention and other mental processes. A size of such movements is distinctly smaller than the size of natural reflexes, on the other hand, their appearance as a correlation to cognitive intensity positively motivates for further studies.

Although pupillary responses are a mixture of reaction to light, closeness of the object and cognitive load, research of pupillary dilations as markers of cognitive load presents attractive non-invasive method of understanding human thoughts. Therefore, pupillary dilations as a possible marker of human intents in problem solving belong also to goals of this work.

Chapter 3

Eye Tracking in HCI

Eye tracking in user interfaces can play various roles, from an analytical device to a game controller. The following chapter explains basic principles of how the eye tracker works, methods of eye tracking used in applications and, finally, types of applications. At the end the limitations of eye tracking are mentioned as a motivation of improvements.

3.1 Eye Tracking Principles and Methods

Eye tracker as a device for recording eye movements is able to detected the eye using two approaches. In the first one a position of the eye is compared to the position of a head and relationship between these positions is used for deduction of eye direction. The second way is evaluating orientation of the eye independently to head position and movements, shortened known as a Point Of Regard (POR). The second approach is used in the eye tracker of my research, so this section looks more closely to its principle.

Remote eye tracking is based on identifying eye features that help to estimate how the position of the eye changed. Video based combined pupil and corneal reflections are commonly employed in eye tracking methods. The detected features are a pupil centre and reflection of cornea, which are gained by reflecting of light source, e.g. infrared light. Thanks to curvature of the eye, four reflections, also called Purkinje images, appear on the cornea. Figure 3.1 shows how the reflected beams according to cornea shape are created.

During an online analysis the eye tracker records the first Purkinje image and the pupil cornea and according to these information it computes a difference between their positions. Figure 3.2 displays how the position of the first reflection changes when the eye looks at nine calibration points at the screen. This way the eye tracker is able to estimate the eye position independently to head movements. By employing the fourth Purkinje image, the eye tracker is able to distinguish between translation and rotation of eyes, so that an accuracy of detection rises [16].

Interaction Methods

Usage of the eye tracker allows us to gain precise information about a gaze position, on the other hand thanks to microsaccadic movements a signal of the eye is quit chaotic and precise data are hardly applicable. An approximation of the signal is able to smooth an output gaze signal so that it reminds a gaze path from a human point of view. Filtered data are a base line for a future design of interactive methods.

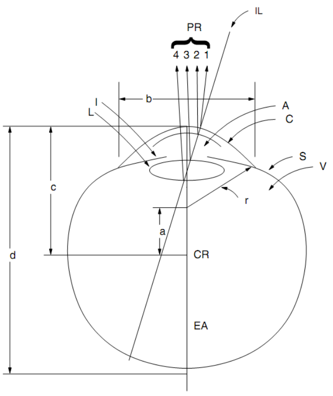


Figure 3.1: Purkinje reflections. 1, reflection from front surface of the cornea; 2, reflection from rear surface of the cornea; 3, reflection from front surface of the lens; 4, reflection from rear surface of the lens. Taken from [16].

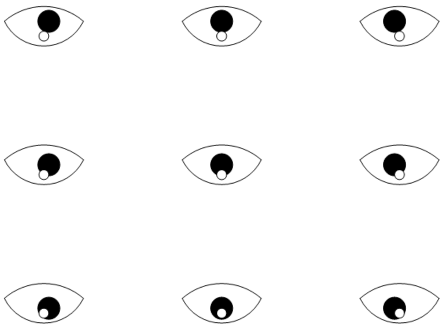


Figure 3.2: The first Purkinje image during calibration. Taken from [16]
.

Eye tracking HCI methods could be roughly divided into gaze-based and gaze-added groups. Gaze-based applications use eye tracking and the gaze as a sole source of user input for the computer. Among gaze-added methods the eye tracker stays as another computer controller like a mouse or a keyboard [16].

Blinking

In user interfaces blinking seems to be a natural substitution for mouse clicking. Research has found out that a continuous usage of this method has been tiring after while because the performed blink had to be longer than common one. Setting the intentional blink duration to longer time period has been decided as prevention of the Midas touch. However, this solution makes interaction significantly slower [30].

Dwell time

A method of dwell time is based on fixation detection. As it is mentioned in the previous chapter the eye naturally focuses on the object for 150 – 600 ms and while focusing on the object the eye oscillates around the angle of 5°. In user interaction this feature is used when users focus on a defined object in purpose for a while, the application creates response, e.g. by pressing a button. Three alternatives of dwell time usage have been found out according to behaviour of a time counter [30].

In continuous dwell time, the counter waits for a defined time and after its expiration the action is fired. If the user looks away before the set time is reached, the counter is reset and the user needs to wait for another time interval. This problem is sufficiently solved in an accumulated dwell time activation where the time counter is incremented by each eye gaze independently on the time period and the action is launched after filling up of the whole counter. The last method uses an adaptive dwell time activation, which adjusts setting of the counter period according to each user, e.g. it presumes that duration of fixations is unique to every user and according to their length it adjusts the counter duration.

Disadvantage of the dwell time is in the Midas touch effect and user friendliness. Because of the visual angle a design of buttons needs to count in larger sizes than usual. Thus an application interface is limited by number of buttons because of their size. Another challenge rises up with a position of buttons since they are commonly located on sides of the screen. Using gaze buttons causes a distraction of user visual attention by need of concentration at screen buttons and lowers user friendliness of the application [27].

Gaze gestures

Gestures described here and in example applications later refer to an interactive method which employs intentionally performed eye movements. This definition differs from the topic of this thesis, in which gaze gestures are meant as involuntary eye movements evoked during a performed task. However, gaze gestures as the interactive method is described here as a standard method, like dwell time and blinking.

From interaction point of view, a gesture is defined as a sequence of eye movements, which are performed in defined time [15]. In the field of eye tracking the definition is applied to the sequence of controlled saccades or series of saccades [23]. An application based on eye gestures monitors these sequences and launches an event after their completion. Gaze gestures are presented as a hands-free interactive method, a detection of fixations is used for a definition of the gesture shape or distinguishing saccades information. Gestures are specified by short time fixations, lasting around 100ms, while long time fixations are used for resetting the gesture, since the gesture takes around 1000ms [15].

A design of gaze gestures offers a wide range of alternatives. Usually a screen is divided into active and passive areas, where the gestures can be performed [25]. The sequence of eye movements starts in a centre point and the gesture is ended while the gaze enters the central point again. Optionally the gesture is designed as scalable and position independent. In this case a virtual grid for monitoring the gesture shape covers the screen, so that the gesture could be fired anywhere. Research of gestures focuses on a size of the gesture and suggests various possibilities [15].

Benefits of gestures is in preventing the Midas touch effect, since a probability of the unintentional eye gesture is lower than in dwell time method [30]. Moreover gestures are proof against a wrong eye tracker calibration and accuracy. Finally, since gestures are connected to the shape, a tracked gaze path, jitter movements do not cause so many difficulties as in the dwell time method.

Obvious limitations can occur in a design of gestures, since too complicated gestures are hard for remembering. Also a fatigue of the eyes is mentioned after a longer time of using an application. The negative effect is connected to amount of gesture repetition that is needed for gesture memorizing, on the other hand causes the eyes tiring [25]. Balance between complexity of the gesture design and easiness of the gesture learning needs to be considered.

3.2 Example Applications Using Eye Tracking

According to the purpose of an application two main groups of an eye tracker usage are involved. Diagnostic applications passively employ the eye tracker as a recorder of an eye gaze during an experiment and further analysis is computed after recording. A field of psychology and marketing is concerned to this opportunity. A common example of the marketing usage concentrates on observations of the best location for advertising, so that it is the first spot which will catch the user's attention [16].

The eye tracker can be also used in interactive way as an input device, which is mainly suitable for disabled people as a communication tool. In this case the eye gaze plays role of a mouse cursor, a keyboard while entering a piece of text, or as another game controller used in game interface. This chapter shows features of such applications that employ the gaze gestures in interaction.

Use of eye movements gestures in web browsing

Juang et. al. has implemented an application of Internet browser with its basic functions, which have been substituted by linear gaze gestures. An approach of gaze gestures has presumed that a gesture should be as simple as possible to naturally simulate human intentions with web content. According to this hypothesis the gesture has been recognized in the case that gaze has entered a predefined hot zone [1].

Hot zones have been designed as parts of a screen with controlling buttons and logical meaning. The application has got the benefit of expected human behaviour, e.g. when the user was trying to scroll down on a page, s/he tended to look down, so the gesture for moving down has been implemented by passing the gaze to the bottom hot zone and up. A timer between hot zone selection has been set up for 100 ms to avoid unwanted selections,

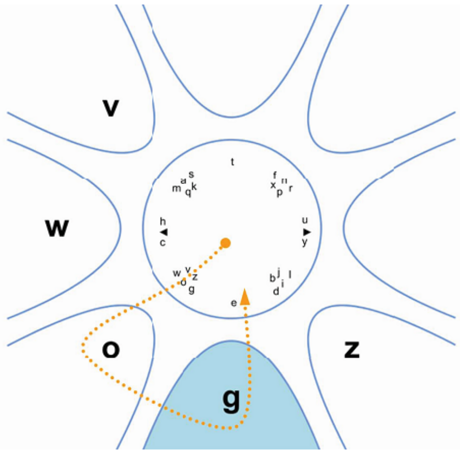


Figure 3.3: An example of the typing gesture. Taken from [3]

a longer time interval has been refused for a possible effect of slowing down an interaction speed.

Benefits of such an approach have been discovered in preventing the Midas Touch effect, faster interaction than with the mouse and in users' popularity. The gestures have been considered as ideal for simple and repeating operations, since they have represented a logical mapping between the gesture and meaning of the gesture [1].

Eye typing

A program based on original QuickWriting has taken over ideas from continuous eye writing [39], which can be the closest approach to a natural gaze. In a former application a stylus was not supposed to lift from a typing surface and the stylus did not stop to touch it. A principle of the stylus has easily fit to the gaze, since human gaze could not be switched of too.

The application has divided a screen into an inner part connected to groups of characters and into an outer part where the letters of each group have been displayed. A gesture has been designed as a curve from the central point to a location of the inner circle and the outer circle and back to the starting central point. Figure 3.3 demonstrates the described rules. Besides gestures another functions to avoid Midas Touch have been implemented. The eye tracker has also detected whether the user has followed the text part and as an event, the typing regions has been disabled.

An advantage of the application from gesture point of view is in the lack of gesture memorizing since hints have been visible as a part of application. The text field has been placed in the middle of the typing circle, so that the user attention has not been split up between typing and the output area [3].

EyeWrite

Another application for eye typing, called EyeWrite, has used a Graffity style of gestures that have been based on a shape of handwritten or printed characters. A principle of the gestures has been set up on four corners of a window, served as hot spots. Each gesture has started in one of corners and has continued to other corners according to the shape of a letter. In Figure 3.4 the gestures for basic alphabet are demonstrated. The program

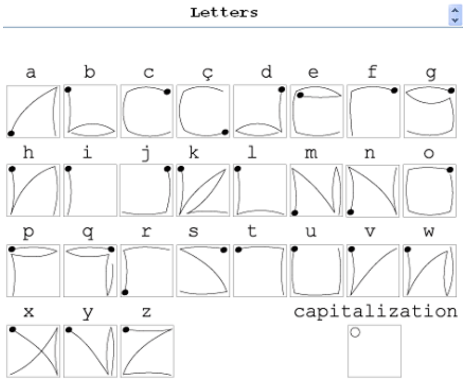


Figure 3.4: An example of character's gesture in EyeWrite. Taken from [17]

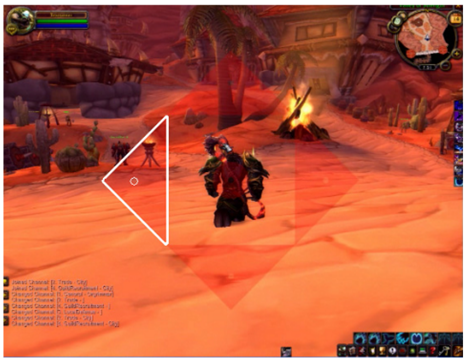


Figure 3.5: Screenshot from the World of Warcraft. Taken from [27]

has been fitted well for alphabets with a small amount of letters. Compared to a virtual keyboard, managed by dwell time method, performing the gestures has been faster after training.

Practicing of gestures has played important role in controlling of typing thank to the muscle memory, which has significantly improved a speed of the gestures. The design of gaze gestures has allowed this kind of improvements, since no fixed waiting time has not been needed as in case of dwell time keyboard [17].

Gaze gestures designed for gaming

A research of Istance et. al. has focused on feasibility of eye gesture in a 3D action game, the World of Warcraft. The game controller has been mapped to gaze gestures and after that also has been tested. The gaze gestures have been designed as 2 or 3–legged movements. The detected sequence has consisted of saccade–fixation–saccade, in the case of 2–legged gesture, and saccade–fixation–saccade–fixation–saccade in the case of 3–legged gesture. A time interval for shorter gesture has been set up to 800ms and 1600ms for the longer one. In contrast to the dwell time method, the time duration of the gesture has been comparable or shorter.

During an experiment a screen has been divided into three zones. Figure 3.5 shows a diamond shape with a circle inside as an active area for gaze gestures. Each gesture has started and ended in the central circle while passing through side triangles. The described

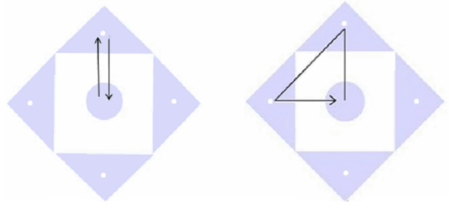


Figure 3.6: Design of gaze gestures. Taken from [27]

gesture is shown at Figure 3.6. Two versions of guided shape have been implemented and tested. The first diamond shape has allowed vertical and horizontal gestures, the second one – a square shape has used oblique gestures. According to an error rate connected to unintentional gestures, the diamond shape has been assigned to more unintentional moves and therefore more errors.

For a gaming purposes, gaze gestures have been more suitable for a specific command, like spell casting, rather than for incessant ones, e.g. controlling movements. As a result of the observation gaze gestures have been suggested as a gaze-added interface, since the gaze-based interaction would be too tiring thanks to game complexity and great amount of functions [27].

Invisible Eni

A game Invisible Eni is a gaze-based application that employs dwell time, blinking and also pupil diameter to control the game play. A task of the game considers picking up and keeping eye on butterflies and running and hiding from the enemy. The interface has been designed and implemented to reflect player's natural reactions, e.g. Blinking is mapped to hiding or escaping power since the scared player tends to close eyes, while a position of fixations is evaluated to relatively to the actual avatar position and used as the next move direction.

Newly, the game tries to utilize information gained from pupil size to perform magic connected to opening a flower. During dwelling at the flower, changes in pupil size are evaluated according to a set baseline. In case, the player's pupil dilates enough above the baseline, the flower blooms. As the pupil size could be influenced by mental effort and emotions, participants have been instructed to perform some kind of willpower to open the flower.

A pilot study has focused on a relationship between a degree of pupil dilation and the type of mental process. While positive emotions caused fair stable performance, the negative ones or feel of pain created a large variation in pupil diameter, the were not suitable for game control. On the other hand, general mind request "open the flower„, which has been tried mostly by novice players, have not affect the play according to their wish.

It has been concluded that the bottleneck of the correct pupil size classification is also in convenient pupil diameter normalization and called for further studies. Combination of global and local baseline and employing adaptive baseline have been recommended as in research [20].

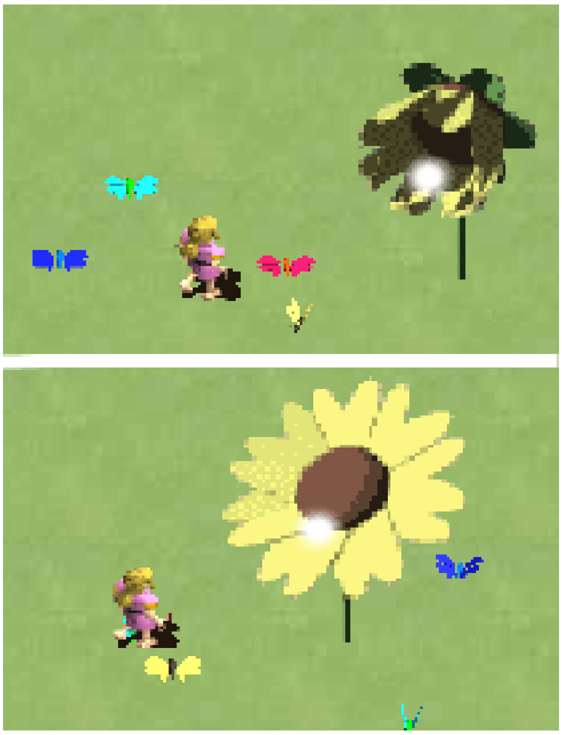


Figure 3.7: Opening the flower according to pupil dilation and willpower. Taken from [20].

3.3 Challenges of Eye Tracking

Although eye tracking provide a wide range of opportunities to improve a computer interaction, there are also a few difficulties. The Midas Touch problem, which made interaction uncomfortable and fatigue, and accuracy of the eye tracker belong to most discussed challenges and will be described further in following subsections.

Midas Touch Effect

An intuitive idea of eye tracking utilization has desired to map the eye gaze to a mouse controller so that the mouse cursor would have acted according to the gaze position. The side effect of this feature has evoked continuous reaction of interface since it is not possible to switch of the gaze. Moreover, interface cannot recognize whether the gaze has been meant as an intentional command or not. Hence, the user has lost his/her freedom of scanning objects at the screen. Permanent popping out of interface events and user's need to pay attention to it has invoked pent-up feelings. As whatever king Midas touched was turned into gold, the same effect has appeared in eye tracking as interface overreaction to the gaze [30]. The problem has represented the main challenge of interactive eye tracking interfaces. Thus, development focuses on searching for a solution through various interactive methods. The commonly used methods are mentioned in Section 3.1.

Accuracy of the Eye Tracker

The second most discussed limitation of eye tracking is connected to the eye tracker accuracy. For a sharp human vision the visual angle of 1° is sufficient, however inside of the visual angle area microsaccadic movements are also embedded. As consequences the eye tracker is not able to detect more finer eye movements in real-time since it would not recognize difference between intentional and jitter movements. In his research Jacob suggested averaging over fixations to improve the eye tracker accuracy, but such an analysis would not be obtainable in real time [30]. As an after-effect of the accuracy limitation eye tracking applications commonly consist of control elements larger than usual [24]. As it is mentioned in the section Eye Tracking Methods, gaze gestures have brought optimistic results in the case of accuracy, since gestures could depend on the visual shape of the gesture and not on the accurate position.

3.4 Summary

This chapter shows that gaze-based and gaze-added interactive methods can offer various solutions that are more proof against the Midas touch effect and problematic accuracy. Due to optimistic results and users' enthusiasm about new interactive interfaces, eye tracking grants a wide range for improvements.

Intuitively, such a betterment should be build on an innovative method resistant against the Midas touch effect on one hand and easily accepted by users on the other. As it is presented in this chapter, simple interfaces are predisposed to over-reacting behaviour while more sophisticated ones indicate the need of user's patience connected to practicing or memorizing. Thus, finding a balance between simplicity in controlling and robustness against negative side effects presents a holly grail of HCI and eye tracking.

The essence of user unfriendliness and disturbing influences is based on the fact that the computer is not able to distinguish between intentional and unintentional user's eye movements. Thus, an analysis and estimation of the intentional movements will be observed in the following chapter as another hypothetical source for innovation.

Chapter 4

Analysis of eye movements data: signal processing view

During tracking of human gaze, the eye tracker provides large amount of data and therefore a logical need for their filtration and expression by characteristic features. A concept of eye tracker feature extraction is presented here as an important step before further analysis. Generally, feature sets involve information gained form fixation and saccade position and duration as well as from pupil diameter signal. These features describe eye movements that involuntary occurred during human intents, as well as during moments without intents. After extracting feature sets, machine learning is employed as a tool that allows distinguishing between intentional and non-intentional vectors via training and parameter optimization. Figure 4.1 illustrates above-mentioned steps as a process diagram which states are detailed in this chapter.

4.1 Intentional and non-intentional eye movements

In this work, a definition of an intentional gaze gesture is related to observed game interface used in 8Puzzle game [6], which represents logical problem solving. An example of the game play is shown in Figure 4.2. In the game, a player is supposed to move with tiles 1-8 to arrange them in specific correct order. Used game interfaces consist of one gaze-based and one gaze-added. The gaze-based interface is based on on adaptive dwell time, more details about this method are given in Section 3.1. The gaze-added interface works with a gaze position for tile selection and a key button press as a choice confirmation.

On the basis of gaze augmented interface functionality, an intentional eye movements are related to user's intentions to move with a puzzle tile. The tile movement is performed when the gaze is fixated on a valid tile, that means the selected tile is able to move to a free position, and the button press (event) is performed. Afterwards, a corresponding sequence of eye tracking data is related to this event.

4.2 Eye movement preprocessing

A representation of intentional and non-intentional sequence can be characterized by a fixed window of consequential samples. However, this choice would raise a problem with a choice of the sample frequency and sequence overlapping. Thus, the sequence is described by amount of samples contained in a specified number of fixations.

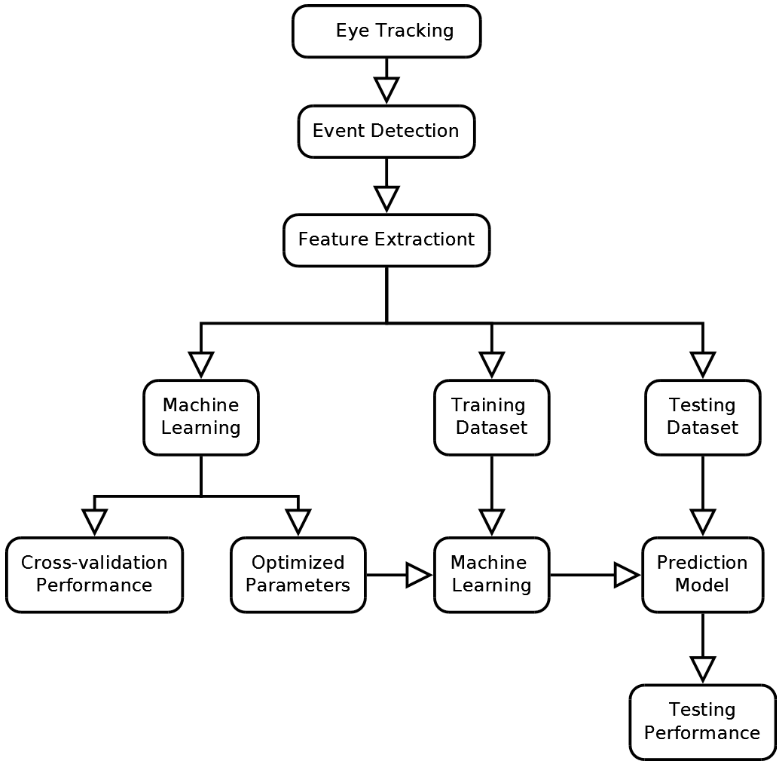
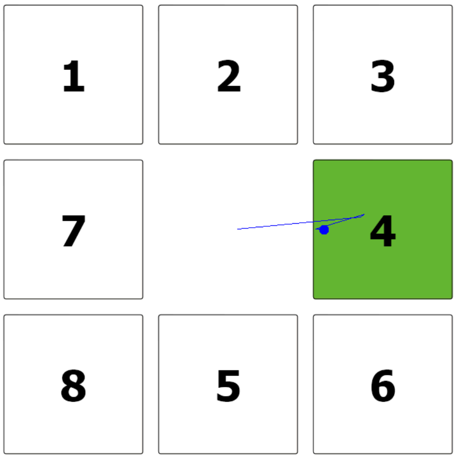


Figure 4.1: A diagram describing process flow from intent extraction to predictive model gathrering



1	2	3
7		4
8	5	6

Figure 4.2: Screenshot of 8Puzzle interface

In this work, it has been decided to study eye behaviour during two fixations before the event and one fixation after, as it is illustrated in Figure 4.3. A small amount of fixations has been set up to prevent event overlapping. The sequence of samples during these three fixations is therefore describing intentional gaze gesture. Further observations and analysis are working with this eye movement data representation.

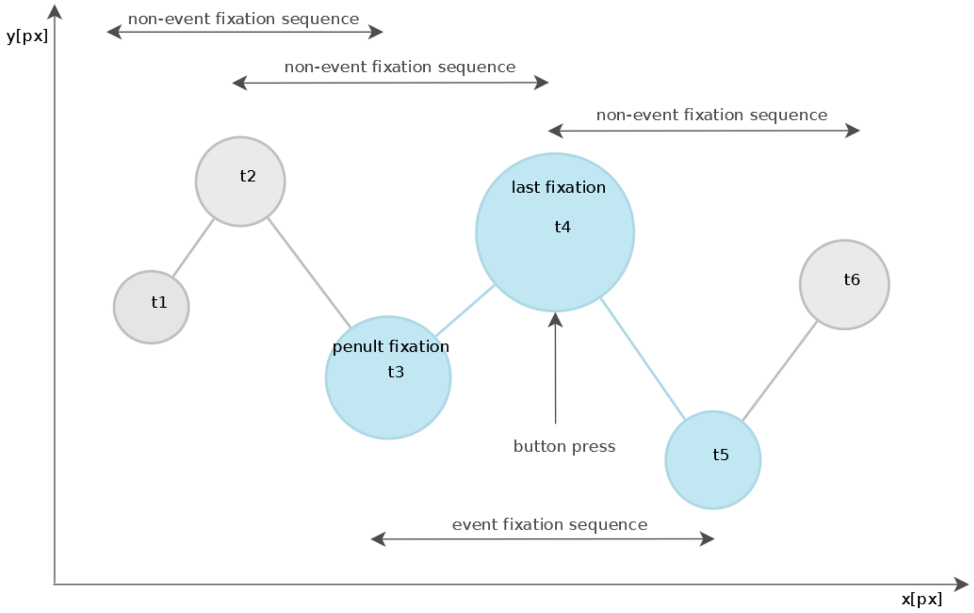


Figure 4.3: Intentional and non-intentional three fixation sequences, diameter represent represents the duration of a fixation.

Independently on the brand of a used binocular eye tracker, each eye is described by a basic set of parameters, recorded in every time period. In this work, a Tobii eye tracker is used and thereby mentioned parameters, which are also given in Table 4.1, are tightly connected to its manufacture. Generally, a dataset consists of a real position, where the eye is gazing to, and the screen position, where the gaze is mapped to, according to an initial calibration [43]. Next parameters describe physical properties of eyes, concretely a distance between the eye and the eye tracker and also a pupil size. A validity code is the last and very valuable parameter that specifies how well data is recorded and thus, how reliable they are.

Besides gaze data, information about fixations and events can be extracted from recordings, parameters used in this work are shown in Table 4.2. ClearView software, used for eye movements analysis, offers embedded filters for automatic gaze data reduction [42]. An eye filter, based on validity code, allows removing faulty recorded data as well as an approximation, in case only monocular recording has been available instead of binocular. A fixation filter allows cutting out fixations according to adjustable fixation algorithm. Also event data can be extracted and its description consists of the event string, connected to a type of event, the event code, giving more details about events, and other data.

All above-mentioned parameters present an input dataset, which is parsed and processed to three-fixation sequences as it is suggested at the beginning of this chapter. After this preparation phase, computations of eye movement metrics are employed and set as a feature vector that characterizes each fixation sequence.

Table 4.1: Eye tracking raw data describing the human gaze [43]

Parameter	Description
Timestamp	Timestamps in seconds and microseconds
Gaze point X	Horizontal coordinate of gaze data target position
Gaze point Y	Vertical coordinate of gaze data target position
Cam X	Horizontal position as seen by eye tracker
Cam Y	Vertical position as seen by eye tracker
Distance	Eye distance
Pupil	Eye pupil size
Validity	Eye validity code

Table 4.2: Output metrics provided by event and fixation filters

Parameter	Description
Event	Name connected to event happening in particular time
Event key	Number specifying event
Fixation number	Number of the detected fixation
Gaze point X	Horizontal coordinate of fixation target position
Gaze point Y	Vertical coordinate of fixation target position

4.3 Eye movement features

As a common knowledge, eye movements and their analysis can provide additional information about cognitive activity like user's actual intents and thoughts [31]. A proper selection of eye movement description to model specific hypothesis is a hard task since the interpretation of the observed effect varies according to an experiment task. For example, a longer mean fixation duration can be reasoned as attention rising, in case of web browsing [31], difficulty or lack of understanding, when reading subtitles, as well as deeper cognitive load and processing in case of problem solving [19]. Thus, a choice of describing parameters presents theory driven investigation because no standard parameters are provided according to the studied task [31].

Fixation and saccade features

In the gaze augmented interface, where the position of gaze is determining the choice of the tile, we expect that this selecting, also called last fixation can be longer since the player makes effort to target his/her desired tile. Also, surrounding fixations of intentional sequences can take more time than free sequences, the fixations without any intentions. A penultimate fixation before selecting one may be explained as the one, in which the player already decided about his/her future choice of the tile. Therefore, cognitive activity should be higher than during, for example free scanning. The next fixation after selecting can be interpreted as assuring of the tile performance, that means that the tile has been placed to a desired position. Hence, features connected to fixation duration are performed and summarized in Table 4.3.

Similarly to the fixation duration, a saccade duration can be related to intentional eye movements, therefore parameters given in Table 4.4 are related to summation, mean and last saccade before the button press. Hypothetically, a saccade duration should be shorter when the user is decided about the next move, since s/he is only performing his/her internal plan. On the other hand, free screen scanning or thinking about next tile movement combinations should last longer because of making a decision. On assumption of differing saccade, saccade durations, lengths and speeds are taken as features distinguishing intentional and non-intentional user's eye movements.

Table 4.3: Eye movements features computed from fixations

Eye movement feature	Description
Mean fixation duration	The average time of fixation duration in the observed sequence
Sum fixation duration	Sum of fixation duration in the observed sequence
Last fixation duration	Duration of the fixation during the ongoing event
Penult fixation duration	Duration of the fixation before event occurrence

Table 4.4: Eye movements features computed from saccades

Eye movement feature	Description
Mean saccade duration	The average time of saccade duration in the observed sequence
Sum saccade duration	Sum of saccade duration in the observed sequence
Last saccade duration	Duration of the fixation before event occurrence
Mean saccade length	The average distance of saccade in the observed sequence
Sum saccade length	Sum of saccade distances in the observed sequence
Last saccade length	Distance of the saccade before event occurrence
Mean saccade speed	The average speed of saccades in the observed sequence
Last saccade speed	Speed of the saccade before event occurrence
Mean saccade acceleration	Acceleration of saccade during the observed sequence

Pupillary responses

As it is mentioned in Section 2.2, cognitive load can be observable via task-evolved pupillary responses. Thanks to precision of eye tracker cameras, pupil diameter changes present attractive source of information, moreover easily obtainable as a one of output parameters.

For example, movement-related pupillary responses (MRPR) were studied according to self-triggered finger flexes, concretely the physical effort for pressing a button and complexity of the movements. Significant differences in the pupil diameter were found out 1.5s before and peaked 0.5s after the performed movement [40], an example of this pupillary responses is given in Figure 4.4. Related to the analysis of the consequent fixations, MRPR should be perceptible in intentional eye movement sequences since the mentioned finger flex is represented here as the button presses, confirming the request for moving the puzzle tile. By comparison with intentional, non-intentional sequences should contain less or none of such a responses since the finger flex is not employed here.

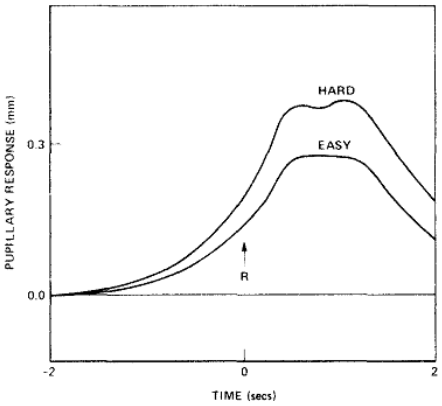


Figure 4.4: Example of the movement pupillary response. Taken from [40].

Another influence of pupil changes is cognitively related to attention driven changes in brain activity. Thus, pupil dilations as reporter variables are fired by external stimuli, as well as by emotional and mental processes which includes increase in intentional efforts [5]. A well-know example of such responses is concerned about a connection between a positive and negative stimulus [38].

Similarly, variances in task related pupillary responses (TEPR) have been studied in problem-solving multiplication task. In this case, differences have been observed in changing difficulty of the problems and therefore, they could be interpreted as a marker of intelligence [4]. A higher mental effort via TEPR have been also examined in search and map reading where fixation sequences have been divided into the controlling and targeting group, according to how far participant gazes have fixated from the given target [34]. Results have confirmed a distinction between fixation sequences that were tagged as far away from the target and the closer ones.

The second experiment has concentrated on map and legend reading which combined a pattern memorizing, while reading the legend, and visual search of this pattern in the map. Symbol memorizing has caused decrease in the pupil diameter and revealed a significant difference between reading the legend and searching in the map [34]. These findings are related to this work since 8 Puzzle game can potentially contain pupillary dilations distinguishing intents to move the puzzle tile and other thoughts related to the game play.

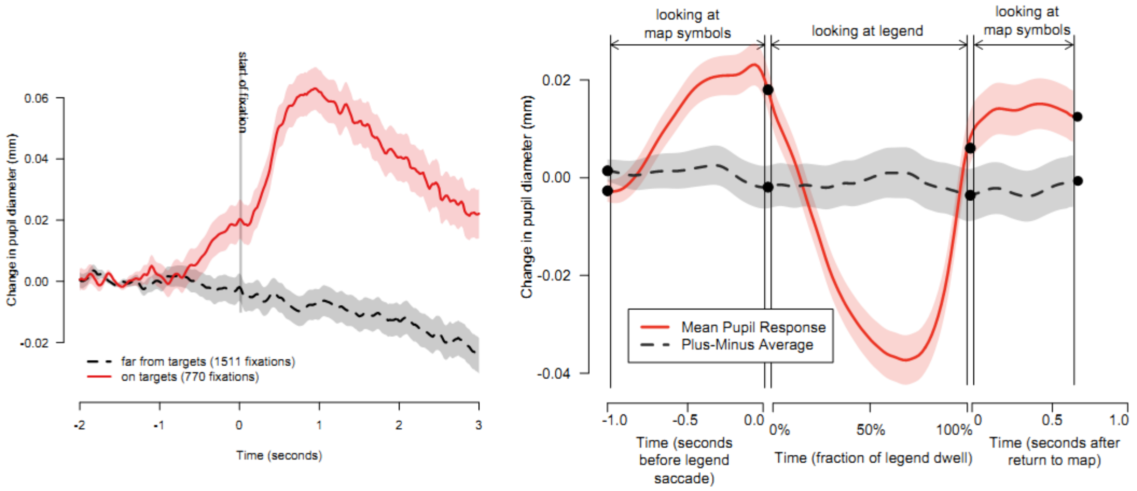


Figure 4.5: An example of task-related pupillary response. Taken from [34].

Pupillary response warping and normalization

Pupillary dilations are also studied via settings of three following fixations, however, raw pupil diameter as input signal needs preliminary arrangement. The need of pupil size preprocessing is given by the experiment setting where the fixation sequence cannot be same long since the fixation duration fluctuates according to situations and also among participants. Thus, pupil alignment and warping is used for sequence unification [34].

The desired length of the sequence has been set up by an empirical observation and averaging among whole tested set. During processing of sequences, fixation beginnings have been stored and averaged for each user separately and afterwards, also for the whole dataset. Procedure is formally described in Equation (4.1) and Table 5.2 shows the empirical results, which consist of the average length and fixation beginnings of each participant, as well as target fixation beginnings. The sequence duration is set according to the end of the last fixation. These fixation beginnings define milestones for fixation sequence warping.

$$g_j = \frac{1}{n} \sum_{i=1}^{n} g_{i,j} \tag{4.1}$$

Piecewise linear warping allows us to stretch or compress input pupil diameter signals, so that they fit into the selected window. In this case the length of the window is set up by the average fixation sequence duration. A way of sequence warping is detailed in Equation (4.2). Input parameters of the procedure consist of $g_{i,j}$, the jth fixation start of ith pupillary response, g_j, desired starts of warped fixations and $P_i\left(t\right)$ the pupil diameter at the particular time stamp. On output of the function is the time stamp defining the original pupil diameter that will occur in the warped sequence. In this way, compressing of the signal is implemented as diameters releasing and stretching as diameter repeating, while preserving original signal properties. Example average pupil diameter changes of intentional and non-intentional fixation sequence are shown in Figure 4.6 where the red line illustrates the intentional, while blue one the non-intentional sequence.

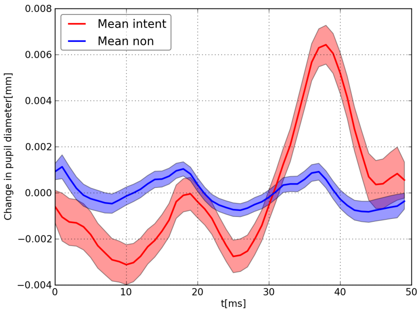


Figure 4.6: Average pupillary responses during intent occurrence.

$$W\left[P_i\left(t\right)\right] = P_i\left(g_{i,n-1} + \left(g_{i,n} - g_{i,n-1}\right)\frac{t - g_{n-1}}{g_n - g_{n-1}}\right) \qquad for \;\; g_{n-1} \leq t < g_n \tag{4.2}$$

The other limitation related to the pupil size, which needs to be taken in account, is connected to its ability of dilation and contraction according to situation, human biological and emotional characteristics as well as according to actual illumination. Functionality of iris muscles is influenced by persons' age, genetics disposal and eye defects [5], hence an absolute size of the pupil diameter requires normalizing transformation which unites pupil sizes to the same level. Various types of normalization have been tested since research generally cannot provide an unambiguous recommendation which way of data normalizing has the best descriptive properties.

Computing a baseline and subtracting it from the input pupillary response perform the first step in pupil size normalization. The baseline is defined as a mean of pupil diameters in the observed sequence [34], illustrated in Equation (4.3), or as a mean pupil diameter evaluated from short time initial trail [5]. This way, relative pupillary response is obtained from absolute values but still expressed in millimeters. In this work, computing the pupil size mean from whole recording set, for each person extra, has been employed since it can bring a valuable comparison between sequence and whole-set normalizing influence.

$$B\left[P_i\left(t\right)\right] = P_i\left(t\right) - \frac{\Delta t}{b_2 - b_1}\sum_{t=b_1}^{b_2} P_i\left(t\right) \tag{4.3}$$

The baseline subtraction also plays an important role in computing the percentage change in pupil size, which is widely used measure in pupillometrics. The percentage change in pupil size (PCPS), taken in Equation (4.4), is counted as a subtraction of the baseline from the input pupil size, followed by its dividing [4]. In this way, variations among users should be reduced and pupillary responses well perceptible. Analogously, APCPS is related to the average PCPS and represents the average pupil dilation inside the observed window of diameters [28]. A challenging characteristic of PCPS is taken in the fact that the metric does not take in account diverse level of pupil noise and can be potentially affected by changing luminance [26].

$$PCPS = \frac{\frac{X-\mu}{\sigma}}{\mu} \tag{4.4}$$

More preferable way of pupil dynamics standardization utilizes Z-scoring, also called standard score and illustrated in Equation (4.5), which expresses the normalized signal as a subtraction the baseline from the diameter and dividing the partial result by signal standard deviation [26]. An advantage of Z-scoring is in lowering differences across participants and measurements by considering varying level of noise during measurements [26].

$$Z = \frac{X - \mu}{\sigma} \tag{4.5}$$

Features derived from pupillary responses

Finally, after intentional and non-intentional sequence extraction, warping and normalizing, features given in Table 4.5 have been calculated. Firstly, Fourier Transform has been performed on normalized pupil signal to gain a power spectrum. The spectrum [46] has capability to reveal slow and fast dynamic changes in pupil size, expressed as lower and higher frequencies [7], and therefore a potential to describe the symptoms of pupillary responses. Consequently, a calculation of the power cepstrum represents potentially valuable feature since it carries information about the rate of change in different spectrum groups [8].

The first and the second differences have been computed for revealing the stability of the pupil signal. Consequently, one-dimensional histograms have been created and served as input feature vectors.

As another feature, a distance between reference and tested pupil sequences has been chosen as a degree of their similarity. The reason of concentrating on this feature lays in the hypothesis that intentional fixation sequences can diverse from the non-intentional ones also in the shape of the pupil diameter curve. Therefore, Dynamic Time Warping as an algorithm for measuring such a curve similarity has been employed [41].

The last extracted feature concerns the above mentioned average percentage change in pupil size because according to assumption that intentional fixation sequences can have perform pupil dilations more often than non-intentional ones, also average changes should reacted according to the type of the sequence.

Table 4.5: Features derived from pupillary responses

Pupillary response features	Feature description
Spectrum	Power spectrum of the pupil diameter signal
Cepstrum	Power cepstrum of the pupil diameter signal
First difference	Histogram of the first differences
Second difference	Histogram of the second differences
DTW distance	Degree of pupil signal similarity to reference set
APCPS	Average percentage change in pupil size over the fixation sequence

4.4 Design of prediction

Above defined features has served for creating a predictive model. The goal of the model is to classify an input feature vector to one of two possible classes, intentional or non-intentional movement and therefore, the task concerns a binary classification. For this purpose, Rapidminer with support vector machines has been employed [36] and this section concerns a design of the model.

Support vector machines (SVM) presents a method based on supervised learning algorithms, commonly used in classification or regression. Originally, SVM was designed for linear separable classes but nowadays it also enables find a solution for non-linearly separable classes. Basically, an object of each class is defined by a set of attributes, so-called feature vectors, and the SVM aims to split off the input vectors by an optimal hyperplane with maximum margin, as is shown in Figure 4.7. Hyperplanes $H1$ and $H2$, defining the borders of the margin, are created via specific feature vectors, the support vectors, which are able to divide the most of features vectors to appropriate classes. The characteristics of SVM is based on the fact that the number of support vectors, defining maximum margin, are generally lower than original set of input vectors [29].

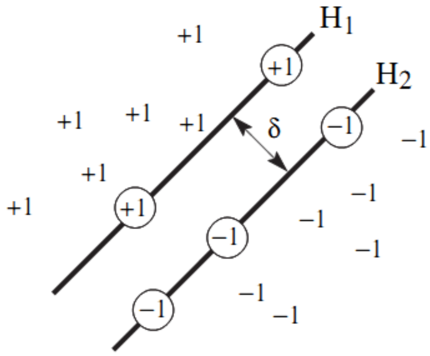


Figure 4.7: Separating classes by hyperplanes based on support vectors. Taken from [29].

Another benefit, specific for SVM, is concerned about the non-linear separable classes since SVM allows nonlinear mapping of original input vectors into the higher dimensional feature space where it is possible to split the classes by the linear hyperplane. Mapping into feature space and direct computing the hyperplane can be computationally challenging, therefore this procedure is done by kernel function. The kernel function makes computations in the input space and the result solution is reached as weighted sum of kernel functions which processed the support vectors [29].

The type of kernel function also determines the feasibility of the predictive model. For example, use of linear kernel for non-linear separable objects can give a strongly inferior model compared to non-linear kernel. On the other hand with higher level of complexity and just a little ability of generalization. This case is illustrated in Figure 4.8.

For training a predicting model based on eye movement and pupil features, radial basis function (RBF) has been chosen as the kernel function $K(x_i, x) = exp\left(-\gamma \|x_i - x\|^2\right)$. The function is based on Euclidean distance between the support vector x_i and the input vector x and the adjustable parameter γ. SVMs with RBF kernel performs analogous results as k-nearest classifier because it predicts classification to the classes according to weighted average of support vector labels, which Euclidean distance was close to zero [21].

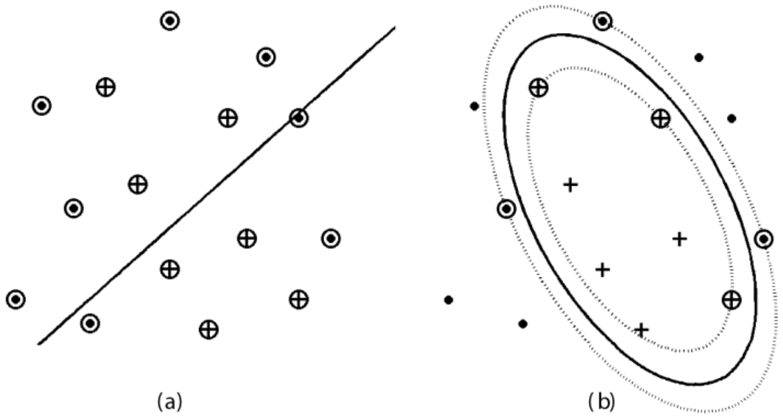


Figure 4.8: Comparison between linear and polynomial kernel function. Taken from [29].

Each set of features is unique when searching for best hyperparameters C and γ. Therefore, a sophisticated training and testing when multiple types of features are used. A design of such model is illustrated in Figure 4.9. At the beginning, the root process performs dataset loading and normalizing of feature vectors, in this case by Z-transform [37].

In the next step, input vectors are separated into training and validation parts to preserve objectivity of training. This goal is achieved by use of cross-validation which splits the dataset into number of validation subsets [36]. Used stratified sampling guarantees the same ratio of classes vectors in the split subsets as it is in the original input dataset. The further training and testing is done step-by-step on each subset exclusively.

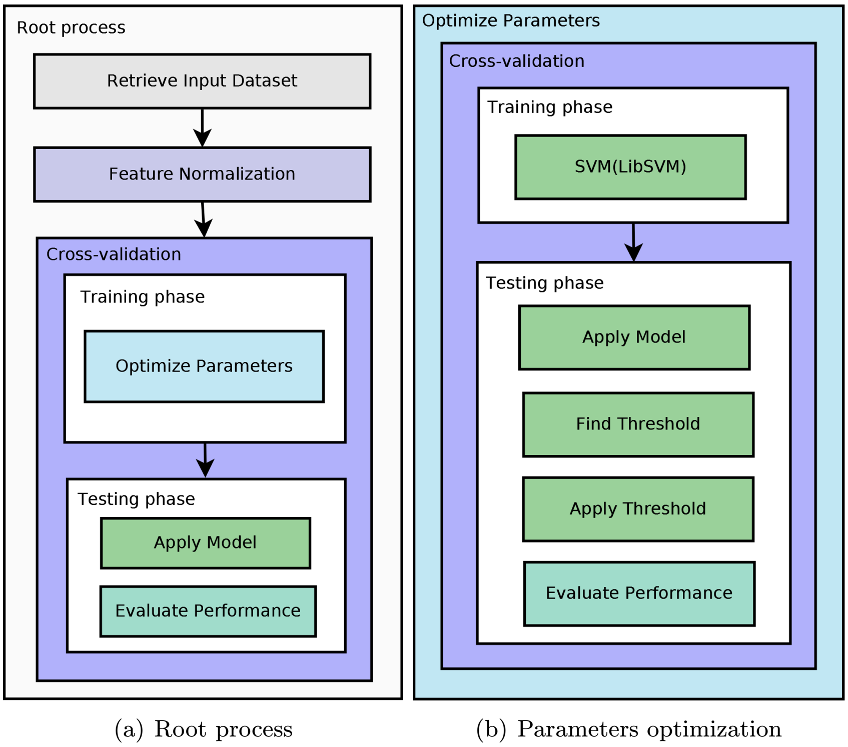
During training, the model is created and after that applied on the validation subset. Testing part of the cross-validation evaluates a performance of the predicted model and continues in next iteration and subset. The simple cross-validation is well suited for a preliminary analysis, thanks to its lower time and computational demands. Clearly superior results are gained by nested cross-validation which uses parameters optimization via a grid search operator, also shown in Figure 4.9.

At the first level of the nested cross-validation, the input dataset is split to training and testing part, as in the previously mentioned process. While the testing subset is stored for overall performance evaluation, the training subset is divided again into training and testing part for the use of parameters optimization. Specifying the parameters, as the first step, and performing grid search while evaluation the each loop performance realize the optimization.

The grid search exhaustively seek for optimal parameters, which are set up by intervals and steps. In each iteration, it selects one combination of parameters, defined as a grid point, and evaluates learning performance. In this manner, the parameters combination with highest accuracy is found out, examined on the entire training set and applied on the predictive model generating [13].

The hyperparameters for the SVM with Gaussian kernel are C, which reguralizes the SVM solution, and γ, which determines the size of the kernel. The performance can be measured for example by Area under the Curve [18]. The AUC has been more emphasized than classification accuracy since the input dataset consists of unbalanced classes. The AUC is able to reflects to this fact and thus, the model counts with the rare class.

Setting the class weights in SVMs so that the rare class has been taken as the more



(a) Root process

(b) Parameters optimization

Figure 4.9: Design of predictive model based on optimization and nested cross-validation.

important than the other one has also solved the challenge of ratio between positive and negative vectors. A similar effect can be achieved by setting a threshold that determines costs for a class miss-classification. Thereby, assigning of the rare class representative to the incorrect class is higher penalized than the opposite case and thus, leads to the balanced classification [36].

Chapter 5

Evaluation

Experiments with feature sets and normalization types, proposed in Chapter 4, are examined in this chapter. Firstly, the employed hardware and software tools are introduced as background information about preparation phase of experiments. Afterwards, the set of experiments is presented, as well as metrics used for evaluation. Results are summarized in following section together with brief comments on effects of chosen feature sets and normalization, which were observed by performance evaluation. In the end, a discussion part refers to overall result performance and future possibly based on employing these features in real time analysis.

5.1 Apparatus and material

Tobii eye tracker

During recording of game play the table-mounted eye tracker Tobii ET 1750 was employed, an illustration of an example eye tracker is given in Figure 5.1. The eye tracker is built of a 17“ monitor with 1280x1024 resolution, an embedded camera with high resolution and led diodes, which emits the infrared light. Diodes are hidden behind optical filter so that the user would not be disturbed by them [16]. Thanks to properties of the eye tracker, a viewing distance allows to come up to 60cm from the screen and head movements employ the space of 30 x 16 x 12cm, which allows covering common head motions. Another benefit related to use of the table mounted, apart from head-mounted solution, is inherent in freedom of facial expression since it does not affect position of the camera while the camera in the head-mounted solution is influenced by every eyebrows lift. The eye tracker grants a sample rate 50Hz binocularly and accuracy around 0.5°. An ability of recording the human gaze is limited with users wearing the bifocal type of glasses that causes misleading corneal-reflections.

A schematic configuration of eye tracker, host and server machines, as a common eye tracker connection, is illustrated in Figure 5.2. The server side provides routines for setting connection establishing and calibration, in which users follow calibrating points and according to their unique properties of corneal-reflections, the eye tracker is adjusted. After calibration, synchronization with client application is done and data streaming, capturing data and delivering gaze coordinates to the client application is ready start. Via Tobii SDK and ETUDriver, the client application takes controls of all the mentioned routines which grants more freedom for application development than in previous versions [16, 44, 45].

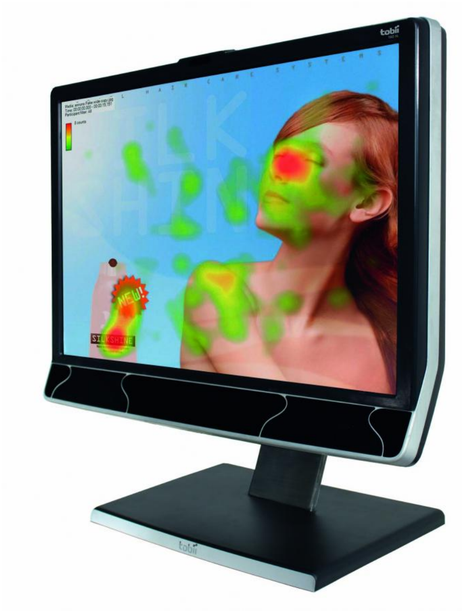


Figure 5.1: Table mounted eye tracker. Taken from [43].

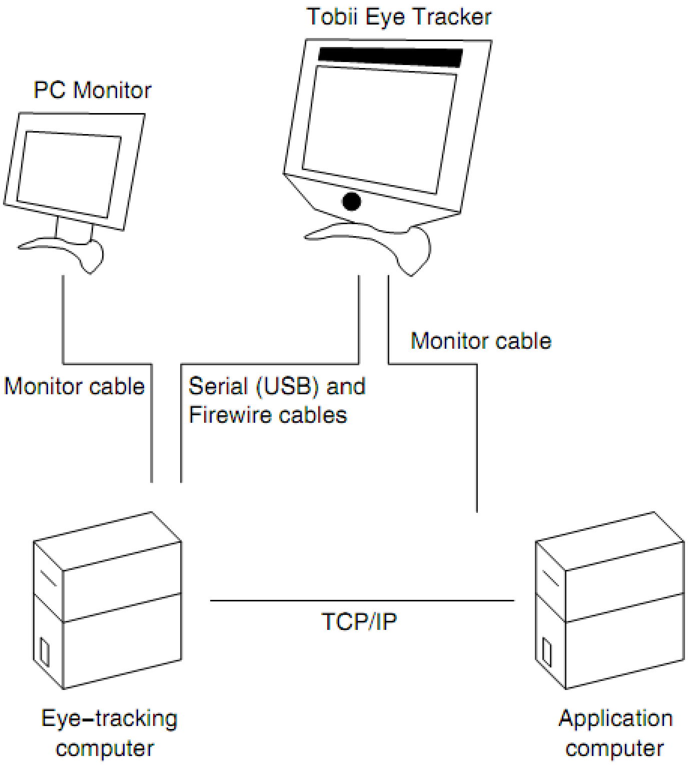


Figure 5.2: Schematic configuration of table-mounted eye tracker. Taken and adapted from [16].

Input dataset

Observed datasets were taken from previous experiments with a logical game—8 puzzle [6]. The game provided three different game interfaces - mouse only, gaze augmented and gaze-based. For the purpose of human intent analysis, the gaze-augmented interface was chosen since intentional movements have been easily marked via button press events. Interaction in the gaze-augmented variant allows players to select the tile by the eye gaze and confirm the selection by a hardware button.

The chosen dataset consisted of 13 participants with normal or corrected-to-normal vision, in the age range from 21 to 53. During experiments, participants played one trial game and three full games. The trial game was designed as warm-up and therefore, excluded from the analysis. A dataset of one participant was also eliminated from the dataset since the calibration was wrongly performed. This observation was done by studying intentional fixation sequences since the button press event should have been surrounded by the last fixation. In the case of wrong calibration, there was significant time delay between the last fixation and the button press event. Via gaze replay, the participant had also complained about the slower response of the application, which can have been influenced by the calibration.

The resulting dataset of selected participants consisted of three fixation sequences extracted as intentional and non-intentional, more details are given in Section 4.1. The overall counts of extracted sequences are illustrated in Table 5.1.

Table 5.1: Input dataset for analysis

Participant	Intentional sequences	Non-intentional sequences	Total count
1	383	1797	2180
2	18	1082	1100
3	130	2030	2160
4	73	953	1026
5	305	1601	1906
6	379	1575	1954
7	352	1687	2039
8	318	974	1292
9	119	1145	1264
10	104	849	953
11	148	1666	1814
12	161	1390	1551
Total count	2490	16749	19239
Total count[%]	12.94	87.06	100

As it is mentioned in Section 4.3 and seen in Table 5.2, length of fixation sequences varied among participants. For experiment purpose, sequences were warped according to Equation (4.2). Table 5.2 shows the empirical results, which consist of the average length and fixation beginnings of each participant, as well as the result fixation beginnings. These fixation beginnings define milestones for fixation sequence warping. Sequence length was set up according to the end of the last fixation.

Table 5.2: Mean fixation beginnings in aligned sequences

Participant	Penultime fixation [n]	Last fixation [n]	Next fixation [n]	Next fixation-end [n]
1	0	18	37	52
2	0	17	34	44
3	0	19	39	53
4	0	21	43	60
5	0	17	35	48
6	0	17	36	50
7	0	19	41	57
8	0	17	37	49
9	0	13	27	38
10	0	23	47	56
11	0	16	34	44
12	0	19	40	50
Average [n]	0	18	37	50

Analytical tools

ClearView was employed as an eye movement analysis tool, which allowed collecting recorded data from various stimuli, Windows desktop included. Among various functionality, which enable studying of human behaviour, e.g. participant responses to developed commercials stimuli, Data filtering, Area of Interest definition tool and Text Export tool were utilized as raw data gaining [42]. The details about used filters and tools are given in Section 4.1.

Fixation sequences extracting and features set preparation was implemented via set of Python scripts with contribution of Numpy, Scipy and Matplotlib libraries which consist of mathematical algorithms and functions for scientific computing and visualizations [32]. Preliminary analysis were done via R statistical tool [12], which was selected as an open source environment and alternative to Matlab tool.

The design of predictive model and procedure of parameters optimisation was made in RapidMiner, an open source solution supporting machine learning and data mining [36]. RapidMiner represents graphical tool for sophisticated experiment constructions, data and parameters handling, as well as wide range for results visualizations. The task is realized via operators that encapsulate data managements, data mining and machine learning routines, which are put together into result operator chains and trees. RapidMiner is Java based and thus, the multi-platform application, which allows server utilization. In this manner, large computational demands are covered at the server site. The predictive model was built on support vector machines, which was realized as the RapidMiner operator with support of LibSVM.

LibSVM presents a library for support vector machines that supports various SVM formulations for classifications, regressions and distribution estimations. Since the task of predictive model considered binary classification, C-SVC was used as the type of SVM. LibSVM also solves the unbalance dataset problem by setting the class weights [13].

Tested experiments were performed at the server side with eight processors, 2.93GHz Intel Xeon X3470, and 10GB RAM, running Ubuntu Linux distribution.

5.2 Method

Difference between intentional and non-intentional sequences was examined by the predictive model training and testing, as it was proposed in Section 4.4. Following experiments were systematically performed with chosen dataset, normalization and a way that the normalization was done. Unit of normalization and choice “Sequence” referred to normalizing the sequence over the actual fixation sequence, setting “Dataset” performed normalization over each participant dataset separately.

1. Experiments with pupil size features:
    - Spectrum
    - Cepstrum
    - Histogram of the first derivative
    - Histogram of the second derivative
    - Distance gained by applying Dynamic Time Warping algorithm with random selection of reference vectors
    - Average percent changes in pupil size
2. Experiments with fixation and saccades features:
    - Features derived from saccades and fixations
3. The combination of pupil size and fixation, saccades features
    - Fusion of fixation, saccade features and APCPS
    - Fusion of fixation, saccade features and the DTW distance
    - Fusion of fixation, saccade features, APCPS and DTW distance
    - Fusion of fixation, saccade features and histogram of the second difference

In case of pupil diameter based features, normalization influence were considered and also tested via following variants while each variant was computed over participant dataset as well as over each sequence.

- Baseline subtraction
- Z-score
- Percent changes in pupil size

Metrics

Designed experiments were based on binary classification therefore performance of the classifier was expressed by confusion matrix. The resulting confusion matrix contained information about predicted intentional and non-intentional vectors in comparison to real vectors. On basis of these results, metrics like accuracy of prediction are computed. In this case, accuracy as a metric was not reliable on since the examined datasets were unbalanced and accuracy would not reflect to this fact. This is caused by sum of true positives a true negatives and its division, thereby overall accuracy could be high even though one class was totally missed and classified as the second one [18]. Thus, another way of classifier evaluation receiver operating characteristic graph (ROC) was employed.

The ROC graph is based information from the confusion matrix since it represents relation between false positive rate, on the X-axis, and true positive rate, on the Y-axis. The ROC curve is defined by pairs [FP,TP] which characterize performance of all tested classification thresholds. Thus, the ROC curve itself provides well-suited visualisation of correctly classified positive instances and negative instances that were misclassified as members of the positive class. Another advantage of ROC curve utilization is in independence on class distribution or error cost.

Based on ROC curve, Area under Curve (AUC) is considered as more convenient metric since it express the probability that a randomly chosen positive instance would be classified with higher score than the negative one. The classifier with AUC equal to 0.5 is interpreted as a random, while AUC=1.0 refers to the perfect one [18].

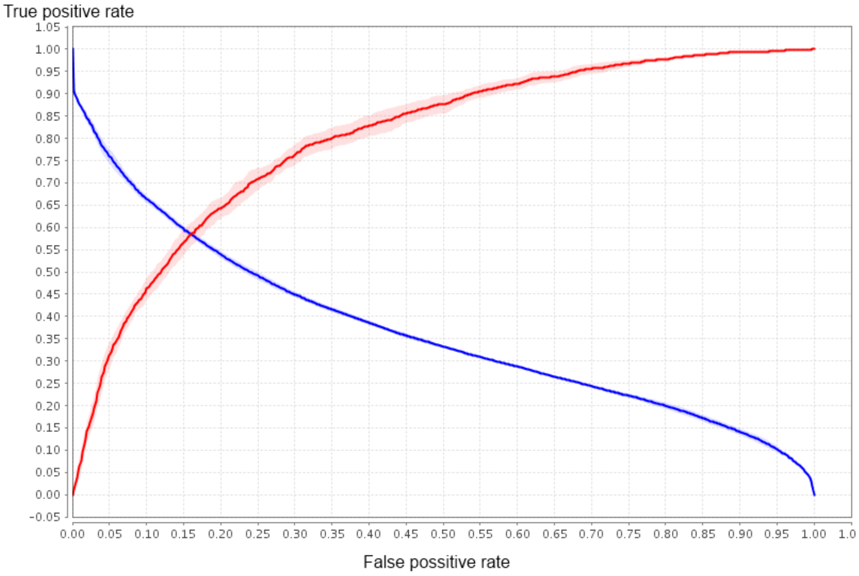


Figure 5.3: Example of ROC curve of predictive model based on fixation and saccade dataset

5.3 Results

Results consisted of AUC a SVM parameters C and γ and summarized in Table 5.3. The feature sets based on fixation and saccades were found to be better discriminative than sets grounded on measurements of pupil size. The best AUC of pupil diameter changes was represented by the histogram of the second difference, which was gained by PCPS normalization over the sequence, and reached up to 0.624. See Appendinx [16] for the list of all metrics computed in this thesis.

Type of feature	**Type of normalization**	**Area of normalization**	**AUC**	**SVM.C**	**SVM.gamma**
Fixation and saccades	Z-transformation	-	0.799	900	0.00050
+DTW	Z-transformation	Sequence	0.800	900	0.00050
+APCPS	Z-transformation	Dataset	0.806	900	0.00050
+DTW+APCPS	Z-transformation	Dataset/Sequence	**0.807**	900	0.00050
+DIFF2	Z-transformation	Dataset	0.799	900.000	0.00050
	Total Average	-	0.802		
APCPS	baseline substraction	Sequence	0.570	94.868	0.000
		Dataset	**0.577**	**94.868**	**0.001**
	PCPS	Sequence	0.624	94.868	0.000
		Dataset	**0.577**	**94.868**	**0.001**
	Z-score	Sequence	0.530	94.868	0.000
		Dataset	**0.577**	**292.201**	**0.000**
	Average	Sequence	0.575		
	Average	Dataset	**0.577**		
	Total Average	-	0.576		
1st difference	baseline substraction	Sequence	0.541	900	0.00050
		Dataset	0.541	900	0.00050
	PCPS	Sequence	0.554	900	0.00050
		Dataset	0.553	900	0.00050
	Z-score	Sequence	**0.572**	900	0.00050
		Dataset	0.541	94.87	0.00050
	Average	Sequence	**0.556**		
	Average	Dataset	0.545		
	Total Average	-	0.550		
2nd difference	baseline substraction	Sequence	0.598	900	0.00050
		Dataset	0.596	900	0.00050
	PCPS	Sequence	0.540	292.2	0.00050
		Dataset	**0.604**	900	0.00050
	Z-score	Sequence	0.601	900	0.00050
		Dataset	0.589	900	0.00050
	Average	Sequence	0.580		
	Average	Dataset	**0.596**		
	Total Average	-	0.588		
DTW	baseline substraction	Sequence	0.501	292.201	0.000
		Dataset	0.510	900.000	0.000
	PCPS	Sequence	0.540	900.000	0.001
		Dataset	0.540	94.868	0.000
	Z-score	Sequence	**0.562**	**30.801**	**0.000**
		Dataset	0.501	900.000	0.000
	Average	Sequence	**0.534**		
	Average	Dataset	0.517		
	Total Average	-	0.526		
spectrum	baseline substraction	Sequence	-	-	-
		Dataset	-	-	-
	PCPS	Sequence	-	-	-
		Dataset	-	-	-
	Z-score	Sequence	-	-	-
		Dataset	-	-	-
	Average	Sequence	-	-	-
	Average	Dataset	-	-	-
	Total Average	-	-	-	-
cepstrum	baseline substraction	Sequence	-	-	-
		Dataset	-	-	-
	PCPS	Sequence	-	-	-
		Dataset	-	-	-
	Z-score	Sequence	-	-	-
		Dataset	-	-	-
	Average	Sequence	-	-	-
	Average	Dataset	-	-	-
	Total Average	-	-	-	-

By comparison with pupil dilation based dataset, the feature set based on fixation and saccades ascended to AUC equal to 0.81 and the results were not affected by added pupil dilation feature sets, thus also independent on the performed normalization. Both resulted AUCs were performed with same setting kernel parameters, thus they should not have influenced final performance. It appears that pupil based features, as they were chosen during experimental design, do not show such an effect in classification task as the feature and saccade based features. This behaviour can be caused by overlapping of extracted sequences, as well as by deeper cognitive load during game play.

On the other hand, extracted spectrum and cepstrum as feature set could not have been evaluated at all since designed parameter optimisation was even unable to find separable hyperplanes during the training phase. Such a performance may be explained as a lack of specific signal changes groups that would distinguish intentional and non-intentional classes.

A comparison based on pupil size measurements and influence of the chosen normalization is illustrated in Figure 5.4 and 5.5 where different normalizations over each sequence and whole dataset are also shown. Among all extraxted pupil based features, the best performance was generally created by the second difference. On the other hand, the classifier based on the distance, computed by DTW, was almost evaluated as a random classifier. Such results can be interpreted as a fact that the pupillary response shape is not characterizing enough for this experimental setting.

Figure 5.4 presents evaluation of various normalization which were performed over each sequence. The best performance while normalizing histogram of first difference was achieved by z-score normalization, similar result was gained also by normalizing the second difference histogram and DTW distance. On the other hand, the highest AUC in case of APCPS dataset revealed by PCPS normalization while z-score created the lowest AUC in this group. Results illustrate that none of normalizations performed over sequence could have been evaluated as favored since the performance differed according to the type of the input dataset.

Similar observations are shown in Figure 5.5 where normalisation over each participant dataset were performed on pupillary response based features. In contrast to sequence normalization, the best performance was achieved via PCPS normalization almost in all feature types. In case of APCPS and normalization over dataset, the resulted AUCs were equal and thus, independent on used normalization. By comparison with previous sequence performances, difference in AUC related to type of normalization differs just a little while in case of sequence normalization, divergence between types is more significant. These results can be interpreted as a dataset normalization behaviour that removes more important signal characteristics than in case of sequence normalization.

Figure 5.6 illustrates evaluation of feature sets which joined promising features to achieve better performance. In comparison with fixation and saccades feature set, generally better AUC was gained by employing fusion of DTW distance and APCPS together with fixation and saccades. On the other hand, added features did not cause meaningfully higher AUC values. This results may reveal that discriminative features are more notable in case of fixation and saccades than in cased of pupillary response based features.

5.4 Discussion

The purpose of this work was focused on opportunity to find gaze gestures, interpreted as eye movement sequences which occurred during specific situations or events. The original hypothesis expected that human intents were observable in human gaze via specific eye

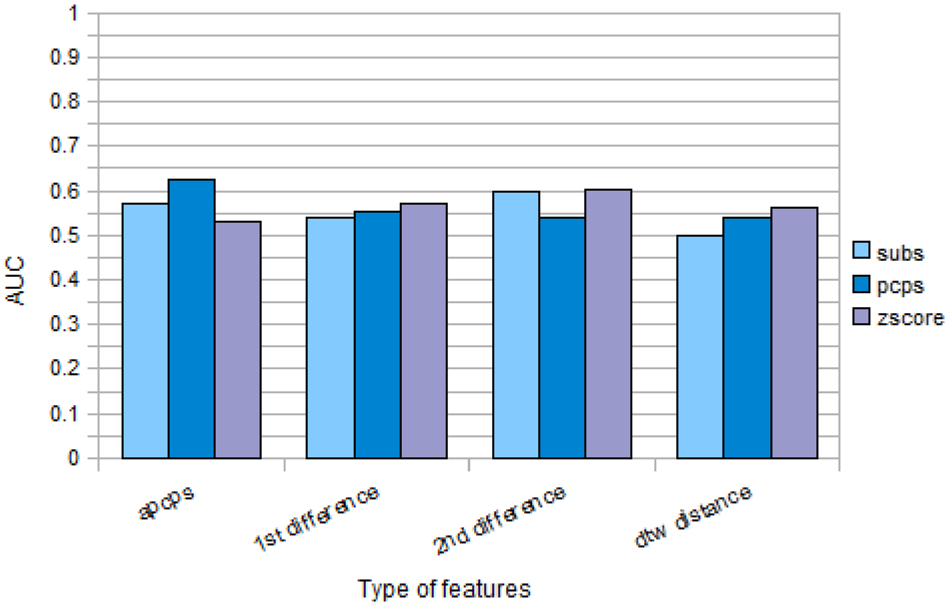


Figure 5.4: Performance of pupillary response feature sets based on normalization over sequence.

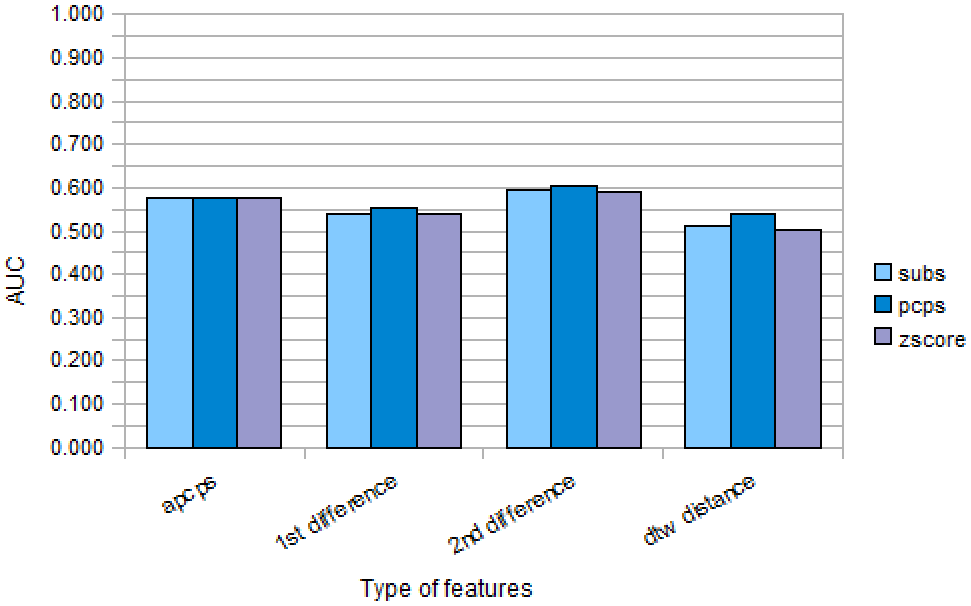


Figure 5.5: Performance of pupillary response feature sets based on normalization over dataset.

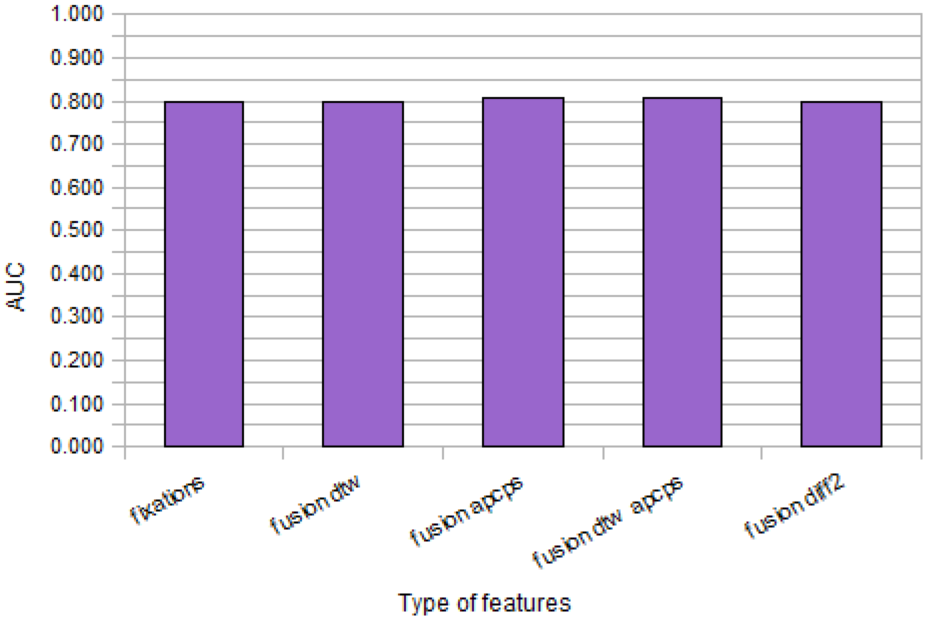


Figure 5.6: Performance of feature fusions

movements, concretely fixations, saccades and pupillary dilations, and therefore, possibly parameterized by the set of features as well as trained and tested via the designed predictive model. Special emphasis was taken on use of pupillary dilations as a promising parameter since in previous research the changes in pupil size were evaluated as task-dependent and tightly connected to higher cognitive load. Another motivation was given in opportunity to compare pupillary responses to fixation and saccade since this was not done before and hence, it could brought brand-new insight into the field of eye movement metrics and features.

The findings, however, confirmed this theory just partially. Features computed from fixation and saccade positions and durations revealed as well-distinguishing in case of intentional and non-intentional eye movement sequences. The AUC of such predictive model reached up to 0.8 which presented a fair-well classification of intentional and non-intentional sequences. On the other hand, extracted features from pupillary responses did not achieve such performance; the most solid AUC was measured around 0.6 and obtained by the histogram of the second difference. On contrary, cepstrum and spectrum of pupil diameter changes could not have been evaluated at all since the classifier could not have been even trained which was quiet surprising.

Also fusions of fixation, saccades and pupillary dilation based set were observed to find out how extra added features, related to pupil diameter changes, would influence the result performance. Although, the higher performance tended towards fusion feature sets, a numerical difference between results moved around 1%, which could be interpreted as mistake of measurements. Thus, pupillary dilations did not prove the hypothesis about discriminative characteristics extracted from task-evoked pupillary responses. In this manner, results of performed experiments did not correspond with findings of previous research that studied cognitive load via pupillary responses only [34].

Although the study was about to eliminate the influence of normalization by testing various settings and choosing the perfect one, overall results showed that the recommended normalizations differed according to the type of the feature. This finding also disapproves with research, which prefers one type as the best one [21, 26].

There are several interpretations, which can explain revealed performance. One general possible conclusion of the lower classification is in the fact that cognitive load was employed during pressing the button as representation of the participant intent, but also could have been found in non-intentional sequences while participant was planning and imaging desired steps in the game play. Similarly, intentional fixations could express non-intentional sequences. This effect could reveal in case that the participant plans his or her steps in the game and afterwards, pressing button represents only executing the decided strategy. Thus, this sequence of button pressing can consist of lower cognitive load than others and therefore, with lower eye movement responses. It is presumable, that overall intents during game are significantly higher than performed button presses, which should influence the classification.

A basic explanation of performance of pupillary responses would rely on the general fact that task-evoked responses are minor compared to responses caused by light and focus reflex. However, experiments were lead in the laboratory under stable lighting conditions and the sole application did not evoked situations, like shade and contrast changes, which would lead to abnormal pupil size changes. Of course, these changes can have revealed when the participant looked outside of the screen and thus, pupils adjusted to different surrounding. On the other hand, a number of such a voluntary gazes away from the game was markedly lower than overall number of in screen gazes. Thereby, the mistake caused by off-screen gazes was not counted.

Another explanation of minor performance of pupil size based features can be found in the fact that percent changes in pupil size do not differ so much as in case of fixation and saccades position and duration and thus, they may be worse recognized.

This study took unexplored and innovative way of human intents classification via eye movement features. Thereby nothing like best practices in case of choosing feature set could have employed. It is probable that some of chosen features could be replaced by better-distinguishing feature set, especially in case of pupil response. In addition, methodological challenges related to unbalanced dataset and its overcoming also limit interpretation of results. Finally, each predictive model was employed for whole dataset, which was computationally demanding and time needed for each experiment was also influenced by the task.

Currently, chosen feature set and its performance do not fulfill conditions for the real time classification and its use in interactive game interface. The implemented extension should improve the accuracy in recognition of intentional player's requests and this way, lower the number of interaction mistakes and unwanted fired events. With measured AUC, hypothetical interface would miss intended requests and moreover, fired another false events. Another mistake generating interface is not needed and therefore, on the basis of resulted evaluation, real time classification is not recommended.

Another future steps can include more studies of DTW distance since creating a bank of pupillary responses could improve the classification performance. Source sequences that correspond to events, attitudes and cognitive load can lead to higher performance since the reference dataset would be chosen on a prior knowledge rather than by random selection, as it was done in this experiments.

Chapter 6

Conclusion

This work concentrated on involving eye tracking in HCI and its possibilities of improvements. The need of advanced iteration methods is related to the often-mentioned side effect Midas touch, which causes system over-reacting to human gaze and user's frustrations due to loss of gazing freedom. Improvements, suggested in this work, were based on the hypothesis that finding proper eye movements characteristics, which arise during human intents. It was assumed that well-chosen set of features would allow distinguishing between intentional and non-intentional player's look and thus, lower the over-reacting behaviour.

Methodology of the experiment consisted of data preprocessing, suggesting feature sets and way of their normalization, as well as design of predictive model and metrics for its evaluation. Eye movements characterizing human intents in problem solving were defined on the background of 8Puzzle game, in which player's desires were interpreted as button press confirming movement of the chosen puzzle tile. Features representing a sequence of eye movements were based on fixation and saccades position and duration, and in addition, pupillary response based features were also employed as promising features.

Prediction of sequence assigning was realized by predictive model training and testing, based on Support vector machines with the RBF kernel. Since the dataset consisted of two classes, evaluation of binary classification was done by confusion matrices, ROC curves and mostly by Area under the Curve. This metric was chosen since the input dataset was strongly unbalanced and thus, more sophisticated measurements than accuracy had to be employed.

Side research was done by observation how a chosen dataset normalization influences overall performance. The motivation for this study was found in a lack of standards and best practices in the field of eye tracking. There were several suggestions about which normalization was the best describing, while suggestions differed according to source. Thus, pupillary response based feature set were normalized by three types with two possible settings and evaluated.

Evaluation and findings pointed out significantly better performance in classification of fixation and saccades based features while performance of pupillary responses were at the level of random classifier. Also results of normalization perfomence cannot provide recommendation for universal way of normalization since the best performance was strongly dependent on a combination of the chosen type of feature, normalization and unit of normalization.

However, the findings encourage for further research of human intents via eye movements, even though their use in real time classification still could not be reliable. A benefit of this work could be achieved by employing extracted intentional and non-intentional features in other classification tasks which provide more information about eternal human thoughts. Thinking aloud and intent detection during problem solving presents a next and recommended step of this work.

Bibliography

[1] J. Ahn, A. Duchowski, F. Jasen, K. Juang, and A. Katrekar. Use of eye movement gestures for web browsing. unpublished manuscript.

[2] S. Alkan and K. Cagiltay. Studying computer game learning experience through eye tracking. *British Journal of Educational Technology*, 38(3):538–542, 2007.

[3] E. Andre and N. Bee. Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In *PIT '08 Proceedings of the 4th IEEE tutorial and research workshop on Perception and Interactive Technologies for Speech-Based Systems: Perception in Multimodal Dialogue Systems*, pages 111–122. Springer, 2008.

[4] J. Beatty. Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychol Bull*, 91(2):276–292, Mar. 1982.

[5] J. Beatty and B. Lucero-Wagoner. *The pupillary system*, chapter 6. Cambridge University Press, 2000.

[6] R. Bednarik, T. Gowases, and M. Tukiainen. Gaze interaction enhances problem solving: Effects of dwell-time based, gaze-augmented, and mouse interaction on problem-solving strategies and user experience. *Journal of Eye Movement Research*, 3(1):1–10, 2009.

[7] R. Bednarik, T. Kinnunen, A. Mihaila, and P. Fränti. Eye-movements as a biometric. *Image Analysis*, pages 780–789, 2005.

[8] B. Bogert, M. Healy, and J. Tukey. The quefrency alanysis of time series for echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum and Saphe Cracking. In *Proc. Symp. on Time Series Analysis*, pages 209–243, 1963.

[9] R. H. S. Carpenter. *Movements of the eyes.* Pion, London, 1988.

[10] A. Cavender, A. Hornhof, and R. Hoselton. Eyedraw: A system for drawing pictures with eye movements. In *Assets '04 Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility*, pages 86 – 93. ACM New York, NY, USA, 2004. ISBN 1-58113-911-X.

[11] Central Sydney Eye Surgeons Pty Ltd. Mechanism of the eye. http://www.centralsydneyeye.com.au/eyeworks.html, 2010.

[12] J. M. Chambers. *Software for Data Analysis: Programming with R.* Springer, New York, 2008. ISBN 978-0-387-75935-7.

[13] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[14] I. D. E. *Visual Memory Within and Across Fixations*, pages 146–165. New York: Springer-Verlag, 1992.

[15] H. Drewes and A. Schmidt. Interacting with the computer using gaze gestures. In *INTERACT 2007*, pages 475–488. Springer, 2007.

[16] A. Duchowski. *Eye Tracking Methodology, Theory and Practice, Second Edition.* Springer, 2007. ISBN 978-1-84628-608-7.

[17] A. Duchowski, J. Rubinstein, M. Sawyer, and J. Wobbrock. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *ETRA '08 Proceedings of the 2008 Symposium on Eye-Tracking Research and Applications*, pages 11–18. ACM, 2008. ISBN 978-1-59593-982-1.

[18] J. P. Egan. *Signal Detection Theory and ROC Analysis.* Academic Press, 1975.

[19] S. Eivazi and R. Bednarik. Predicting Problem-Solving Behavior and Performance Levels from Visual Attention Data. pages 9–16, 2011.

[20] I. M. Ekman, A. W. Poikola, and M. K. Mäkäräinen. Invisible eni: using gaze and pupil size to control a game. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI EA '08, pages 3135–3140, New York, NY, USA, 2008. ACM.

[21] C. Elkan. Predictive analytics and data mining, 2010.

[22] A. Gail, J. Hansen, M. Lillholm, and E. Mollenbach. Single gaze gestures. In *ETRA '10 Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications*, pages 177–180, 2010. ISBN 978-1-60558-994-7.

[23] A. Gale, J. Hansen, M. Lillholm, and E. Mollenbach. Single stroke gaze gestures. In *CHI '09 Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4555–4560, 2009. ISBN: 978-1-60558-247-4.

[24] T. Gowases. Gaze vs. mouse: An evaluation of user experience and planning in problem solving games. Master's thesis, University of Joensuu, 2007.

[25] H. Heikkilä and K. J. Räihä. Speed and accuracy of gaze gestures. *Journal of Eye Movement Research*, 3(2):1–14, 2009.

[26] J.-M. Hupé, C. Lamirel, and J. Lorenceau. Pupil dynamics during bistable motion perception. *Journal of vision*, 9(7):10, Jan. 2009.

[27] A. Hyrskykari, L. Immonen, H. Istance, S. Mansikkamaa, and S. Vickers. Designing gaze gestures for gaming: an investigation of performance. In *ETRA '10 Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications*, pages 323–330. ACM New York, NY, USA, 2010. ISBN 978-1-60558-994-7.

[28] S. Iqbal, P. Adamczyk, and X. Zheng. Changes in mental workload during task execution. *Proceedings of the 17th*, pages 1–2, 2004.

[29] O. Ivanciuc. Applications of Support Vector Machines in Chemistry. *Biochemistry*, 23:291–400, 2007.

[30] R. J. K. Jacob. What you look at is what you get: Eye movement-based interaction techniques. In *ACM CHI '90: Conference on Human Factors in Computing Systems*, pages 11–18, 1990. ISBN: 0-201-50932-6.

[31] R. J. K. Jacob and K. S. Karn. Commentary on section 4. eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, pages 573–605. Elsevier Science, 2003.

[32] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–2011.

[33] D. Kahneman. *Attention and effort*. Englewood Cliffs, Nj: Prentice-Hall, 1973.

[34] J. Klingner. *Fixation-aligned pupillary response averaging*, volume 1. ACM Press, New York, New York, USA, 2010.

[35] H. Koesling, A. Kenny, A. Finke, H. Ritter, S. McLoone, and T. Ward. Towards intelligent user interfaces: anticipating actions in computer games. In *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, NGCA '11, pages 4:1–4:8, New York, NY, USA, 2011. ACM.

[36] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale: Rapid prototyping for complex data mining tasks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA, 2006. ACM.

[37] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-time signal processing (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1999.

[38] T. Partala and V. Surakka. Pupil size variation as an indication of affective processing. *International Journal of Human-Computer Studies*, 59(1-2):185 – 198, 2003. Applications of Affective Computing in Human-Computer Interaction.

[39] K. Perlin. Quikwriting: continuous stylus-based text entry. In *UIST '98 Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 215–216, 1998. ISBN:1-58113-034-1.

[40] F. Richer and J. Beatty. Pupillary dilations in movement preparation and execution. *Psychophysiology*, 22(2):204–207, 1985.

[41] H. Sakoe and S. Chiba. Readings in speech recognition. chapter Dynamic programming algorithm optimization for spoken word recognition, pages 159–165. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[42] Tobii Technology AB. Clearview 2.7 eye gaze analysis software. http://www.tobii.com/scientific_research/, 2008.

[43] Tobii Technology AB. Tobii api overview. http://www.tobii.com/scientific_research/, 2008.

[44] Tobii Technology AB. Tobii sdk. http://www.tobii.com/scientific_research/, 2008.

[45] University of Tampere. *Eye-Tracking Universal Driver (ETUDriver, COM interface description, Version 1.25*, 2006.

[46] S. A. T. W. H. Press, B. P. Flannery and W. T. Vetterling. *Numerical Recipes – The Art of Scientific Computing.* Cambridge: Cambridge University Press, 1986.

[47] E. Wästlund, K. Sponseller, and O. Pettersson. *What you see is where you go: testing a gaze-driven power wheelchair for individuals with severe multiple disabilities*, pages 133–136. ACM, 2010.

Appendix A

Content of the CD

A CD-ROM included in this thesis contains following folder and files:

- **Feature extraction** folder contains set of Python scripts which were used for fixation sequence parsing and computing of output feature sets. For script utilization, Numpy, Scipy and Matplotlib libraries needs to be installed at your computer.
- **Experiments and datasets** folder consist of used feature sets and related RapidMiner scripts. Each experiment was adjusted for the input set which is saved in the data-folder and output of experiments will appear in results-folder.
- **Meausurements** folder consists of resulting figures in higher resolution and tables with measurements, equations and charts.

Appendix B

Measurements and pupillary response figures

This chapter contains all measured performances, illustrated in Table B1.

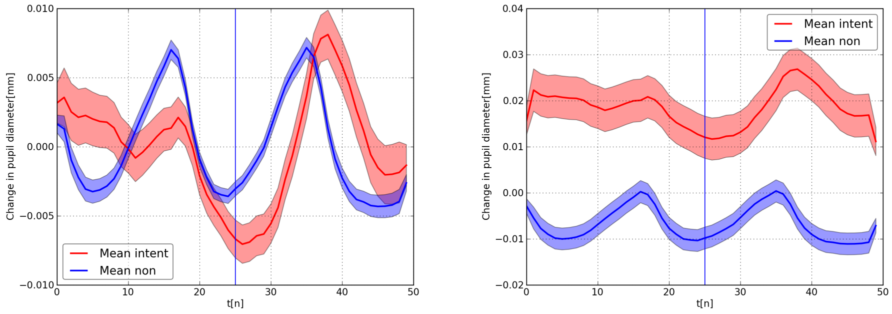
Following illustrations B.1, B.2 and B.3 provides visual compartments between types of normalizations and units over which normalizations were performed. Figures consist of mean representation of intentional eye movemenets sequences, a red line, as well as non-intentional ones, a blue line.

The last set of figures B.4, B.5 and B.6 contains mean pupillary dilations of intentional eye movement sequences, a red line, compared to the level of noise. Amount of noise were performed as plus-minus average of all intentional sequences and is shown as a grey line.

Table B.1: Results of measurements

Type of feature	Type of normalization	Unit of normalization	Accuracy	AUC	FPR	FNR	TPR	TNR	Precision	Recall	F-measure	G-mean1	G-Mean2
spectrum	baseline subtraction	Sequence	-	-	-	-	-	-	-	-	-	-	-
		Dataset	-	-	-	-	-	-	-	-	-	-	-
	PCPS	Sequence	-	-	-	-	-	-	-	-	-	-	-
		Dataset	-	-	-	-	-	-	-	-	-	-	-
	Z-score	Sequence	-	-	-	-	-	-	-	-	-	-	-
		Dataset	-	-	-	-	-	-	-	-	-	-	-
	Average	Sequence	-	-	-	-	-	-	-	-	-	-	-
	Average	Dataset	-	-	-	-	-	-	-	-	-	-	-
	Total Average	-	-	-	-	-	-	-	-	-	-	-	-
cepstrum	baseline subtraction	Sequence	-	-	-	-	-	-	-	-	-	-	-
		Dataset	-	-	-	-	-	-	-	-	-	-	-
	PCPS	Sequence	-	-	-	-	-	-	-	-	-	-	-
		Dataset	-	-	-	-	-	-	-	-	-	-	-
	Z-score	Sequence	-	-	-	-	-	-	-	-	-	-	-
		Dataset	-	-	-	-	-	-	-	-	-	-	-
	Average	Sequence	-	-	-	-	-	-	-	-	-	-	-
	Average	Dataset	-	-	-	-	-	-	-	-	-	-	-
	Total Average	-	-	-	-	-	-	-	-	-	-	-	-
1st difference	baseline subtraction	Sequence	0.278	0.541	0.810	0.130	0.870	0.190	0.138	0.870	0.238	0.346	0.406
		Dataset	0.260	0.541	0.834	**0.106**	**0.894**	0.166	0.137	**0.894**	0.238	**0.351**	0.385
	PCPS	Sequence	0.365	0.554	0.698	0.210	0.790	0.302	0.144	0.790	0.244	0.337	0.489
		Dataset	0.392	0.553	0.662	0.240	0.760	0.338	0.146	0.760	0.245	0.333	0.507
	Z-score	Sequence	**0.432**	**0.572**	**0.613**	0.268	0.732	**0.387**	**0.151**	0.732	**0.250**	0.332	**0.532**
		Dataset	0.329	0.541	0.741	0.200	0.800	0.259	0.138	0.800	0.236	0.333	0.455
	Average	Sequence	**0.358**	**0.556**	**0.707**	0.203	0.797	**0.293**	**0.144**	0.797	**0.244**	0.338	**0.476**
	Average	Dataset	0.327	0.545	0.746	**0.182**	**0.818**	0.254	0.141	**0.818**	0.240	**0.339**	0.449
	Total Average	-	0.343	0.550	0.726	0.192	0.808	0.274	0.142	0.808	0.242	**0.339**	0.462
2nd difference	baseline subtraction	Sequence	0.479	0.598	0.554	0.299	0.701	0.446	0.158	0.701	0.258	0.333	0.559
		Dataset	0.481	0.596	0.552	0.299	0.701	0.448	0.159	0.701	0.259	0.334	0.561
	PCPS	Sequence	**0.510**	0.540	**0.517**	0.303	0.697	**0.483**	**0.167**	0.697	**0.269**	0.341	**0.580**
		Dataset	0.493	**0.604**	0.539	0.294	0.706	0.461	0.163	0.706	0.265	0.339	0.571
	Z-score	Sequence	0.457	0.601	0.586	0.253	0.747	0.414	0.159	0.747	0.263	**0.345**	0.556
		Dataset	0.412	0.589	0.643	**0.221**	**0.779**	0.357	0.153	**0.779**	0.255	**0.345**	0.527
	Average	Sequence	**0.482**	0.580	**0.552**	0.285	0.715	**0.448**	**0.161**	0.715	**0.263**	**0.340**	**0.565**
	Average	Dataset	0.462	**0.596**	0.578	**0.271**	**0.729**	0.422	0.158	**0.729**	0.260	0.339	0.553
	Total Average	-	0.472	0.588	0.565	0.278	0.722	0.435	0.160	0.722	0.262	0.339	0.559
DTW	baseline subtraction	Sequence	0.201	0.501	0.909	0.056	0.944	0.091	0.134	0.944	0.234	0.355	0.293
		Dataset	0.207	0.510	0.899	0.080	0.920	0.101	0.132	0.920	0.231	0.348	0.304
	PCPS	Sequence	0.282	0.540	0.732	0.500	0.500	0.268	0.042	0.500	0.077	0.145	0.366
		Dataset	0.317	0.540	**0.692**	0.500	0.500	**0.308**	0.036	0.500	0.067	0.134	0.392
	Z-score	Sequence	**0.351**	**0.562**	0.718	0.183	0.817	0.282	**0.145**	0.817	**0.246**	0.344	**0.480**
		Dataset	0.176	0.501	0.941	**0.037**	**0.963**	0.059	0.132	**0.963**	0.232	**0.357**	0.238
	Average	Sequence	**0.278**	**0.534**	**0.786**	0.247	0.753	**0.214**	**0.107**	0.753	**0.186**	**0.281**	**0.380**
	Average	Dataset	0.233	0.517	0.844	**0.206**	**0.794**	0.156	0.100	**0.794**	0.177	0.280	0.311
	Total Average	-	0.256	0.526	0.815	0.226	0.774	0.185	0.103	0.774	0.181	0.280	0.346
APCPS	baseline subtraction	Sequence	0.257	0.570	0.831	**0.157**	**0.843**	0.169	0.131	**0.843**	0.227	**0.333**	0.378
		Dataset	**0.517**	0.577	**0.481**	0.496	0.504	**0.519**	**0.135**	0.504	0.213	0.261	**0.512**
	PCPS	Sequence	0.314	**0.624**	0.758	0.202	0.798	0.242	0.135	0.798	**0.231**	0.328	0.439
		Dataset	**0.517**	0.577	**0.481**	0.496	0.504	**0.519**	**0.135**	0.504	0.213	0.261	**0.512**
	Z-score	Sequence	0.296	0.530	0.776	0.215	0.785	0.224	0.131	0.785	0.224	0.320	0.419
		Dataset	**0.517**	0.577	**0.481**	0.496	0.504	**0.519**	**0.135**	0.504	0.213	0.261	**0.512**
	Average	Sequence	0.289	0.575	0.788	**0.191**	**0.809**	0.212	0.132	**0.809**	**0.227**	**0.327**	0.412
	Average	Dataset	**0.517**	**0.577**	**0.481**	0.496	0.504	**0.519**	**0.135**	0.504	0.213	0.261	**0.512**
	Total Average	-	0.403	0.576	0.635	0.343	0.657	0.365	0.134	0.657	0.220	0.294	0.462

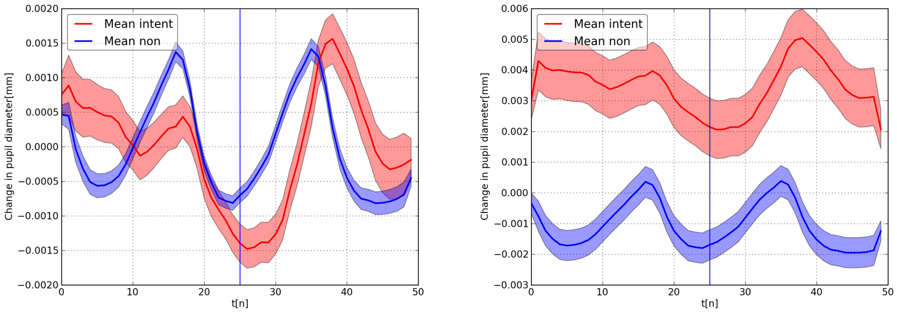
Type of feature	Type of normalization	Sequence / Dataset	Accuracy	AUC	FPR	FNR	TPR	TNR	Precision	Recall	F-measure	G-mean1	G-Mean2
Fixation, saccades	Z-transformation	-	0.751	0.799	0.240	0.305	0.695	0.760	0.301	0.695	0.420	0.457	0.727
+APCPS	Z-transformation	Dataset	0.758	0.806	**0.232**	0.308	0.692	**0.768**	0.307	0.692	0.425	0.461	0.729
+DTW+APCPS	Z-transformation	Dataset/Sequence	**0.759**	**0.807**	**0.232**	0.306	0.694	**0.768**	**0.308**	0.694	**0.427**	**0.462**	**0.730**
+DTW	Z-transformation		0.753	0.800	0.239	**0.302**	0.698	0.761	0.303	0.698	0.423	0.460	0.729
+DIFF2	Z-transformation	Dataset	0.743	0.799	0.250	0.300	**0.700**	0.750	0.294	**0.700**	0.414	0.453	0.724
	Average	-	0.753	0.802	0.239	0.304	0.696	0.761	0.302	0.696	0.422	0.459	0.728



(a) Sequence

(b) Dataset

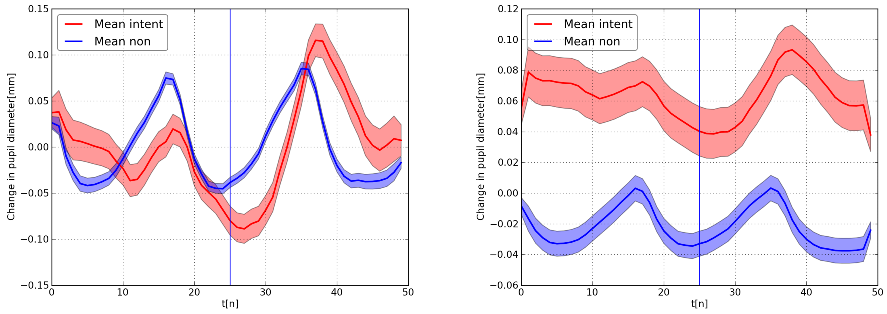
Figure B.1: Baseline subtratction



(a) Sequence

(b) Dataset

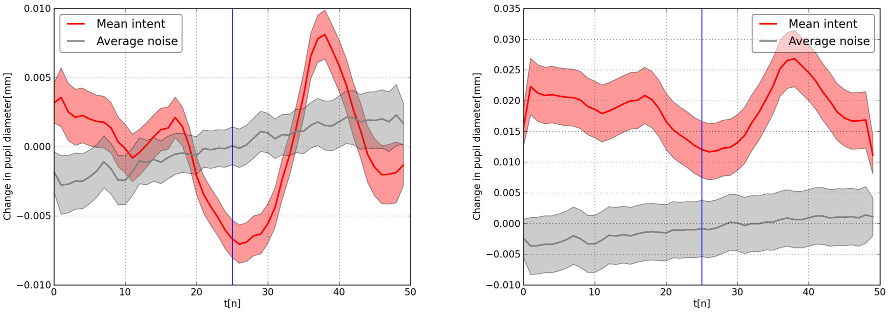
Figure B.2: PCPS



(a) Sequence

(b) Dataset

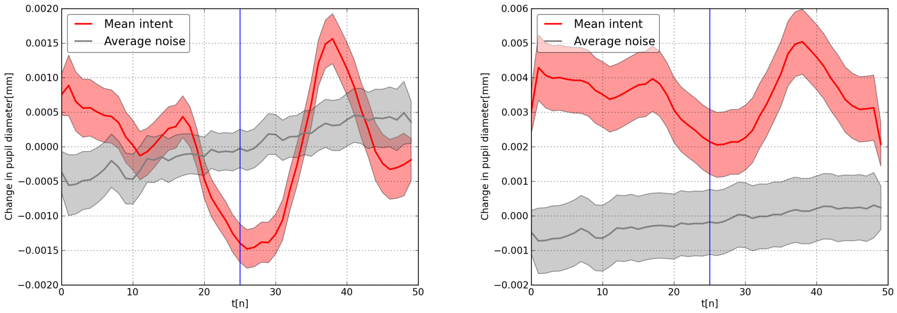
Figure B.3: Z-score



(a) Sequence

(b) Dataset

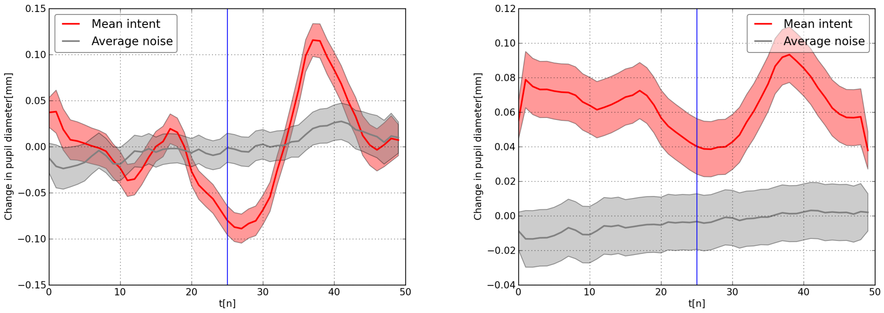
Figure B.4: Baseline subtraction, plus-minus average



(a) Sequence

(b) Dataset

Figure B.5: PCPS, plus-minus average



(a) Sequence

(b) Dataset

Figure B.6: Z-score, plus-minus average