

Ophthalmic Diagnostics Using Eye Tracking Technology

RAFAEL ALDANA PULIDO



**KTH Information and
Communication Technology**

Degree project in
Communication Systems
Second level, 30.0 HEC
Stockholm, Sweden

Ophthalmic Diagnostics Using Eye Tracking Technology

Master of Science Thesis

Rafael Aldana Pulido
rafaelap@kth.se

Supervisor: Gustaf Öqvist Seimyr
Examiner: Prof. Gerald Q. Maguire Jr.

School of Information and Communication Technology (ICT)
KTH Royal Institute of Technology
Stockholm, Sweden

February 15, 2012

Abstract

Human eyes directly reflect brain activity and cognition. The study of eye movements and gaze patterns can therefore say a lot about the human brain and human behavior. Today eye tracking technology is being used to measure acuity of toddlers, to rehabilitate patients in intensive care, to detect if a person is lying or not, and to understand the cognitive level of a non-verbal person.

Current vision testing is mostly based on manual observation and subjective methods. With eye tracking technology eye movements can be tested in an automated way that increases reliability and reduces variability and subjectivity.

Eye tracking technology allows for measuring eye movements and therefore quantitative comparisons of the progress in treatment can be made over the course of a patient's therapy – enabling more effective therapy. This technology also enables standardized and automated processes that are more time- and cost-efficient. The most important advantages of this technology is that it is non-invasive and it is not necessary to stabilize the subject's head during testing. These advantages greatly extend the set of subjects that can be studied and reduce the cost and skills required for studying eye movements and gaze patterns.

This thesis has developed and evaluated a novel calibration procedure for an eye tracker. The development phase has included programming and integration with the existing application programming interfaces. The evaluation phase included reliability and validity testing, as well as statistical analysis in terms of repeatability, objectivity, comprehension, relevance, and independence of the performance of the *Tobii T60/T120 Eye Tracker* on *healthy* subjects.

The experimental results have shown that the prototype application gives the expected benefits in a *clinical* setting. A conclusion of this thesis is that eye tracking technology could be an improvement over existing methods for screening of eye alignment and diagnostics of ophthalmic disorders, such as strabismus (crossed eyes) or amblyopia (lazy eye). However, applying this technology to clinical cases will require further development. This development is suggested as future work.

Sammanfattning

En människas ögon speglar direkt hennes hjärnaktivitet och kognition. Att undersöka ögonens rörelsemönster kan därför säga mycket om den mänskliga hjärnan och om mänskligt beteende. Idag används eye tracking-teknik för att mäta synskärpa hos småbarn, för att rehabilitera patienter i intensivvård, för att upptäcka om en person talar sanning eller ej, och för att utvärdera den kognitiva nivån hos icke-verbala personer.

Nuvarande syntest bygger främst på manuella observationer och subjektiva metoder. Med eye-tracking teknik kan ögonens rörelsemönster testas på ett automatiserat sätt som ökar tillförlitligheten och minskar variabilitet och subjektivitet.

Eye tracking-tekniken tillåter kvantifiering av rörelsemönstrena och möjliggör jämförelser samt uppföljning av progression av behandlingar på ett effektivare sätt. Tekniken möjliggör även standardiserade och automatiserade processer som är mer tids- och kostnadseffektiva. De mest intressanta fördelarna med denna teknik är att den är icke-invasiv och att det inte är nödvändigt att stabilisera subjektets huvud under testningen.

I denna tes har en ny metod för att kalibrera eye-tracking utrusning utvecklats och utvärderats. Utvecklingsfasen bestod bland annat av programmering samt integration med befintliga API. I utvärderingsfasen ingick testning av reliabilitet och validitets samt statistisk analys för att undersöka repeterbarhet, objektivitet, förståelse, relevans och oberoende av *Tobii T60/T120 Eye Tracker* på friska försökspersoner.

Resultat av experiment har visat att prototypen ger de förväntade fördelarna i en klinisk miljö. En slutsats av denna avhandling är att eye tracking-teknik kan förbättra befintliga metoder för screening av ögats anpassning och för diagnostik av oftalmologiska sjukdomar, till exempel skelning eller amblyopi, om tekniken ytterligare utvecklas och förbättras.

Acknowledgements

I would like to sincerely thank my supervisors Jenny Grant, Peter Tiberg, and Gustaf Öqvist Seimyr. They introduced me to an interesting medical application of software engineering and shared their good experience with me. We have had interesting discussions and they have provided feedback during the project, which I have enjoyed so much.

I would also like to thank my supervisor at KTH, Professor Gerald Q. Maguire Jr. I am glad he accepted to be my academic examiner since he considered this project very interesting. Gerald's guidance has also been essential in some steps of this thesis, such as the testing procedure of the prototype application and the analysis and presentation of results.

There are more people who also deserve great thanks. They are my colleagues at *Tobii Technology*, all the involved people from *Karolinska Institutet* and *S:t Eriks Ögonsjukhus*, and all the test participants.

Finally I would like to thank my friends and my family, all of whom have been encouraging me during my stay in Stockholm.

Thank you all.

Contents

Contents	vii
List of Acronyms and Abbreviations	viii
List of Code Listings	x
List of Figures	xi
List of Tables	xii
1 Introduction	1
1.1 The problem statement	2
1.2 Goals	2
1.3 Limitations	3
1.4 Audience	4
1.5 Organization of the thesis	4
1.5.1 Research	5
1.5.2 Development and prototype building	5
1.5.3 Structure of this report	6
2 Strabismus and amblyopia	9
2.1 Concomitant strabismus	10
2.2 Incomitant strabismus	10
2.3 Amblyopia	11
3 Previous work	13
3.1 Scleral search coil technique	13
3.2 Image-processing techniques, video-oculography system	14
3.3 Performance evaluations of remote and binocular 2D eye tracking systems	15
3.4 Study of errors introduced by eye tracking systems	16
4 Hardware and experimental setup	17
4.1 The <i>IEyetracker</i> interface	19

4.2	Coordinate system	21
4.3	Available gaze data	22
4.4	Configuring the screen	26
4.5	Calibration	28
4.5.1	Calibration procedure	28
4.5.2	Calibration plot	29
4.5.3	Calibration buffers	30
4.6	Tracking	32
4.7	Magnitudes and units	32
4.8	Test procedure	35
4.8.1	Environment	35
4.8.2	Equipment	37
4.8.3	Workflow	38
5	Description of the prototype application	41
5.1	Specification of the desired report	41
5.2	Workflow of <i>Tobii Alignment</i>	44
5.2.1	Calibration routines	47
5.2.2	Tracking routine	52
6	Evaluation	53
6.1	Study I – Measurement of <i>2D gaze points</i> on the tracking area	53
6.2	Study II – Measurement of angles of deviation between gaze and target vectors	56
6.3	Study III – Tobii Alignment validation against scleral search coil technique	66
6.3.1	Principles of operation of a scleral search coil	66
6.3.2	Validation procedure	69
6.3.3	Transformation from voltage to angle of deviation	70
6.3.4	Interpolation and decimation of a set of samples	71
6.3.5	Correlation between the <i>Tobii T120 Eye Tracker</i> and SSC performances	74
6.3.6	Graphical comparison between the <i>Tobii T120 Eye Tracker</i> and SSC performances	77
6.3.7	Influence of the changes in the pupil size	88
6.3.8	Discussion	89
7	Conclusions and Future work	91
7.1	Discussion of the results	91
7.2	Future work	92
	Bibliography	95
A	A diagram of the human eye	99

List of Acronyms and Abbreviations

API	Application Programming Interface
CALIB	calibration data file
cm	centimeter
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
ESO	esotropia
ETT	Eye Tracking Technology
EXE	executable application
EXO	exotropia
FFT	Fast Fourier Transform
GB	gigabyte
GHz	gigahertz
HYPO	hypotropia
HYPER	hypertropia
Hz	hertz
i.e.	that is
IP	Internet Protocol
IR	infrared
kHz	kilohertz
LAN	Local Area Network
mA	milliampere
Mbps	megabit per second
ms	millisecond
nm	nanometer
OD	Oculus Dexter, right eye
ORTHO	orthotropia
OS	Oculus Sinister, left eye
PC	Personal Computer
rms	Root Mean Square
SDK	Software Development Kit
s	second
SSC	Scleral Search Coil

TFT	Thin-Film Transistor
TRACK	tracking data file
UCS	User Coordinate System
USB	Universal Serial Bus
VGA	Video Graphics Array
VOG	video-oculography

List of Code Listings

4.1	The <i>IEyetracker</i> interface [36].	20
4.2	The <i>Point2D</i> class [36].	22
4.3	The <i>Point3D</i> class [36].	22
4.4	The <i>GazeDataItem</i> class [36].	23
4.5	The <i>XConfiguration</i> class [36].	27
4.6	The <i>CalibrationPlotItem</i> class [36].	31
4.7	The <i>Calibration</i> class [36].	32
4.8	Precision of angular magnitude.	34
5.1	The <i>binocular calibration</i> routine	48
5.2	The <i>monocular calibration on the left eye</i> routine	49
5.3	The <i>monocular calibration on the right eye</i> routine	50
5.4	The <i>Merge Calibrations</i> routine	51
5.5	The <i>Start Tracking</i> routine	52
6.1	Transformation from voltage to angle of deviation.	70
6.2	Linear interpolation.	72
6.3	Sampling.	73
6.4	Correlation in the changes in the signals.	75
6.5	Calculation of the correlation in the changes in two signals.	76
6.6	Calculation of the average discarding not valid samples.	76

List of Figures

1.1	The planned development process.	4
2.1	Graphical examples of concomitant strabismus.	10
3.1	The scleral search coil setting. Picture taken at <i>Bernadottelaboratoriet</i> at <i>S:t Eriks Ögonsjukhus</i> during the <i>Tobii Alignment</i> validation against SSC technique.	14
4.1	Pupil center corneal reflection. Adapted from [6].	18
4.2	A drawing of two subjects with bright and dark pupils each one. Adapted from [6].	18
4.3	Synchronization scenario. Adapted from [34, figure 12].	19
4.4	The <i>User Coordinate System</i> on a <i>Tobii T60/T120 Eye Tracker</i> . Adapted from [34, figure 6].	21
4.5	Schematic head movement box. Adapted from [34, figure 16].	24
4.6	Gaze point and gaze vector. Adapted from [34, figure 7].	24
4.7	Screen configuration schematic. Adapted from [34, figure 17].	26
4.8	A 9-point calibration plot example with data for the left eye.	30
4.9	A 9-point calibration plot example with data for the right eye.	31
4.10	Angle of deviation decomposed into ϕ and θ . Adapted from [2, p. 6]. . .	33
4.11	The room used for the first two sets of tests.	36
4.12	The location of the light sources. Adapted from [2, figure 22].	37
4.13	A light meter is required in order to determine the illumination in the test room. Adapted from [2, figure 22].	38
4.14	The <i>Eyetrackers Found on the Network</i> and the <i>Eyetracker Status</i> boxes.	40
5.1	The 9-point calibration pattern. The points are randomly shown one by one.	42
5.2	A graphical generalization of the <i>Tobii Alignment</i> report.	44
5.3	The basic screen layout of <i>Tobii Alignment</i>	45
5.4	An example of the appearance of the <i>Gaze Plotter</i> . Note that boxes of the same row have the same origin in the horizontal axis, whereas boxes of the same column have the same origin in the vertical axis.	47

6.1	Monocular calibration on subject #4 with 750 nm wavelength IR transparent lens.	54
6.2	Monocular calibration on subject #11 with 800 nm wavelength IR transparent lens.	54
6.3	Binocular calibration on subject #10.	55
6.4	Average ϕ_L deviations for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.	57
6.5	Average θ_L deviations for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.	58
6.6	Average ϕ_R deviations for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.	59
6.7	Average θ_R deviations for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.	60
6.8	Average <i>vergence</i> angle ($\phi_L - \phi_R$) for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.	61
6.9	Linear regression for binocular calibration. All measurements are expressed in degrees.	63
6.10	Linear regression for MONO LR calibration. All measurements are expressed in degrees.	64
6.11	Linear regression for MONO RL calibration. All measurements are expressed in degrees.	65
6.12	Relationship between samples in a linear interpolator for $L = 3$	72
6.13	Impulse response of a linear interpolator for $L = 3$	73
6.14	Effect of a sampler for $M = 8$	74
6.15	Correlation between both measurement techniques.	77
6.16	Binocular calibration and ϕ tracking on subject #1.	78
6.17	Binocular calibration abd θ tracking on subject #1.	79
6.18	Monocular calibration and ϕ tracking on subject #1.	80
6.19	Monocular calibration abd θ tracking on subject #1.	81
6.20	<i>Vergence</i> angle for subject #1.	82
6.21	Binocular calibration and ϕ tracking on subject #2.	83
6.22	Binocular calibration abd θ tracking on subject #2.	84
6.23	Monocular calibration and ϕ tracking on subject #2.	85
6.24	Monocular calibration abd θ tracking on subject #2.	86
6.25	<i>Vergence</i> angle for subject #2.	87
6.26	Comparison between the original and the <i>modified</i> T120 outputs.	88
A.1	A diagram of the human eye [14].	99

List of Tables

4.1	Data validity codes.	25
5.1	The location of the calibration points in 2D and 3D coordinates.	42
5.2	The location of the calibration points in 2D coordinates and degrees.	43
6.1	Mean and standard deviation values with respect to the corresponding target point for the nine cardinal positions and all the subjects. Values in red are out of the desired range (a deviation within the order of magnitude of a microsaccade may be ignored, and the implication of a positive <i>vergence</i> value is <i>crossed fixation</i>).	62
6.2	Binocular calibration procedure in a SSC setting.	69
6.3	Monocular calibration procedure in a SSC setting.	69
6.4	Correlation between both measurement techniques.	77
6.5	Mean difference between both measurements expressed in degrees.	89

Chapter 1

Introduction

Today eye tracking is a very innovative technology by which computers are able to estimate with high accuracy where users are looking. Eye tracking technology (ETT) can be used in both analytical and interactive applications. Some of these applications are assessments, diagnostics, and rehabilitation within the field of clinical ophthalmology.

An eye tracker estimates the point of gaze with high accuracy using image sensor technology and mathematical algorithms that find the subject's eyes. In other words, the tracker works much like an ophthalmologist would if he/she face a patient and estimates where the subject is looking just by observing the subject's eyes.

For clinical use there is a range of eye movement data that contains valuable diagnostic information. This eye movement data can be objectively quantified and compared to published data. Deviations that are related to a particular diagnosis or impairment(s) can be identified.

In the field of clinical ophthalmology applying ETT at specialist centers or optometrists can provide a superior way of studying ocular motor behavior and vision deficiencies in comparison to current clinical techniques. The quantifiable data could include:

- Saccadic performance, such as latency, overshoot, and undershoot.
- Fixation stability, such as drift.
- Smooth pursuit, such as asymmetries between eye movement and stimuli.
- Alignment, such as strabismus (uncontrolled inward or outward eye movement) or amblyopia (lazy eye) – further details are given in chapter 2.
- Nystagmus (rapid eye movements), such as the slope of the slow phase and amplitude of the fast phase – details of these phases can be found in [19, p. 28].

1.1 The problem statement

In the field of ophthalmology, oculomotor behavior, cognitive visual function, and vision deficiencies have been studied from a *qualitative* eye-movement analysis approach. In consequence, methodologies that rely on observation have been manual and variable, generally resulting in the identification of disease taking place at a late stage of disease progression.

As the dynamics of eye movement have been subjectively and qualitatively analyzed, it has been hard to compare such data to published data or deviations related to a particular diagnosis. As a result, impairments can not be easily identified.

Eye movement studies in a subjective and manual way are variable and unreliable. Standardized but manual processes are not time- or cost-efficient. In the same way, comparison, follow-up, and progression studies also need to be automated so that they too can be more effective.

The method that has been categorized as the gold standard is the scleral search coil technique (SSC) [22, p. 137-145]. SSC provides an accurate, but expensive and invasive method to study eye movements and eye movement deficits. Moreover, the subject is required to place their chin on a chin rest in order to be stabilized during testing, what limits head movement. The result is a neither natural nor relaxed test situation, this is especially importunate in research or diagnoses that involve children.

1.2 Goals

This thesis was motivated by the following goals:

- **Study of true eye movements**

An eye tracker measures at high sampling rates compared to a microsaccade¹ interval both the direction of gaze as well as the position of the eyes in space with high accuracy. This enables the calculation of head movements and, by extension, calculation of the true eye movements within the orbit of the eye.

- **Quantification of eye movement data**

ETT allow ophthalmologists to *quantify* the dynamics of eye movement data. This is useful when studying eye movement data as this data that contains valuable diagnostic information. The possibility of objectively quantifying such data facilitates comparison to published data and deviations related to a particular diagnosis or that identification of specific impairments.

¹*Microsaccades* are small and involuntary eye movements that move the gaze back to the target after a drift. They can be as large as 1 degree and last for 0.3 seconds [18].

- **Reduction of variability and increased of efficiency**

ETT enables eye movement studies in an objective and automated way that increases reliability and reduces variability. In vision research, as well as in clinical assessments of eye movements, automated processes could replace manual observation. ETT can be used to implement standardized and automated processes which are more time- and cost-efficient. The ability to quantify eye movement data makes comparison, follow-up, and progression studies more effective.

- **Provision of a non-invasive method with improved validity**

ETT does not require placing any equipment on the subject's head and allows for a great deal of free head movement. The result is a natural and relaxed test situation, which is imperative in research or diagnoses that involve children.

In order to reach these goals, the following steps involved were undertaken in this thesis:

- Investigation of the requirements for a clinical tool to test alignment with an eye tracker.
- Development of the required calibration procedure of an eye tracker and a prototype application, *Tobii Alignment*, programmed in C#. The application is self instructive and very simple to use in a clinical situation.
- Evaluation of the prototype in a clinical environment for a *Tobii T60/T120 Eye Tracker*. This lead to a verification of the expected benefits.

1.3 Limitations

This thesis excludes:

- Market research

However, the incidence of strabismus is 5-8% in the general population [10, p. 96] and amblyopia has been estimated to affect between 1% and 5% of the world population [3, p. 365–375], hence there should be a substantial potential market in the area of clinical alignment testing methods.

- Cost analysis

The cost of a clinically suitable eye tracker has not been estimated. The cost of the necessary development work (either in this thesis or for any further development) has not be estimated.

- A complete description of the ophthalmic disorders considered in this work

This thesis focuses on determining the new methods potential to address a specification defined in conjunction with experts and clinicians in the area at *Karolinska Institutet*. Although some information about these disorders is given in chapter 2 to facilitate the reader's understanding of this thesis – details of the ophthalmic disorders are outside the scope of this work.

- Adjustments for different eye trackers

The application developed in this thesis works **only** with *Tobii Technology's* eye trackers. Technology provided access to their code and experts to enable the development of this application.

- Evaluation of the prototype system in a clinical environment with patients

This thesis focuses on the development of *a new method* to provide repeatability, objectivity, comprehension, relevance, and independence. An evaluation with patients was outside the scope of the project's definition.

1.4 Audience

Vision researchers and software engineers interested in developing applications for ETT hardware for clinical purposes could benefit from this thesis. This thesis may also be interesting for others who may wish to use ETT in their applications, such as for games or assessing if the user is becoming tired.

1.5 Organization of the thesis

The overall development process for the prototype in this thesis is illustrated in figure 1.1.

ID	Apr 2011		May 2011		Jun 2011		Jul 2011		Aug 2011		Sep 2011		Oct 2011	
	1—15	16—30	1—15	16—31	1—15	16—30	1—15	16—31	1—16	16—31	1—15	16—30	1—15	16—31
1	Planning													
2		Method												
3			Literature study and research											
4			Development											
5					Prototype									
6									Testing and evaluation					
7													Documentation	

Figure 1.1: The planned development process.

1.5.1 Research

Some limited research in ophthalmology, vision test methods, and eye tracking has been performed to gather background information about ophthalmology in general, vision evaluation tests, and gaze tracking. This has been done by reading relevant literature, which is included in the bibliography, learning about the vision test procedures at *Berdanottelaboratoriet* at *S:t Eriks Ögonsjukhus*, and asking questions of the staff at *Karolinska Institutet* and *Tobii Technology*.

1.5.2 Development and prototype building

The prototype development process was divided into three major parts because of the vision test methodology: operant conditioning, testing, and showing results.

Requirement analysis – *Specifying the report that the application must return*

A few requirements were already defined by researchers at *Karolinska Institutet* before this thesis project started had a clear vision of what they wanted from the new vision test. This new test was supposed to produce at least the same results as the tests that are used today. Therefore, a small study of the methodology that is currently being used was performed. The requirements were modified a bit during the application development. The final requirement specification is given in chapter 5.

Design – *Specifying what the parts will be and how they will fit together*

Some sketches and sequence diagrams were made before the start of the development phase. The sequence diagrams were used to lay out the parts of the vision test and how they should be built. See section 5.2.

Implementation – *Writing the code*

The prototype application has been developed in *Microsoft Visual Studio 2010*, using *C# .NET* as the programming language. The *Tobii Eye Tracker Software Development Kit* (SDK) has been used to interface with the eye tracker during the test. The prototype was tested and evaluated as it was being developed. Finally, the entire prototype was tested and evaluated against the gold standard method. See sections 3.1, 5.2, and 6.3.

Testing – *Executing the application with test data for input*

The prototype was tested during development with three sets of tests. First, tests were performed on sixteen adults to see if the tests would produce the same results when replicated. In this testing there was no calculation of angle of deviation, the results were simply the *2D gaze point* sets as output by the SDK. Subsequently, tests were performed on twelve adults, most of whom had participated in the previous test. This testing included calculation of angle of deviation. Finally, tests were performed on two adults in a SSC setting to see if there was correlation between the gaze data returned by both procedures. The first two sets of tests

were performed at *Tobii Technology* and the final set of tests were performed at *Berdanottelaboratoriet* at *S:t Eriks Ögonsjukhus*. The conditions of these tests are explained in section 4.8.

Evaluation – *Study of reliability and validity as well as statistical analysis*

The results have been analyzed using *MATLAB R2010b*. All the gaze data were analyzed to compute the mean and standard deviation values for each eye and the target point in every isolated test. The analysis shows that the tests produce the same results when replicated on the same subject when considering both 2D gaze points got from the SDK and angles of deviation obtained from computing gaze and target vectors. See chapter 6.

Maintenance – *Repairing software defects and adding capability*

Software maintenance has been performed to a limited extent during the development. Error testing of the different parts of the prototype has been performed in sufficiently for a complete prototype, but the complete user interface has not been tested. The development and testing of a more complete user interface will mostly occur after this thesis project, therefore further details of this user interface are excluded from the thesis.

1.5.3 Structure of this report

Chapter 2 outlines the various types of strabismus and ocular motility disturbances that have been considered in this thesis. It contains basic information on each topic, but excludes in-depth discussion as this is beyond the scope of this work – for details the reader is referred to [3, 4, 5, 7, 10, 15, 23, 27, 29, 41].

Chapter 3 reports on previous work which has been done in the area of alignment testing using ETT, including verification of system performance and methods for data analysis. SSC, the method considered to be the gold standard, is presented here.

Chapter 4 introduces the *Tobii T60/T120 Eye Tracker* and its SDK. This chapter includes a description of the magnitudes utilized and their units as well as the input and output parameters.

Chapter 5 presents in detail the desired report that the software should output. This report's contents were defined in the beginning of the thesis in conjunction with experts and clinicians in the area at *Karolinska Institutet*. The prototype system architecture designed and implemented during this thesis is explained in this chapter.

Chapter 6 contains the evaluation performed for the system architecture and prototype presented in chapter 5. Three different sets of tests were made.

Chapter 7 presents a discussion of the results presented in chapter 6 and draws some conclusions. Finally, the excluded scope of this thesis is re-considered and presented as suggestions for future work.

Chapter 2

Strabismus and amblyopia

This chapter outlines the various types of ophthalmic disorders that have been considered in this thesis. In regard to figures in which both eyes appear, note that the eyes are placed as an ophthalmologist would observe when *facing* a patient. This is why the left eye is shown on the right and viceversa.

Strabismus is a condition in which one or more visual axis is not directed towards the fixation point [10, p. 96]. In fixing a distant point, the visual axes of both eyes normally are aligned [33, p. 3]. This disorder may occur in one of two major forms: *concomitant* or *incomitant*.

In concomitant strabismus, the deviation is, within physiologic limits and for a given fixation distance, the same in all directions of gaze. In this type of strabismus, all eye muscles function normally; there is simply an error in the position of the eyes. In incomitant strabismus, there is a paralysis of the nerve innervating extraocular muscle. The deviation therefore varies in different directions of gaze, but is larger when the eyes are turned in the direction of action of the underacting or paralytic muscle [7, 33].

In other words, we speak of *concomitance* when, in spite of the deviation, one eye accompanies the other in all its excursions whereas *incomitance* implies that the angle of deviation changes with different positions of gaze.

Amblyopia signifies weak vision. At present, the word “amblyopia” is only rarely used in connection with organically caused decreases in vision, such as tobacco-alcohol amblyopia or quinine amblyopia. Usually, amblyopia is considered to be functional, meaning that it is reversible. However, there can also be a combination of organic and functional causes as well and only when vision loss is reversible by exercises, is there proof of the functional nature of this amblyopia [33].

2.1 Concomitant strabismus

The following briefly explains the different types of concomitant strabismus that have been considered in this thesis. Note that the *absence* of is technically referred to as orthotropia.

Esotropia is a condition in which one or other eye deviates nasally [10].

Exotropia is a condition in which one or other eye deviates outwards [10].

Hypertropia is a disorder in which one or other eye is rotated upwards [10].

Hypotropia is a disorder in which one or other eye is rotated downwards [10].

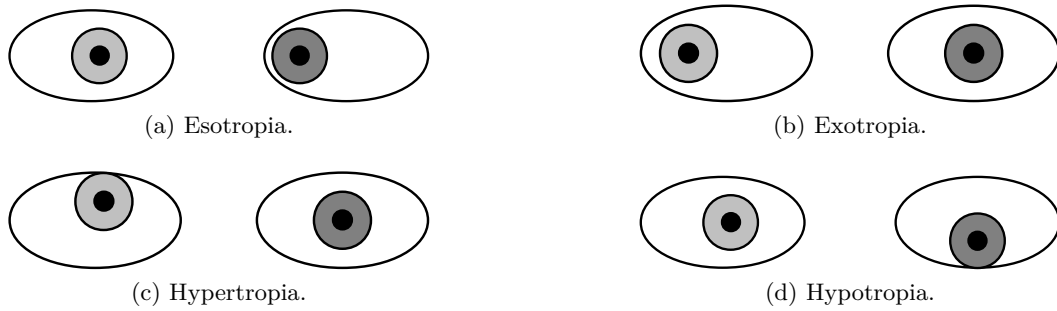


Figure 2.1: Graphical examples of concomitant strabismus.

2.2 Incomitant strabismus

Incomitant strabismus is associated with defective movement of the eye or with asymmetrical accommodative¹ effort [10]. This condition may be found in association with many disorders. It can be congenital or acquired.

Congenital incomitance can be caused by abnormal innervation, trauma either during gestation or delivery, inflammation either neonatal or antenatal, developmental abnormality, muscle insertions to the eyeball or to the orbital contents, fibrosis, adhesions either intermuscular or muscle to orbit, etc.[10]

Acquired incomitance can be caused by trauma, multiple sclerosis, hypertension, space-occupying lesions, diabetes, fractures, neuromuscular junction, muscle fibre membrane, muscle fibre contents, etc.[10]

¹*Accommodation* is the process by which the refractive power of the eyes is altered to ensure a clear retinal image [7].

2.3 Amblyopia

Amblyopia is a condition of diminished visual form sense which is not a result of any clinically demonstrable anomaly of the visual pathway and which is not relieved by the elimination of any defect that constitutes a dioptric obstacle to the formation of the foveal² image [10].

Amblyopia is caused by inadequate stimulation of the visual system during the critical period of visual development in early childhood, less than eight years. This is most marked under the age of two years [31]. Amblyopia may be unilateral or bilateral and the cause may be anyone or a combination of the following factors:

Light deprivation. There is no stimulus to the retina³. This is uncommon as it is likely that some light enters the eye even in dense cataract⁴.

Form deprivation. The retina receives a defocused image as with refractive errors.

Abnormal binocular interaction. Non-fusible images fall on each fovea as with strabismus.

The prognosis for achieving good visual acuity decreases when more than one of these factors are present together [10].

²The *fovea* is a part of the eye, located in the center of the macula region of the retina [26, p. 1698–1705][30]. The fovea is responsible for sharp central vision, which is necessary in humans for fixating when visual detail is of primary importance.

³The *retina* is a light-sensitive tissue lining the inner surface of the eye [29].

⁴A *cataract* is a clouding that develops in the crystalline lens of the eye or in its envelope, varying in degree from slight to complete opacity and obstructing the passage of light [11].

Chapter 3

Previous work

A lot of work has been carried out in the area of ETT solutions for assessment and diagnosis within ophthalmology. This chapter presents a small sample of this work to provide content for the work done in this thesis project.

3.1 Scleral search coil technique

The method that until today has been categorized as the gold standard is the scleral search coil technique (SSC) [22, p. 137-145]. It provides an accurate, but expensive and invasive method to study eye movements and eye movement deficits. It does this by recording horizontal, vertical, and torsional eye movements binocularly. It requires applying a coil attached to a silicon contact lens to the subject's sclera (white part of the eye) that picks up an induced voltage according to its position in a magnetic field. The SSC apparatus is shown in figure 3.1.

Because the coil is attached to a lens attached to the subject's sclera, the subject needs to be administered a local anesthesia and is required to place their chin on a chin rest or use a bite bar in order to stabilize his/her head during testing. Moreover, SSC causes effects such as ocular discomfort, hyperaemia¹ of the bulbar conjunctiva (a clear mucous membrane that covers the sclera and lines the inside of the eyelids), and reduction in visual acuity, among other effects. The result is a neither natural nor relaxed test situation, this is especially importunate in research or tests that involve children or test that require a long time to perform.

As the SSC is invasive and complex to setup, this method has been limited exclusively to a strict research settings. In contrast, using ETT in a more clinical environment the requirements of the procedure are different, i.e., the quantity of patients increases – hence the time for setup, calibration, and tracking procedures diminishes.

¹*Hyperaemia* or *hyperemia* is an increase of blood flow. It is a regulatory response, allowing changes in blood supply through vasodilation. It manifests as redness of the eye, because of the engorgement of vessels with oxygenated blood [27].

This thesis compares the scleral search coil and the *Tobii T120 Eye Tracker* using tests on two volunteers aged 26 and 36 years old to analyze its performance in terms of linearity, accuracy, and testing with both monocular and binocular tests.



Figure 3.1: The scleral search coil setting. Picture taken at *Bernadottelaboratoriet* at *S:t Eriks Ögonsjukhus* during the *Tobii Alignment* validation against SSC technique.

3.2 Image-processing techniques, video-oculography system

Another system for measuring eye movements in three dimensions is the video-oculography system (VOG). This technique stems from photographic and cinematographic recordings of the eye for measuring pupil size some decades ago. In the late 1950s this method was refined and also used for measuring eye movements, in addition to be a pupillometer [4, p. 47–48]. The sampling frequency was limited to 25 Hz, which made it difficult to measure saccades [18]. VOG operates by tracking the pupil center as an indicator of eye position.

The term VOG has been used for the image-processing techniques that also measure torsional eye movements². Today VOG systems work with small video cameras mounted in a mask that also includes some IR-emitting diodes to provide the necessary lighting. Analysis of the data from these cameras utilizes one of two different principles: a two-dimensional correlation technique requiring the selection of at least two landmarks on the surface of the iris or a polar correlation technique which measures (and saves) a natural luminance profile from a sector of the iris. This saved luminance profile is then compared with the images frame-by-frame and the degree of torsion is measured [4, p. 47–48].

The performance of the VOG system is about 0.02° , 0.01° , and 0.025° for the vertical, horizontal, and torsional components respectively. The system behaves robustly within an angular range of $\pm 25^\circ$ for horizontal and vertical movements. The system's biggest disadvantage is its low sampling frequency. Unfortunately, this sampling frequency is too low to properly study rapid movements like nystagmus³ and saccades [4, p. 47–48]. This disadvantage is overcome by using systems with high speed cameras, specifically the *Tobii T60/T120 Eye Tracker*.

3.3 Performance evaluations of remote and binocular 2D eye tracking systems

Multiple studies have been performed to analyze the performance of eye trackers on healthy subjects in terms of linearity, accuracy, and testing with monocular and binocular setups. The purpose was to understand the potential use of ETT in different clinical applications.

A performance evaluation of a remote and binocular 2D eye tracking system, including a linearity testing, optical filter testing, and a comparison between the SSC and the eye tracker on many test subjects between 16-30 years old with no vision disabilities has been carried out by Alexander Hultman in [24]. In all tests the stimuli were presented in 48 areas using a 17-inch TFT monitor. Each stimulus was presented with a duration of less than 3 seconds. Two methods were used to describe linearity. One method described linearity as a line of regression plus the R^2 -value. For each area the mean, of approximately 120 recorded samples, described the accuracy of the eye tracker. The second method described linearity as the error of linearity as a percentage.

The correlation between the amplitude of each stimulus and the measured eye movement was determined to be linear, which is close to the theoretically optimal curve in both the horizontal and vertical plane. Even though the SSC has a significantly lower standard deviation, the true line of sight could not be accessed

²The prototype application implemented in this thesis only reports horizontal and vertical deviations.

³*Nystagmus* is a condition in which the eyes cannot be held still but have a constant tremulous movement [33, p. 128].

more accurately, due to the uncertainty of fixation stability, as determined with the SSC rather than the eye tracker. As a conclusion was, ETT can be used for accurate tracking of 2D eye movements. This presupposed an acceptable calibration of each test subject the method could be configured with different stimulus configurations that could be used in medical experiments, such as visual acuity tests.

The thesis presented in this document also performed linearity testing and optical filter testing. Performances of optical transparent lenses of 750 and 800 nm wavelength were compared with each other for a monocular setup. However, the stimuli were presented only in 9 areas with a duration of 2.2 seconds each, as as specified by researchers at *Bernadottelaboratoriet* at *S:t Eriks Ögonsjukhus*. Gaze data were sampled only during one second in the middle of such interval. For each area the mean and the standard deviation of approximately 60 or 120 recorded samples, depending on whether the eye tracker used was a *Tobii T60* or *T120*, described their accuracy.

3.4 Study of errors introduced by eye tracking systems

Measuring of eye movements regarding which types of errors different eye tracking systems introduce was done by Magnus Olsson [13]. He showed how one can use curve fitting to get a general idea of how the data has been altered by the eye tracking system and also how one can correct it afterwards.

The result was a program called *LINEA* that has been developed together with researchers at *Bernadottelaboratoriet* at *S:t Eriks Ögonsjukhus*. The goal was to correct the errors introduced by various eye tracking systems in a general way. As such this software managed to correct errors independent of which system was the cause or the magnitude and type of the error. Many attempts were made to solve the problem. One of them was to describe the non linearity using a surface $z = f(x, y)$. This was presented as the optimal solution, but it failed in one major way – as the author was not able to fit a surface z to a set of arbitrary fixation points. Another attempt was to skip pseudo axes completely and only use the base axes as references, but this resulted lower accuracy since the skewness was also less accurately described.

In the thesis presented in this document the angles of deviation and vergence⁴ were calculated disregarding the possible errors the eye tracker could introduce. This work sought to describe linearity and accuracy by looking at the mean and standard deviation values of all gaze point sets, assuming an accuracy and a drift of 0.5 and 0.1 degrees respectively – as specified by the vendor [37].

⁴The *vergence* angle is defined as the difference between the rotation about the vertical axis of the left and the right eye. Using the magnitudes described in section 4.7, the *vergence* angle is defined as $v = \phi_L - \phi_R$.

Chapter 4

Hardware and experimental setup

This chapter explains the knowledge needed to develop the prototype using the *Tobii T60/T120 Eye Tracker*. There is also a description of the input and output parameters, as well as the magnitudes utilized and their units.

Figure 4.1 illustrates how an eye tracker works. The *Tobii T60/T120 Eye Tracker* is based on the principle of corneal-reflection tracking¹ [6]. This process involves the following steps:

- One or several near-infrared illuminators, invisible to the human eye, create reflection patterns on the cornea of the eyes.
- At high sampling rates, one or more image sensors record an image of the subject's eyes.
- Image processing is used to find the eyes, detect the exact position of the pupil and/or iris, and identify the correct reflections from knowledge of the illuminators and their exact positions.
- A mathematical model of the eye is used to calculate the eyes' position in space and the point of gaze.

In this thesis project, infrared (IR) light transparent lenses were used to conduct monocular calibration in such a way the subject cannot fixate with the occluded eye whereas the eye tracker's sensors can still find that eye. Two IR transparent lenses with different wavelength (750 nm and 800 nm respectively) provided by production staff working at *Tobii Technology*, who recommended these wavelength values, were used to occlude the subject's corresponding eye.

¹The *cornea* is the transparent front part of the eye that covers the iris, pupil, and anterior chamber. Together with the lens, the cornea refracts light, with the cornea accounting for approximately two-thirds of the eye's total optical power [15].

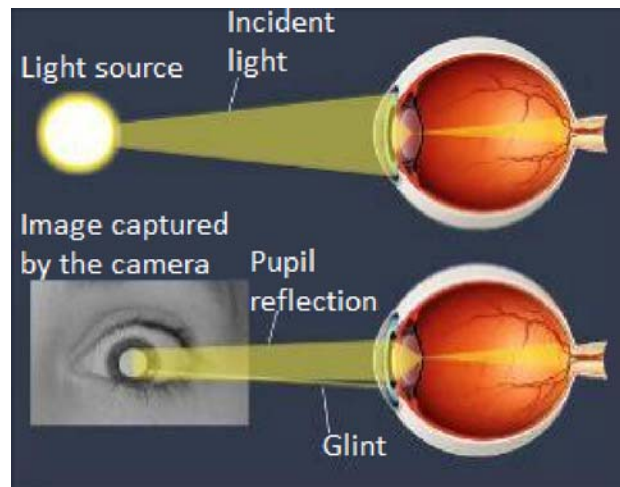


Figure 4.1: Pupil center corneal reflection. Adapted from [6].

Figure 4.2 shows two cases of different pupil color intensity. The *Tobii T60/T120 Eye Tracker* uses two different illumination setups with pupil corneal reflection eye tracking to calculate the gaze position. It can accommodate larger variations in experimental conditions and ethnicity than an eye tracker that uses only one of the techniques described below[6]:

- *Bright pupil eye tracking.* Illuminators are placed close to the optical axis of the imaging sensor, which causes the pupil to appear lit up (this is the same phenomenon that causes red eyes in photos).
- *Dark pupil eye tracking.* Illuminators are placed away from the optical axis causing the pupil to appear darker than the iris.

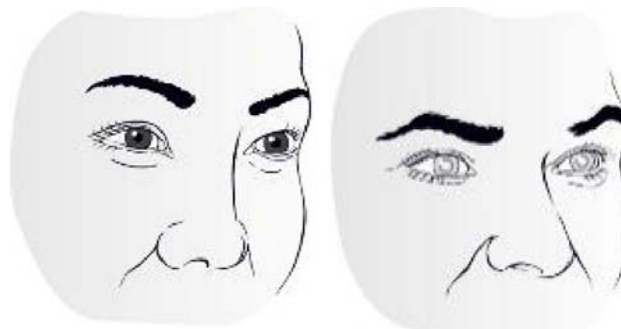


Figure 4.2: A drawing of two subjects with bright and dark pupils each one. Adapted from [6].

Figure 4.3 illustrates the scenario where the prototype application and the eye tracker are linked each other. Note that the application controls the display, while the ETT computes the subject's gaze.

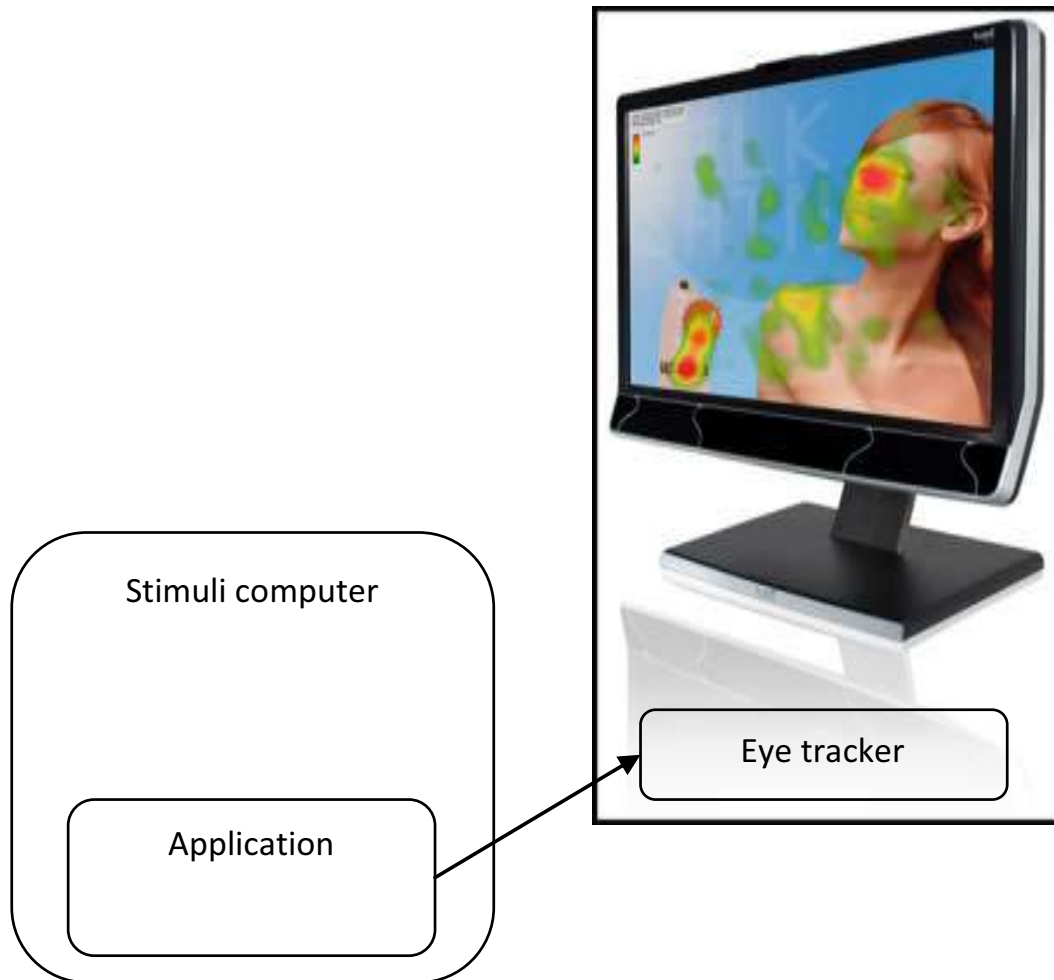


Figure 4.3: Synchronization scenario. Adapted from [34, figure 12].

4.1 The *IEyetracker* interface

The *Tobii* SDK consists of a core library with a C-style interface. On top of this core library they have created language specific bindings that make it possible to access the eye tracker functionality from multiple languages [34, p. 10]. They have implemented many interfaces, but the most important interface to this project is the *IEyetracker* interface. Listing 4.1 shows this interface.

Listing 4.1: The *IEyetracker* interface [36].

```

#region Assembly Tobii.Eyetracking.Sdk.dll, v2.0.50727
#endregion

using System;
using System.Collections.Generic;

namespace Tobii.Eyetracking.Sdk
{
    public interface IEyetracker : IDisposable
    {
        bool RealTimeGazeData { get; set; }

        event EventHandler<CalibrationStartedEventArgs> CalibrationStarted;
        event EventHandler<CalibrationStoppedEventArgs> CalibrationStopped;
        event EventHandler<ConnectionErrorEventArgs> ConnectionError;
        event EventHandler<FramerateChangedEventArgs> FramerateChanged;
        event EventHandler<GazeDataEventArgs> GazeDataReceived;
        event EventHandler<HeadMovementBoxChangedEventArgs> ←
            HeadMovementBoxChanged;
        event EventHandler<XConfigurationChangedEventArgs> ←
            XConfigurationChanged;

        void AddCalibrationPoint(Point2D pt);
        void AddCalibrationPointAsync(Point2D pt, EventHandler<←
            AsyncCompletedEventArgs<Empty>> responseHandler);
        void ClearCalibration();
        void ComputeCalibration();
        void ComputeCalibrationAsync(EventHandler<AsyncCompletedEventArgs<←
            Empty>> responseHandler);
        void DumpImages(int count, int frequency);
        void EnableExtension(int extensionId);
        IList<float> EnumerateFramerates();
        AuthorizeChallenge GetAuthorizeChallenge(int realmId, IList<int> ←
            algorithms);
        IList<Extension> GetAvailableExtensions();
        Calibration GetCalibration();
        byte[] GetDiagnosticReport(int includeImages);
        IList<Extension> GetEnabledExtensions();
        float GetFramerate();
        HeadMovementBox GetHeadMovementBox();
        bool GetLowblinkMode();
        PayPerUseInfo GetPayperuseInfo();
        UnitInfo GetUnitInfo();
        string GetUnitName();
        XConfiguration GetXConfiguration();
        void RemoveCalibrationPoint(Point2D pt);
        void SetCalibration(Calibration calibration);
        void SetFramerate(float framerate);
        void SetLowblinkMode(bool enabled);
        void SetUnitName(string name);
        void SetXConfiguration(XConfiguration configuration);
        void StartCalibration();
        void StartTracking();
        void StopCalibration();
        void StopTracking();
        void ValidateChallengeResponse(int realmId, int algorithm, byte[] ←
            responseData);
    }
}

```

4.2 Coordinate system

All of the data available from *Tobii Eye Trackers* that describe spatial coordinates are given in the *User Coordinate System* (UCS). This coordinate system uses units of millimeters from an origin at the center of the frontal surface of the eye tracker.

The coordinate axes are oriented as shown in figure 4.4: the x -axis points horizontally towards the user's right (green axis), the y -axis points vertically towards the user's up (red axis), and the z -axis points towards the user (blue axis), perpendicular to the filter surface [34, p. 22–23]. Note that the eye tracker itself is located in the monitor's bezel below the middle of the display's screen.



Figure 4.4: The *User Coordinate System* on a *Tobii T60/T120 Eye Tracker*. Adapted from [34, figure 6].

4.3 Available gaze data

The possibility to collect gaze data is probably the most interesting functionality provided by the SDK. The SDK's API makes it possible to subscribe to a stream of data that will arrive at the eye tracker's sampling frequency, which can be set to either 60 or 120 Hz.

This data will be used as input parameters when calculating angles of deviation. Some of these data are 2D and 3D point objects represented by two or three *double* values respectively. They are implemented as using the *Point2D* and *Point3D* classes (see listing 4.2 and listing 4.3 respectively).

Listing 4.2: The *Point2D* class [36].

```
#region Assembly Tobii.Eyetracking.Sdk.dll, v2.0.50727
#endregion

using System;

namespace Tobii.Eyetracking.Sdk
{
    public struct Point2D
    {
        public Point2D(double x, double y);

        public double X { get; set; }
        public double Y { get; set; }

        public override string ToString();
    }
}
```

Listing 4.3: The *Point3D* class [36].

```
#region Assembly Tobii.Eyetracking.Sdk.dll, v2.0.50727
#endregion

using System;

namespace Tobii.Eyetracking.Sdk
{
    public struct Point3D
    {
        public Point3D(double x, double y, double z);

        public double X { get; set; }
        public double Y { get; set; }
        public double Z { get; set; }

        public override string ToString();
    }
}
```

Each sample consists of a *GazeDataItem* class object [34, p. 32–34]. The *GazeDataItem* class is shown in listing 4.4. The elements of such an object are described below.

Listing 4.4: The *GazeDataItem* class [36].

```
#region Assembly Tobii.Eyetracking.Sdk.dll, v2.0.50727
#endregion

using System;

namespace Tobii.Eyetracking.Sdk
{
    public class GazeDataItem : IGazeDataItem
    {
        public GazeDataItem(DataTree xdsGazeData);

        public Point3D LeftEyePosition3D { get; }
        public Point3D LeftEyePosition3DRelative { get; }
        public Point2D LeftGazePoint2D { get; }
        public Point3D LeftGazePoint3D { get; }
        public float LeftPupilDiameter { get; }
        public int LeftValidity { get; }
        public Point3D RightEyePosition3D { get; }
        public Point3D RightEyePosition3DRelative { get; }
        public Point2D RightGazePoint2D { get; }
        public Point3D RightGazePoint3D { get; }
        public float RightPupilDiameter { get; }
        public int RightValidity { get; }
        public long TimeStamp { get; }
    }
}
```

- The *Time Stamp* is a value that indicates the time when a specific gaze data sample was obtained by the eye tracker. This value should be interpreted as a 64 bit unsigned microsecond timer starting from some unknown point in time. The source of this time stamp is the internal clock in the eye tracker hardware. In a typical use case, such as the one implemented in this prototype application, this time stamp is used to synchronize the gaze data stream when the first stimuli was presented on the computer screen. How this is done is described in chapter 5.
- The *Eye Position* is provided for the left and right eye individually and describes the position of each eyeball in 3D space. Three floating point values are used to describe the x , y , and z coordinate respectively. The position is specified in the UCS coordinate system. If a subject wearing glasses and has a cornea that diverts by a few percent this will highly affect the accuracy – by up to 17% [37].

- The *Relative Eye Position* is provided for the left and right eye individually and gives the relative position of the eyeball in the head movement box volume as three normalized coordinates (see figure 4.5). The head movement box is an imaginary box in which a user can move his/her head and still be tracked by the device. This information can be used to help the user position himself/herself in front of the tracker.

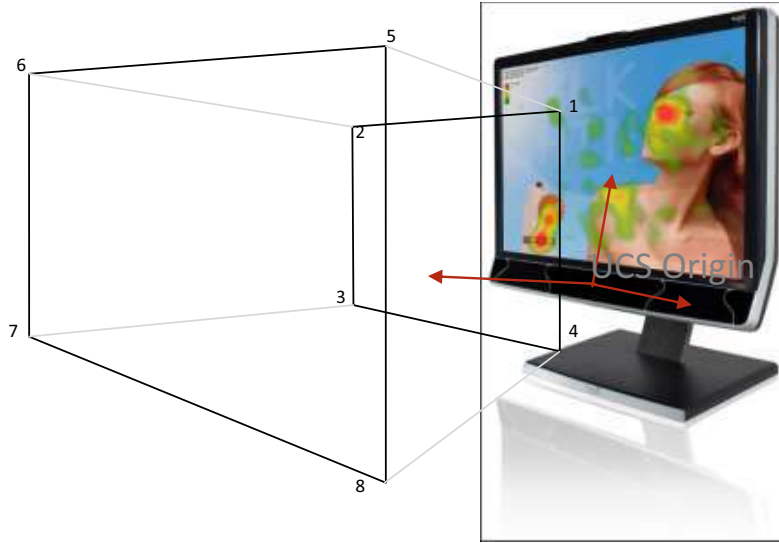


Figure 4.5: Schematic head movement box. Adapted from [34, figure 16].

- The *Gaze Point* is provided for the left and right eye individually and describes the position of the intersection between a line originating from the eye's position in the same direction as the gaze vector and the tracking plane. This is illustrated in figure 4.6. The gaze vector is not explicitly provided in the gaze data stream, but can easily be computed by subtracting the 3D gaze point and the 3D eye position and normalizing the resulting vector.

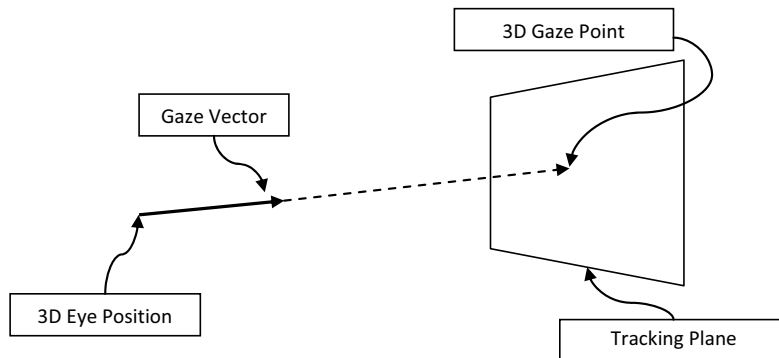


Figure 4.6: Gaze point and gaze vector. Adapted from [34, figure 7].

- The *Relative Gaze Point* is provided for the left and right eye individually and corresponds to the two dimensional position of the gaze point within the tracking plane. The coordinates are normalized to $[0,1]$ with the point (0,0) in the upper left corner from the user's point of view. The x -coordinate increases to the right and the y -coordinate increases towards the bottom of the screen.
- The *Validity Code* is an estimate of how certain the eye tracker is that the data given for an eye really originates from that eye. When the tracker finds two eyes in the camera image, identifying which one is the left and which one is the right eye is very straightforward. Similarly the case when no eyes are found at all is really simple. The most challenging case is when the tracker only finds one eye in the camera's image. When that happens, the image processing algorithms tries to deduce if the eye in question is the left or the right one. This is done by referring to previous eye positions, the position in the camera sensor and certain image features. The validity codes describe the outcome of this deduction. The validity codes can only appear in certain combinations. These combinations and their interpretations are summarized in table 4.1. The validity codes can be used to filter out data that is most likely incorrect. Normally it is recommended that all samples with validity code 2 or higher are removed or ignored, but only samples with validity code 0 are considered in the prototype application implemented.

Table 4.1: Data validity codes.

Left \ Right	0	1	2	3	4
0	Found two eyes.				Found most probable the left eye only.
1				Found probably the left eye only.	
2			Found one eye without any certainty.		
3		Found probably the right eye only.			
4	Found most probably the right eye only.				No eyes found.

- The *Pupil Diameter* data is provided for the left and the right eye individually and is an estimate of the pupil size in millimeters. The *Tobii Eye Trackers* try to measure the true pupil size, i.e., the algorithms take into account the magnification effect given by the spherical cornea as well as the distance to the eye.

4.4 Configuring the screen

To compute angles between vectors described by the corresponding eye position and any point on the eye tracker screen, it is necessary to configure the eye tracking area geometry. The SDK contains functions to learn the so called *x-configuration* parameters. Before starting to calculate such angles, it is necessary to know the relative position of the tracking area in relation to the eye tracker screen. The tracking plane is described by three points in space, the upper left, upper right and lower left corner points of the eye tracker screen. These three points must be described in the UCS coordinate system as illustrated by the red vectors in figure 4.7.

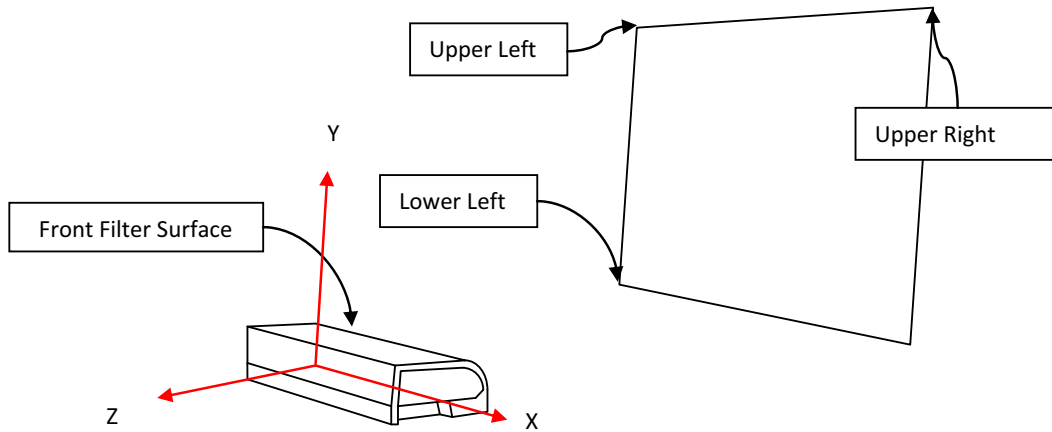


Figure 4.7: Screen configuration schematic. Adapted from [34, figure 17].

As the *Tobii T60/T120 Eye Tracker* includes a 17-inch TFT monitor, these three points are mechanically fixed and their positions already measured, thus the coordinates of these three points can be returned by the SDK. They will be used as input parameters in conjunction with the gaze data when calculating angles of deviation. The *XConfiguration* class is shown in listing 4.5.

Listing 4.5: The *XConfiguration* class [36].

```

#region Assembly Tobii.Eyetracking.Sdk.dll, v2.0.50727
#endregion

using System;

namespace Tobii.Eyetracking.Sdk
{
    public class XConfiguration
    {
        public XConfiguration();

        public Point3D LowerLeft { get; set; }
        public Point3D UpperLeft { get; set; }
        public Point3D UpperRight { get; set; }
    }
}

```

The *GetXConfiguration()* method of the *IEyetracker* interface returns the *x-configuration* parameters. Let P , Q , and R be the upper left, upper right, and lower left corner 3D points of the eye tracker's screen respectively. After calling this method on the *Tobii T60/T120 Eye Tracker*, the following 3D points are returned:

$$P(x, y, z) = (-169, 287.6, 93.78)$$

$$Q(x, y, z) = (169, 287.6, 93.78)$$

$$R(x, y, z) = (-169, 31.8, -4.41)$$

Let O be the origin in the UCS. Let $\vec{H}$ and $\vec{V}$ be the horizontal and vertical directions on the eye tracker screen respectively. $\vec{H}$ and $\vec{V}$ are calculated as follows:

$$\vec{H}(x, y, z) = \vec{PQ} = \vec{OQ} - \vec{OP} = (Q_x - P_x, Q_y - P_y, Q_z - P_z) \quad (4.1)$$

$$\vec{V}(x, y, z) = \vec{PR} = \vec{OR} - \vec{OP} = (R_x - P_x, R_y - P_y, R_z - P_z) \quad (4.2)$$

Looking at the values of P , Q and R for the *Tobii T60/T120 Eye Tracker* we can see that:

$$P_x = -Q_x = R_x \quad (4.3)$$

$$P_y = Q_y \quad (4.4)$$

$$P_z = Q_z \quad (4.5)$$

This implies that $\vec{PQ}$ is parallel to the x -axis and equations 4.1 and 4.2 can be simplified as follows:

$$\vec{H} = (2Q_x, 0, 0) \quad (4.6)$$

$$\vec{V} = (0, R_y - P_y, R_z - P_z) \quad (4.7)$$

P , Q , and R describe the tracking plane. Let S be the lower right corner 3D point of the eye tracker's screen. The position S is calculated as follows:

$$S(x, y, z) = P + \vec{H} + \vec{V} = (P_x + 2Q_x, P_y + R_y - P_y, P_z + R_z - P_z) = (Q_x, R_y, R_z) \quad (4.8)$$

Finally the tracking area, which is a rectangle, is described by the planar area enclosed by P , Q , R , and S . In this way, any 3D point on the tracking area can be computed if we know its normalized 2D coordinates on the tracking area.

Let T and U be the same target point, but given in normalized 2D coordinates as explained in the *Relative gaze points* description and in 3D coordinates respectively. U can be computed as follows:

$$U(x, y, z) = P + T_x \cdot \vec{H} + T_y \cdot \vec{V} = (P_x + 2 \cdot T_x \cdot Q_x, P_y + T_y \cdot (R_y - P_y), P_z + T_y \cdot (R_z - P_z)) \quad (4.9)$$

For example:

$$T = (0.9, 0.1) \implies U = (135.2, 262.02, 83.961)$$

4.5 Calibration

In order to compute the gaze point and gaze direction with high accuracy, the eye tracker firmware adapts the algorithms to the subject sitting in front of the tracker. This adaptation is done during the calibration process, when the subject is looking at points located at a known set of coordinates. The calibration is initiated and controlled by the prototype application using the SDK [34, p. 37–44].

4.5.1 Calibration procedure

The calibration of the eye tracker is done as follows:

1. A small object is shown on the screen to catch the subject's attention.
2. The position of this object must be stationary for enough time to give the subject a chance to focus.

3. Tell the eye tracker to start collecting data for the specific calibration point.
4. Wait for the eyetracker to finish calibration data collection at the current position.
5. Clear calibration point.
6. Repeat steps 1-5 for all desired calibration points.

The object in step 1 should be long enough for the subject to easily focus on it. Otherwise the firmware may not be able to get a good calibration result as the subject has not had enough time to focus their gaze on the target before the eye tracker starts collecting calibration data.

The normal number of calibration points are 2, 5, or 9. A large number of points can be used, but the result will not significantly increase calibration accuracy for more than 9 points. The calibration points should span an area that is as large as the area in which targets will later be shown on the screen (a large area should be used in order to ensure good interaction with the subject). The calibration points must also be positioned within the area that is trackable by the eye tracker.

To be able to perform a calibration the client application must first enter the calibration state. It is entered by calling the method *StartCalibration()* and is left by calling *StopCalibration()*. Some operations can only be performed during the calibration state, such as *AddCalibrationPoint()* [step 3], *ClearCalibration()* [step 5], and *ComputeAndSetCalibration()* [last step]. Other operations such as *SetCalibration()* or *GetCalibration()* work at any time. Listing 4.1 on page 20 includes these methods in the interface provided by the *IEyetracker* interface.

Chapter 5 will describe the calibration points in the prototype application, where they are located, how long they are shown, and why these specific points have been chosen. Both binocular and monocular procedures will also be explained too.

4.5.2 Calibration plot

The calibration plot is a simple yet concise representation of a performed calibration and it usually looks something like what is shown in figures 4.8 and 4.9. The calibration plot shows the offset between the mapped gaze samples and the calibration points based on the best possible adaptation of the eye model to the values collected by the eye tracker during calibration.

The samples are drawn as small red and green lines representing the left and right eye respectively. The gray circles are the actual calibration points. Each sample also has a mapped point and validity per eye. In figures 4.8 and 4.9 each red and green line is a connection between the true point and a mapped point filtered by validity. The validity code can take the following values: -1: eye not found, 0: found but not used, and 1: used.

When validity is -1 neither a gray circle nor connecting lines are shown. When validity is 0 only a gray circle will be drawn. When validity is 1 both a gray circle and connecting lines are shown.

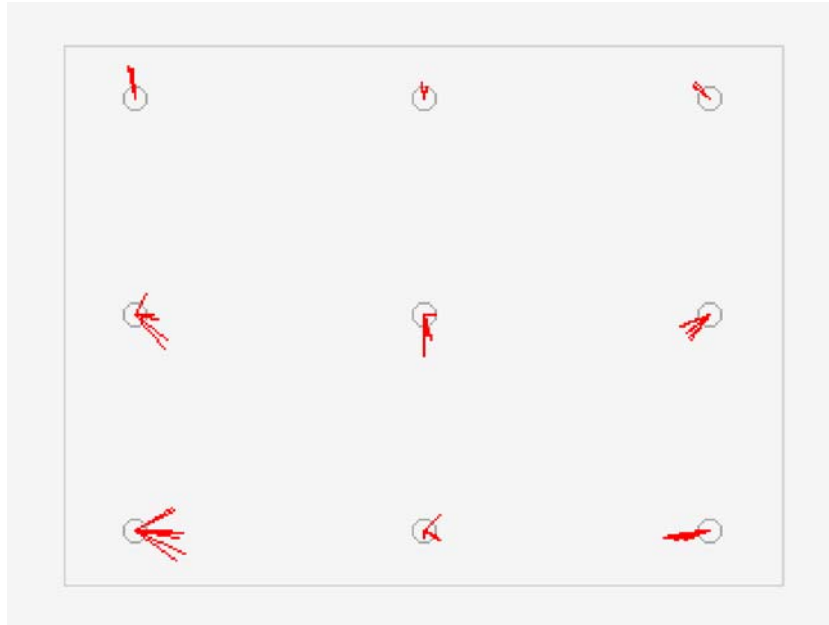


Figure 4.8: A 9-point calibration plot example with data for the left eye.

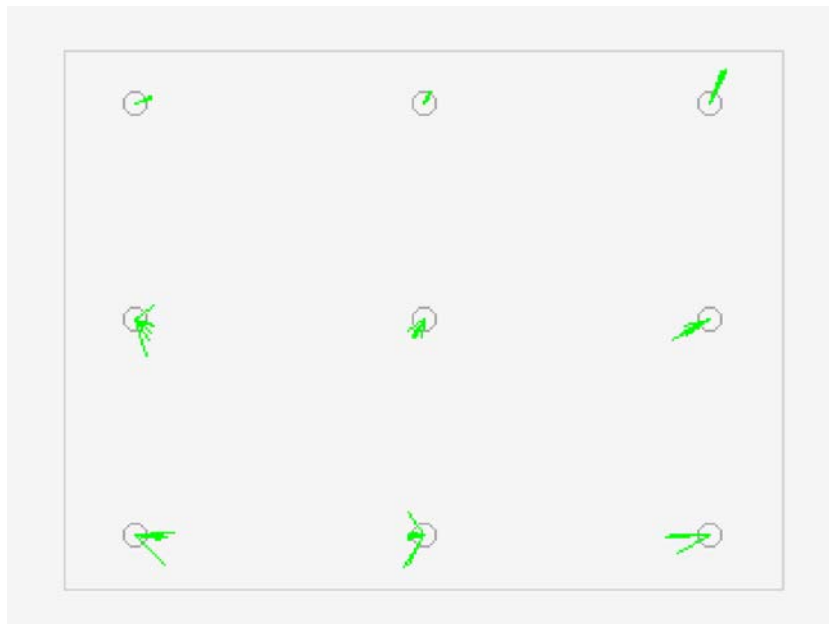


Figure 4.9: A 9-point calibration plot example with data for the right eye.

4.5.3 Calibration buffers

The eye tracker firmware uses two buffers to keep track of the calibration data:

- *The Temporary Calibration Buffer* is only used during calibration. This is where data is added or removed during calibration.
- *The Active Calibration Buffer* contains the data for the calibration that is currently set. This buffer is modified either by a call to *SetCalibration()*, which copies data from the client application to the buffer or a successful call to *ComputeCalibration()*. *ComputeCalibration* computes the calibration parameters based on the data in the temporary buffer and then copies the data into the active buffer.

The *CalibrationPlotItem* and *Calibration* classes are shown in listings 4.6 and 4.7 respectively. The Temporary Calibration Buffer and Active Calibration Buffer are each an instance of a *Calibration* class object. The data within the buffer are each instance of *CalibrationPlotItem* objects.

Listing 4.6: The *CalibrationPlotItem* class [36].

```
#region Assembly Tobii.Eyetracking.Sdk.dll, v2.0.50727
#endregion

using System;

namespace Tobii.Eyetracking.Sdk
{
    public class CalibrationPlotItem
    {
        public CalibrationPlotItem();

        public float MapLeftX { get; set; }
        public float MapLeftY { get; set; }
        public float MapRightX { get; set; }
        public float MapRightY { get; set; }
        public float TrueX { get; set; }
        public float TrueY { get; set; }
        public int ValidityLeft { get; set; }
        public int ValidityRight { get; set; }
    }
}
```


Listing 4.7: The *Calibration* class [36].

```

#region Assembly Tobii.Eyetracking.Sdk.dll, v2.0.50727
#endregion

using System;
using System.Collections.Generic;

namespace Tobii.Eyetracking.Sdk
{
    public class Calibration
    {
        public Calibration(byte[] rawData);

        public List<CalibrationPlotItem> Plot { get; }
        public byte[] RawData { get; }
    }
}

```

4.6 Tracking

If everything went well during the calibration procedure, then gaze points can be computed with high accuracy. In the prototype application implemented in this thesis, the tracking procedure is similar to the calibration procedure in the sense that the workflow of the sequence of stimuli is exactly the same. The difference lies in the fact that during the tracking procedure we record the gaze patterns rather than perform a calibration.

As we obtain the input gaze data items and we know where each input target point is shown on the tracking area, then angles of deviation in the subject's gaze patterns can be computed as output parameters by using trigonometry. The workflow of this method will be described in chapter 5.

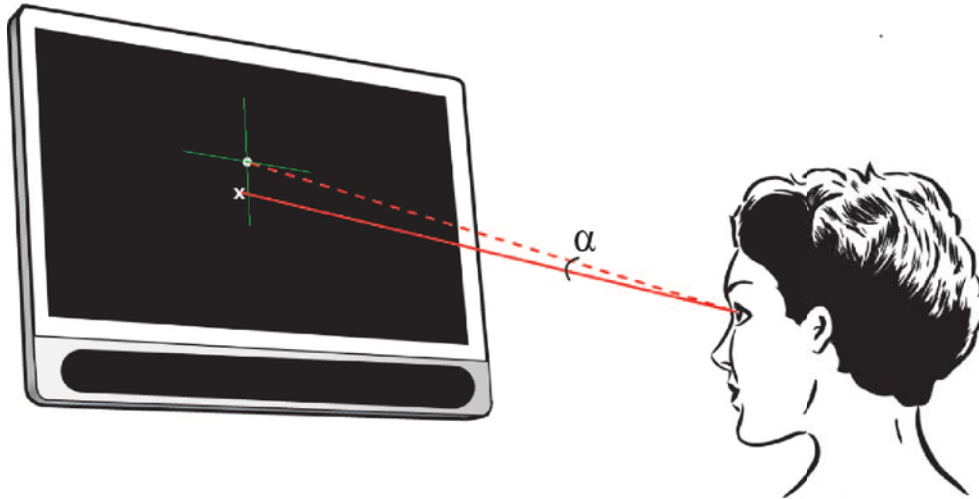
4.7 Magnitudes and units

In summary, the input parameters are 2D and 3D points. All of the 2D points are normalized to $[0,1]$ with the point $(0,0)$ in the upper left corner from the subject's point of view. The x -coordinate increases to the right and the y -coordinate increases towards the bottom of the screen. This left handed coordinate system is widely used in computer graphics. In contrast, the 3D points are described in the UCS coordinate system and their position is expressed in units of millimeters.

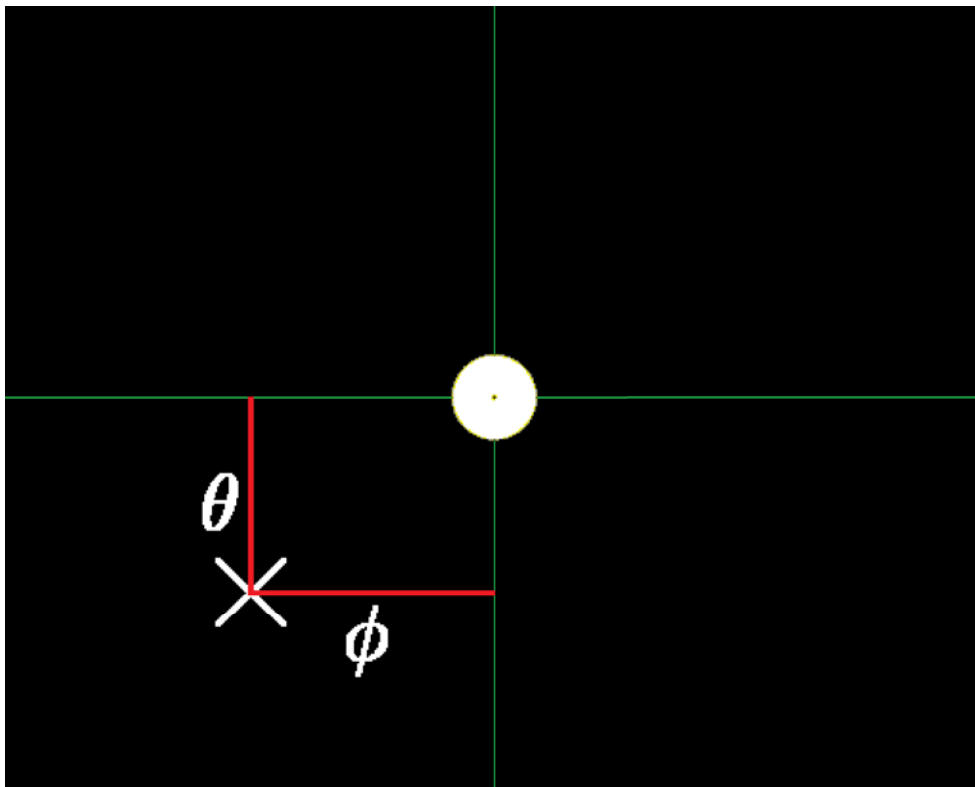
Now that we have collected all the input parameters, the output parameters can be computed. These outputs will be horizontal and vertical angles of deviation in the subject's gaze patterns for each stimuli area. All of these computations are expressed in degrees. In the next paragraph, the relevant angles are defined.

Let α be the angle between the vector from the eye's position to the gaze point (gaze vector) and the vector from the eye's position to the target point (target vector). Let ϕ and θ be the angles obtained after decomposing α such that these

new angles are expressed with respect to the horizontal and vertical axes of the target point respectively as shown in figure 4.10.



(a) Angle between the *gaze vector* (solid line) and the *target vector* (dashed line).



(b) The ϕ and θ angles.

Figure 4.10: Angle of deviation decomposed into ϕ and θ . Adapted from [2, p. 6].

To calculate how many significant digits those values have, we considered an ideal situation in which the average eye position in space is 60 cm directly in front of the eye tracker filter's surface, as recommended for the *Tobii T60/T120 Eye Tracker* [37, p. 13]. Let E be that average eye position. Let U be the set of 3D points, $u_{i,j}$, on the tracking area such that there are $1280 \times 1024 = 1\,310\,720$ points, one for each pixel of the monitor. The precision of values that can be achieved for ϕ and θ are the smallest angles among all of the vectors described by E and two neighboring pixels on the same horizontal and vertical direction respectively, included in U . These angles will be compared relative to the angles between the vectors described by E and the farthest pixel on the tracking area and its closest horizontal and vertical neighbors respectively.

To find these values, a script in *MATLAB R2010b* was implemented and run. The code is included in listing 4.8. It implements what is described in equations 4.1, 4.2, and 4.9 for all the pixels of the eye tracker screen. For each iteration, the distance between E and $u_{i,j}$ is calculated. The row and column indices returned when we call the instruction *find* looking for the maximum value among all the distances tells us which is the farthest pixel. Then, the angles between the vectors described by E and the farthest pixel and its closest horizontal and vertical neighbors respectively, are calculated as follows:

$$\Delta\phi = 0.023197392294608^\circ$$

$$\Delta\theta = 0.023298105218762^\circ$$

Thus, even though all angles of deviation were computed with the precision given by *double* precision values in *C#*, the mean and standard deviation values for each cardinal stimulus will be presented with one significant floating point digit (since both delta $\Delta\phi$ and $\Delta\theta$ are less than 0.1 degrees), results that will be presented in chapter 5.

Listing 4.8: Precision of angular magnitude.

```
function [DeltaPhi, DeltaTheta] = precision(P, Q, R, width, height)
% Angles are initialized to infinity since some could be not assigned
% during call to the script and we are looking for the minimum value.
angleN = Inf; angleE = Inf; angleS = Inf; angleW = Inf;
H = Q - P; V = R - P; % See equations 4.1 and 4.2.
% Ideal average eye position in space is 650 millimeters straightly in
% front of the eye tracker filter surface.
E = [0 0 600];
% As many 3D points as pixels the eye tracker has.
U = zeros(width,height,3);
% Distance between E and each pixel of the eye tracker screen.
D = zeros(width,height);
for i = 1:width
    for j = 1:height
        % See equation 4.9.
        U(i,j,:) = P + (i-0.5)/width*H + (j-0.5)/height*V;
```

```

        D(i,j) = sqrt(((E(1)-U(i,j,1))^2 + (E(2)-U(i,j,2))^2 + (E(3)-U(i,j,3))^2));
    end
end
[Tx, Ty] = find(D == max(max(D)));
u = P + Tx(1)/width*H + Ty(1)/height*V; % See equation 4.9.
Eu = u - E;
if Ty(1) > 1
    neighborNorth = [Tx(1) Ty(1)-1];
    % See equation 4.9.
    uN = P + neighborNorth(1)/width*H + neighborNorth(2)/height*V;
    EuN = uN - E;
    angleN = 180/pi * acos(sum(Eu.*EuN) ./ (sqrt(Eu(1)^2+Eu(2)^2+Eu(3)^2) *←
        sqrt(EuN(1)^2+EuN(2)^2+EuN(3)^2)));
end
if Tx(1) < width
    neighborEast = [Tx(1)+1 Ty(1)];
    % See equation 4.9.
    uE = P + neighborEast(1)/width*H + neighborEast(2)/height*V;
    EuE = uE - E;
    angleE = 180/pi * acos(sum(Eu.*EuE) ./ (sqrt(Eu(1)^2+Eu(2)^2+Eu(3)^2) *←
        sqrt(EuE(1)^2+EuE(2)^2+EuE(3)^2)));
end
if Ty(1) < height
    neighborSouth = [Tx(1) Ty(1)+1];
    % See equation 4.9.
    uS = P + neighborSouth(1)/width*H + neighborSouth(2)/height*V;
    EuS = uS - E;
    angleS = 180/pi * acos(sum(Eu.*EuS) ./ (sqrt(Eu(1)^2+Eu(2)^2+Eu(3)^2) *←
        sqrt(EuS(1)^2+EuS(2)^2+EuS(3)^2)));
end
if Tx(1) > 1
    neighborWest = [Tx(1)-1 Ty(1)];
    % See equation 4.9.
    uW = P + neighborWest(1)/width*H + neighborWest(2)/height*V;
    EuW = uW - E;
    angleW = 180/pi * acos(sum(Eu.*EuW) ./ (sqrt(Eu(1)^2+Eu(2)^2+Eu(3)^2) *←
        sqrt(EuW(1)^2+EuW(2)^2+EuW(3)^2)));
end
DeltaPhi = min(angleE,angleW); % Precision for Phi.
DeltaTheta = min(angleN,angleS); %Precision for Theta.

```

4.8 Test procedure

This section describes the workflow of a test procedure for the measurement of gaze deviations of a subject as well as the scenario into consideration, i.e., the test conditions and the tools needed.

4.8.1 Environment

In order to correctly perform the test, a clinical environment is crucial to avoid distraction on the test subjects (as these distractions could greatly impact the outcome of the test). The testing room must be isolated from disturbances and the *Tobii* display must be positioned steadily on the table. Light conditions have the greatest effect on test results, therefore they must be carefully controlled. Firstly,

the light intensity must be measured (and if necessary adjusted) to correspond with the level agreed with experts and clinicians at *Karolinska Institutet*. This ambient light level is between 200 and 300 lux. In addition, it is imperative to avoid any light reflections in the room and that all light sources are wide spread across the room (i.e., that the light is diffuse). The light sources must not be angled towards the eye tracker nor towards the participant in order to avoid disturbing the tracker and the test subject respectively. A further prerequisite for an adequate testing environment is that the test room remain isolated from sunlight and other uncontrolled light sources. If there are windows in the room, these must be covered so that all light sources are controlled exclusively by the test leader [2].

The room used for the first two sets of tests is shown in figure 4.11. The light sources were *Cromalite* model *TIZ50W230V* halogen lamps located in windows of light 40×60 cm sized as represented in figure 4.12. The room used for the final set of tests is shown in figure 3.1. The light source was ordinary light located in the ceiling of the room and generating between 200 and 300 lux.



Figure 4.11: The room used for the first two sets of tests.



Figure 4.12: The location of the light sources. Adapted from [2, figure 22].

4.8.2 Equipment

There are several tools needed for conducting these tests. An eye tracker and a computer are required to conduct the tests. In the measurements reported in this thesis we have used a *Tobii Technology T60 Eye Tracker* for the first two sets of tests (see sections 6.1 and 6.2) and a *Tobii Technology T120 Eye Tracker* for the final set of experiments (see section 6.3). An *Intel® Pentium®* model *Dual-Core CPU T4400 @ 2.20 GHz* PC running the *Microsoft Corporation® Windows® 7 Professional* operating system and equipped with 2 GB of memory and a *Mobile Intel®* model *4 Series Express Chipset Family* graphics card connected to the Tobii display (with eye tracker) using a VGA cable has been used for all the studies. The eye tracker needs to be connected to the computer using either a USB or Ethernet cable. The eye tracker can also be connected to an office LAN or to a separate network router. When using an Ethernet cable to connect the eye tracker the host, router, or other device connected to the same network segment must be configured to automatically give the eye tracker an IP address via DHCP as the eye tracker will make a request for an IP address when it is powered on [37]. In the tests reported in this thesis we have directly connected the PC and eye tracker via a cross over cable. The Ethernet interface of the PC was configured to operate in full duplex mode at 100 Mbps.

In addition, a light meter, such as the one shown in figure 4.13, is needed in order to ensure the correct light intensity for each of the tests. In the tests conducted in this thesis project we have used a *Clas Ohlson* model *ST-1300* light meter. This meter was calibrated by Fangfang Helenius at *Tobii Technology* on 2011.08.10.



Figure 4.13: A light meter is required in order to determine the illumination in the test room. Adapted from [2, figure 22].

4.8.3 Workflow

The following is the outline of the workflow that the specialist would carry out in a clinical environment. Note that although we refer to a “patient” in this description, the actual test subjects for all of the tests conducted in this thesis project were eighteen volunteers from among technical staff working at *Tobii Technology* (sixteen of them for the first set of tests and twelve of them for the second set of tests) and the author of this thesis and the author’s supervisor in the case of the final set of experiments. Informed consent was obtained from all of these volunteers. We assume that as these persons were all working with the *Tobii* eye trackers that they were fully aware of the operation of the device. All of the volunteers were also informed about the purpose of these tests, before they agreed to participate. All of the participants were above the age of 25. As this age was greater than 18, they did not need to have parental permission to participate.

- Before patient arrives:
 1. Start computer, eye tracker, and the application.
 2. Make sure that the eye tracker is connected and works. A simple way for the specialist to ensure this is looking for the eye tracker in the “*Eyetrackers Found on the Network*” box and clicking on “*Connect to Eyetracker*” button. See figure 4.14.
 3. Enter patient ID (in the tests conducted in this thesis project the ID was simply a sequentially assigned number).
- After the patient has arrived, position the patient in front of the eye tracker. Make sure that the patient is placed in the correct position relative to the eye tracker. A very simple way for the specialist to make sure that the patient is placed properly is looking at “*Eyetracker Status*” box in which both eyes are represented by white circles on a black background. The average distance of the eyes to the front filter surface is shown at the upper left corner of this box. If the bar at the bottom of the box is green, it means that the patient is placed at a distance of between 60 and 65 cm. Otherwise, the bar is red. The patient should be positioned so that both eyes are shown and the box is green. See figure 4.14.
- If the binocular calibration procedure is chosen, conduct binocular calibration. The calibration will be automatically saved and uploaded to the eye tracker.
- If the monocular calibration procedure is chosen:
 1. Occlude left eye (OS) with IR light transparent lens and conduct monocular calibration on the right eye (OD). The calibration will be automatically saved waiting for the other monocular calibration result.
 2. Occlude right eye (OD) with IR light transparent lens and conduct monocular calibration on the left eye (OS). The calibration will be automatically saved waiting for the specialist to order the application to merge both calibrations.
 3. Order the application to merge both calibrations. The result will be automatically saved and uploaded to the eye tracker.
- Start the tracking test.

The monocular calibration procedure explained above can be swapped in the sense that calibration on the right eye (OD) could be done first. After a test is performed, the result can be examined in the application, the gaze points considered as signal samples regarding fixation are plotted on the screen in continuous time, sample by sample, see section 5.2. The result is also exported as a bitmap image and a text document according to the specification presented in section 5.1.

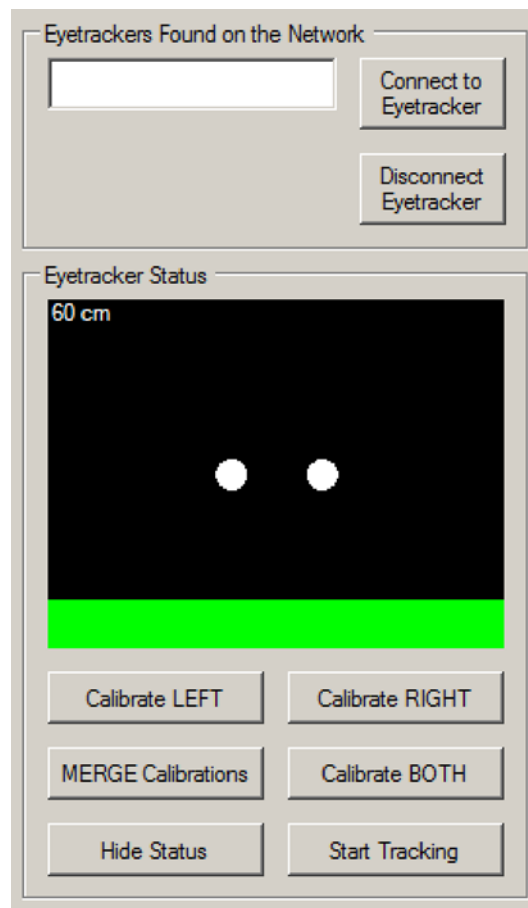


Figure 4.14: The *Eyetrackers Found on the Network* and the *Eyetracker Status* boxes.

Chapter 5

Description of the prototype application

This chapter presents in detail the desired report that the software should output as well as the prototype system architecture designed and implemented during this thesis.

5.1 Specification of the desired report

In the beginning of the thesis project a specification of the desired report of a test was defined in conjunction with experts and clinicians in the area at *Karolinska Institutet*. The document is the output of the prototype application and shows the information about the subject's gaze patterns that are considered most valuable.

As was described in sections 4.5.1 and 4.6, small objects are shown on the screen to catch the subject's attention for both the calibration and tracking procedures. In the prototype application, the objects are randomly shown in nine cardinal positions. The object is a yellow circle with a black center on a black grid over a gray background as shown in figure 5.1.

The location of the calibration points is a combination of the normalized 2D coordinates 0.1, 0.5, and 0.9 in order to ensure both good interaction and trackability. These locations are shown in table 5.1 along with their 3D coordinates in the UCS coordinate system – according to equation 4.9.

In the ideal situation in which the average eye position in space is 60 cm straightly in front of the eye tracker's filter surface, as it is recommended for the *Tobii T60/T120 Eye Tracker* [37, p. 13], the angles between the vectors described by that average eye position and the center of the screen and left/right and up/down axes respectively, are calculated by trigonometry. Table 5.2 shows the corresponding *standard* angle pairs expressed as (ϕ, θ) .

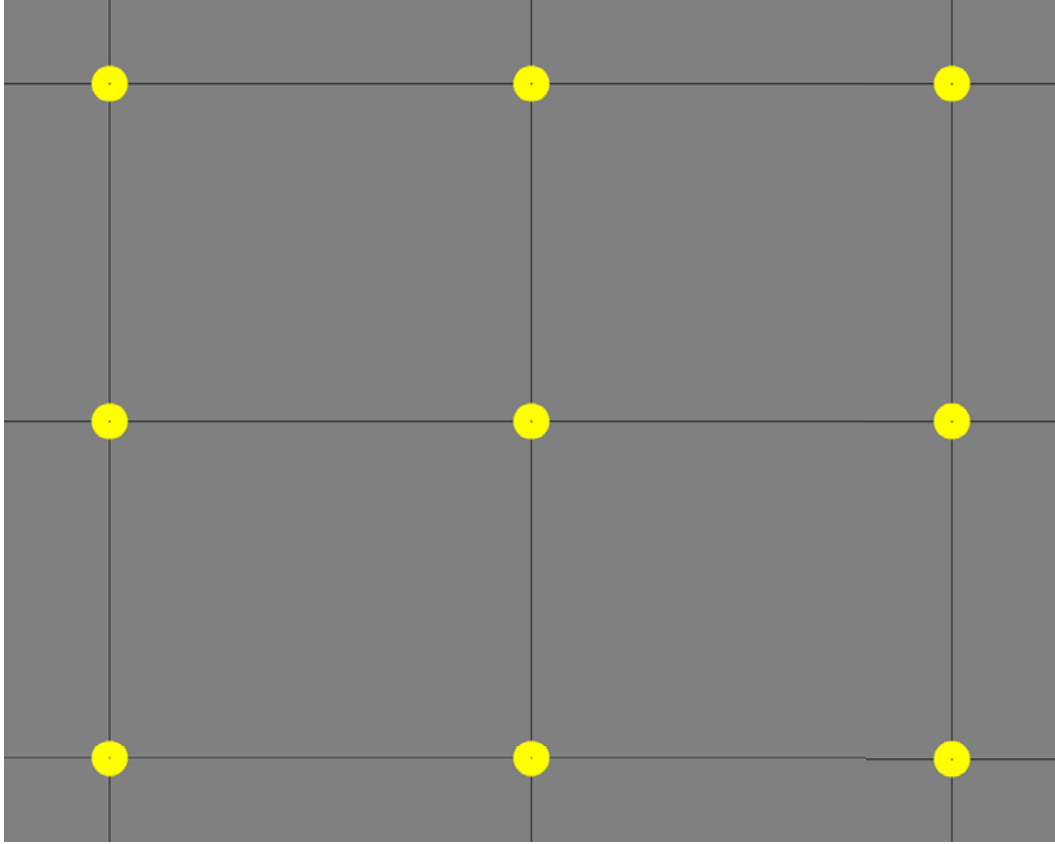


Figure 5.1: The 9-point calibration pattern. The points are randomly shown one by one.

Table 5.1: The location of the calibration points in 2D and 3D coordinates.

Point	2D coordinate	3D coordinate (mm, mm, mm)
1	(0.1, 0.1)	(-135.2, 262.02, 83.961)
2	(0.5, 0.1)	(0, 262.02, 83.961)
3	(0.9, 0.1)	(135.2, 262.02, 83.961)
4	(0.1, 0.5)	(-135.2, 159.7, 44.685)
5	(0.5, 0.5)	(0, 159.7, 44.685)
6	(0.9, 0.5)	(135.2, 159.7, 44.685)
7	(0.1, 0.9)	(-135.2, 57.38, 5.409)
8	(0.5, 0.9)	(0, 57.38, 5.409)
9	(0.9, 0.9)	(135.2, 57.38, 5.409)

Table 5.2: The location of the calibration points in 2D coordinates and degrees.

Point	2D coordinate	(ϕ, θ) (degrees, degrees)
1	(0.1, 0.1)	(-13.2, 10.9)
2	(0.5, 0.1)	(0, 10.9)
3	(0.9, 0.1)	(13.2, 10.9)
4	(0.1, 0.5)	(-13.2, 0)
5	(0.5, 0.5)	(0, 0)
6	(0.9, 0.5)	(13.2, 0)
7	(0.1, 0.9)	(-13.2, -10.9)
8	(0.5, 0.9)	(0, -10.9)
9	(0.9, 0.9)	(13.2, -10.9)

Thus, the stimuli shown in the nine cardinal positions range from 13.2 degrees from central vertical axis to left and right vertical axes and 10.9 degrees from central horizontal axis to up and down horizontal axes, when the average position of the subject's eyes is 60 cm directly in front of the eye tracker's filter surface.

The report shows a gaze point set for each cardinal position and eye. Each set has the ranges $[\phi_i - 2, \phi_i + 2]$ and $[\theta_j - 2, \theta_j + 2]$, where $\phi_i = \{-13.2, 0, 13.2\}$ and $\theta_j = \{-10.9, 0, 10.9\}$ are taken on their nine possible combinations. The mean and the standard deviation for each gaze point set and eye are shown for both ϕ - and θ -axes. All measurements are expressed in degrees with one significant floating point digit as explained in section 4.7 starting on page 32.

Figure 5.2 illustrates a graphical example of the report generated by the prototype application as implemented in this thesis project. In this example, the circles represent areas in which gaze points could be plotted. Thus, only a mean value and the standard deviation of each gaze point set for both ϕ -axis and θ -axis are shown for both the left and right eye at each point. In this graphical example, standard deviation is not shown since only representative circles which contain possible gaze points are drawn. In this way, it can be estimated how much the subject's gaze patterns deviate from an ideal orthotropia case. The graph shown by default for each point has a range of ± 2 degrees. If some value lies outside this range, then the box is re-scaled up to a range of ± 5 degrees, the labels on the box are shown, and grid lines are added to the current axes such that there is a grid line every 1° in order to allow the ophthalmologist to notice easily that some disorder has been recorded in the test (as simulated in figure 5.4 on page 47).

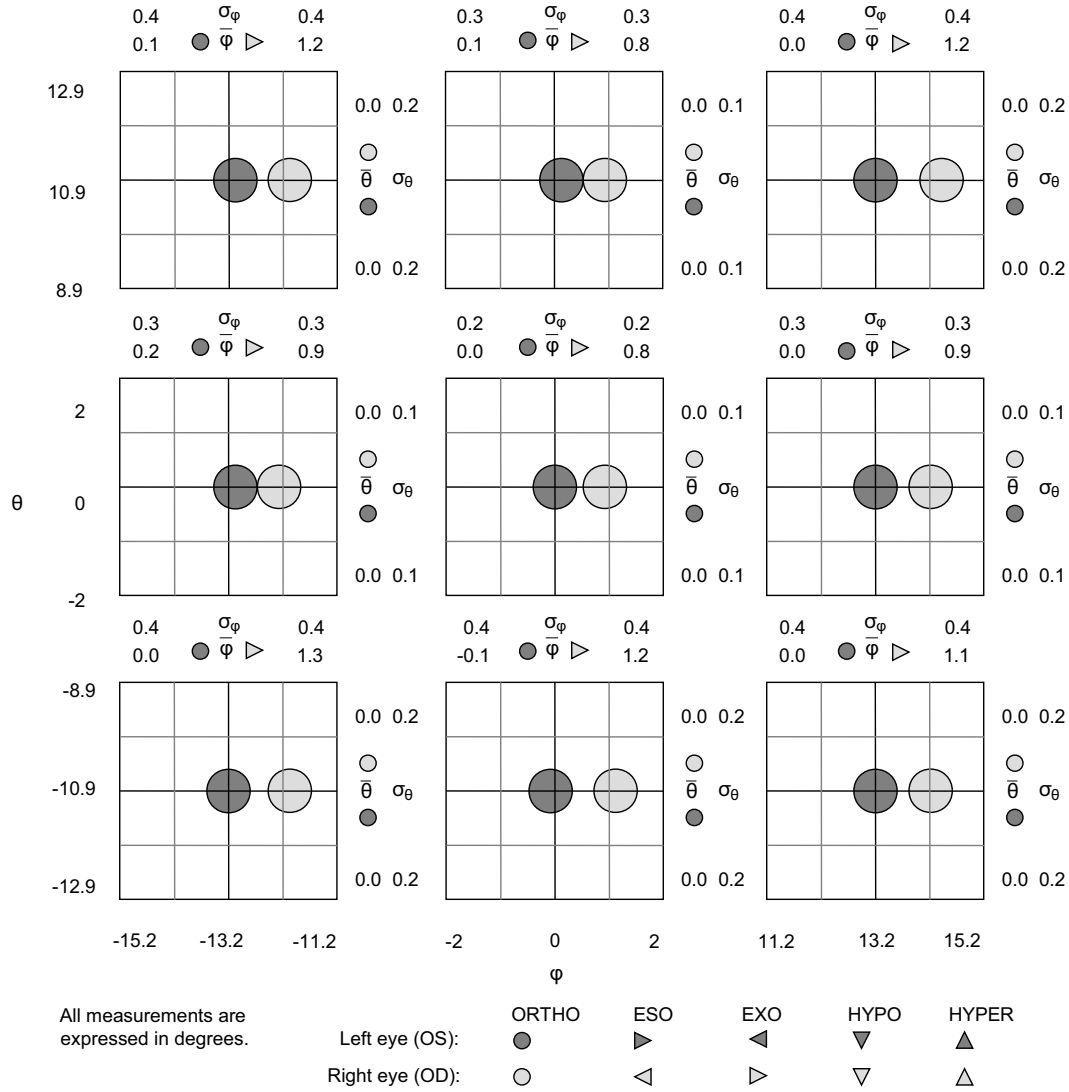
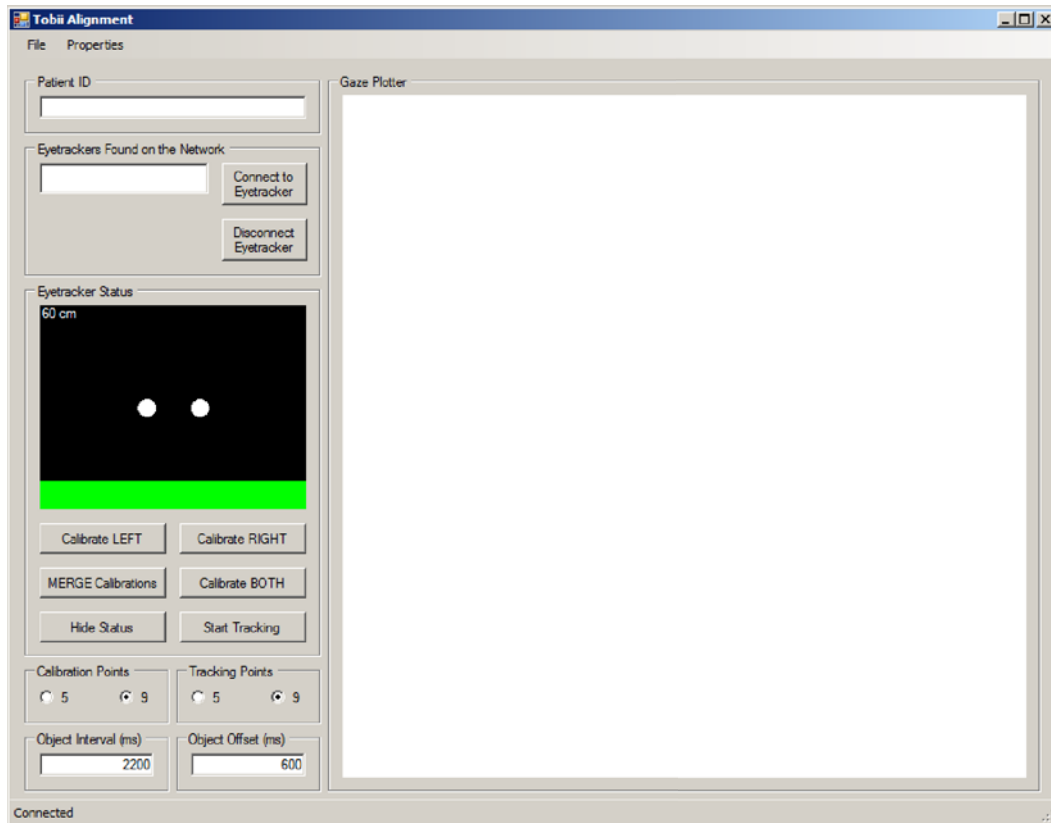


Figure 5.2: A graphical generalization of the *Tobii Alignment* report.

5.2 Workflow of *Tobii Alignment*

The prototype application, which in this thesis will be called *Tobii Alignment*, is self instructive and very simple to use in a clinical situation. It is composed of six menu items and eight boxes: *Save Calibration*, *View Calibration*, *Load Calibration*, *View Gaze Data*, *Load Gaze Data*, *Framerate*, *Patient ID*, *Eyetrackers Found on the Network*, *Eyetracker Status*, *Calibration Points*, *Tracking Points*, *Object Interval*, *Object Offset*, and *Gaze Plotter*. The basic screen layout is shown in figure 5.3.

Figure 5.3: The basic screen layout of *Tobii Alignment*.

The *Save Calibration* item gets calibration data from the *Active Calibration Buffer* (see section 4.5.3) and saves it into a binary *.CALIB* file. The *View Calibration* item gets calibration data from the *Active Calibration Buffer* and plots it as illustrated in figures 4.8 and 4.9 on page 30. The *Load Calibration* item reads calibration data from a binary *.CALIB* file, uploads it to the connected eye tracker, and plots it in the same format as for the *View Calibration* menu item. The *View Gaze Data* item gets tracking data from the current *ArrayList* of serializable *double*, *float*, *int*, and *long* values that describes the entire gaze data of a tracking test (see section 5.2.2), and other information such as how many stimuli were shown, where and in which order, the interval of duration of each one and the interval of duration of the set of samples considered as noise before and after the interval of interest; then the gaze points considered as signal samples regarding fixation are plotted in the *Gaze Plotter* box in real-time time, sample by sample. The data is also automatically saved to a file. The *Load Gaze Data* reads tracking data from a binary *.TRACK* file, copies it into the *ArrayList* mentioned before and plots the data in the same way as the *View Gaze Data* menu item.

The *Patient ID* box is where the patient's identifier is entered. This identifier will be used in conjunction with date, time, and type of calibration to name the calibration, tracking, and report files. For example, if the patient ID is 33565791, the date is October 6, 2012, the time is 12:34:56 and the tracking test is run after a monocular calibration, the name of the files will be 33565791-20121006123456-monocular. In this way, comparing and following up the patient's therapy is facilitated. This file name is generated when the binocular calibration is run or when both monocular calibrations for the left and right eye are merged. The tracking and report files will have the same root file name with the necessary number of suffixes if more than one tracking test is run for a given calibration.

The *Eyetrackers Found on the Network* box lists the eye trackers found on the network. When an eyetracker is found on the network it is added to the list. When it disappears from the network it is removed from the list.

The *Eyetracker Status* box visualizes the position of the eyes. This is mainly a tool to help the user position himself/herself in front of the tracker. This routine uses the relative eye position data from the left and right eye as explained in section 4.3 starting on page 22. This box also includes a set of six buttons as shown in figure 4.14 on page 40. The *Calibrate LEFT* and *Calibrate RIGHT* buttons run calibrations, assuming that the specialist is occluding the patient's right and left eye with an IR light transparent lens respectively. Then the calibration is saved in memory. The calibrations of the left and right eye will be saved into *.CALIB* binary files with the suffix "left" and "right" respectively when the user clicks on the *MERGE Calibrations* button. This button calls an external *.EXE* application provided by *Tobii Technology* which gets as input two *.CALIB* files, each one for each eye, and returns as output a *.CALIB* file after merging both eye calibration models. The file will be given a name in such a way it will be easily related to the input. The *Calibrate BOTH* button runs a binocular calibration, assuming that no IR light transparent lens is being used. Then the calibration is saved. The *Hide/Show Status* button disables/enables the *Eyetracker Status* box. The *Start Tracking* button runs a tracking test (see section 5.2.2).

The *Calibration Points* and *Tracking Points* boxes let the user choose how many stimuli will be shown in the calibration and tracking procedures respectively. The *Object Interval* and *Object Offset* boxes let the user decide the interval of duration of each of those objects and the interval of duration of the set of samples considered as noise before and after the interval of interest.

The *Gaze Plotter* box is the most important one. After a tracking test is run, left and right eye's samples are plotted in red and green respectively in real-time. There are 9 cardinal positions similar those it is illustrated in figure 5.2. The percentage of samples that were validated is printed as well. It was agreed with the supervisor of this thesis project to remove the labels regarding ophthalmic disorders by allowing the ophthalmologist to have the final say by observing the computed statistics. An example of the appearance of the *Gaze Plotter* is shown in figure 5.4.

96,11% samples were validated.

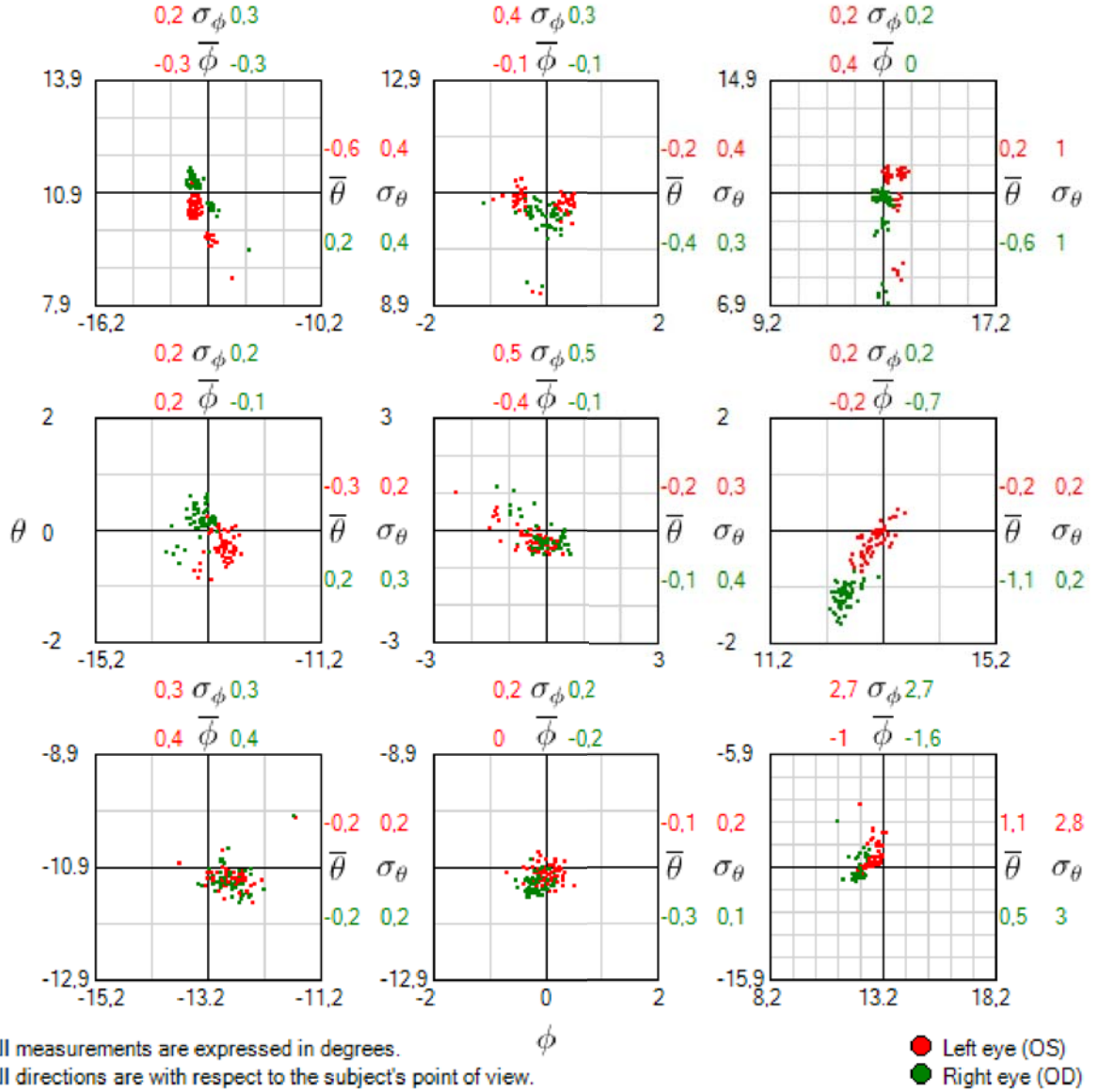


Figure 5.4: An example of the appearance of the *Gaze Plotter*. Note that boxes of the same row have the same origin in the horizontal axis, whereas boxes of the same column have the same origin in the vertical axis.

5.2.1 Calibration routines

The following are the calibration routines developed in this application. Note that many of the methods are not explained in detail due to confidentiality.

Listing 5.1: The *binocular calibration* routine

```

private void _calibrateBinocularButton_Click(object sender, EventArgs e)
{
    if (!_isTracking)
    {
        // Unsubscribe from gaze data stream
        _connectedTracker.StopTracking();
        _isTracking = false;
    }
    UpdateUIElements(); // Update user interface elements

    var runner = new CalibrationRunner();

    try
    {
        // Start a new calibration procedure
        Calibration binocularCalibration = runner.RunCalibration(↵
            _connectedTracker, _howManyCalibrationPoints);

        // Show a calibration plot if everything went OK
        if (binocularCalibration != null)
        {
            // Save the calibration on the right eye
            binocularFileName = participantTextBox1.Text + '-' + ↵
                UpdateTodayNow() + '-' + _howManyCalibrationPoints + "↵
                points-binocular.calib";
            SaveCalibration(binocularCalibration, binocularFileName);

            var resultForm = new CalibrationResultForm();
            resultForm.SetPlotData(binocularCalibration);
            resultForm.Text = "BINOCULAR Calibration Result";
            resultForm.ShowDialog();
            loadedCalibration = "binocular";
        }
        else
        {
            MessageBox.Show("Not enough data to create a calibration (or ↵
                calibration aborted).");
        }
    }
    catch (EyetrackerException ee)
    {
        MessageBox.Show("Failed to calibrate. Got exception " + ee,
            "Calibration Failed",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    finally
    {
        rightCalibration = null;
        leftCalibration = null;

        if (!_isTracking)
        {
            // Start subscribing to gaze data stream
            _connectedTracker.StartTracking();
            _isTracking = true;
        }
        UpdateUIElements(); // Update user interface elements
    }
}

```

Listing 5.2: The *monocular calibration on the left eye* routine

```

private void _calibrateLeftButton_Click(object sender, EventArgs e)
{
    // Hide the cursor when the mouse pointer enters the button.
    Cursor.Hide();

    if (_isTracking)
    {
        // Unsubscribe from gaze data stream
        _connectedTracker.StopTracking();
        _isTracking = false;
    }
    UpdateUIElements(); // Update user interface elements

    var runner = new CalibrationRunner(objectInterval);

    try
    {
        // Start a new calibration procedure
        leftCalibration = runner.RunCalibration(_connectedTracker, ←
            howManyCalibrationPoints);

        // Show a calibration plot if everything went OK
        if (leftCalibration != null)
        {
            var resultForm = new CalibrationResultForm();
            resultForm.SetPlotData(leftCalibration);
            resultForm.Text = "Monocular Calibration on the LEFT Eye Result←
                ";
            resultForm.ShowDialog();
            loadedCalibration = null;
        }
        else
        {
            MessageBox.Show("Not enough data to create a calibration (or ←
                calibration aborted).");
        }
    }
    catch (EyetrackerException ee)
    {
        MessageBox.Show("Failed to calibrate. Got exception " + ee,
            "Calibration Failed",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    finally
    {
        if (!_isTracking)
        {
            // Start subscribing to gaze data stream
            _connectedTracker.StartTracking();
            _isTracking = true;
        }
        UpdateUIElements(); // Update user interface elements

        // Show the cursor when the mouse pointer leaves the button.
        Cursor.Show();
    }
}

```

Listing 5.3: The *monocular calibration on the right eye* routine

```

private void _calibrateRightButton_Click(object sender, EventArgs e)
{
    if (!_isTracking)
    {
        // Unsubscribe from gaze data stream
        _connectedTracker.StopTracking();
        _isTracking = false;
    }
    UpdateUIElements(); // Update user interface elements

    var runner = new CalibrationRunner();

    try
    {
        // Start a new calibration procedure
        rightCalibration = runner.RunCalibration(_connectedTracker, ←
            _howManyCalibrationPoints);

        // Show a calibration plot if everything went OK
        if (rightCalibration != null)
        {
            var resultForm = new CalibrationResultForm();
            resultForm.SetPlotData(rightCalibration);
            resultForm.Text = "Monocular Calibration on the RIGHT Eye ←
                Result";
            resultForm.ShowDialog();
            loadedCalibration = null;
        }
        else
        {
            MessageBox.Show("Not enough data to create a calibration (or ←
                calibration aborted).");
        }
    }
    catch (EyetrackerException ee)
    {
        MessageBox.Show("Failed to calibrate. Got exception " + ee,
            "Calibration Failed",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    finally
    {
        if (!_isTracking)
        {
            // Start subscribing to gaze data stream
            _connectedTracker.StartTracking();
            _isTracking = true;
        }
        UpdateUIElements(); // Update user interface elements
    }
}

```

Listing 5.4: The *Merge Calibrations* routine

```

private void _mergeCalibrationsButton_Click(object sender, EventArgs e)
{
    DateTime now;
    string strCmdLine;

    if (_isTracking)
    {
        // Unsubscribe from gaze data stream
        _connectedTracker.StopTracking();
        _isTracking = false;
    }
    UpdateUIElements(); // Update user interface elements

    todayNow = UpdateTodayNow();

    rightFileName = patientTextBox1.Text + '-' + todayNow + '-' + ↵
        howManyCalibrationPoints + "points-right.calib";
    leftFileName = patientTextBox1.Text + '-' + todayNow + '-' + ↵
        howManyCalibrationPoints + "points-left.calib";
    targetFileName = patientTextBox1.Text + '-' + todayNow + '-' + ↵
        howManyCalibrationPoints + "points-merged.calib";

    SaveCalibration(rightCalibration, rightFileName);
    SaveCalibration(leftCalibration, leftFileName);

    try
    {
        if (File.Exists(rightFileName))
            if (File.Exists(leftFileName))
            {
                now = DateTime.Now;
                while (DateTime.Now.Second - now.Second < 1) { }

                // The following usage with the "-2" optional parameter implies two↵
                // -eyed calibrations
                strCmdLine = "/c merge-calibrations " + leftFileName + ' ' + ↵
                    rightFileName + ' ' + targetFileName + " -2";
                System.Diagnostics.Process.Start("cmd", strCmdLine);
                LoadAndViewMergedCalibration();
            }
        catch (Win32Exception ee)
        {
            MessageBox.Show("Failed to merge calibrations. Got exception " + ee↵
                ,
                "Error",
                MessageBoxButtons.OK,
                MessageBoxIcon.Error);
        }
        finally
        {
            if (!_isTracking)
            {
                // Start subscribing to gaze data stream
                _connectedTracker.StartTracking();
                _isTracking = true;
            }
            UpdateUIElements(); // Update user interface elements
        }
    }
}

```

5.2.2 Tracking routine

After the calibration is set, the *Start Tracking* button is enabled. If the user clicks on this button, the routine that will be called is shown in listing 5.5. Note that many of the methods are not explained in detail due to confidentiality.

Listing 5.5: The *Start Tracking* routine

```
private void _trackRecordButton_Click(object sender, EventArgs e)
{
    // Hide the cursor when the mouse pointer enters the button.
    Cursor.Hide();

    if (_isTracking)
    {
        // Unsubscribe from gaze data stream
        _connectedTracker.StopTracking();
        _isTracking = false;
    }
    UpdateUIElements(); // Update user interface elements

    var runner = new TrackingRunner(objectInterval);

    UpdateScreen(); // Get the x-configuration of the connected tracker

    // Create queue of Tracking Points
    _pointQueue = CreatePointQueue(howManyTrackingPoints);

    gazeDataItems = new ArrayList();

    _isRecording = true;
    objectInterval = runner.RunTracking(_connectedTracker, _pointQueue);
    _isRecording = false;

    if (gazeDataItems != null)
    {
        SaveTracking();
        isGazeDataLoadedFromFile = false;
        gazePlot1.updateGazePlot(gazeDataItems, howManyTrackingPoints, ←
            objectInterval, objectOffset);
    }
    else
    {
        MessageBox.Show("Not enough data to create a tracking plot (or ←
            tracking aborted).");
    }

    if (!_isTracking)
    {
        // Start subscribing to gaze data stream
        _connectedTracker.StartTracking();
        _isTracking = true;
    }
    UpdateUIElements(); // Update user interface elements

    // Show the cursor when the mouse pointer leaves the button.
    Cursor.Show();
}
```

Chapter 6

Evaluation

This chapter contains the evaluation performed for the system architecture and prototype presented in chapter 5. Three different sets of tests were made. An important precaution regarding reliability is to measure all subjects with the same device and firmware for each study in order to avoid possible sources of variance.

6.1 Study I – Measurement of *2D gaze points* on the tracking area

The purpose of this study was to measure *relative gaze points* of sixteen subjects above the age of 25 as a first concept of solution of the prototype application. Both binocular and monocular calibrations were conducted at *Tobii Technology* by using a *Tobii T60 Eye Tracker*. The IR transparent lenses mentioned on page 17 were used to conduct monocular calibration.

It turned out that crossed fixations¹ were obtained at distances of between 60 and 65 cm. For binocular calibration crossed fixations of larger than 40 pixels² were observed in 18.7% of the objects across all of the objects shown to the sixteen subjects. For monocular calibration by using a 750 nm wavelength IR transparent lens the percentage was 38.6%. For monocular calibration by using a 800 nm wavelength IR transparent lens crossed fixations of larger than 40 pixels on the tracking plane were collected in 45.6% of the objects across all of the objects shown to the sixteen subjects as well.

Figures 6.1, 6.2, and 6.3 show examples of crossed fixations obtained from 750 nm monocular, 800 nm monocular, and binocular calibrations for subjects #4, #11, and #10 respectively.

¹A *crossed fixation* occurs when the left eye is located further to the right than the right eye.

²A segment of 40 pixels long on the diagonal of a 17-inch monitor with a resolution of 1280×1024 pixels observed from a point that is 60 cm straightly in front of the screen describes an angle of 1° . This threshold of 1° could discard crossed fixations since it is of the same order of magnitude as a microsaccade [18].

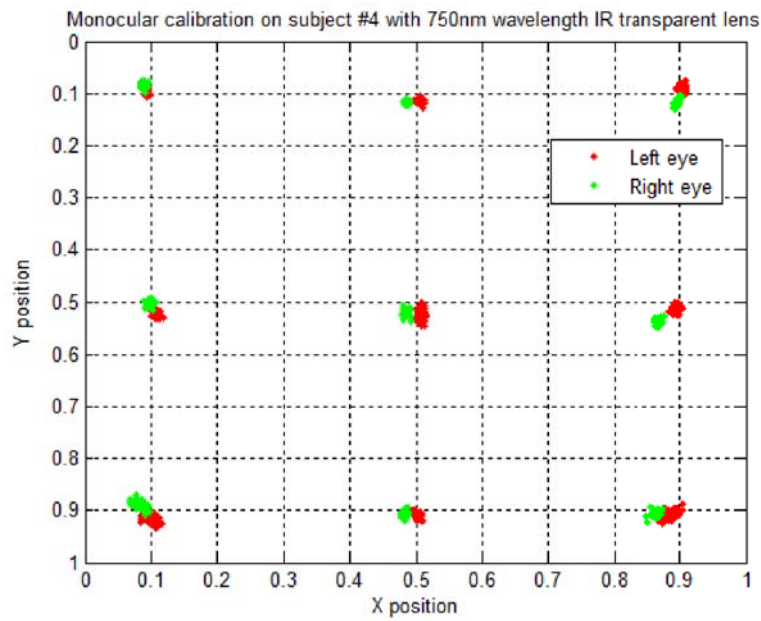


Figure 6.1: Monocular calibration on subject #4 with 750 nm wavelength IR transparent lens.

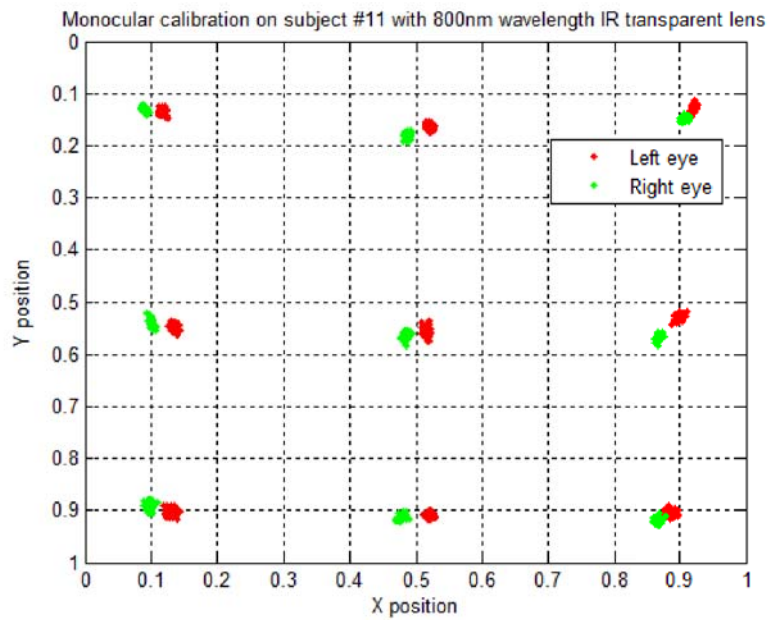


Figure 6.2: Monocular calibration on subject #11 with 800 nm wavelength IR transparent lens.

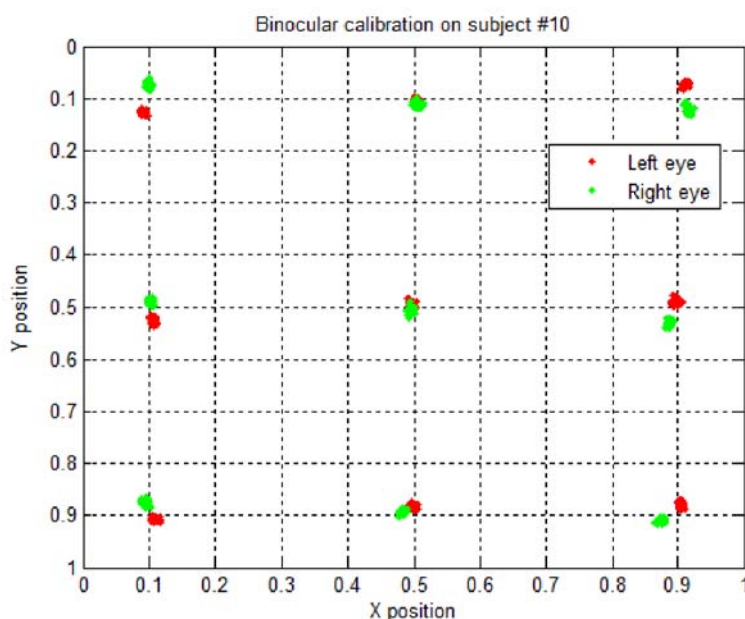


Figure 6.3: Binocular calibration on subject #10.

These results show that the 750 nm wavelength IR transparent lens produces better results than the 800 nm IR transparent lens. This can be reasonable since a typical human eye will respond to wavelengths from about 390 to 750 nm although some people may be able to perceive wavelengths from 380 to 780 nm [8]. Thus, some of the sixteen subjects who had participated in this study may have fixated unconsciously with the eye occluded by the 750 nm wavelength IR transparent lens as well as the not occluded eye. This might be why the results obtained after 750 nm monocular calibration are more similar to the obtained after binocular calibration (without any IR transparent lens) than the obtained after 800 nm monocular calibration.

This observation motivated a series of tests using VOG with monocular calibration, which has been performed by researchers at *Bernadottelaboratoriet* at *S:t Eriks Ögonsjukhus*. Crossed fixations at distances of between 60 and 65 cm were obtained there as well (these tests are not included in this report). The results from the VOG looked very much like the results from the round of tests that I have conducted with the *Tobii T60 Eye Tracker* in Study I.

This finding matched the results expected from [17]. Binocular eye movements of normal adult readers were examined as they read single sentences by Antje Nuthmann and Reinhold Kliegl in this work. Analyses of horizontal and vertical fixation disparities carried out in Nuthmann's and Kliegl's work indicated that the most prevalent type of disparate fixation were **crossed** as well.

Either the distance to the display of the *Tobii T60 Eye Tracker* or that a display is used affect the results. This motivated studies II & III.

6.2 Study II – Measurement of angles of deviation between gaze and target vectors

The study II has been carried out after implementing a new version of the prototype application. This phase of testing included calculation of angle of deviation (as was explained in section 5.1). The purpose of this study was to measure gaze deviations of twelve subjects above the age of 25 using a second version of the prototype application. Both binocular and monocular calibrations were conducted at *Tobii Technology* by using a *Tobii T60 Eye Tracker*. A 750 nm wavelength IR transparent lens provided by production staff working at *Tobii Technology* was used to occlude the subject's corresponding eye since this lens performed better than the 800 nm wavelength lens did in the study I (see section 6.1).

In this study eighteen tracking tests were performed on every subject. The tests were divided into two rounds that took place on two different days and at different times (one round in the morning, one round in the afternoon). Each round consisted of nine tracking tests: three tests after a binocular calibration, three tests after a monocular calibration run first on the left eye (MONO LR), and three tests after a monocular calibration run first on the right eye (MONO RL). For the first round, the MONO LR procedure was performed first, followed by the MONO RL procedure. For the second round, the order of the monocular calibration was reversed. In this way, repeatability and independence of results obtained after different combinations of procedures at different times were analyzed and evaluated.

Figures 6.4 to 6.7 show average deviation with respect to the corresponding target point for every subject, calibration, session, and test. Looking at these figures, the reader may see that many tests produced similar results when replicated on a given subject. Thus, there is a good grade of repeatability and independence for every pair of sessions.

Figure 6.8 shows the average *vergence* angle values for every subject, calibration, session, and test. Looking at this figure, the reader may see that most of the *vergence* angle values are negative. Since the origin is the vertical axis of the corresponding target point, when we have $\phi_L - \phi_R < 0$, there is **uncrossed** fixation on average across all of the objects shown for each subject for the corresponding calibration, what implies that uncrossed fixations were more frequent than crossed fixations.

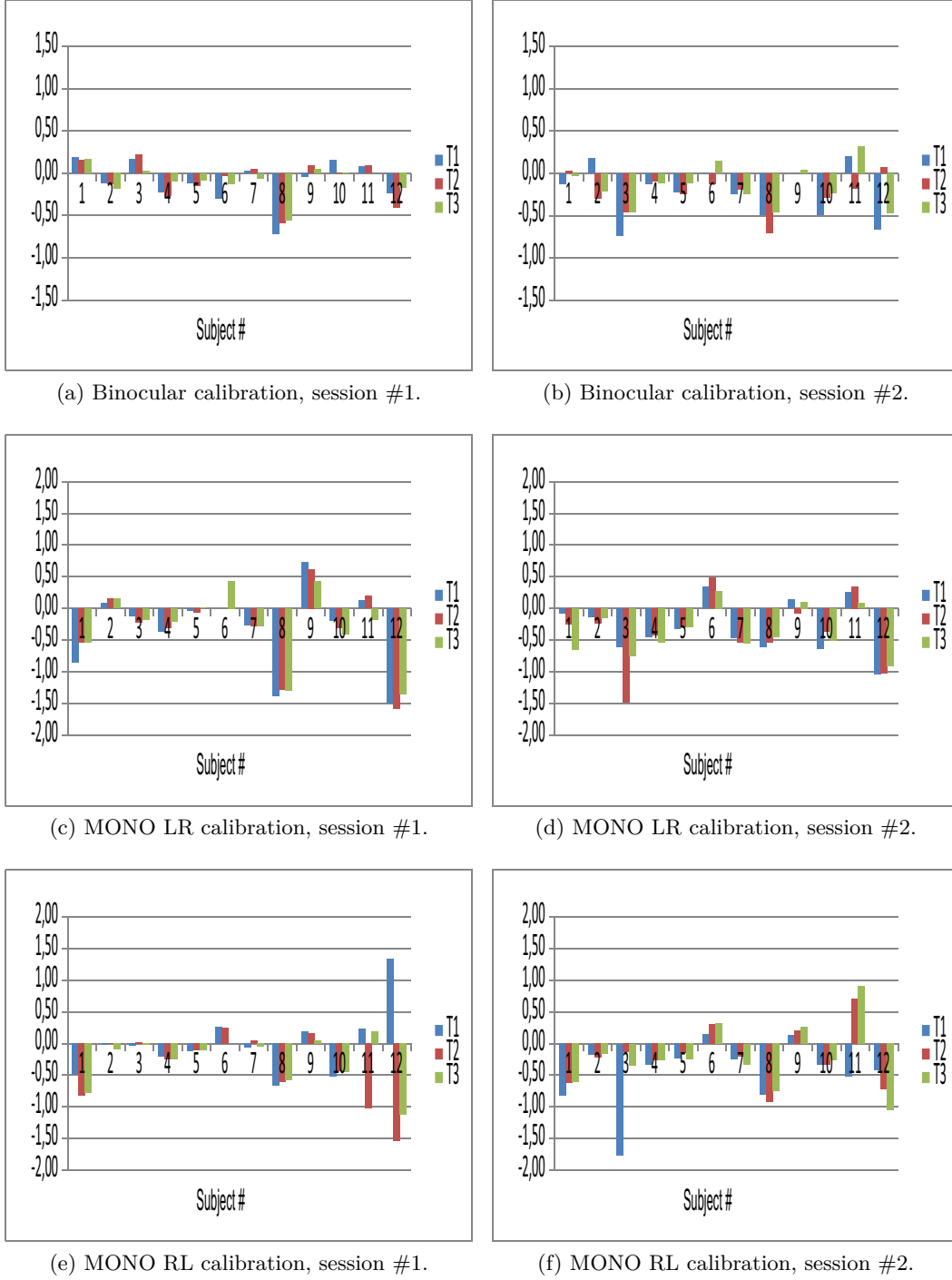
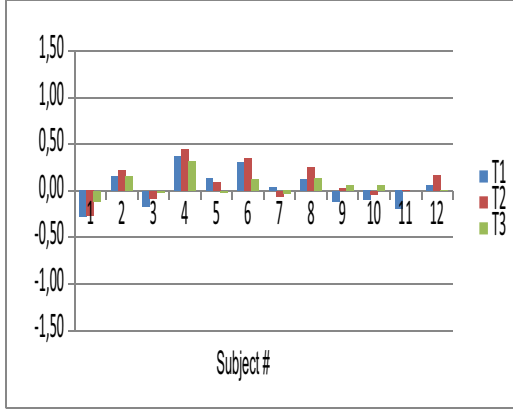
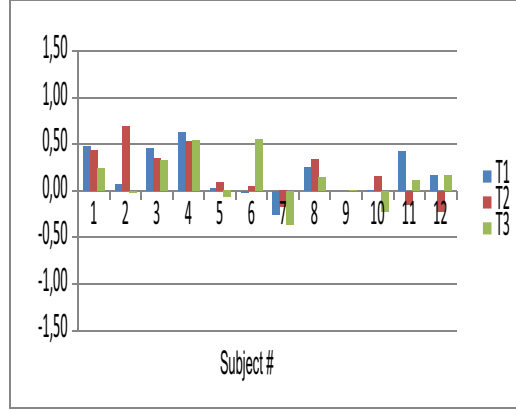


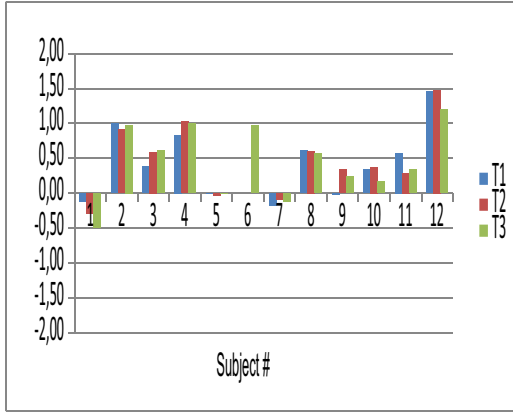
Figure 6.4: Average ϕ_L deviations for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.



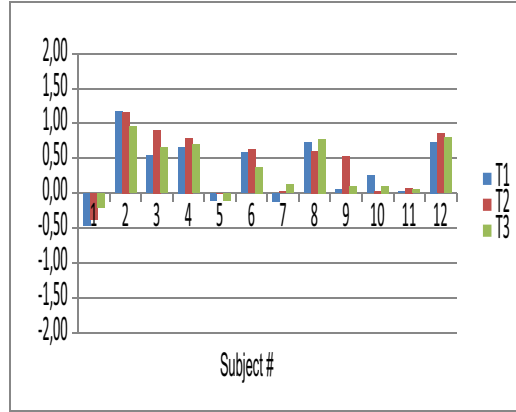
(a) Binocular calibration, session #1.



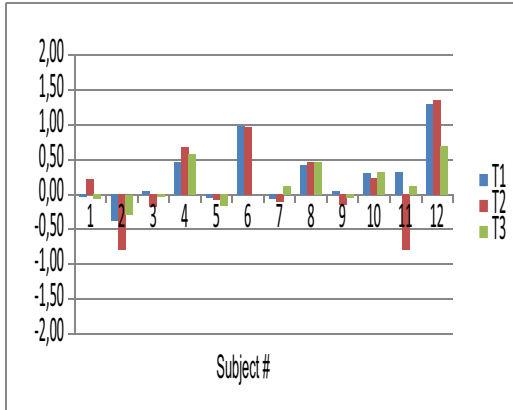
(b) Binocular calibration, session #2.



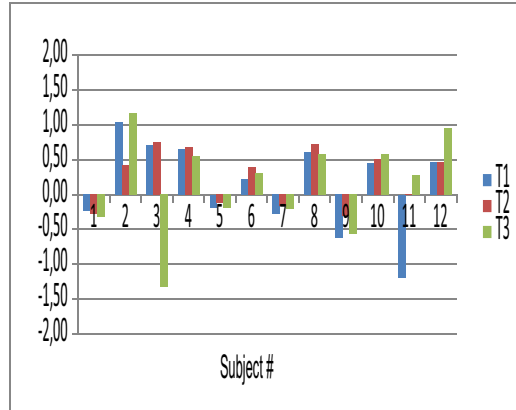
(c) MONO LR calibration, session #1.



(d) MONO LR calibration, session #2.



(e) MONO RL calibration, session #1.



(f) MONO RL calibration, session #2.

Figure 6.5: Average θ_L deviations for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.

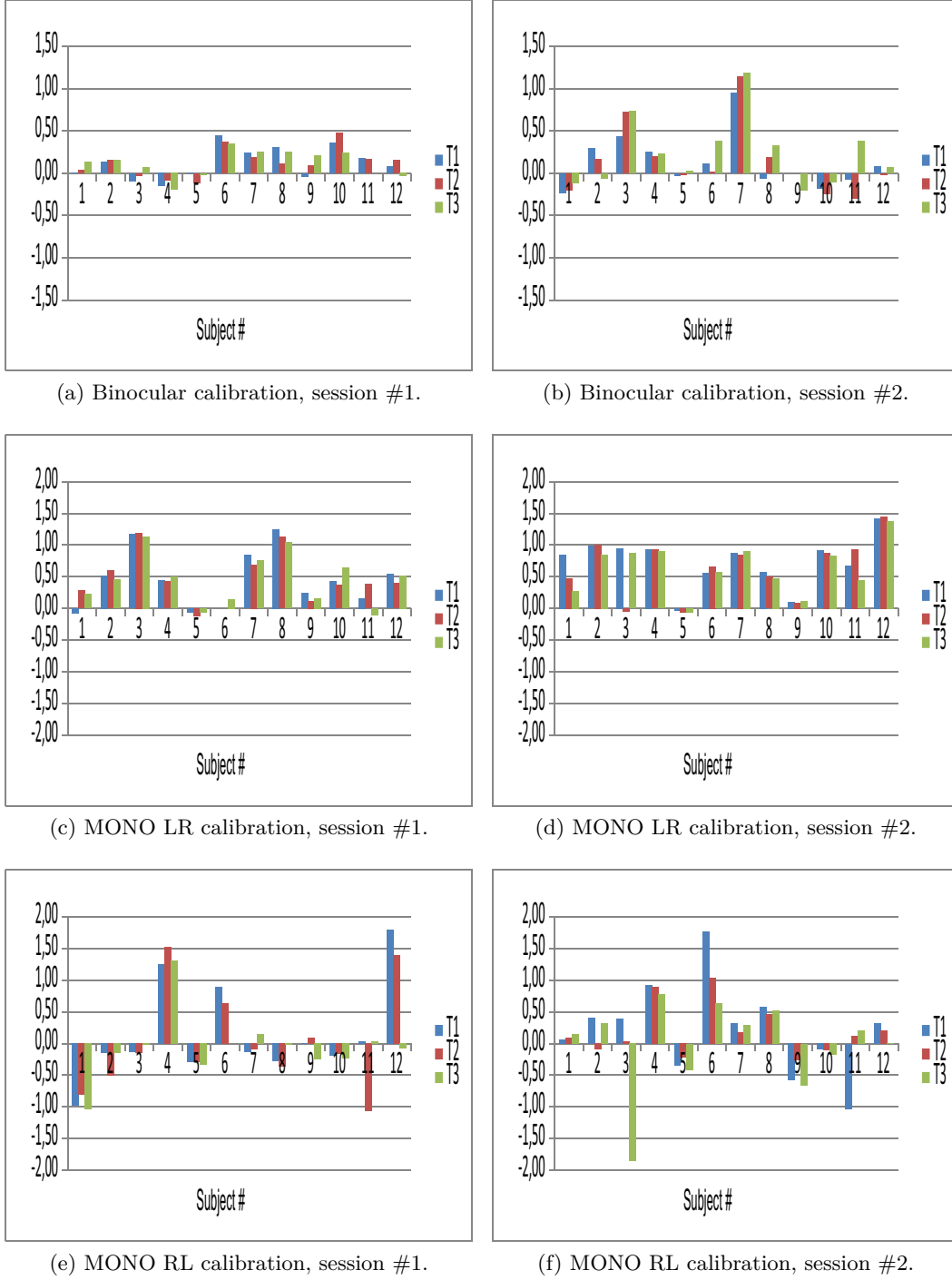
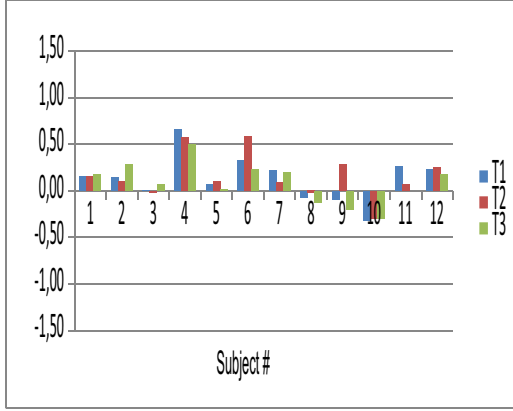
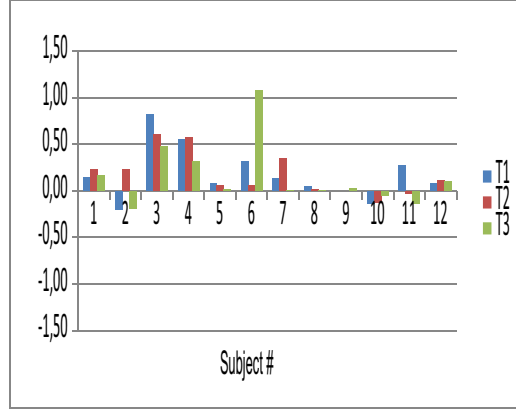


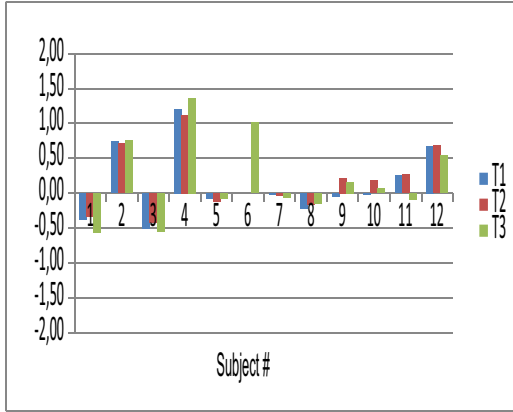
Figure 6.6: Average ϕ_R deviations for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.



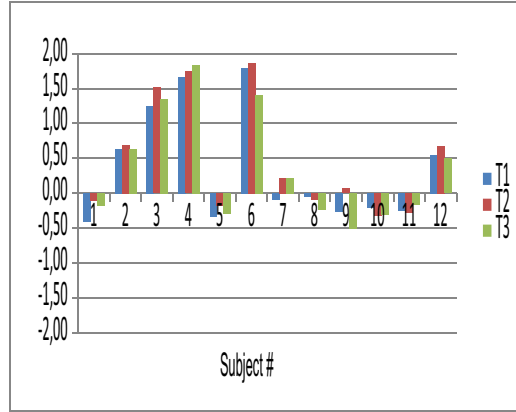
(a) Binocular calibration, session #1.



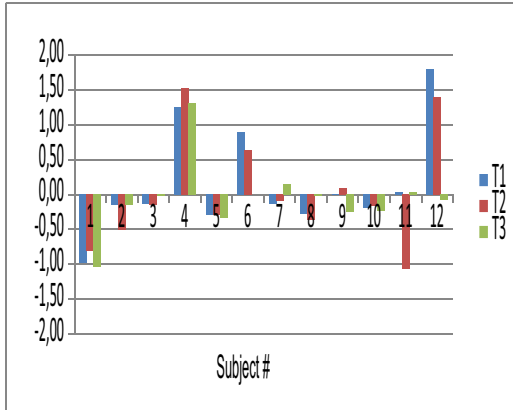
(b) Binocular calibration, session #2.



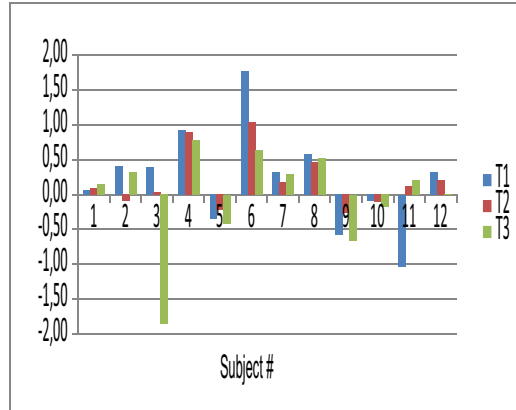
(c) MONO LR calibration, session #1.



(d) MONO LR calibration, session #2.

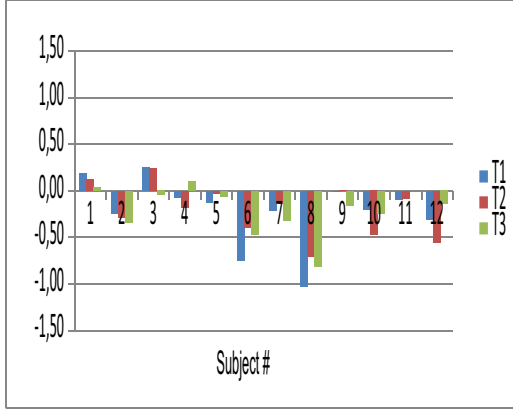


(e) MONO RL calibration, session #1.

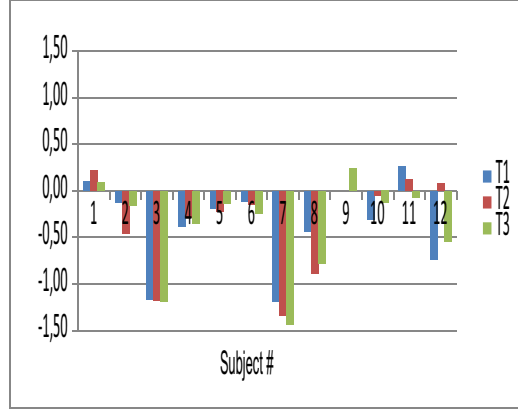


(f) MONO RL calibration, session #2.

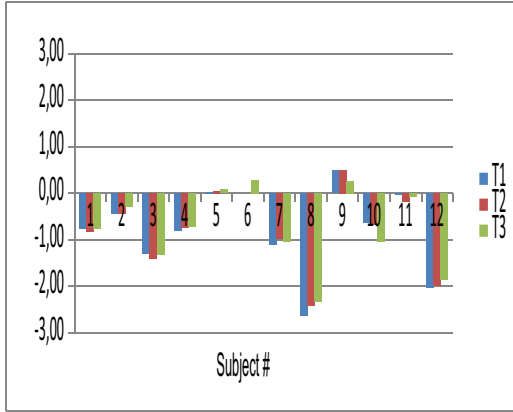
Figure 6.7: Average θ_R deviations for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.



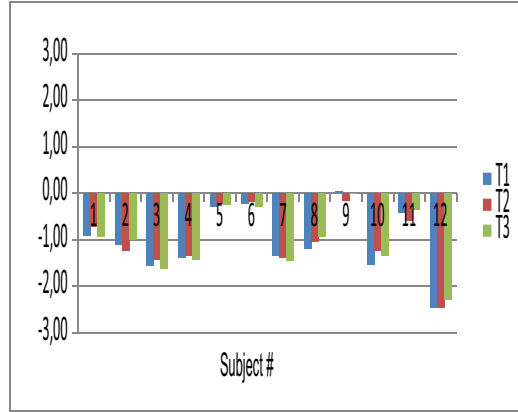
(a) Binocular calibration, session #1.



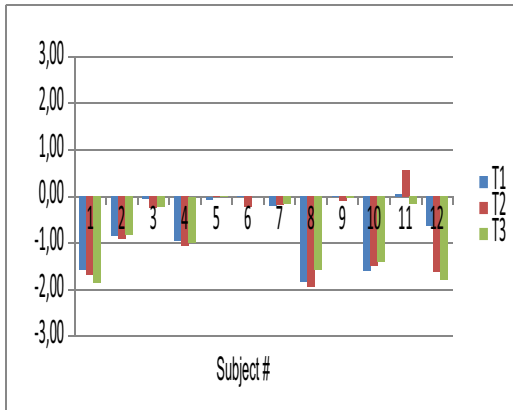
(b) Binocular calibration, session #2.



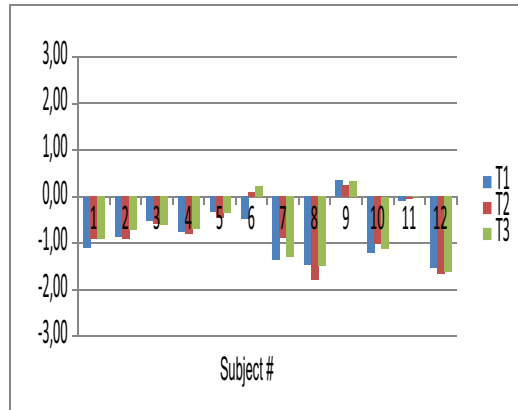
(c) MONO LR calibration, session #1.



(d) MONO LR calibration, session #2.



(e) MONO RL calibration, session #1.



(f) MONO RL calibration, session #2.

Figure 6.8: Average *vergence* angle ($\phi_L - \phi_R$) for every subject, calibration, session, and test. All measurements are expressed in degrees. Note that the plots of binocular calibration are scaled to a different range.

However, the values shown before are the average of all the deviations with respect to the corresponding target point for a given test. If we calculate the average of the deviations with respect to each target point separately between all the subjects, then the crossed fixations are obtained as observed in the study I. This is why positive *vergence* angle values are obtained (see table 6.1).

Table 6.1: Mean and standard deviation values with respect to the corresponding target point for the nine cardinal positions and all the subjects. Values in red are out of the desired range (a deviation within the order of magnitude of a microsaccade may be ignored, and the implication of a positive *vergence* value is *crossed fixation*).

	$\overline{\phi_L}$	$\overline{\phi_R}$	$\overline{v}$	$\overline{\theta_L}$	$\overline{\theta_R}$	σ_{ϕ_L}	σ_{ϕ_R}	σ_{θ_L}	σ_{θ_R}
NW	0.3	-0.3	0.6	-0.7	-0.1	0.4	0.4	0.3	0.3
N	0.3	-0.2	0.5	-0.5	-0.5	0.2	0.2	0.2	0.2
NE	0.1	-0.2	0.3	-0.2	-0.7	0.2	0.2	0.2	0.2
W	0.7	0.0	0.7	-0.5	0.0	0.2	0.2	0.2	0.2
C	0.2	-0.5	0.7	-0.4	-0.4	0.2	0.2	0.3	0.2
E	-0.2	-0.8	0.6	-0.2	-0.6	0.2	0.2	0.2	0.2
SW	1.0	0.3	1.3	0.1	0.4	0.4	0.4	0.3	0.3
S	0.4	-0.6	1.0	0.1	0.3	0.4	0.3	0.1	0.3
SE	-0.5	-1.3	0.8	0.2	0.2	0.4	0.4	0.4	0.3

Figures 6.9, 6.10, and 6.11 show the linear regression for ϕ_L , ϕ_R , θ_L , and θ_R for average results of binocular, MONO LR, and MONO RL calibrations in the two different sessions for all the subjects. In these figures, each blue dot represents the average angle of deviation across all of the objects shown for each subject for the corresponding calibration and variable. This offers a better approach to see how valid the results are regarding repeatability and independence. Results show R^2 -values up to 0.7566.

These results in conjunction with the illustrated in figures 6.4 to 6.7 suggest that there is a good grade of repeatability and independence of results obtained after different combinations of procedures at different times.

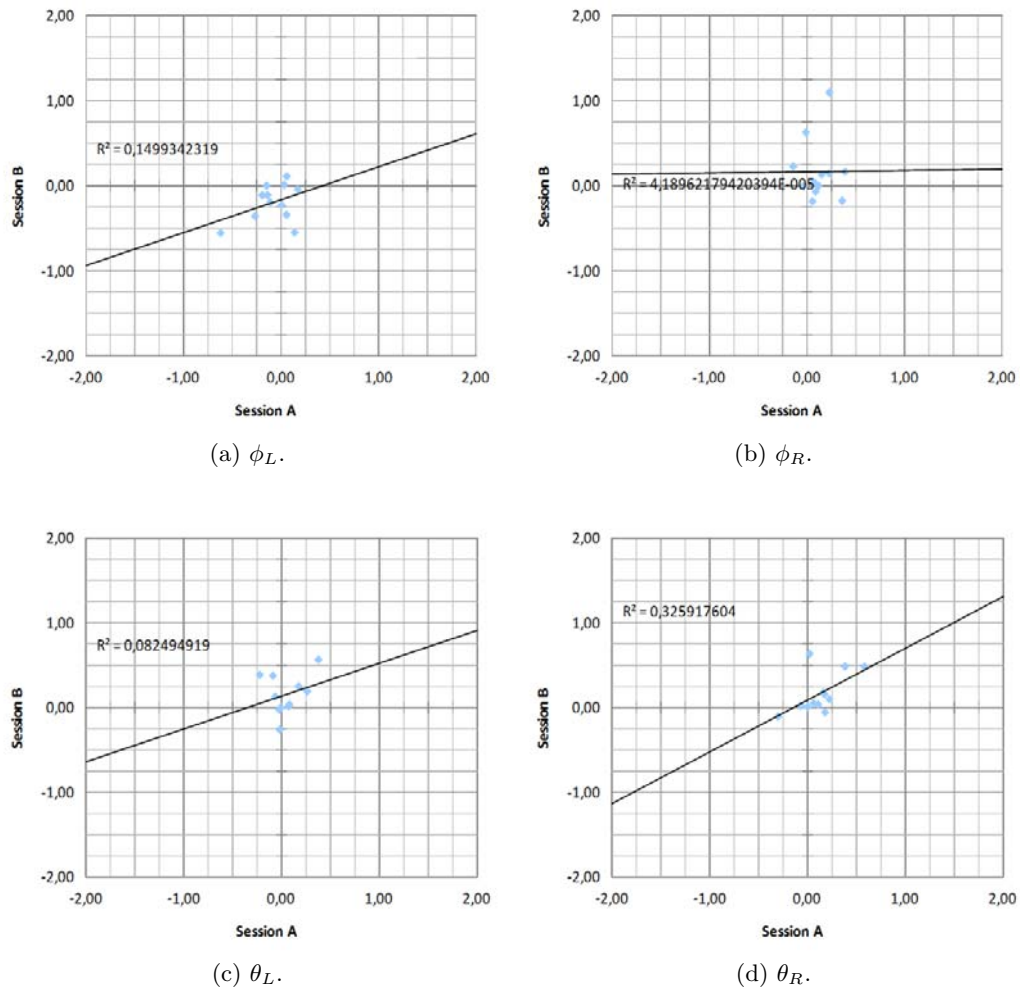


Figure 6.9: Linear regression for binocular calibration. All measurements are expressed in degrees.

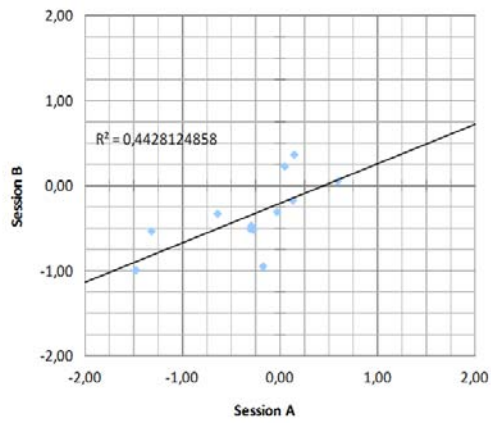
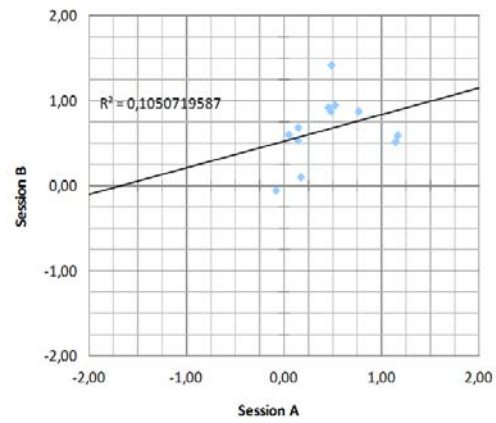
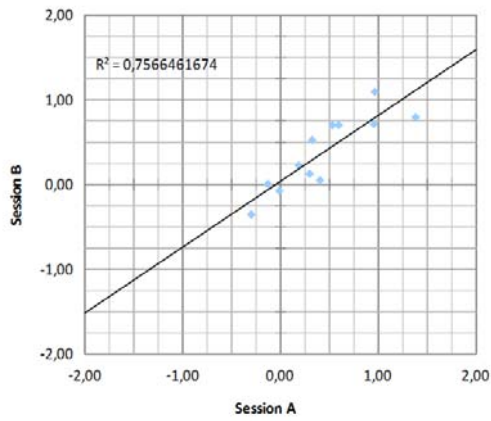
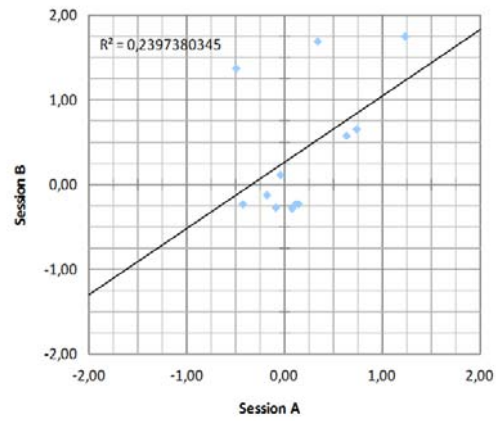
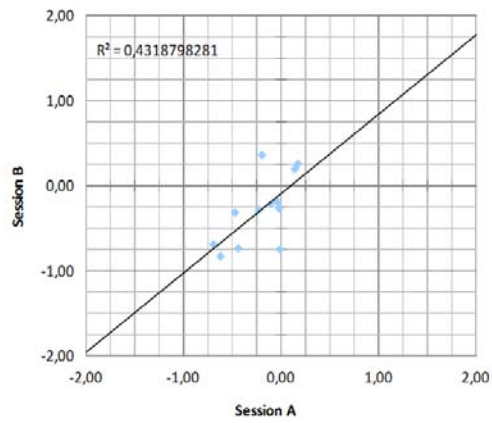
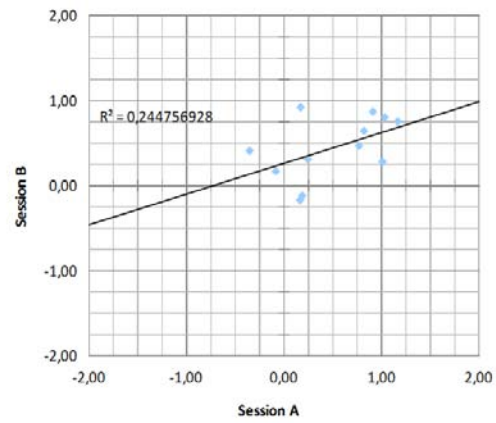
(a) ϕ_L .(b) ϕ_R .(c) θ_L .(d) θ_R .

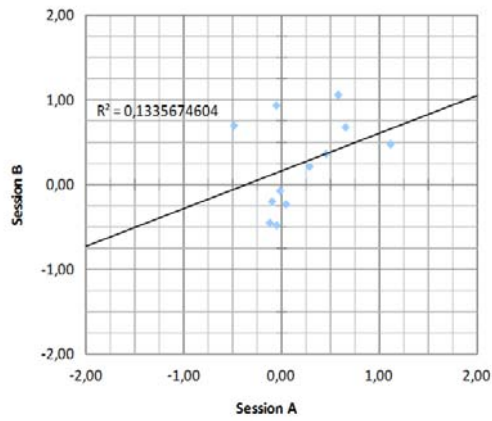
Figure 6.10: Linear regression for MONO LR calibration. All measurements are expressed in degrees.



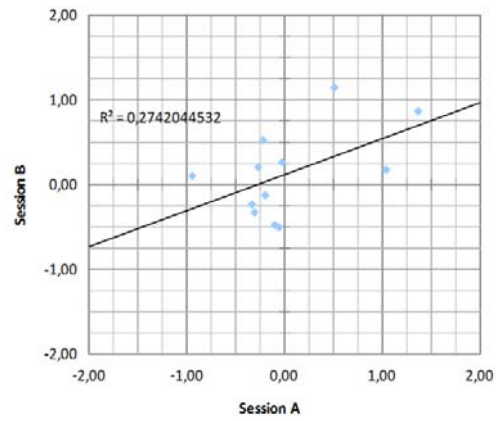
(a) ϕ_L .



(b) ϕ_R .



(c) θ_L .



(d) θ_R .

Figure 6.11: Linear regression for MONO RL calibration. All measurements are expressed in degrees.

6.3 Study III – Tobii Alignment validation against scleral search coil technique

In the study III several scleral search coil experiments were conducted, where a *Tobii T120 Eye Tracker* and the SSC were used simultaneously. The SSC is considered the gold standard in eye tracking, thus this experiment shows precisely how close the results of the prototype application with the *Tobii T120 Eye Tracker* is to this gold standard.

This section describes the validation procedure carried out to compare the new method with the SSC. Although, for this study, all angles of deviation were expressed with respect to the center of the screen instead of the corresponding target point (as were the angles of deviation in the earlier studies). Since the results obtained in the study II show that the test results were independent of the different combinations of procedures conducted at different times, thus in Study III only MONO LR calibration was run before each set of tracking tests. Before describing this procedure, the principles of operation of the SSC installed at *Bernadottelaboratoriet* at *S:t Eriks Ögonsjukhus* are detailed in the following subsection.

6.3.1 Principles of operation of a scleral search coil

The basic principles of the electromagnetic recording used in this measurement technique originate from research conducted in [22, p. 137–145] and later in [25, p. 447–450]. A small induction coil, with N turns, is embedded in a flexible ring of silicone rubber which adheres to the limbus (the border of the cornea and the sclera) of the adult human eye. Around the head of the subject an alternating horizontal H_x and vertical H_y magnetic field is generated and consequently an alternating voltage E_i will be induced in the coil in accordance with Faraday's law:

$$E_i = -\frac{d\Psi}{dt} \quad (6.1)$$

If all turns contain the same flux (ϕ) we find with the coupled flux:

$$\Psi = \sum_{k=1}^N \phi_k \quad (6.2)$$

that:

$$E_i = -N \cdot \frac{d\phi}{dt} \quad (6.3)$$

Flux is represented by:

$$\phi_A = \int_A \mu H \cdot dA. \quad (6.4)$$

In which μ is the absolute permeability ($\mu = \mu_0\mu_r$) and H_n is the magnetic induction component perpendicular to the total area of A.

In case of a homogeneous magnetic field equation 6.4 becomes:

$$\phi = \mu H \cdot A \quad (6.5)$$

When the eye looks up by the angle θ and the vertical magnetic field intensity is given by $H_v = H \cos \omega t$, then the induced voltage for this vertical position will be:

$$E_i = \omega N A \mu H \sin \theta \sin \omega t \quad (6.6)$$

Before we add a second coil to these considerations we define appropriate angles in a coordinate system. We consider a sphere comparable with the terrestrial globe. The eye is positioned in the centre and looking in the direction of the intersection between the earth's equator and the 0° meridian Greenwich.

The axes of rotation are taken as follow:

x -axis

- : 90° West longitude to 90° East longitude.

y -axis

- : North to South.

z -axis

- : 0° longitude to 180° longitude.

As a result of these assumptions the primary positions of the horizontal/vertical coils are in parallel with respectively the $x - y$ and $y - z$ plane. Horizontal deflection can thus be described as a rotation of ϕ degrees about the y -axis. For a vertical deflection we find a rotation of respectively θ degrees about the x -axis.

When we use two magnetic fields spatially and temporally in quadrature and two lens coils in spatial quadrature, it is easy to deduce that:

1. Position of the axis of rotation for vertical deflection changes with horizontal deflection.
2. Position of the axis of rotation for horizontal deflection does not change with vertical deflection.

These deductions lead to two DC signals:

$$E_\phi \propto \sin \phi \cos \theta \quad (6.7)$$

$$E_\theta \propto \sin \theta \quad (6.8)$$

If ϕ and θ are all less than 20° , $\sin \phi$ departs from ϕ in radians by less than 2% and $\cos \theta$ departs from unity by less than 6% so that approximately:

$$E_\phi \propto \phi \quad (6.9)$$

$$E_\theta \propto \theta \quad (6.10)$$

The coil-frame is a cube (70x70x70 cm) in which six coils are embedded. Each coil is individually shielded with copper shielding tape. In this way the electric field, which is generated at the same time as the magnetic field, is considerably reduced. The left and right coils generate a horizontal magnetic field, while the top and bottom coils generate a vertical magnetic field. The current through the horizontal coils is a sine waveform, while the current through the vertical coils is a cosine waveform. Both current values are identical (325 mA rms) and both have the same frequency (20 kHz).

For easy calibration, the coil-frame is supplied with a third set of coils. The current through these coils corresponds to an eye deflection of 10° . The actual value is 325 times the sine of 10° , which is 56.4 mA (rms).

A calibration signal, equivalent to 10 degrees eye deflection, is obtained by pressing the calibration button. Pressing the H-calibration button will induce a sine waveform magnetic field, while pressing the V-calibration button will induce a cosine waveform magnetic field.

With the aid of a calibration device, capable of simulating horizontal and vertical position a registration of a calibration is recorded. A current with a value equal to the sine of 10° times the regular current (325 mA rms) is fed to the calibration magnet coils located in front and at the back of the SSC. The SSC is thus energized by the calibration magnetic field perpendicular to this coil. Whether the coil is not exactly perpendicular to this calibration magnetic field does not influence this

calibration technique much. When for example the primary position of the coil is 5° to the right the induced voltage will only be 0.4% less ($\cosine\ 5^\circ = 0.996$).

6.3.2 Validation procedure

Two tables of events were designed by Gustaf Öqvist Seimyr, the supervisor of this thesis project, in order to study a comparison between the SSC and the *Tobii T120 Eye Tracker* performance for both binocular and monocular calibrations. These events are listed in tables 6.2 and 6.3.

Table 6.2: Binocular calibration procedure in a SSC setting.

Event	Duration (s)	Time (ms)
Start SSC recording	0	0
View far center target (600 cm)	10	10000
Horizontal and vertical SSC calibration signal	5	15000
Place <i>Tobii T120 Eye Tracker</i> in front of the subject	15	30000
View near center target (60 cm)	10	40000
Run binocular calibration	30	70000
Run tracking test three times	90	160000
Stop SSC recording	0	160000

Table 6.3: Monocular calibration procedure in a SSC setting.

Event	Duration (s)	Time (ms)
Start SSC recording	0	0
View far center target (600 cm)	10	10000
Horizontal and vertical SSC calibration signal	5	15000
Place <i>Tobii T120 Eye Tracker</i> in front of the subject	15	30000
View near center target (60 cm)	10	40000
Calibrate left eye (OS)	30	70000
Calibrate right eye (OD)	30	100000
Run tracking test three times	90	190000
Stop SSC recording	0	190000

After all the assumptions made in section 6.3.1, it is possible to calculate all the angles of deviation values from the text file returned by an application which records the voltage values at a framerate of 1 kHz for both the left and right eye from the SSC system. In that text file, voltage values are organized in a matrix in such a way the first column is the time stamp in seconds with a precision of 1 ms, the second and third columns are the right eye's horizontal and vertical deflection

voltages respectively, and the fourth and fifth columns are the left eye's horizontal and vertical deflection voltages respectively. There are as many rows as samples.

6.3.3 Transformation from voltage to angle of deviation

To transform the voltage values to angle of deviation values, the maximum absolute value for each voltage column for rows from samples 10001 to 15000 (see tables 6.2 and 6.3) which corresponds with an eye deflection of 10° is found. This value corresponds to the event when the volunteer was looking at the far center target point, and the calibration button was pressed to obtain the calibration signal. Then, all the voltage values of each column are divided by their corresponding maximum absolute values and finally multiplied by 10° . In this way, all values are transformed to angle of deviation expressed with respect to the vector from the eye's position to the far center target point (far center target vector).

In order to avoid considering a peak voltage value as maximum, we utilize the maximum of the mean of all voltage values for half a second (i.e., 500 ms) after the actual maximum. This can be easily done by finding the actual absolute maximum value for each column and calculating the maximum of the mean. After these calculations, all the voltage values of each column of the original matrix are divided by their corresponding “*maximum*” absolute values and finally multiplied by 10° . In this way, all values are transformed to angle of deviation while avoiding the influence of a peak voltage value.

In addition, all the angle of deviation values have to be subtracted from the existing offsets. These offsets can be approximated as the average values of the angles of deviation when the volunteer is looking at the near center target for 2.2 seconds. Note that the volunteer is aware of the duration of the objects that are shown on the screen during the tests. Thus, the offsets are the average of the values for each angle of deviation column for rows 30001 to 32200 (see tables 6.2 and 6.3). Listing 6.1 shows how “*maximum*” values were calculated and how the offset is subtracted by using *MATLAB*.

Listing 6.1: Transformation from voltage to angle of deviation.

```
function angleData = transformer(voltageData)
% angleData = transformer(voltageData)
% voltageData: "time(sec)", "RE Hor", "RE Ver", "LE Hor", "LE Ver"

% Actual maximum voltage values for the calibration interval
maximum = max(abs(voltageData(10001:15000,:)));
% Get the indexes of the actual maximum voltage values
maxIndex2 = find(abs(voltageData(10001:15000,2)) == maximum(2));
maxIndex3 = find(abs(voltageData(10001:15000,3)) == maximum(3));
maxIndex4 = find(abs(voltageData(10001:15000,4)) == maximum(4));
maxIndex5 = find(abs(voltageData(10001:15000,5)) == maximum(5));
% Let them be the maximum values the mean of all voltage values for half a
% second after the actual maximum for each column
maximum(2) = mean(voltageData(maxIndex2+10001:maxIndex2+10100,2));
maximum(3) = mean(voltageData(maxIndex3+10001:maxIndex3+10100,3));
```

```

maximum(4) = mean(voltageData(maxIndex4+10001:maxIndex4+10100,4));
maximum(5) = mean(voltageData(maxIndex5+10001:maxIndex5+10100,5));

angleData(:,1) = voltageData(:,1); % Time (seconds)
% All the voltage values of each column are divided by their corresponding
% "maximum" absolute values and finally multiplied by 10
angleData(:,2) = voltageData(:,2)/maximum(2)*10;
angleData(:,3) = voltageData(:,3)/maximum(3)*10;
angleData(:,4) = voltageData(:,4)/maximum(4)*10;
angleData(:,5) = voltageData(:,5)/maximum(5)*10;
% All the angle values of each column are subtracted each mean value
% corresponding to the fixation at the near center target (60 cm)
angleData(:,2) = angleData(:,2) - mean(angleData(30001:32200,2));
angleData(:,3) = angleData(:,3) - mean(angleData(30001:32200,3));
angleData(:,4) = angleData(:,4) - mean(angleData(30001:32200,4));
angleData(:,5) = angleData(:,5) - mean(angleData(30001:32200,5));

```

6.3.4 Interpolation and decimation of a set of samples

After carrying out the transformation explained in the previous subsection, we must deal with a problem that often occurs in practice. Suppose that $x_i[n]$ is the set of samples returned by the SSC system, taken with a sampling frequency f_{Si} . We intend to find another set of samples $x_o[n]$ corresponding to the sampling of the same analog signal with a sampling frequency f_{So} equal to the framerate of the *Tobii T120 Eye Tracker*.

We address the problem by studying the case where $f_{So} = (L/M) f_{Si}$, where L and M are integers, i.e., there is a rational relationship between the sampling frequencies. As $f_{Si} = 1000$ Hz and $f_{So} = 120$ Hz, $L/M = 3/25 < 1$. Thus, we talk about decimating by a rational factor.

It is easy to see that concatenating an interpolation and decimation by integer factors L and M respectively the desired goal is achieved. There might be a question of which should be done first: interpolation or decimation. To answer this question we must remember that decimation *destroys information*. For this reason it is necessary to first perform interpolation and then decimation.

To interpolate a signal it is padded with zeros and low-pass filtered with an impulse response shaped filter [39, p. 23]:

$$h[n] = \text{sinc} \frac{n}{L} \quad (6.11)$$

This filter is ideal and, therefore, unfeasible. As a result it is necessary to approximate an impulse response. A commonly used filter is the *linear interpolator*. It is known by this name because the new samples are calculated by drawing straight lines between the neighboring values. Listing 6.2 shows the calculations of the interpolated samples as implemented in *MATLAB R2010b*.

Listing 6.2: Linear interpolation.

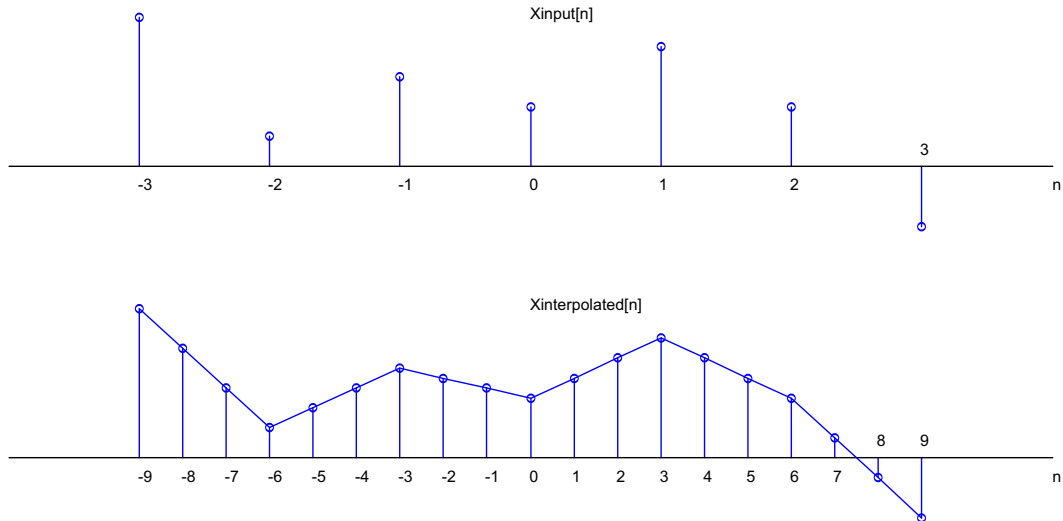
```

function x2 = interpolator(x1, L)
% x2 = interpolator(x1, L)
% x1: set of samples to interpolate.
% L: factor of interpolation.

x2 = x1(1,:); % The first sample of each column of x1 is taken
for i = 1:length(x1)-1
    % For each pair of samples, calculate the slope of the line they
    % describe
    m(1,:) = (x1(i+1,:) - x1(i,:))./L;
    % Interpolate with as many samples as L?1
    for j = 1:L-1
        x2 = [x2
              j.*m(1,:)+x1(i,:)];
    end
    x2 = [x2
          x1(i+1,:)]; % Add the next sample of x1
end

```

An example for $L = 3$ is illustrated in figure 6.12. The reader might ask what relationship exists between this interpolation and ideal low-pass filter. The answer is that linear interpolation is equivalent to a zero padding followed by a triangular impulse response filter, as shown in figure 6.13.

Figure 6.12: Relationship between samples in a linear interpolator for $L = 3$.

It is known from the literature that when the signal is sufficiently sampled (i.e., at least twice the Nyquist rate), the linear interpolator is a good approximation to the ideal low-pass interpolation.

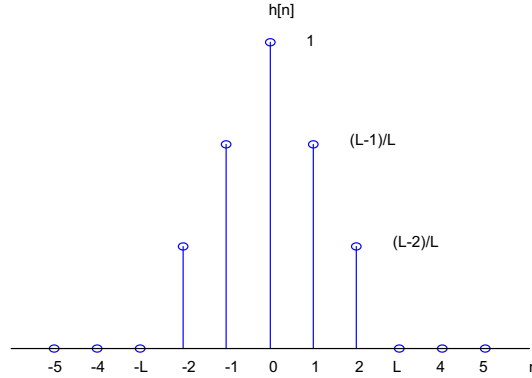


Figure 6.13: Impulse response of a linear interpolator for $L = 3$.

If the final sampling frequency is M times smaller than $f_{So} = L \cdot f_{Si}$, i.e.:

$$f_{So} = \frac{L \cdot f_{Si}}{M} = \frac{f_{Sinterpolated}}{M} \quad (6.12)$$

Decimation is relatively simple. It will be sufficient to take one out of every M samples of the interpolated input signal and copy them into the final output, i.e., the system with the following input/output relationship will compute:

$$x_o[n] = x_{interpolated}[Mn] \quad (6.13)$$

Listing 6.3 shows the calculations of the new sampling that were implemented in *MATLAB R2010b*. Figure 6.14 shows the effect of a sampler for $M = 8$.

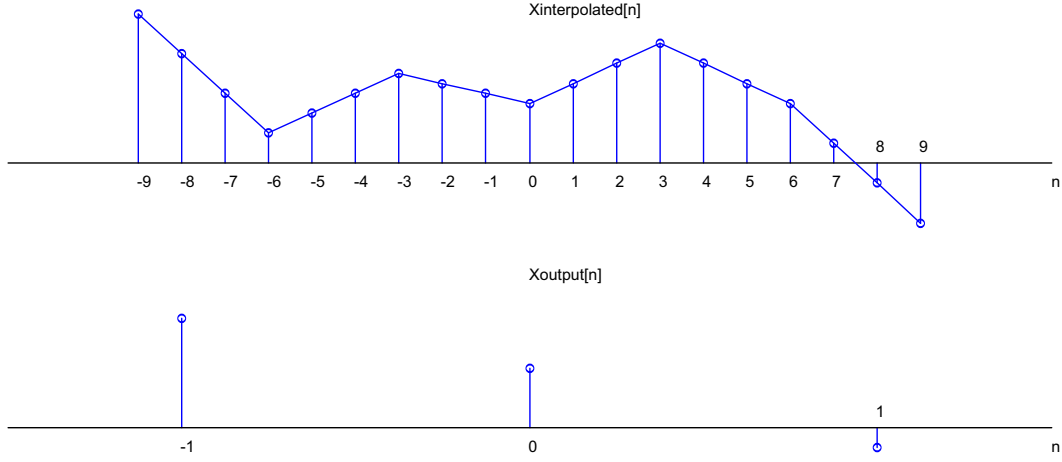
Listing 6.3: Sampling.

```
function x2 = sampler(x1, M)
% x2 = sampler(x1, M)
% x1: set of samples to decimate.
% M: factor of sampling.

x2 = [];

% Take one out of every M samples of the interpolated input signal and copy
% them into the final output.
for i = 1:M:length(x1)
    x2 = [x2
          x1(i,:)];
end
```

Thus, if we perform interpolation for $L = 3$ and then decimation for $M = 25$ to the set of samples $x_i[n]$ with a sampling frequency $f_{Si} = 1000$ Hz, the resulting set of samples $x_o[n]$ corresponds to sampling the same analog signal with a sampling frequency $f_{So} = 120$ Hz.

Figure 6.14: Effect of a sampler for $M = 8$.

6.3.5 Correlation between the *Tobii T120 Eye Tracker* and SSC performances

To figure out how the *Tobii T120 Eye Tracker* performs when compared with the gold standard, a correlation between tracking test data sets and their corresponding SSC sampled and transformed outputs in time was calculated.

After transforming the SSC data set in such a way that its values describe angles of deviation, it was necessary to look for synchronization points in both data sets. The synchronization must be done **before** the decimation, otherwise there will be a phase error. This is a hard task since the validation procedure explained in section 6.3.2 was manual and not that precise.

Interpolations for $L = 3$ to the set of samples from the SSC and $L = 25$ to the set of samples from the *Tobii T120 Eye Tracker* were performed. Thus, sets of samples with a sampling frequency $f_{so} = 3000$ Hz were obtained.

Let x and y be two digital signals with the same sampling frequency and N samples each. The correlation in the changes in these two signals is calculated as:

$$r_{xy} = \frac{\sum_{i=1}^N (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \quad (6.14)$$

Listing 6.4 shows how the correlation in the changes in these signals with a sampling frequency $f_S = 3000$ Hz and their best corresponding indexes in the sets

of samples were determined. In this way, it is possible to see which is the best index to use as a start offset to get the correlation value closest to 1.0, so that both sets can be plotted in the same figure after sampling for $M = 25$. We consider the best start offset as the average of the individual corresponding starting offsets for the left and right eye, with both horizontal and vertical deviations. This script also calculates the ratio of valid samples returned by the *Tobii T120 Eye Tracker*. Listing 6.5 implements equation 6.14 on page 74 for the set of samples of the SSC synchronized with samples from the *Tobii T120 Eye Tracker* as explained before. Listing 6.6 calculates the average of the set of samples discarding the invalid samples returned by the *Tobii T120 Eye Tracker*.

Listing 6.4: Correlation in the changes in the signals.

```
function [r, index, sscRatio, tobiiRatio, best, bestIndex] = ←
    bestCorrelation(ssc,tobii,start)
% [r, index, sscRatio, tobiiRatio, best, bestIndex] = bestCorrelation(ssc,←
    tobii,start)

[rows, columns] = size(ssc);
r0 = -1*ones(1,columns-1);
index0 = ones(1,columns-1);
sscRatio0 = zeros(1,columns-1);
tobiiRatio0 = zeros(1,columns-1);
ssc = transformer(ssc);
ssc = ssc(start:start+9*2200+1000,:);
tobii = interpolator(tobii,25);

for j = 2:columns
    sscJ = interpolator(ssc(:,j),3);
    for i = 1:length(sscJ)-length(tobii)+1
        [r, sscRatio, tobiiRatio] = correlator(sscJ(i:i+length(tobii)-1),←
            tobii(:,j));
        if r > r0(j-1)
            r0(j-1) = r;
            index0(j-1) = i;
            sscRatio0(j-1) = sscRatio;
            tobiiRatio0(j-1) = tobiiRatio;
        end
    end
end

r = r0;
index = index0;
sscRatio = sscRatio0;
tobiiRatio = tobiiRatio0;
best = [start round(sum(r.*index)/sum(r))]; bestIndex = sum(best);
```

Listing 6.5: Calculation of the correlation in the changes in two signals.

```

function [r, sscRatio, tobiiRatio] = correlator(ssc,tobii)
% [r, sscRatio, tobiiRatio] = correlator(ssc,tobii)
% r: sample correlation coefficient
% sscRatio: percentage of valid samples in the ssc data set
% tobiiRatio: percentage of valid samples in the Tobii data set

if length(ssc) ~= length(tobii)
    disp('ssc and tobii have different size')
    r = 0;
    sscRatio = 0;
    tobiiRatio = 0;
    return;
end

sumNumerator = 0;
sumXroot = 0;
sumYroot = 0;
[sscmean, sscRatio] = calcMean(ssc);
[tobiimean, tobiiRatio] = calcMean(tobii);

for i = 1:length(ssc)
    if isnan(ssc(i)) == 0 && isnan(tobii(i)) == 0
        sumNumerator = sumNumerator + (ssc(i)-sscmean)*(tobii(i)-tobiimean);
        sumXroot = sumXroot + (ssc(i)-sscmean)^2;
        sumYroot = sumYroot + (tobii(i)-tobiimean)^2;
    end
end

r = sumNumerator / (sqrt(sumXroot)*sqrt(sumYroot));

```

Listing 6.6: Calculation of the average discarding not valid samples.

```

function [m, validityRatio] = calcMean(x)
% [m, validityRatio] = calcMean(x)
% m: mean of x discarding not valid elements
% validityRatio: percentage of valid samples returned by the Tobii Eye Tracker

calcSum = 0;
k = 0;

for i = 1:length(x)
    if isnan(x(i)) == 0
        calcSum = calcSum + x(i);
        k = k + 1;
    end
end

m = calcSum ./ k;
validityRatio = k / length(x);

```

Correlation between both measurements for every volunteer, test, and variable are summarized in table 6.4 and illustrated in figure 6.15. All these correlation results are over 0.9799 (the value in red), hence the reader may consider that there is a very good correlation between both measurement techniques.

Table 6.4: Correlation between both measurement techniques.

Subject #	Test #	$r_{\phi L}$	$r_{\phi R}$	$r_{\theta L}$	$r_{\theta R}$
Binocular #1	1	0.9952	0.9937	0.9977	0.9930
	2	0.9956	0.9934	0.9975	0.9934
	3	0.9951	0.9918	0.9957	0.9913
Binocular #2	1	0.9945	0.9946	0.9971	0.9949
	2	0.9951	0.9916	0.9925	0.9933
	3	0.9967	0.9936	0.9960	0.9966
Monocular #1	1	0.9962	0.9927	0.9980	0.9931
	2	0.9966	0.9931	0.9982	0.9940
	3	0.9962	0.9930	0.9981	0.9944
Monocular #2	1	0.9959	0.9799	0.9882	0.9950
	2	0.9974	0.9818	0.9890	0.9957
	3	0.9968	0.9895	0.9900	0.9947

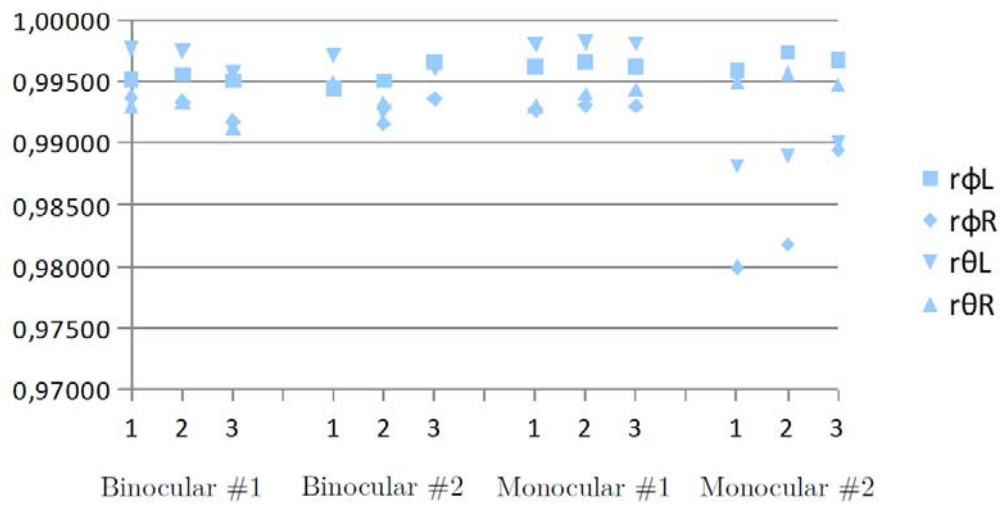
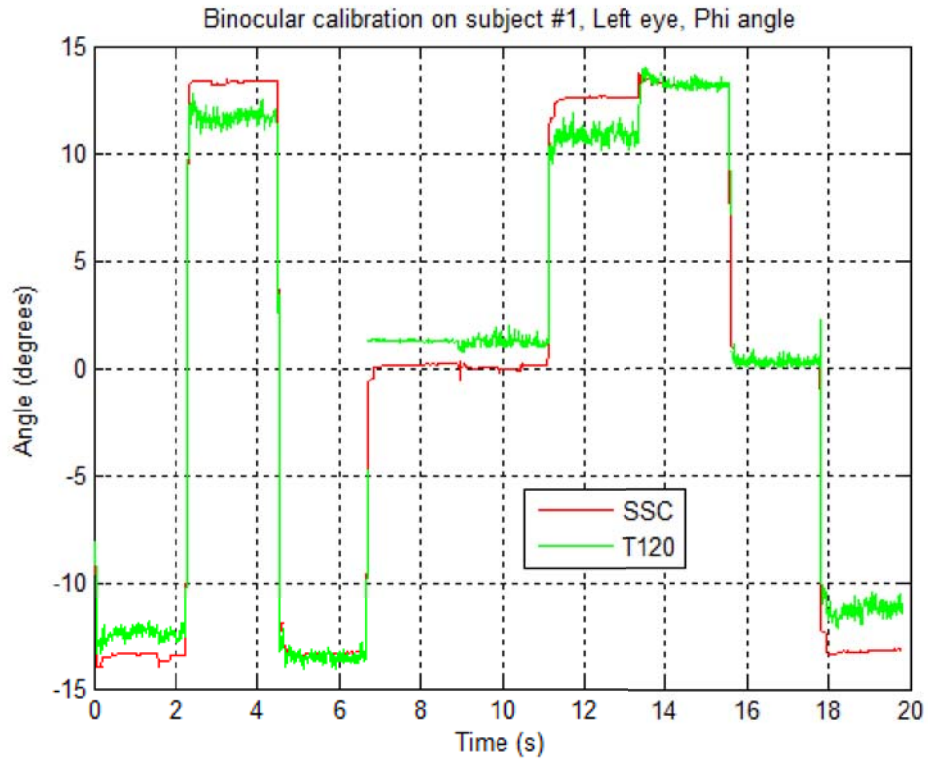
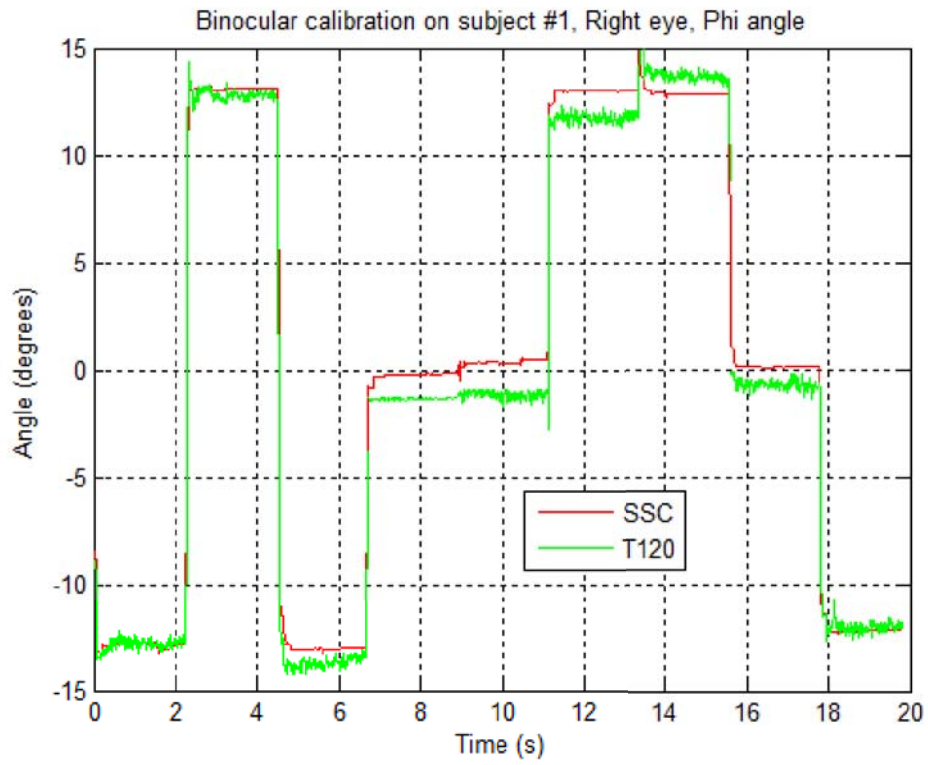
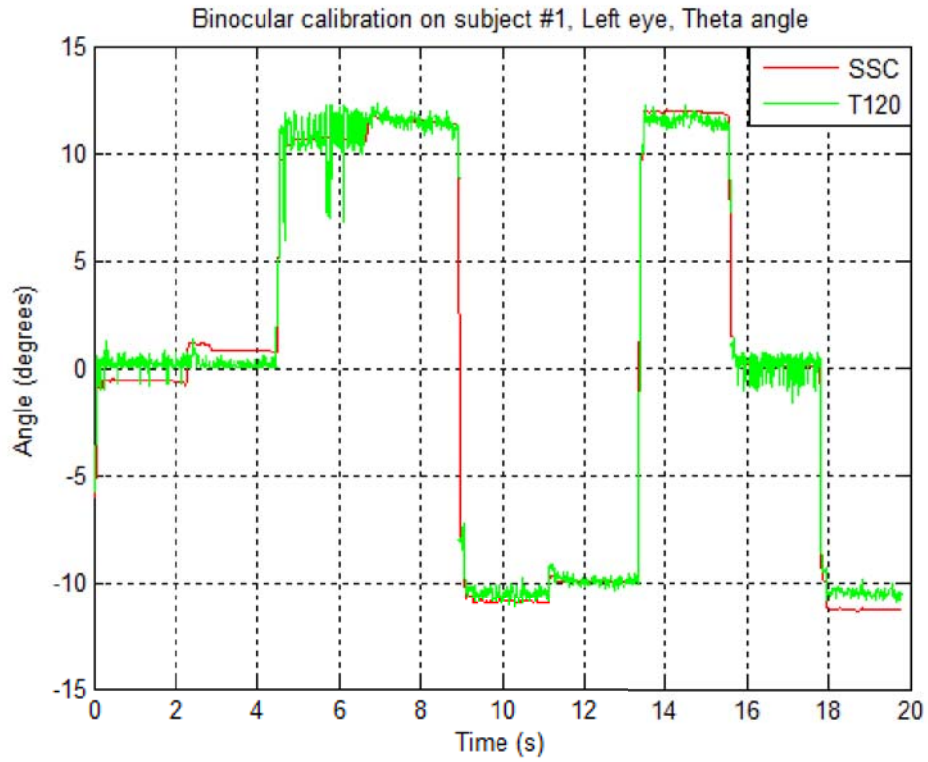


Figure 6.15: Correlation between both measurement techniques.

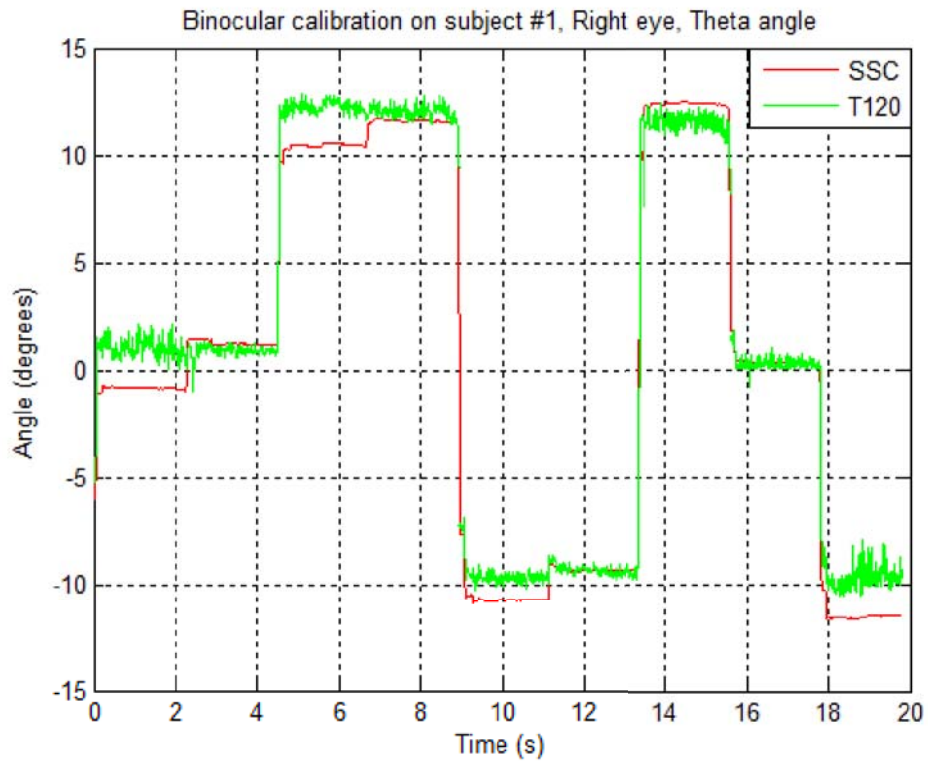
6.3.6 Graphical comparison between the *Tobii T120 Eye Tracker* and SSC performances

Figures from 6.16 to 6.25 show examples of graphical comparisons between the SSC and the eye tracker performances for every volunteer, calibration, and output variable. A comparison between calculations of *vergence* angle is included.

(a) ϕ_L tracking.(b) ϕ_R tracking.Figure 6.16: Binocular calibration and ϕ tracking on subject #1.

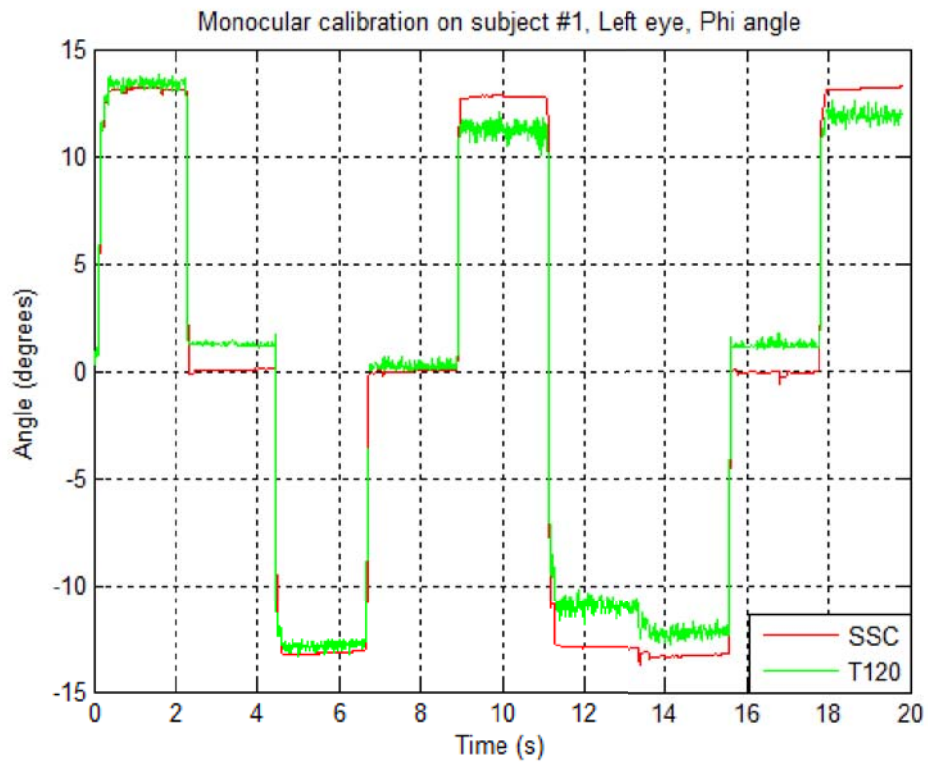
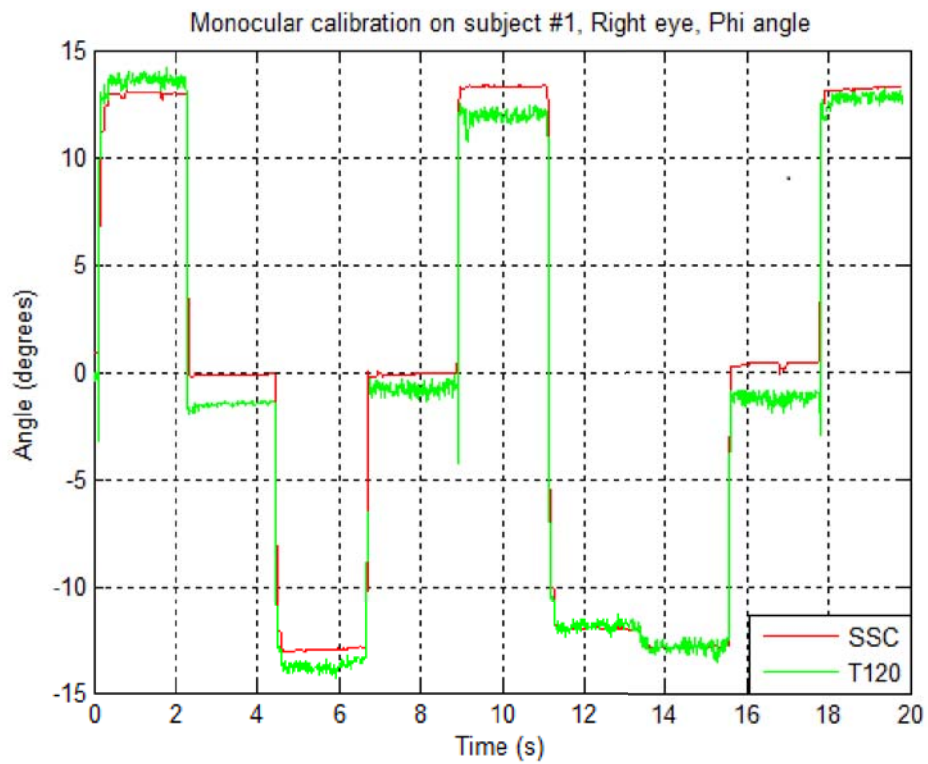


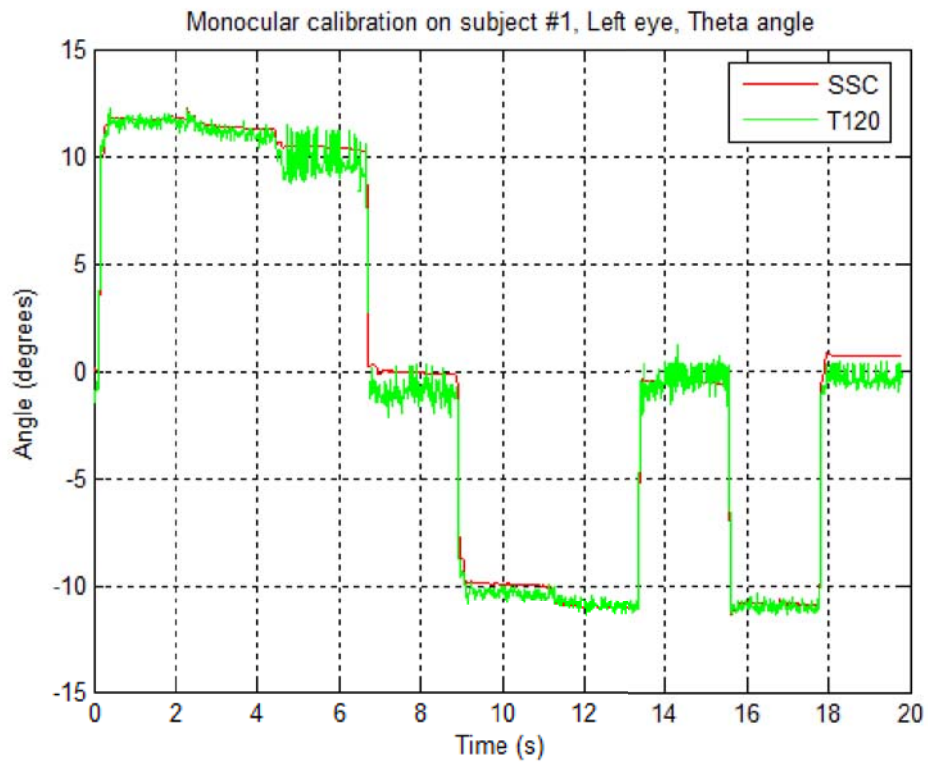
(a) θ_L tracking.



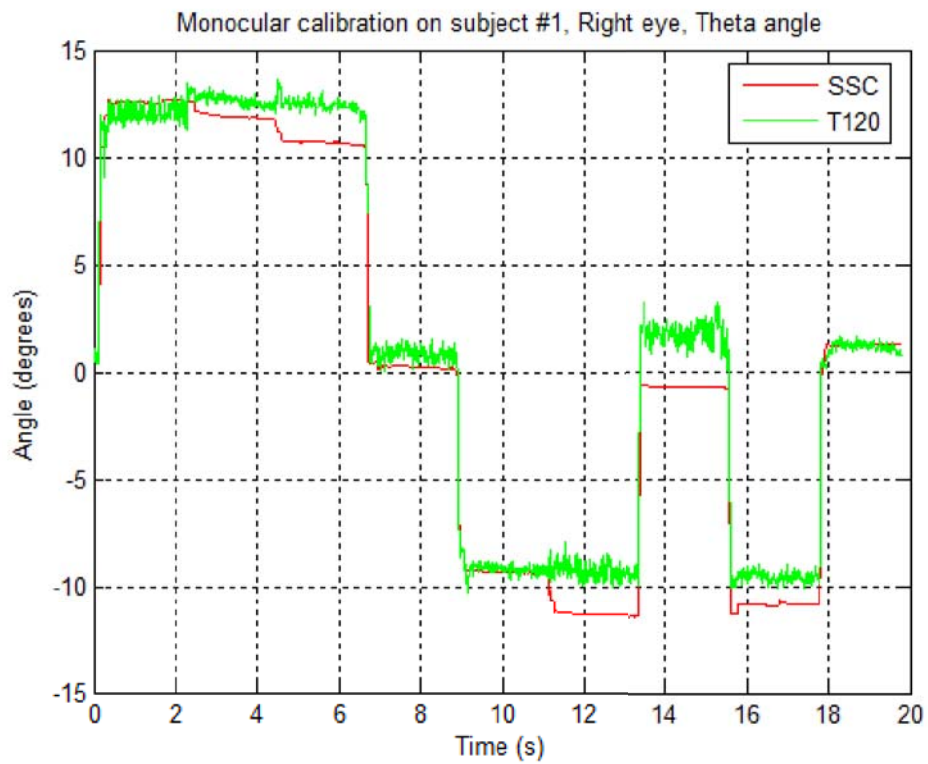
(b) θ_R tracking.

Figure 6.17: Binocular calibration and θ tracking on subject #1.

(a) ϕ_L tracking.(b) ϕ_R tracking.Figure 6.18: Monocular calibration and ϕ tracking on subject #1.

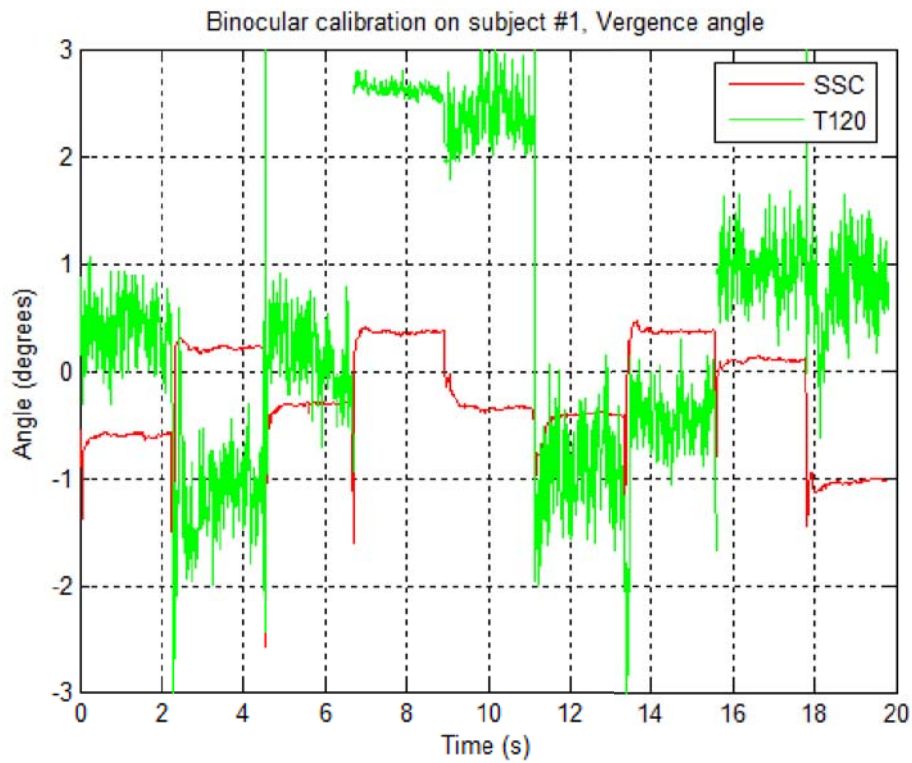


(a) θ_L tracking.

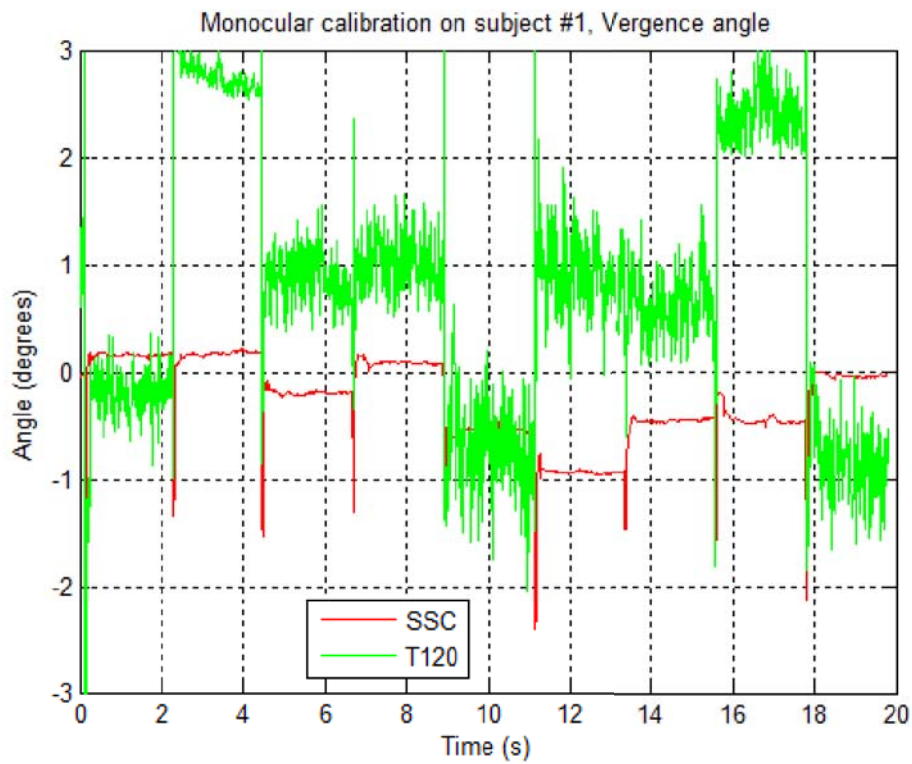


(b) θ_R tracking.

Figure 6.19: Monocular calibration and θ tracking on subject #1.

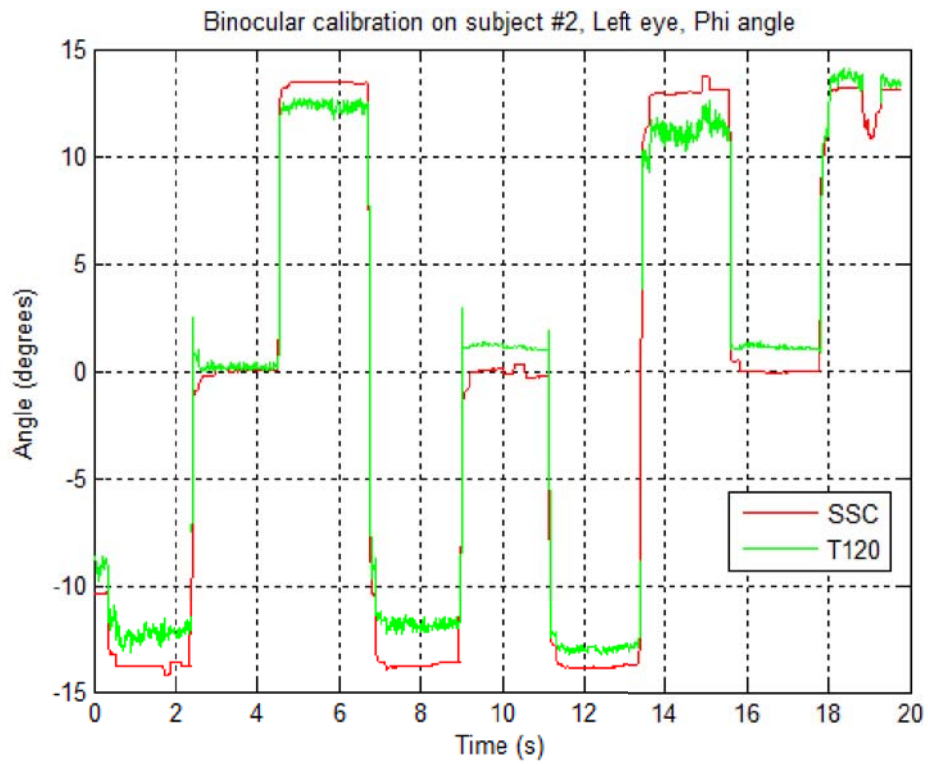


(a) Binocular calibration.

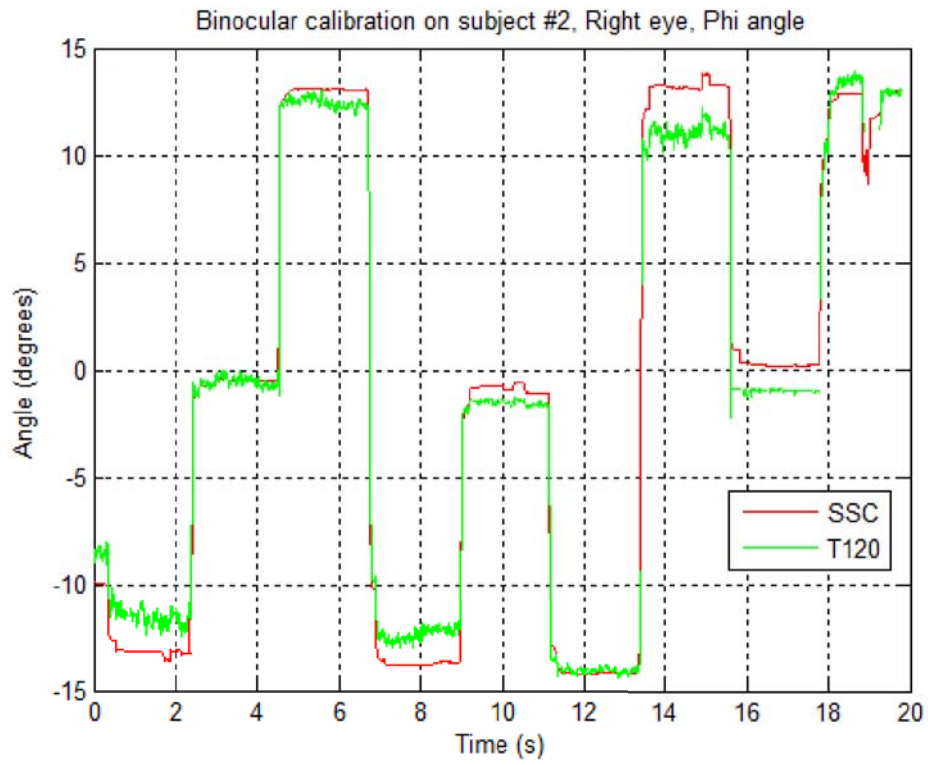


(b) Monocular calibration.

Figure 6.20: *Vergence* angle for subject #1.

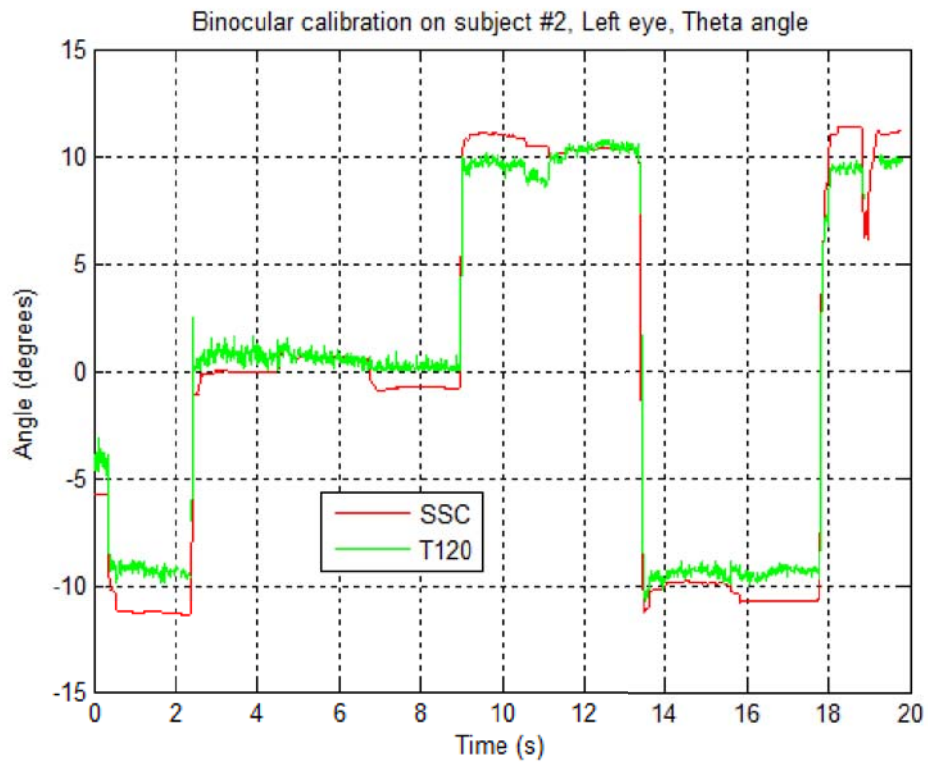
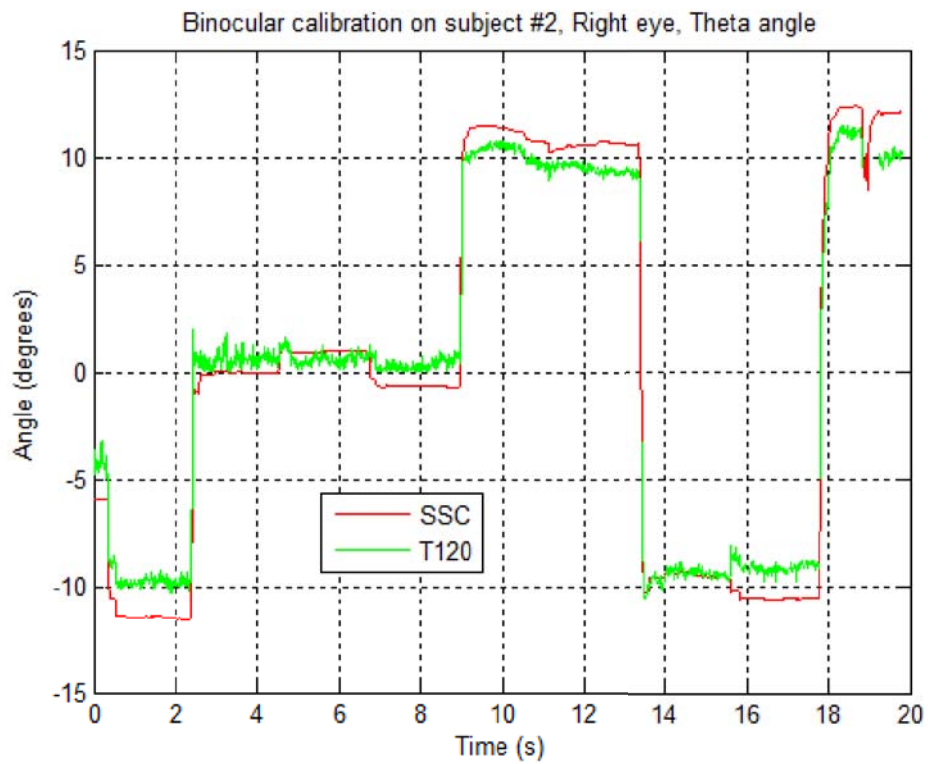


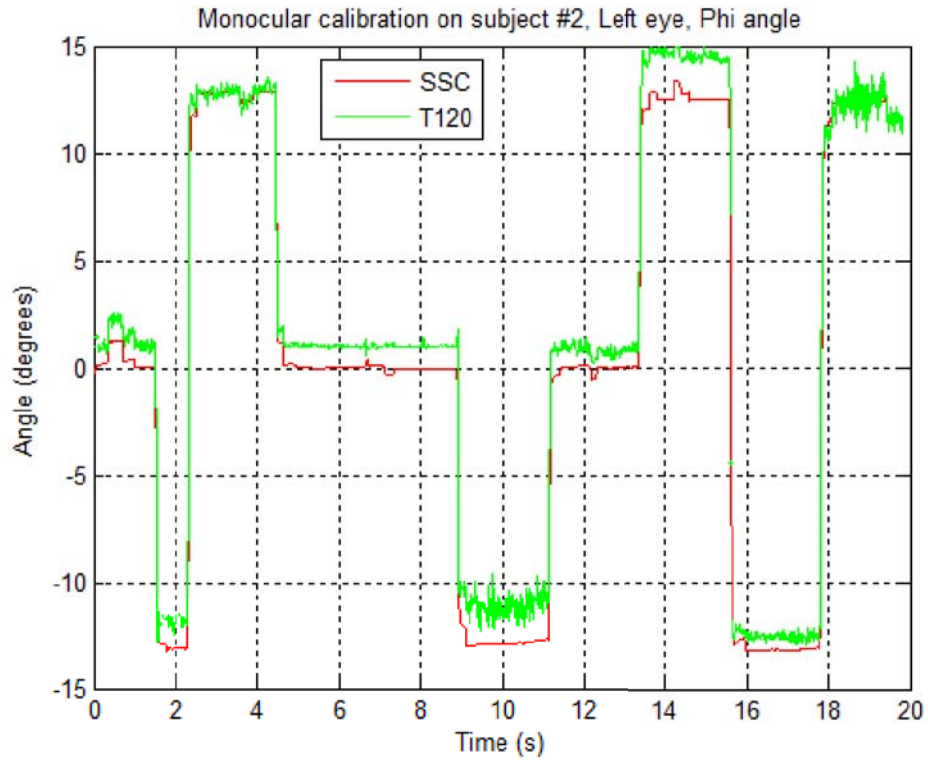
(a) ϕ_L tracking.



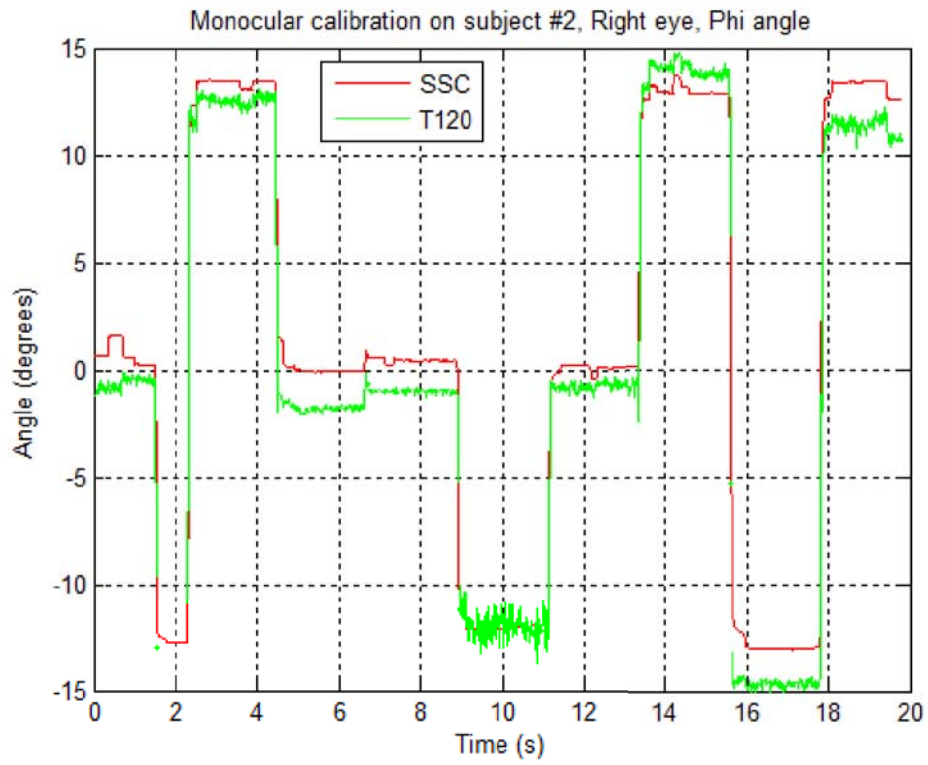
(b) ϕ_R tracking.

Figure 6.21: Binocular calibration and ϕ tracking on subject #2.

(a) θ_L tracking.(b) θ_R tracking.Figure 6.22: Binocular calibration and θ tracking on subject #2.

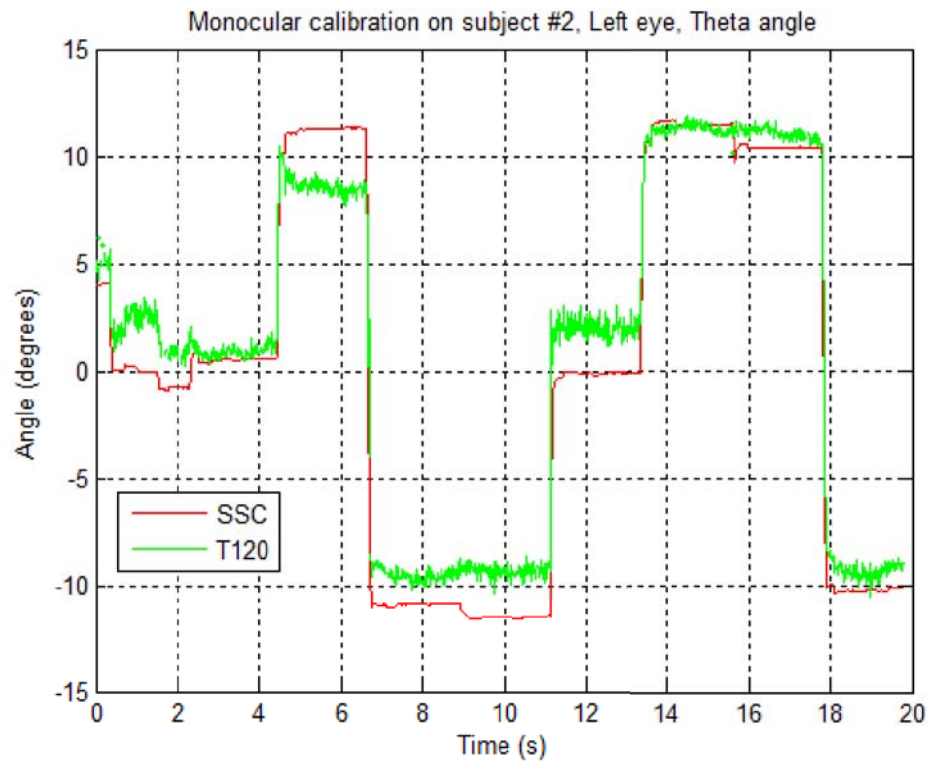
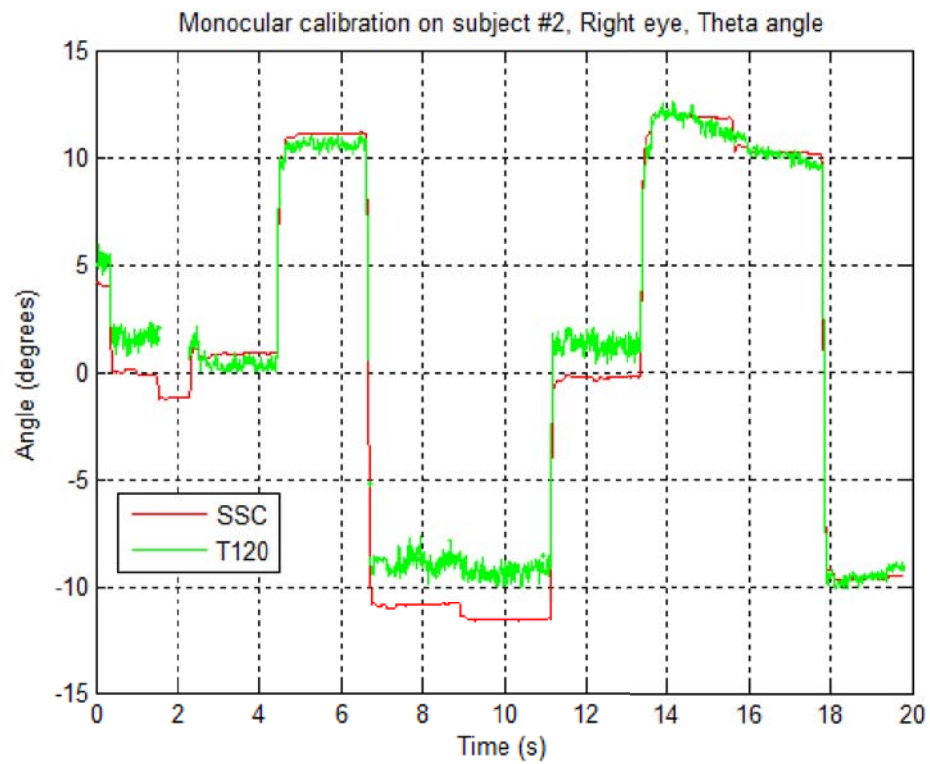


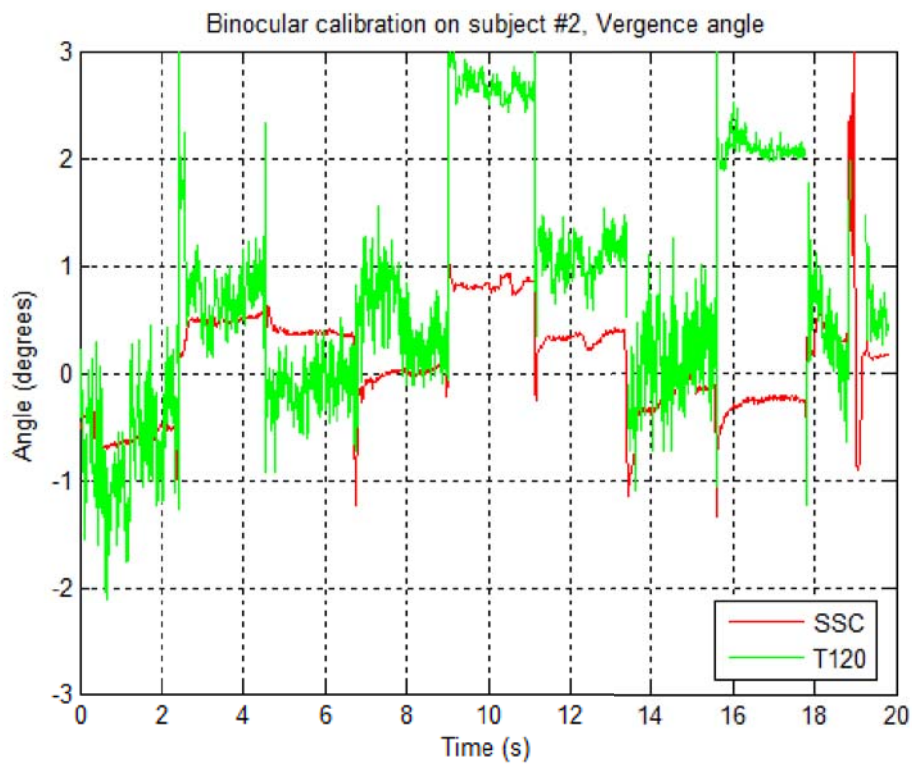
(a) ϕ_L tracking.



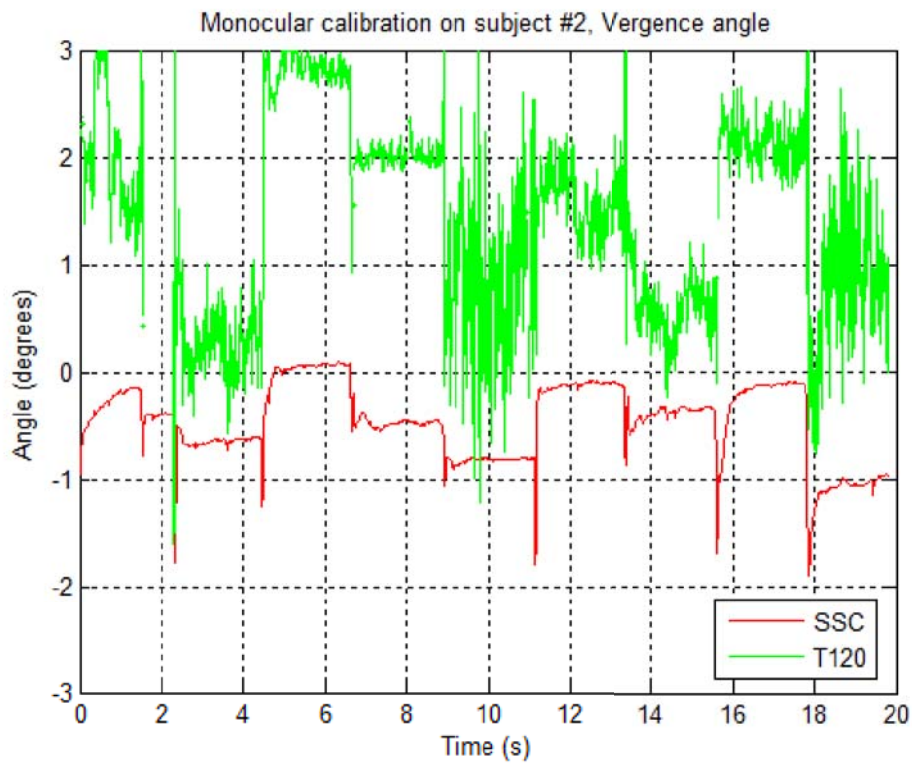
(b) ϕ_R tracking.

Figure 6.23: Monocular calibration and ϕ tracking on subject #2.

(a) θ_L tracking.(b) θ_R tracking.Figure 6.24: Monocular calibration and θ tracking on subject #2.



(a) Binocular calibration.



(b) Monocular calibration.

Figure 6.25: *Vergence* angle for subject #2.

6.3.7 Influence of the changes in the pupil size

Looking at figures in section 6.3.6, the reader may see that there is a very good phase synchronization between the SSC and the *Tobii T120 Eye Tracker* performances. However, the reader may also see that there is no linear relationship between the amplitudes of the two measurements for an object shown during a given test – sometimes the SSC amplitude is higher than the eye tracker’s one, and other times the opposite happens. Moreover, the output from the eye tracker is much noisier, but this is expected since the SSC filters the analog signal in order to smooth it. This filtering by the SSC could explain why measurements of *vergence* angle by the *Tobii T120 Eye Tracker* are quite noisy and do not give stable information.

It was conjectured in conjunction with experts and clinicians in the area at *Karolinska Institutet* that the changes in the pupil size during saccades from an object to another object could affect the measurements returned by the eye tracker. Figure 6.26 shows a plot in which every output sample of the eye tracker for a given test is divided by its corresponding pupil diameter normalized to the mean.

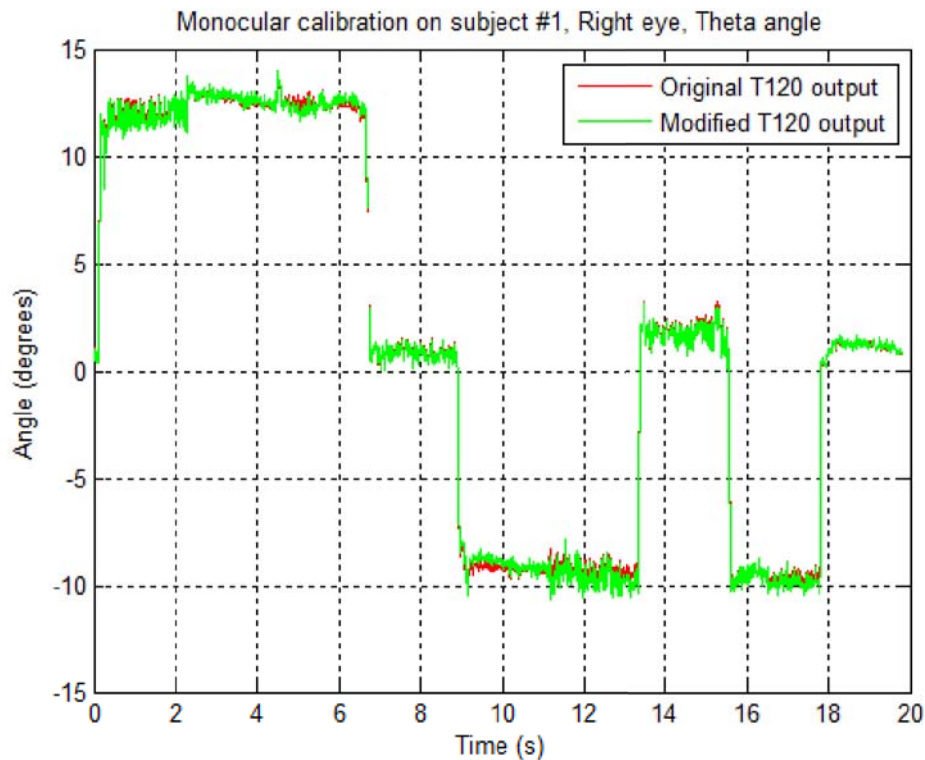


Figure 6.26: Comparison between the original and the *modified* T120 outputs.

6.3.8 Discussion

Neither the comparison illustrated in the previous section nor the corresponding ones for tests (these plots are not included in this report) reveal any significant difference between the original and the *modified* outputs. Table 6.5 shows mean differences between both measurements for every volunteer, calibration, and output variable. All the values are under 1.5 degrees (the value in red) and most of them are less than or equal to 1 degree, which can be considered a reasonable difference since this is of the same order of magnitude as a microsaccade [18]. Microsaccades occur during prolonged visual fixation in both **healthy** subjects and those who present ophthalmic disorders [18]. Thus, mean differences between both measurements that are less than or equal to 1 degree may be considered to be small and negligible independent of the subject.

Table 6.5: Mean difference between both measurements expressed in degrees.

Subject #	Test #	$\overline{\Delta\phi_L}$	$\overline{\Delta\phi_R}$	$\overline{\Delta\theta_L}$	$\overline{\Delta\theta_R}$
Binocular #1	1	1.1	0.8	0.5	1.0
	2	1.0	0.8	0.5	1.0
	3	1.1	0.9	0.7	1.1
Binocular #2	1	1.2	0.9	1.0	1.1
	2	1.2	1.0	0.9	1.1
	3	1.2	0.9	1.0	1.0
Monocular #1	1	1.0	0.9	0.5	1.1
	2	1.0	0.8	0.5	1.1
	3	1.1	0.9	0.6	1.0
Monocular #2	1	0.9	1.1	1.5	1.1
	2	1.0	1.1	1.5	1.0
	3	1.0	1.3	1.4	1.0

This study has been made within a framework in which it is assumed that the *Tobii T120 Eye Tracker* was already tested and evaluated regarding accuracy and precision as described in [2]. As a result of this assumption and the results of my own comparison, it seems that the differences between the amplitudes of both measurements are due to the fact that the primary positions of the SSC coils were not perfectly in parallel with the $x - y$ plane of the eye tracker. As the SSC and the *Tobii T120 Eye Tracker* were not interacting through any communication medium, it was impossible to compute the relation between these two planes in order to obtain a correlated effect which can be corrected for.

Chapter 7

Conclusions and Future work

In this chapter a discussion of the results of chapter 6 is made and following this some conclusions are presented in section 7.1. Finally, some ideas that were considered but not implemented in this thesis project and some new ideas that have come up in the course of this project are suggested as future work in section 7.2.

7.1 Discussion of the results

The tests carried out in the three studies carried out during this thesis project show that the prototype is able to produce significant results once the participant understands the procedure and does his/her best during a test. The new eye tracking test may become successful, but requires further development and improvements.

The application worked successfully during the tests. However, the application also had a few problems. It does not respond correctly after minimizing and restoring the window because there are errors regarding paint events of the graphics of the *Gaze Plotter*, which might be time consuming since the application needs to be restarted and calibration and/or tracking files loaded from the disk. This has to be corrected and carefully tested.

Although definite conclusions from the tests **cannot** be made, it seems that the prototype met the requirement specification in all the aspects.

Results from the current tests of the application suggest that the methodology may be successful, but will require further development and improve. This methodology, could lower variability, subjectivity, and time. The application is faster than testing based on manual observation or subjective methods *if the participant is attentive*. In addition, the tester does not need to be an educated and well trained ophthalmologist in order to perform the eye tracking test. All these benefits could result in a change of how vision testing is performed. Perhaps Eye Tracking Technology could be used at medical centers as a standard method of vision testing.

The application is able to save and load eye tracker calibrations for each participant. Although for all the tests carried out in the three studies calibrations were run before a set of tracking tests, it is possible to start a new test without needing to calibrate the eye tracker if a participant gets tired and needs to rest for a while. In such a case the application simply loads an earlier saved calibration file. However, there is still the need to calibrate the system for each participant.

Deviations larger than 1 degree occurred in some subjects who are apparently **healthy** and many of them involved crossed fixations. For actual patients, this threshold of 1 degree could discard crossed fixations by allowing the ophthalmologist to have the final say. However, this would once again require the participation of a trained ophthalmologist.

The results obtained in the study I were contrasted with a series of tests conducted at *Bernadottelaboratoriet* at *S:t Eriks Ögonsjukhus* using a video-oculography system with monocular calibration. This series of tests suggested that the prototype application offers a certain degree of reliability. The results obtained in the study II show that results are independent of different combinations of procedures conducted at different times. Regarding repeatability, the tests produce similar results under these different combinations of procedures as shown by the R^2 -values in section 6.2 starting on page 56. The results obtained in the study III show that the prototype application can be validated by comparison with the SSC gold standard, since the phase synchronization is very good as illustrated in figures 6.16 to 6.25 starting on page 78.

Evaluations of the prototype application shows that vision testing with eye tracking technology has great potential for success with real patients, but requires that it be further developed, improved, and evaluated.

7.2 Future work

In the beginning of this thesis project, it was agreed with experts and clinicians in the area at *Karolinska Institutet* to consider stimulus 18.5 degrees from central position left, right, up-left, up-center, up-right, down-left, down-center, and down-right. Unfortunately, using a *Tobii T60/120 Eye Tracker* reduced the actual range of stimuli to 13.2 and 10.9 degrees for horizontal and vertical axes respectively. These limitations were required in order to ensure good interactivity between the subject and the system.

The eye tracker should be placed so that the gaze angle will not exceed approximate 35.0 degrees to any point on the screen [37, p. 24]. This gaze angle is the angle between the gaze vector and the vector described by the average subject's eye position and the origin of the UCS. This front filter surface vector and the gaze vector coming from the ideal average UCS 3D eye position (0,0,600) and pointing to the normalized 2D coordinates either (0,0) or (1,0) on the tracking area describe an angle of approximate 31.4 degrees; according to the corner 3D points returned

by the method *GetXConfiguration()* when using the *Tobii T60/T120 Eye Tracker*, according to equation 4.9 and trigonometry. In this way, good trackability was ensured according to [37, p. 24].

Thus, the limitation ensures good interactivity between the subject and the system. That is why an eye tracker with a larger screen and, consequently, larger tracking area should be used for testing *Tobii Alignment*. The *Tobii T60 XL Eye Tracker* has a 24-inch TFT monitor and a maximum gaze angle of approximate 42.0 degrees [38].

Another method could be, for example, using the *Tobii T60/T120 Eye Tracker* with a physical board overlay laid on the screen in such a way its trackability is fully exploited. All the stimuli would be printed on that board emulating normalized 2D coordinates from (0,0) to (1,1). Then, when either the calibration or tracking procedure were performed, the specialist would tell to the subject which stimulus he/she should look at.

In this thesis project, the first and last 600 milliseconds long intervals were considered to be noise samples. The purpose was to ignore the samples during an interval twice as long as a maximal microsaccade lasts [18], before and after this interval a second period were used for signal samples. A criteria not implemented in the prototype application was to consider the farthest third of samples from the mean gaze point for each gaze point set and eye as noise. This criteria could be implemented later.

The evaluation presented in this work does not consider situations where the test participant subjects present ophthalmic disorders. These subjects were explicitly excluded as it was agreed with experts and clinicians in the area at *Karolinska Institutet*, that this thesis project would aim to describe a novel procedure and examine its validity regarding repeatability, objectivity, comprehension, relevance, and independence on **healthy** subjects. This prototype application was compared to the gold standard in the study III, but in the future a study should consider this comparison for patients with ophthalmic disorders.

Two binocular calibrations without any IR transparent lens could be performed and we could merge the model of the left eye from one calibration run with the model of the right eye from the calibration run, to see if occluding the eyes affects the results of the tracking tests.

To see just what filter the people who built the SSC use (see section 6.3.7), the Fast Fourier Transform (FFT) of each output from the SSC and the *Tobii T120 Eye Tracker* may be computed to decompose both sequences of values into components of different frequencies [12]. Then, the cross-correlation can be obtained as the inverse transform of the product of the individual transforms [12] to figure out how the *Tobii T120 Eye Tracker* performs when compared with the gold standard by taking into account the filtering by the SSC.

Finally, it would be very interesting to use a projector as the tracking area. Again, it is necessary to know the projector's area geometry, what can be determined by using a *motion sensing input device*. Today there are several such devices on the market. They are based around a webcam-style add-on peripheral with an infrared projector and camera and a special microchip to track the movement of objects and individuals in three dimension [21]. Using this kind of devices three of the four corner points of the projector and the eye tracker's front filter surface can be tracked such that the relative position of the tracking plane can be configured in relation to the origin of the eye tracker's coordinate system and uploaded to the eye tracker. For example, the geometric quality of depth data obtained by the *Microsoft's Kinect* sensor decreases with increasing distance to the sensor, and ranges from a few millimeters up to about 4 cm at the maximum range of the sensor, which is 5 meters [1]. In this way, the eye tracker could be placed so that the gaze angle does not exceed the limitations required in order to ensure good interactivity between the subject and the system, but the subject would fixate to further targets, allowing the gaze pattern to be more stable.

Moreover, this system could be used to record gaze patterns from several subjects sequentially from one subject to the next such that all samples would be recorded at the same framerate, but at different phases. A non-medical application could be studying effectiveness of a commercial in some sample audience, such as whether the audience was attentive to the commercial and/or which elements they were looking at.

Bibliography

- [1] K. Khoshelham, *Accuracy Analysis of Kinect Depth Data*. ITC Faculty of Geo-information Science and Earth Observation, University of Twente, 2011. Available from: http://www.isprs.org/proceedings/XXXVIII/5-W12/Papers/ls2011_submission_40.pdf.
- [2] *Accuracy and Precision Test Method for Remote Eye Trackers*. Tobii Technology, 2011. Available from: <http://www.tobii.com/en/eye-tracking-research/global/about-tobii/eye-tracking/test-method/>.
- [3] J. L. Weber and Joanne Wood, *Amblyopia: Prevalence, Natural History, Functional Effects and Treatment*. Scholar Search. Clinical and Experimental Optometry 88, 2005.
- [4] Gunnar Lennerstrand and Jan Ygge, *Advances in Strabismus Research: Basic and Clinical Aspects*. Portland Press, 2000.
- [5] Gunnar Lennerstrand and Gustaf Öqvist Seimyr, *Advances in Pediatric Ophthalmology Research*. The Sigvard & Marianne Bernadotte Research Foundation for Children Eye Care, 2010.
- [6] *The Basics of Eye Tracking*. Tobii Technology, 2011. Available from: <http://www.tobii.com/eye-tracking-integration/global/eye-tracking/the-basics-of-eye-tracking/>.
- [7] Gunter K. von Noorden, *Binocular Vision and Ocular Motility, Theory and Management of Strabismus, 4th Edition*. The C. V. Mosby Company, 1990.
- [8] Cecie Starr, *Biology: Concepts and Applications*. Thomson Brooks/Cole. ISBN 053446226X.
- [9] Andrew W. Troelsen, *C# and the .NET Platform, 2nd Edition*. Berkeley, Calif.: Apress, 2003.
- [10] Fiona Rowe, *Clinical Orthoptics*. Blackwell Science Ltd, 1997.
- [11] David A. Quillen, *Common Causes of Vision Loss in Elderly Patients*. American Academy of Family Physicians, 1999. Available from: <http://www.aafp.org/afp/990700ap/99.html>.

- [12] Samir S. Soliman and Mandyam D. Srinath, *Continuous and Discrete Signals and Systems*. Prentice Hall. ISBN: 0-13-569112-5, 1998.
- [13] Magnus Olsson, *Correction Software for Measuring of Eye Movements*. Master of Science Thesis. (UPTEC IT 06 045). *Uppsala Universitet*. Uppsala, Sweden, 2007.
- [14] Wikipedia, the free encyclopedia, *A diagram of the human eye*. Available from: <http://en.wikipedia.org/wiki/File:Eye-diagram.svg>.
- [15] B. Cassin and S. Solomon, *Dictionary of Eye Terminology*. Gainesville, Florida: Triad Publishing Company, 1990.
- [16] Howard Anton, *Elementary Linear Algebra, 10th Edition*. Wiley, 2000.
- [17] Antje Nuthmann and Reinhold Kliegl, *An Examination of Binocular Reading Fixations Based on Sentence Corpus Data*. Journal of Vision, volume 9, 2009. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/19757909>.
- [18] K. Holmqvist, M. Nyström, R. Andersson, J. Jarodzka, and J. van de Weijer, *Eye tracking data and dependent variables, Under revision*. 2011.
- [19] Oleg Špakov, *iComponent - Device-Independent Platform for Analyzing Eye Movement Data and Developing Eye-Based Applications*. Academic Dissertation. *University of Tampere*, 2008.
- [20] María de Diego, Alberto González, and Gema Piñero, *Introducción a las Señales Aleatorias*. Editorial Universidad Politécnica de Valencia, 2003.
- [21] Wikipedia, the free encyclopedia, *Kinect*. Available from: <http://en.wikipedia.org/wiki/Kinect>.
- [22] D. A. Robinson, *A Method of Measuring Eye Movement Using a Scleral Search Coil in a Magnetic Field*. IEEE Trans. Biomed. Eng., vol. BME-10, 1963.
- [23] R. John Leigh and David S. Zee, *The Neurology of Eye Movements, 3rd Edition*. Contemporary Neurology Series, 1999.
- [24] Alexander Hultman, *A Performance Evaluation of a Remote and Binocular 2D Eye Tracking System, Tobii ET-1750*. Master of Science Thesis. *Karolinska Institutet*. Stockholm, Sweden, 2005.
- [25] H. Collewyn, F. van der Mark, and T. C. Jansen, *Precise Recording of Human Eye Movements*. Vision Research 14, 1975.
- [26] Masayuki Iwasaki and Hajime Inomara, *Relation Between Superficial Capillaries and Foveal Structures in the Human Retina – Investigative Ophthalmology & Visual Science* (journal) - volume 27, 1986. Available from: <http://www.iovs.org/content/27/12/1698.full.pdf>.

- [27] Jon Aster, Vinay Kumar, Abul K. Abbas, and Nelson Fausto, *Robbins & Cotran Pathologic Basis of Disease, 8th Edition*. Philadelphia: Saunders. ISBN 1-4160-3121-9, (p. 113), 2009.
- [28] Michael T. Heath, *Scientific Computing – An Introductory Survey, 2nd Edition*. McGraw-Hill, 2002.
- [29] *Sensory Reception: Human Vision: Structure and function of the Human Eye* - vol. 27. Encyclopaedia Britannica, 1987.
- [30] Helga Kolb, *Simple Anatomy of the Retina*. Webvision. University of Utah, 2011. Available from: <http://webvision.med.utah.edu/book/part-i-foundations/simple-anatomy-of-the-retina/>.
- [31] S. Awaya, S. Miyake, E. Koizumi, and T. Hirai, *The Sensitive Period of Visual System in Humans*. In: M. Lenk-Schafer, C. Calcutt, M. Doyle and S. Moore (eds) *Transactions of the Sixth International Orthoptic Congress*, British Orthoptic Society, London, (p. 44), 1987.
- [32] Andrew Webb, *Statistical Pattern Recognition, 2nd Edition*. Wiley, 2005.
- [33] Joseph Lang, *Strabismus*. SLACK Incorporated, 1984.
- [34] *Tobii SDK 3.0 – Developers Guide*. Tobii Technology, 2011. Available from: <http://www.tobii.com/eye-tracking-research/global/products/software/tobii-software-development-kit/tobii-software-development-kit/>.
- [35] *Tobii SDK User Manual*. Tobii Technology, 2011. Available from: <http://www.tobii.com/en/eye-tracking-research/global/library/manuals/>.
- [36] *Tobii Eyetracking SDK Library v2.0.50727*. Tobii Technology, 2011. Available from: <http://www.tobii.com/eye-tracking-research/global/products/software/tobii-software-development-kit/tobii-software-development-kit/>.
- [37] *Tobii T60 & T120 User Manual*. Tobii Technology, 2011. Available from: <http://www.tobii.com/en/eye-tracking-research/global/library/manuals/>.
- [38] *Tobii T60 XL User Manual*. Tobii Technology, 2011. Available from: <http://www.tobii.com/en/eye-tracking-research/global/library/manuals/>.
- [39] Antonio Albiol Colomer, Valery Naranjo Ornedo, and Josep Prades Nebot, *Tratamiento Digital de la Señal – Teoría y Aplicaciones*. Editorial Universidad Politécnica de Valencia, 2007.
- [40] Ulrika Sverkersten, *Visually Induced Ocular Torsion*. Doctor of Philosophy Thesis. *Karolinska Institutet*, ISBN 978-91-7457-341-1, Stockholm, Sweden, 2011.

- [41] Jan Ygge, *Ögat och Synen. Karolinska Institutet University Press*, 2011.

Appendix A

A diagram of the human eye

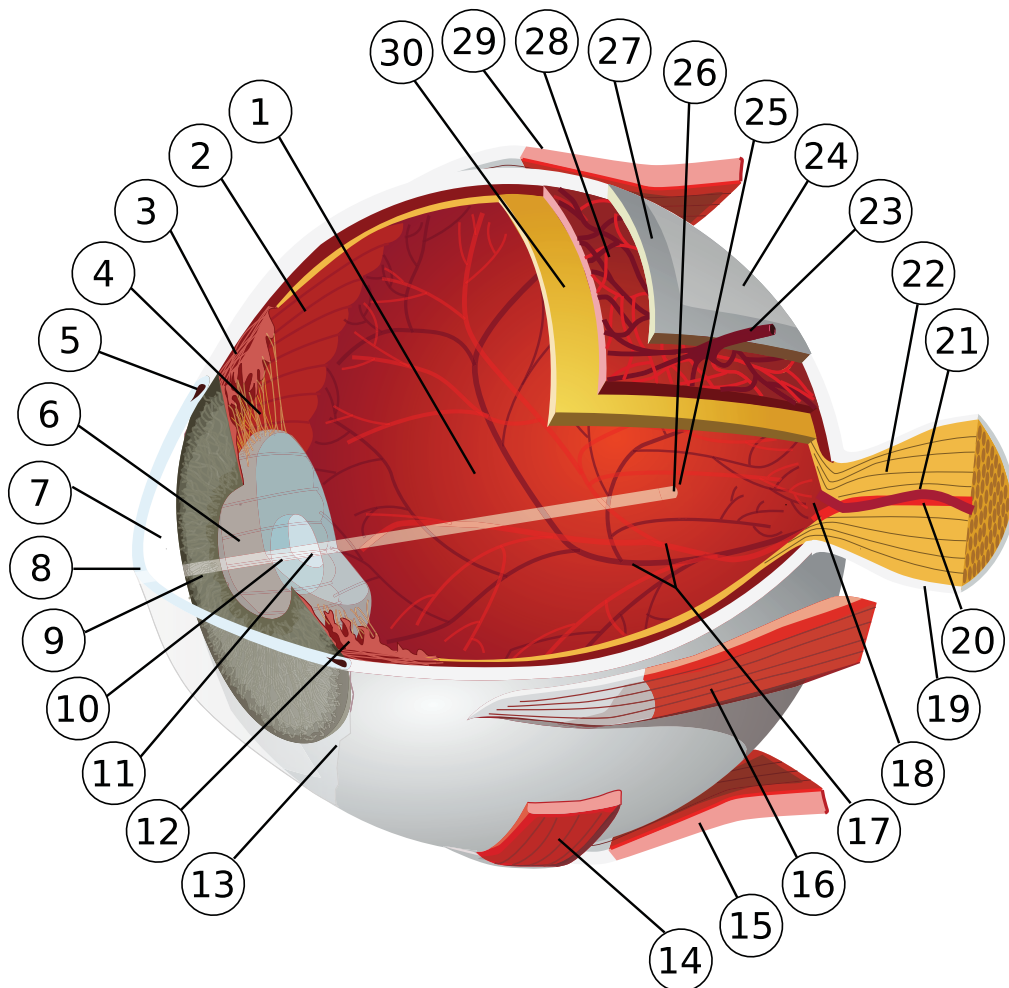


Figure A.1: A diagram of the human eye [14].

1. posterior compartment
2. ora serrata
3. ciliary muscle
4. ciliary zonules
5. canal of Schlemm
6. pupil
7. anterior chamber
8. cornea
9. iris
10. lens cortex
11. lens nucleus
12. ciliary process
13. conjunctiva
14. inferior oblique muscle
15. inferior rectus muscle
16. medial rectus muscle
17. retinal arteries and veins
18. optic disc
19. dura mater
20. central retinal artery
21. central retinal vein
22. optical nerve
23. vorticose vein
24. bulbar sheath
25. macula
26. fovea
27. sclera

- 28. choroid
- 29. superior rectus muscle
- 30. retina

