BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF RADIO ELECTRONICS
ÚSTAV RADIOELEKTRONIKY

A SYSTEM FOR EYE MOVEMENT TRACKING
SYSTÉM SLEDOVÁNÍ POHYBU OČÍ

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR Lucie Urbanová
AUTOR PRÁCE

SUPERVISOR doc. Ing. Jan Mikulka, Ph.D.
VEDOUCÍ PRÁCE

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Elektronika a sdělovací technika**
Ústav radioelektroniky

Studentka: Lucie Urbanová ***ID:*** 151753
Ročník: 3 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Systém sledování pohybu očí

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte řešení kamerového systému pro sledování pohybu očí. Vyberte vhodnou variantu s 1 nebo 2 kamerami umístěnými na pevných brýlích. Zvolte vhodné hardwarové (PC, Raspberry Pi, apod.) a softwarové (OpenCV, ITK/VTK, apod.) řešení.

Realizujte systém pro sledování pohybu očí navržený v semestráním projektu. Zdůvodněte výběr hardwarového a softwarového řešení. Navrhněte a realizujte počítačový program pro experimentální ověření kamerového systému na dostatečném počtu osob. Proveďte sérii zkušebních měření a vyhodnoťe nejistotu měření. Jako výstup měření se předpokládá dvourozměná informace o pohybu očí v čase.

DOPORUČENÁ LITERATURA:

[1] DAWSON-HOWE, Kenneth. A practical introduction to computer vision with OpenCV. ISBN 9781118848456.

[2] JOHNSON, Hans, J. et al. The ITK Software Guide Book 1: Introduction and Development Guidelines. ISBN 1930934270.

Termín zadání: 6. 2. 2017 ***Termín odevzdání:*** 30.5.2017

Vedoucí práce: doc. Ing. Jan Mikulka, Ph.D.
Konzultant:

prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta elektrotechniky a komunikačních technologií
Ústav radioelektroniky

ABSTRAKT

V této bakalářské práci jsou diskutovány možnosti a aplikace systému pro sledování pohybu očí, obsahující i aktuálně na trhu používané produkty. Práce následně uvádí řešení takového systému. Obsahuje návrh a realizaci držáku pro pevné uchycení kamery. Práce obsahuje i realizaci softvéru vytvořeného v C/C++ za využití Visual Studia a knihoven OpenCV. Tato aplikace sestává z procesů detekce očí, testování a vyhodnocení. Jako součást práce je provedeno a vyhodnoceno testování vytvořeného systému. Práce rovněž posuzuje vlastnosti systému a nabízí připadná rozšíření.

KLÍČOVÁ SLOVA

Detekce očí, konstrukce držáku kamery, OpenCV, sledování pohybu očí, testovací aplikace, zpracování obrazu

ABSTRACT

This Bachelor's thesis discusses possibilities and applications of the eye movement tracking system, including currently available products on the market. The thesis then introduces the solution of such eye tracking system. It includes the design and realisation of the camera holder for secure grip of the camera. The thesis includes the realisation of the software written in C/C++ using Visual Studio and OpenCV libraries. This application consists of eye tracking, testing and evaluation processes. As a part of the thesis, the testing of the created system was performed and evaluated. The thesis also evaluates the properties of the created system and discusses possible extensions.

KEYWORDS

Camera holder design, eye detection, eye tracking system, image processing, OpenCV, testing application

BIBLIOGRAPHIC CITATION

URBANOVÁ, L. *A System for Eye Movement Tracking*. Brno: Brno University of Technology, The Faculty of Electrical Engineering and Communication, Department of Radio Electronics, 2017. 49 p., 35 p. attachment. Bachelor thesis supervisor Assoc Prof Jan Mikulka.

PROHLÁŠENÍ

Prohlašuji, že svoji bakalářskou práci na téma Systém sledování pohybu očí jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestně právních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

(podpis autorky)

PODĚKOVÁNÍ

Ráda bych poděkovala doc. Ing. Janu Mikulkovi, Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích. Mé poděkování dále patří Ing. Tomáši Kříži, Ph.D. za pomoc při práci s 3D tiskárnou a nákresem obrouček. Dále bych chtěla poděkovat všem, kteří se zúčastnili měření a Mgr. Agátě Klusové za pomoc při korekci textu.

DECLARATION

I declare that I have written my bachelor thesis on the theme of “A System for Eye Movement Tracking” independently, under the guidance of the bachelor thesis supervisor and using the technical literature and other sources of information which are all cited in the project and detailed in the list of literature at the end of the project.

As the author of the bachelor thesis I furthermore declare that, as regards the creation of this semestral thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone’s personal and/or ownership rights and I am fully aware of the consequences in the case of breaking Regulation S11 and the following of the Copyright Act No 121/2000 Sb. of the Czech Republic, and of the rights related to intellectual property right and changes in some Acts (Intellectual Property Act) and formulated in later regulations, inclusive of the possible consequences resulting from the provisions of Criminal Act No 40/2009 Sb. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

(author's signature)

ACKNOWLEDGMENT

I would like to thank Assoc Prof Jan Mikulka for valuable counsel, objective remarks and helpfulness during consultations. My thanks also belong to Tomáš Kříž, Ph.D. for his help while working with 3D printer and sketch correction. I would also like to thank to everyone, who attended the testing session and to Mgr. Agata Klus, who helped me with text correction.

TABLE OF CONTENT

LIST OF FIGURES

INTRODUCTION

During the last decades, humanity has advanced in technological research more than ever before. In the information age, when everything becomes connected, it is important to gather and use every piece of information in every meaningful way. Thanks to the advancement in computer vision, gathering information about person’s eye movement may be used in security, healthcare, advertisement and countless other scientific fields. The purpose of this thesis was to gather the information on the topic of eye movement tracking and use it to design, construct and evaluate such a system.

At the beginning, the thesis introduces the reader to eye movement tracking methods, applications and currently available solutions. It also sums up the history of the topic.

In the main part of the thesis, the eye tracking system realisation and functionality are described. The system consists of a camera attached to the camera holder mounted on the head of the subject. The camera captures the left eye of the subject and detects the iris. The location the subject is looking at is calculated by using the input video of the eye combined with calibration results.

To evaluate the system, the application was created. It consists of eye tracking, calibration, testing and evaluation process. The testing part includes several tests designed not only to measure the test subject, but also to test the tracking system capabilities and precision. There are six tests in total, each focuses on a different aspect of the application and different qualities of the subject. The evaluation part consists of four different kinds of evaluation. They are created to evaluate the testing process thoroughly, focusing on different aspects of the measuring.

The measurement was performed on a number of subjects and the accuracy of the system was assessed. The measurement and its results are included in the attachment of the thesis. At the end of the thesis the possible extensions of the created eye tracking system are discussed.

1 INTRODUCTION TO THE EYE TRACKING

As the time goes by our world is changing. Everything goes faster and the demands pressured on us are greater. But at the same time everything is developing, growing and improving. This is why we collect as much data as possible in any way we can and we try to get as much information out of them as we may. One kind of data we receive every day while meeting anyone is their eye movement. We notice it when someone looks at us or talks to us but we mostly do not focus on it and detect only a small part of data.

Imagine playing a game of chess where you notice every move of your opponent´s eye. You would know what he is interested in and could predict his next moves. If you were able to process these data in real time, you would win that game easily. Or someone paralyzed communicating through looking at the characters of alphabet displayed on the screen in front of him.

There are many ways of processing these data into information we would find very beneficial. This part of my thesis will focus on possible uses of eye movement tracking, on techniques of its detection and on currently available products that are able to do that.

1.1 THE HUMAN EYE

To be able to track the eye movement, it is crucial to learn the human eye anatomy and physiology as shown in Figure 1.1.

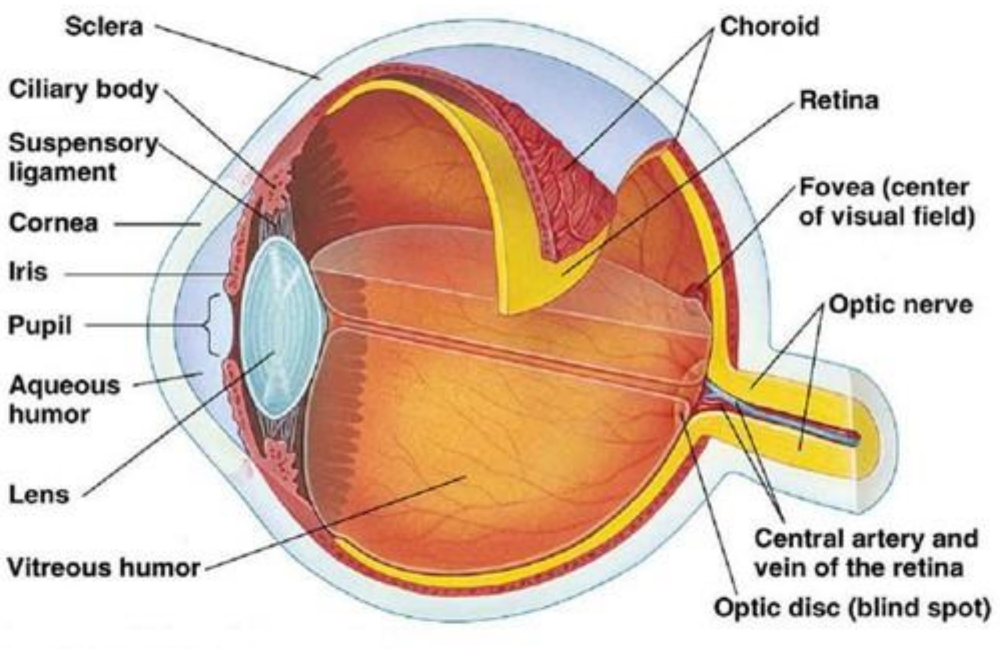


Figure 1.1 Human eye anatomy; taken from [1]

The most significant components of the human eye mentioned in this thesis are:

- Cornea

  Cornea is the transparent circular part on the front side of the human eyeball. It refracts the light entering the eye through the pupil onto the lens [2][3].

- Iris

  The circular coloured part of the eye is called iris. It works as the muscle that adjusts the size of the pupil to regulate the amount of light admitted into the eye. This regulation is called pupillary reflex.

- Pupil

  Pupil is the circular spot in the middle of the iris that enables the light to pass into the eye. Its size is regulated by the pupillary reflex. When the bright light reaches retina, parasympathetic nervous system is stimulated and a muscle tights around the iris contracts, reducing pupil size. It is a regulating and self-protecting system of the human eye.

- Lens

  Lens is located behind the pupil. It refracts the light passing through and focuses it on the retina. Lens is able to change its shape, enabling human eye to focus on different objects at various distances from the eye. This function is called accommodation.

- Retina

  The light passing through the cornea, the pupil and the lens is captured by retina. It consists of rods and cones, the photosensitive cells and their nerve fibres. They convert the light into nerve impulses that are sent to the brain by the optic nerve.

- Sclera

  Sclera is the tough white sheath on the outer layer of the eye. It forms a strong, dense protective layer that maintains the shape of an eye. It is thicker at the back of an eye, resulting in slightly shaping the eye forward. Its colour changes from white to slightly yellow with age.

1.2 BRIEF HISTORY

The experiments trying to describe eye movement began in 19th century with directly observing the subject. After that, Louis Emile Javal discovered that people do not read smoothly across the text, but rather pause and move quickly through some words.

The first eye tracking device was invented by Edmund Burke Huey in early 1900s. He created lenses with holes so the person could see and attached long needles to them. The needles moved as the subject was reading, enabling the study of the eye fixations. This device was very uncomfortable for the tested person, as their eyes had to be under anaesthetics and they could not blink.

In 1937, a psychologist Guy Thomas Buswell built a device that used the light focused on the eye. Depending on the direction that the eye was looking, the reflection changed, as he recorded it on the film. His work showed that there is difference between silent and loud reading.

In 1950s, a Russian psychologist Alfred L. Yarbus conducted several studies on eye tracking research that described relation between eye fixation and the interest of the tested subject [4].

In 1970s and 1980s, thanks to advancing technology, eye tracking devices became less intrusive, more accurate and able to separate head and eye movements. In 1980s began the first eye tracking researches of humans working with computers in real time. In 1990s the commercial applications of eye tracking while browsing the internet began.

1.3 APPLICATIONS

Eye tracking technologies in today's society have a wide scale of applications used in many fields. A few examples are listed below.

- Medical research
  It enables paralyzed people to write as well as to move the mouse of the computer by using their eyes. It is also used while doing laser refractive surgery. Early stroke recognition is currently in development.
- Psychology and cognitive science
  Used for research of human interaction, mood, stress, fatigue detection and lie exposure. It is also used for dyslexia diagnosis as well as schizophrenia, which can be detected with 98.3 % accuracy [5].
- Marketing research
  Eye tracking is widely used in marketing to scan what commercial or product attracts the customer most. It is used to decide what kind of wrapping is the most attractive for customers, where the most frequently viewed area on the shelf is or which parts of web pages are suitable for placing a commercial. One of the companies that is focused on marketing research in the Czech Republic is UXFocus [6].
- Human-computer interaction
- Activity recognition
- Sensors and security
  Some car security systems scan the driver´s blinking to prevent the micro-sleep accidents. Retina scans are used for security measures, as each individual has a unique one, just like fingerprints.
- Virtual reality
  By knowing the direction in which the subject is looking, it is possible to move the camera to capture the same location.
- Trainings and simulations
  For example, Cristiano Ronaldo tested his spacial awareness by using eye tracking system to improve his game and to review his current performance.

1.4 EYE TRACKING TECHNIQUES

All the eye tracking techniques may be divided into two categories. Those that measure the position of an eye relative to the head, and those that measure the orientation of the eye in space, which gives us the point of regard [7].

The first category requires an additional information to pinpoint the point of regard. Those techniques were popular in the past, but thanks to the technological advancement they are being replaced by the second category. The latter ones are commonly used for interactive applications or those where location of an object is the main purpose. Probably the most widely used apparatuses these days are the video-based ones, especially corneal reflection.

Electro-Oculography

Electro-oculography, or EOG, was the most widely applied technique in mid-1970s. It measures eye movement relative to the head position, so it is impossible to detect point of regard without measuring the head movement as well.

It is based on the measurement of electric potential differences, which exist between the front and the back of the human eye, by electrodes placed on the skin surrounding eyes. These potentials are between 15-200 µV and they are plotted in electrooculogram. [7] The eye poses as a dipole in which the front of an eye acts as a positive and the back as a negative pole. The negative electrode is placed near the outer canthus of an eye, the positive one is placed near the inner canthus. Electrodes detect a negative trending change of the potential when the subject's left eye looks left and a positive one as he looks to the right. For the right eye the level of potential difference is inverse.

Scleral Contact Lens/Search Coil

In 1950s, the techniques using contact lenses were developed to improve accuracy of measurement. The lenses are equipped with small objects as mirrors or coils to enable tracking of their movements. The first devices were developed using the reflective lenses. This technique was based on detecting the reflected light. Despite being very precise in those days, this approach was later abandoned because the light focused on the lenses was disturbing the tested subject.

Eye tracking using a scleral search coil is a very precise method able to react to quick eye movements. The coils are either adhered to an eye or attached to lenses. The vertical altering magnetic field is powered by magnets attached above and below the eye. The coil is attached vertically, so when the subject looks directly ahead, the induced current is zero. According to Faraday's law, when the subject looks up, the angle between the magnetic field and the coil wiring increases, therefore the electric voltage is induced. The higher the eye turns, the larger the angle is, resulting in higher voltage induced. As the subject looks down the same angle, the same voltage is induced, only with reversed polarity. Practically, there are two magnetic fields, the horizontal and the vertical one, each altering at different frequency, enabling us tracking of the vertical and horizontal eye movements. [8] The major downside of this method is that the coil must be wired to the measuring station, resulting in the wire attached to the test subject´s eye, making the method very invasive.

Video-Oculography

Video-oculography, or VOG, is a wide category of eye tracking techniques based on recording, focusing on measurement of an eye under rotation and translation, e.g. limbus tracking, apparent shape of pupil or corneal reflections of a closely placed source of light (mostly infrared one).

However, these methods mostly do not resolve into point of regard and usually involve other measurements in order to be precise. Detection of the position of limbus usually demands usage of photodiodes attached to the glasses placed on the subject's face and other source of illumination. These methods may also require visual inspection of eye movement recorded on video, which may be susceptible to error and are limited by frame rate of the camera. Head fixation is almost always required as the method is not able to compensate head movement.

Video-Based Combined Pupil/Corneal Reflection

There is a difficulty in acquiring the point of regard measurement using video-based techniques called head movement. One way to remove that problem is head fixation. Securing the head in one position is highly uncomfortable for the subject and limits the practicability of the study. Another way to ensure the point of regard location is by scanning multiple ocular aspects to distinguish head movements from eye rotation. Pupil and corneal reflection is a technique utilizing the reflectivity of an eye. Along with the location of the pupil, the camera scans the light reflected on the eye from a source with known coordinates.

There are two types of infrared eye tracking. Bright-pupil tracking, when retina is illuminated by the IR light, creates a high contrast difference between iris and retina. It allows to track the movement in darkness, as well as in strongly lighted environments. Dark-pupil tracking does not illuminate the pupil, resulting in the pupil observed as dark. This allows better visualization of corneal reflections, making dark-pupil tracking more suitable for our measuring.

When the head is stationary, the pupil moves, but the light reflection does not change its position. If the head moves, the reflection on the eye moves accordingly, which enables the software to calculate the new location of the subject´s head and its new point of regard. The light source used is typically infra-red to minimize the invasiveness. Corneal reflections are known as the first Purkinje images which are named after Jan Evangelista Purkyně (1787 - 1869), the Czech anatomist and physiologist.

1.5 CURRENTLY USED TRACKERS

These days the eye trackers are mostly of non-invasive type. The video-based combined pupil/corneal reflection category is a widely used one. The base of the trackers are the cameras directed on an eye (or eyes). Sometimes they capture the eye from the distance while being statically secure, e.g. fixed to the bottom of a monitor, sometimes they are attached to the glasses frame and only take a recording of the eye and its close surroundings. This chapter gives a brief overview of several commercially available trackers and companies which sells them.

Medicton Group

There are several manufacturers on the world market focused on the field of eye tracking and other measurements of human responses and conditions. In the Czech Republic, there is only one company so far that develops the eye tracking system. Its name is Medicton Group s.r.o. and its tracking software is called I4Tracking.

It is based on pupil tracking method using a high frame-rate infrared camera (over 150frames per second) that is attached either to the glasses frame – the first version of tracker hardware or to the computer monitor – a newer model. It is designed for tracking eye movement while the subject is watching a computer screen. However, the first version of the product is not adjusted for the active head movement and it may cause error while evaluating the measurement. The newer version is eliminating this problem by detecting the reflected light from the cornea. Beside many other products like blood pressure and temperature meters, this company also develops a contactless computer controller called I4Control [9].

Tobii Pro

Tobii Pro is a Swedish unit within Tobii Group that focuses on the sphere of human behaviour research based on the eye tracking. They developed several types of eye tracking systems including both hardware and software, which makes them one of the leading companies in this field.

Their glasses tracker called Tobii Pro Glasses 2 (see Figure 1.2) is the second generation of wireless wearable trackers. It is based on 4 cameras which are aimed on eyes, a scene camera that has a 90° viewing angle and it also includes a gyroscope and an accelerometer. Those provide information about the head movement so the error of result would not appear. This system captures data at 50 or 100 Hz [10].

Figure 1.2 Tobii Pro Glasses 2; taken from [10]

Some of their other products are screen-based. For example, one of them - Tobii Pro Spectrum (see Figure 1.3) is able to capture gaze data at speed up to 600 Hz with accuracy 0.4° at 65 cm and it also offers freedom of head movement without errors in results. This product is recommended for 23.8" monitor and its tracking method is binocular bright and dark pupil tracking.

Figure 1.3 Tobii Pro Spectrum; taken from [10]

Similar products to Tobii Pro Spectrum is Tobii pro X3-120 or Tobii Pro X2-60.

Their products are supported by their software called Tobii Pro Studio and Tobii Pro Lab, that are optimized for their products.

Other companies

SensoMotoric Instruments (SMI) is a German company manufacturing several types of eye trackers. Their SMI Red 500 is a monitor-based eye tracker. It is placed below the monitor and records eye movement at 500 Hz. It tracks both eyes and enables free head movement [11].

SMI also support Oculus Rift devices for virtual reality and produce their extensions (eye trackers). They also produce wireless eye-tracking glasses and support their products with specialized software.

SR Research is a company from Canada that offers the eye tracking device with frequency 1 kHz while tracking both eyes or 2 kHz while tracking a single one. It is a static tracker located below the computer screen [12].

The Eye Tribe is a Danish company producing a low-cost eye tracking device including its own software for only 99 USD. The tracker works at 30 or 60 Hz and it is possible to connect it with Android or Windows Phones [13].

The company from Latvia named Neurotechnology developed their software called SentiGaze SDK that enables any computer with a web-camera to track eye movement [14].

2 EYE TRACKING REALISATION

This chapter is focused on practical results of the Bachelor's thesis and describes how the algorithm, that was designed, works. It includes the camera holder construction, its application in a video recording session and a real-time working programme for eye detection analysis.

2.1 CHOSEN SOLUTION

From a wide range of possibilities, a solution for construction and programming of application for detection of the point of regard, was chosen. This chapter gives an overview of this chosen method. For outline see Figure 2.1.

The foundation of this design is one camera fastened to a camera holder and aimed on the left eye of the testing subject. This solution makes the solution a cheap, reliable and functional eye tracking system. The camera holder is designed in SolidWorks and printed on 3D printer, attached to the testing subject's head by an elastic band. The camera then passes the captured video to a computer via a one-meter-long cable with a USB 2.0 port.

To capture the whole eye, the camera is attached to a holder in front of the eye on the bottom outer side to minimize the sight blocking. It is attached to the extension part of the holder that connects the forehead frame and the camera. The forehead frame is the base of the camera holder construction. It consists of a cut-out part to fit the forehead and a solid frame allowing the attachment of the extended part and the clips to hold the rubber bands. The clips are designed to hold the rubber band stabilizing the construction on the subject´s head. In order to capture the stabilized video, the construction of the holder has to be made out of strong and thick enough material. An example of the construction and material texture can be found in Chapter 2.2.

The video of the captured eye is proceeded to a computer, where it is processed to detect the eye by a program written in C/C++ via Visual Studio 2013, using OpenCV 3.1 libraries and CLR Windows form. Visual Studio is an integrated development environment for Microsoft. It is used for computer programs, web sites and applications. Open CV (Open Source Computer Vision) is a library of programming functions focused on real-time computer vision [15] [16]. The software solution enables the application to have a user-friendly design with windows form graphics. The OpenCV supports the computer vision library functions with already implemented image filters needed for used solution. The software part of the design must resolve into transforming the input image to the calculated point of regard. In the first stage, the algorithm processes the raw input video, reliably locating the centre of the eye.

Eye detection (described in Chapter 2.4) is realized in several steps. It begins with mirroring the image. After that the image is converted from RGB into a red channel to receive a better ratio of contrast. To smooth the image, a normalized box filter is applied. Then it is easy to detect edges by Canny algorithm from which the circle, created by iris, is detected by Hough transform. The transform also detects the centre of the circle.

The second stage processes the eye centre positions during the time and compares them with calibration process results. The calibration process is based on test subject´s full cooperation. The subject looks at the points placed all over the screen of the calibration program one by one. With each focus of his sight at the concrete point, the subject marks it by clicking on it with the mouse, at the same time he looks at it. The positions of the acquired points are then compared with the position coordinates of the eye, captured by the eye position detection software, at the same time exactly. This way the point of regard is found.

After detecting the point of regard, the testing of the subject may begin. For the testing and evaluating, the testing application called "EyeTrackApp" is designed. The test consists of a reading part, a watching the picture part and an easy brain teaser. The test was repeated with several test subjects to compare the functionality and reliability of the method as well as the accuracy of the detecting method. The output of the testing is a graph connecting the point, where the subject was looking, a heat map, a heat map replay to show the development over time and moving dot replay with the exact point of regard changing in time. The testing application also presents the option of real time detection.

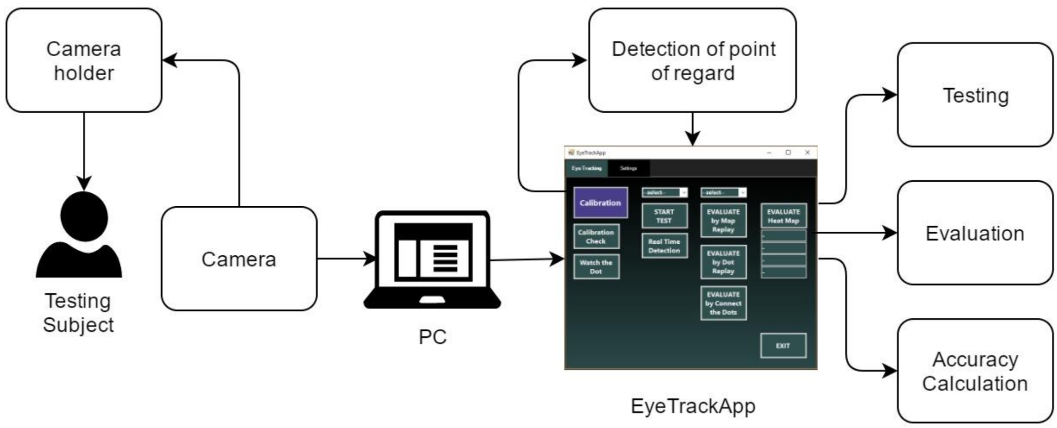


Figure 2.1 Eye Tracing System Overview

2.2 CAMERA HOLDER DESIGN AND REALISATION

To provide a camera standing in the same position at all times, the camera holder has been designed. The design was drawn in SolidWorks and printed on a 3D printer at school. Before drawing the whole camera holder, it is crucial to know the limitations of all technological processes like tolerances, desired precision and dimensions of the parts or the assembly. The assembly designed for this project had to meet several requirements and limitations.

The available 3D printer, type K8200, has 0.3 mm layer height and has from 0.25 mm to 0.5 mm dimension tolerance. Its printing area is only 20 × 20 × 20 cm. The sketch is adjusted to these factors by adding 2mm clearance between the fitting parts and by dividing the sketch into several pieces. The division into several pieces was also done to prevent the printing defects. 3D printer needs a solid base to print onto. As it applies the layers on each other, it requires an area to stick to. The printed parts are not fully filled with the printing material as instead there is a grid (see Figure 2.2) to support the construction as well as to lighten it.

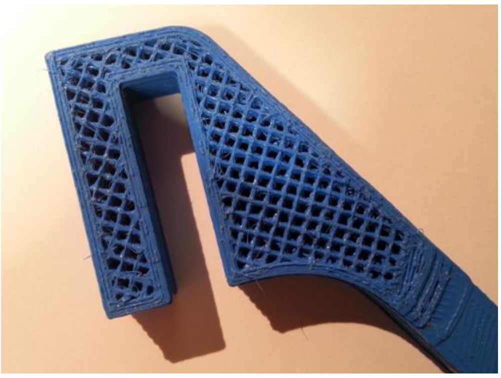


Figure 2.2 The support grid

The sketch (see Figure 2.3) is designed in SolidWorks. SolidWorks is one of the most commonly used 3D CAD software products in the world. It offers a wide variety of functions and interfaces to allow the user to create any shape/ part/ assembly quickly, precisely and to export the draft to process it further.

The holder consists of four main parts: the forehead support, the frontal extension and two clips for attaching the elastic band. The base of the whole construction is the forehead support. It includes grooves for attaching other parts and a curved front for stabilization and comfort of the tested subject, which is also achieved by a small pillow that is glued on it. The frontal extension is attached by adhesive onto the forehead support. Its bottom includes the camera attachment with three holes in it to secure the camera by screws, placed in the position, that minimises blocking of the sight. Securing the whole holder to the subject´s head is done by two clips glued to the forehead support as well. They are equipped with an elastic band that is supposed to go around the head. For the whole assembly see Figure 2.4.

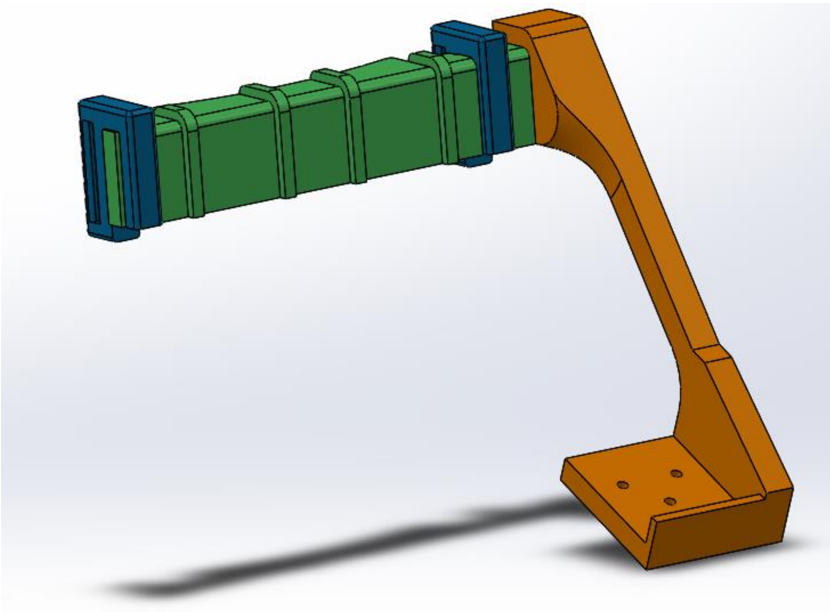


Figure 2.3 Sketch of camera holder made in SolidWorks

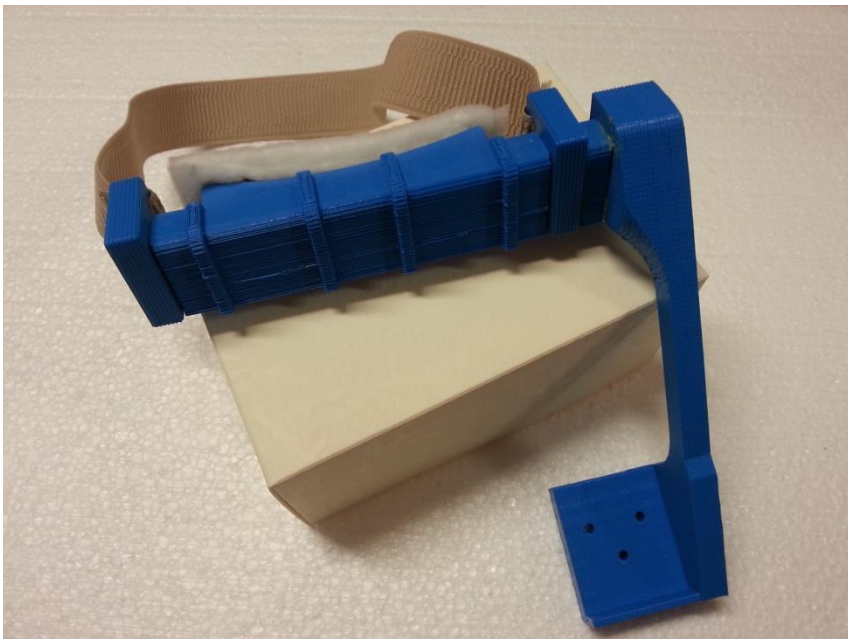


Figure 2.4 Manufactured camera holder

2.3 TESTING APPLICATION

This chapter describes the design and functionality of the testing application. It also gives a brief overview of possible settings. The program is written in C/C++ via Visual Studio 2013, it uses OpenCV 3.1 libraries and is based on CLR Windows form.

This application fully coordinates the functions of the eye tracking, the calibration process, the point of regard finding, the testing process with the option of real-time tracking as well as the evaluation process. Its global inner structure is shown in Figure 2.5.

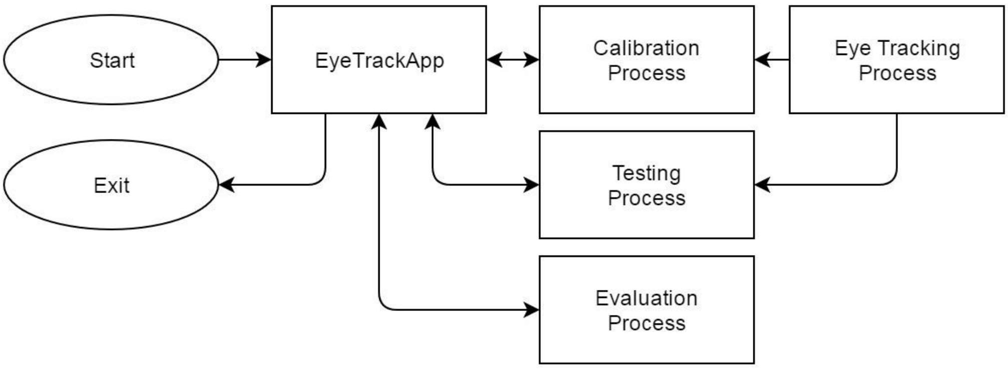


Figure 2.5 The structure of the testing application

After launching the application, the main window called “EyeTrackApp” appears on the screen. It is the coordinator of all application processes, as well as the user’s tool for operating the application. It also includes the application settings. “Eye Tracking

Process” gets the image from the camera, finds the circle around the iris and returns its middle coordinates. “Calibration Process” calculates the initial points of regard by combining the “Eye Tracking Process” data with the data from calibration. After the calibration is complete, the camera-screen recalculation is possible and the testing may be initialized. “Testing Process” may vary depending on the chosen test and real-time testing choice. It combines the calibration data with the “Eye Tracking Process” data in real-time to calculate the current point of regard of the tested subject during testing. It may display the calculated point of regard in real-time as well. “Evaluation Process” processes the data from “Testing Process” and returns the measured test result in the chosen form. The available forms are Heatmap evaluation, Real-time Heatmap evaluation, Connect the Dots evaluation and Replay evaluation. It also returns the average and maximum deviation. The whole EyeTrackApp also contains tooltip comments to help the user with understanding the software and also changes the colour of the button to the lighter shade of colour while placing the mouse over the button.

After launching the testing application, a window as shown below will appear (see Figure 2.6). This is where most of the functions are. It includes a calibration button, another button for calibration check and a button for watch the dot test, two combo boxes to pick a test from (separately for testing and evaluation), start buttons for test’s start and for real-time detection as well as four buttons for different types of evaluations and four text boxes in which the evaluated inaccuracy is given. At the bottom, there is an exit button. On the top, the application may be switched to setting, which will be described later.

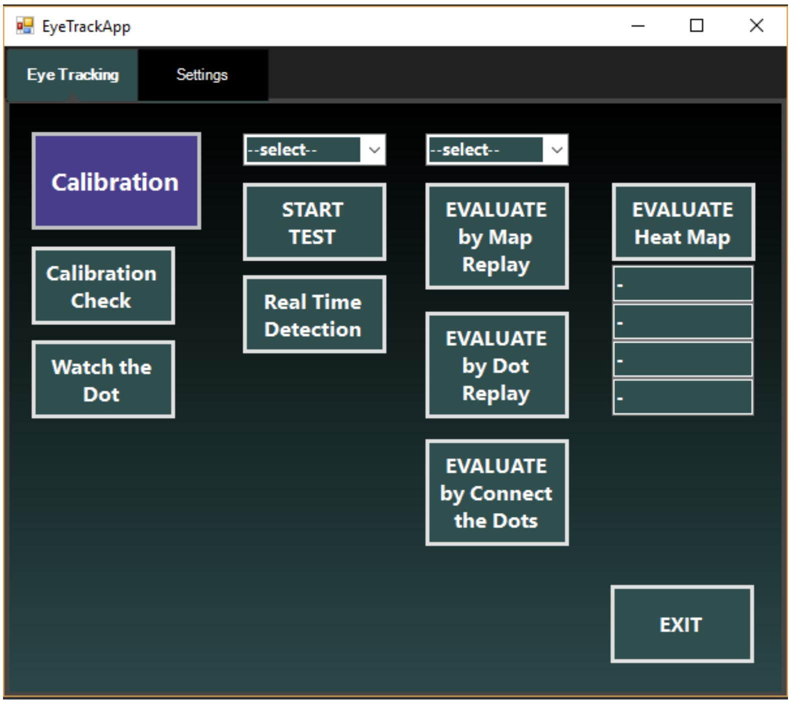


Figure 2.6 EyeTrackApp - main window

After the user clicks the “Calibration” button, the program initializes new Windows Form object called “CalibrationWindow”. It coordinates the calibration process and will be described in detail later. The “Calibration Check” button (see Figure 2.7 left) initializes new Windows Form object which allows the user to check whether the calibration is accurate. The “Watch the Dot” button starts the “Watch the Dot” testing process. Its inner testing process differs from other testing methods, which is the reason why it is not included in the testing section of the form.

Testing combo box (see Figure 2.7 middle) allows the user to select the desired test. “START TEST” button initializes the testing process without real-time detection. The algorithm first checks, which test was selected in the combo box and launches the new testing process accordingly. The “Real Time Detection” button starts the testing process with an additional window with detected point of regard in the testing form. The algorithm checks the selected test and launches “RealTimeTracking” testing algorithm.

Evaluation section of the application includes evaluation combo box (see Figure 2.7 right) in which the user selects the test he wants to evaluate. The “Evaluate Heat Map” button starts the algorithm, which results in displaying the heat map on top of the selected test. While selecting “TEST 5” in the combo box, the deviations from that test are calculated as well. The textboxes display (from top) the average deviation in pixels, average deviation in millimetres on the screen, maximum deviation in pixels and maximum deviation in millimetres. The “EVALUATE by Map Replay” button initializes the algorithm which results in an animation of a heat map generation over time of the selected test. “EVALUATE by Dot Replay” button starts a function, which replays the calculated point of regard of the selected test over time. The “EVALUATE by Connect the Dots” button starts the algorithm which results in connected points of regard, as they were measured, displayed in the selected test.

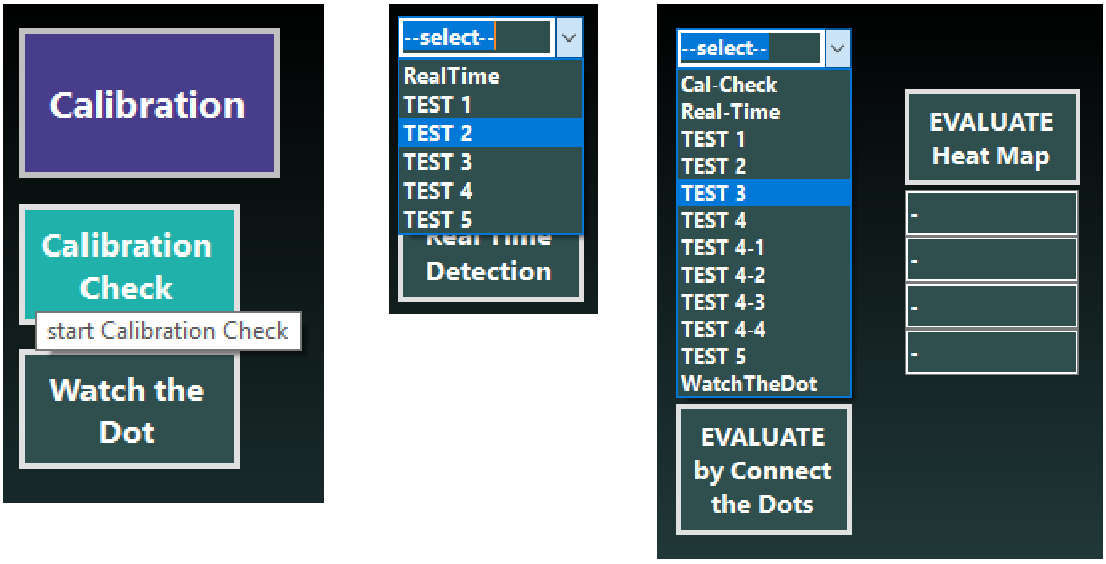


Figure 2.7 EyeTrackApp features including (left) calibration section with visible tooltip and lighting of button, (middle) testing combo box with visible selection and (right) evaluation combo box with visible selection, evaluation buttons and text boxes for deviation results.

The “Settings” tab (see Figure 2.8) of the application allows the user to modify the parameters of the eye tracking algorithm in “Set Hough Values” section, to change the colour scheme of the application in “Set Background Colour” section and to check the eye tracking settings functionality. The “Set Hough Values” allows the user to modify the input variables of the “HoughCircles” function from the OpenCV used in the algorithm.

The first parameter “dp” sets the accumulator size for half the size of the camera image input. This option increases the algorithm speed significantly. The second parameter “param1” modifies the edge sharpness values. “Param2” coordinates the algorithm to detect or ignore the circle. The smaller “param2” value is, the more of false circles may be detected. “Minimum radius” and ”maximum radius” determine the radius size boundaries, which the detected circle may have [15]. The setting of these parameters may be useful while using this system with a different camera. The “START” button initializes the eye tracking algorithm with currently set Hough Values and shows the window with the captured frame from the camera with drawn circles inside to check the functionality of the settings. The “STOP” button cancels the eye tracking algorithm and exits the camera window.

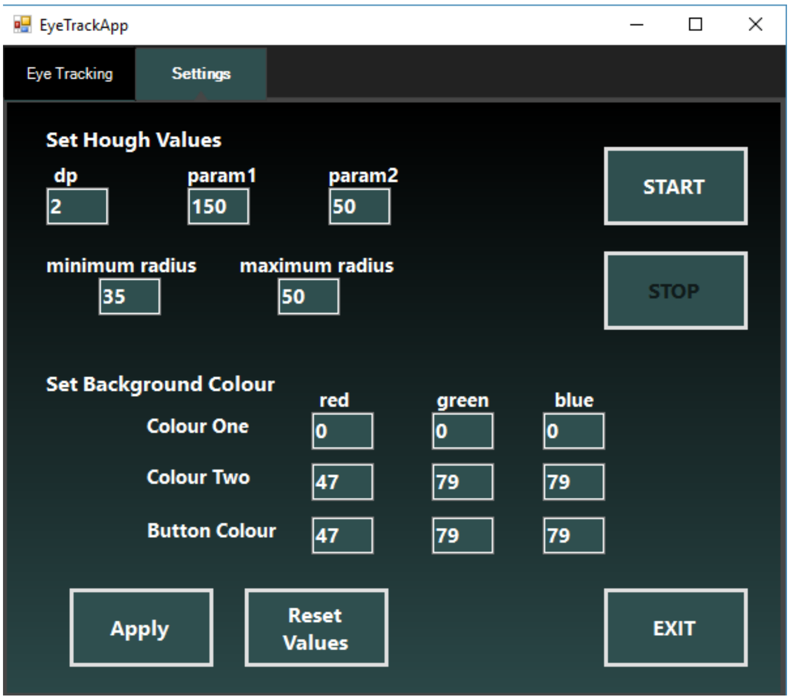


Figure 2.8 Settings tab of the EyeTrackApp window

To set the new colour scheme to the application, “Colour One” values determine the RGB intensities of the top background colour, ”Colour Two” values determine the colour of the bottom background colour and “Button Colour” determines the background colour of the buttons (for examples see Figure 2.9). To apply the newly set Hough Values and Background Colours the user needs to click the “Apply” button.

Figure 2.9 The example of colour schemes of the application

The “Apply” button reads the values set in all textboxes in the tab, converts them to integer values and sets the desired background colours to the appropriate values. The button also sets the values of the global structure “HoughConstants” containing five float values to match the values in textboxes. The “HoughConstants” are utilized by the eye tracking algorithm while finding the eye circle. The “Reset Values” button returns the values of both Hough Values and Background Colour to the default settings. The “EXIT” button ends the EyeTrackApp application.

2.4 EYE TRACKING AND CALIBRATION PROCESS

This chapter describes the part of the testing application concerning the eye tracking process and calibration. It also includes used camera characteristics and used software. The whole process of the eye detection is captured in Figure 2.10.

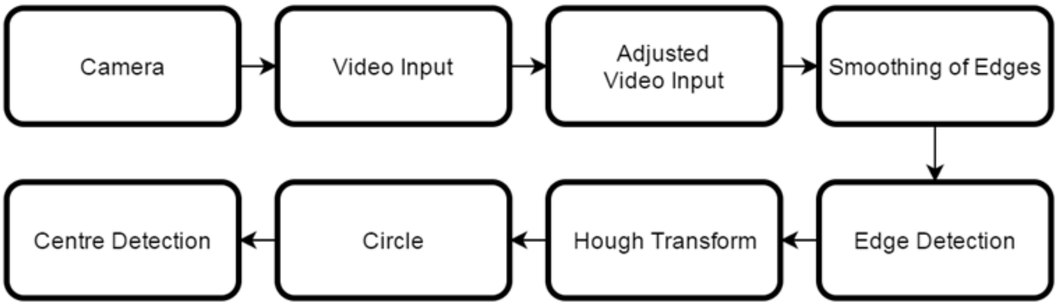


Figure 2.10 Eye detection process diagram

The camera that is used for video capture of the eye tracking system is a web camera from A4tech, type PK-910H, with Full HD 2 Mpx scanner, it captures video in up to 1920×1080 px resolution, with maximum of 30 fps (frames per second) and supports auto focus. It is connected to the computer by USB 2.0. The camera is placed on the camera holder extension. It is aimed at the left eye, capturing it from the outer side of the head. This enables the system to capture the whole eye without blocking the vision of the tested person significantly. The whole setup is shown in Figure 2.11.

The “Eye Tracking Process” consists of the algorithm, that gets the raw input from the camera and returns the centre coordinates of the detected iris with the radius of the detected circle. The input video is processed using commands from the OpenCV libraries. While the video runs real-time, the frames from the camera are processed separately as single pictures. To simplify the work with the video, the image is first mirrored to create a mirror-like environment (see Figure 2.12). This operation will later enable us to find the point of regard from the processed image.

Figure 2.11 Camera and its holder - application in video recording session

Figure 2.12 Example of mirrored image

The following operation divides the RGB picture into three separate matrixes, each including one colour channel. The application then processes the red channel matrix because the eye is the most distinguishable there compared to the other colour channels and to the greyscale of the original frame. The red channel has the best contrast proportions. Example can be found in the Figure 2.13 which compares blue (part 1), green (part two) and red (part three) channel together with greyscale (part four).

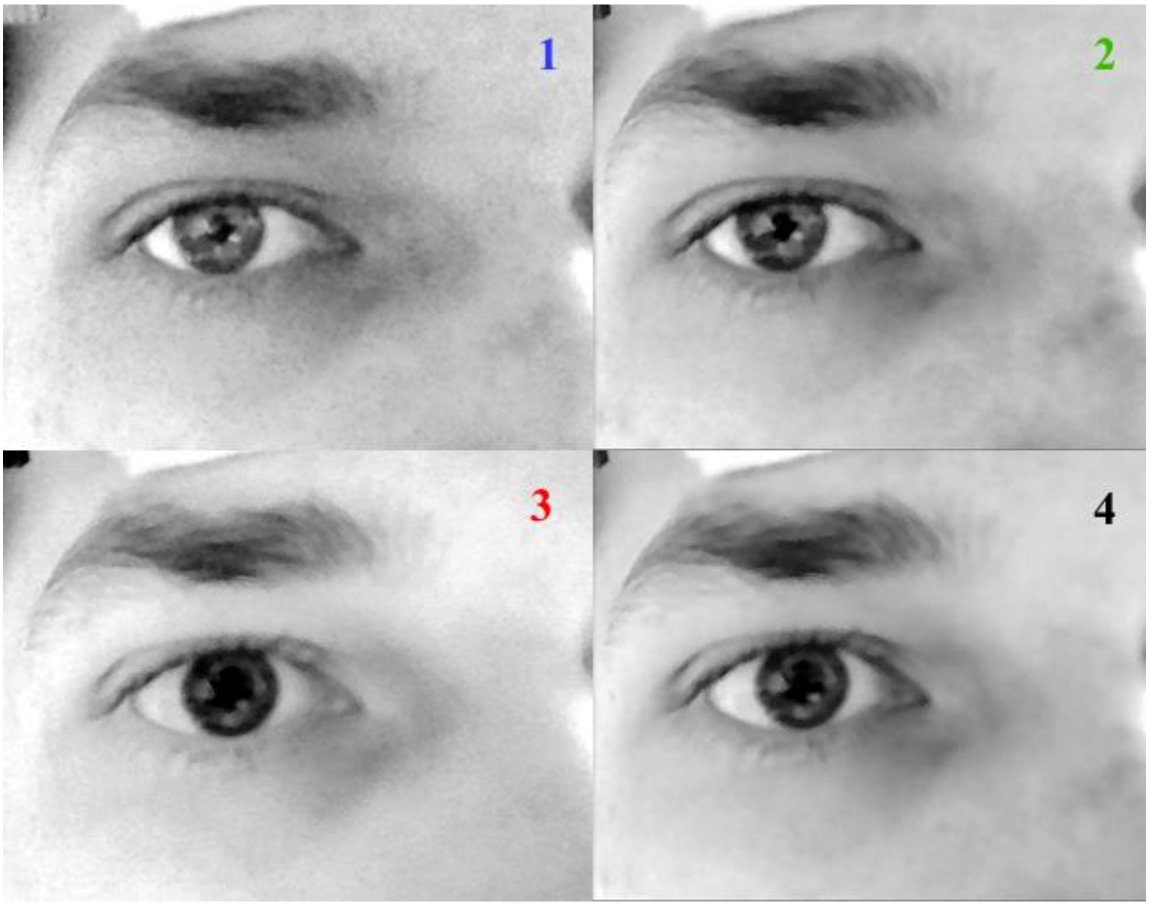


Figure 2.13 Example of blue (1), green (2), red (3) colour channel and greyscale image (4)

Then the image is filtered to blur the homogeneous areas using the normalized box filter. It smoothens the image and it also reduces less significant edges. A linear box filter is a spatial linear filter in which each pixel in the resulting image has a value equal to the average value of its neighbouring pixels in the input image. It is a form of low-pass filter that reduces noise [15].

The function smoothens the image using the convolution matrix - kernel **K** (see Equations 2.1, 2.2, 2.3):

$$\mathbf{K} = \alpha \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \qquad \text{(Equation 2.1)}$$

where

$$\alpha = \frac{1}{m \cdot n} \qquad \text{(Equation 2.2)}$$

where *m* stands for height of kernel and *n* for its width.

This program uses the 3 × 3 kernel:

$$\mathbf{K} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \text{(Equation 2.3)}$$

The result of the blur is captured in Figure 2.14.

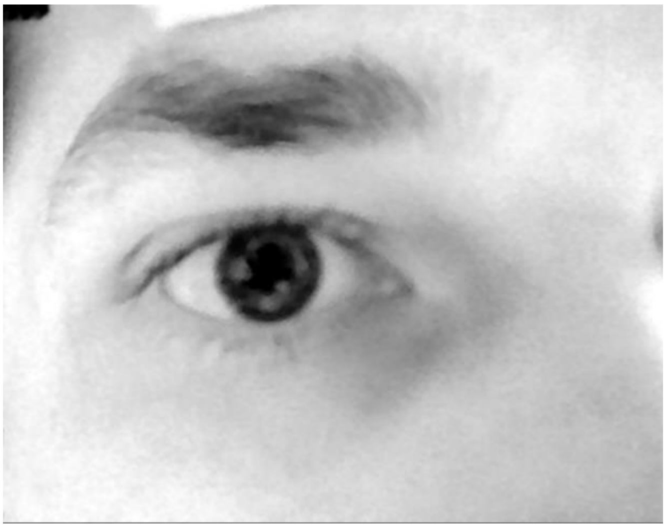


Figure 2.14 Example of blurred image

To find the edges in blurred image the Canny algorithm is used (see Figure 2.15). The Canny algorithm reduces the amount of data that needs to be processed, accurately finds the edge points in the centre of the edge and prevents the image noise to create false edges [15].

The Canny algorithm first applies the Gaussian filter to remove the noise, then finds the edges using the first derivate filter in horizontal and vertical directions. The edge detection algorithm then keeps only the maximum intensity edge values to locate the true maxima. The double threshold filter then removes the non-edge positive results. The final filter separates the weak edges that remained, keeping the ones that are connected to the strong ones while the unconnected noise edges are suppressed [15].

Figure 2.15 Image of detected edges

From the image of the edges it is easy to detect the circle that defines the iris by using Hough circle transformation. It finds circles in imperfect images by finding the circle "candidates" by "voting" procedure in the Hough parameter space. Candidates are obtained as local maxima in accumulator matrix, which is explicitly constructed for computing the transform. The parameters which define the found circle are *x* and *y* position of the circle centre and the circle radius.

For sake of efficiency, OpenCV implements a detection method slightly more complicated than the standard Hough Transform. It is called the Hough gradient method. It uses the local gradient direction to reduce the number of operations [15] [16].

See example of the found circle and its centre in Figure 2.16.

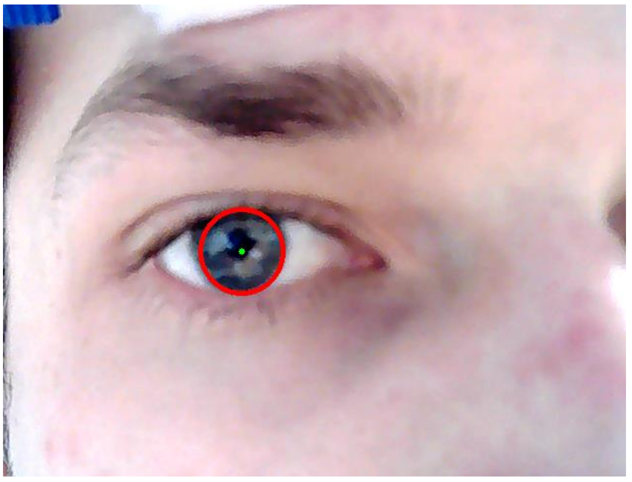


Figure 2.16 Image of detected iris

Thanks to Hough transform's ability to detect a circle even when it is not fully visible, it is able to identify the eye when it moves in any direction. See Figure 2.17. It shows detection during the eye movement in up (part 1), down (part 2), left (part 3) and right (part 4) direction.

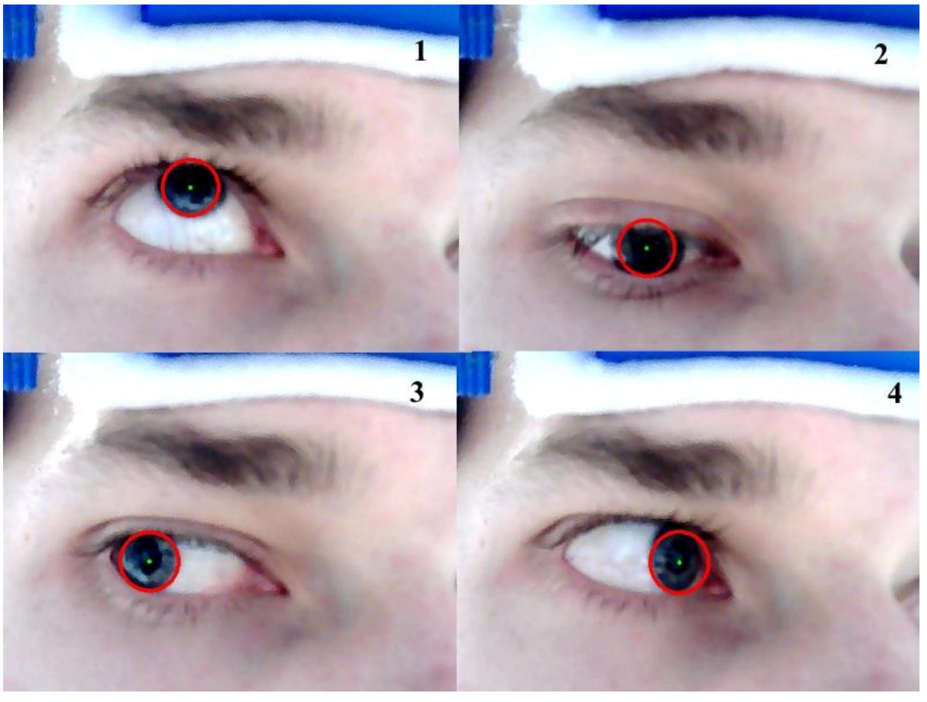


Figure 2.17 Detection during the eye movement - up (part 1), down (part 2), left (part 3) and right (part 4) direction

Using this whole method, the program is able to detect the eye in real time. The video of the subject´s eye, taken by the camera mounted in the camera holder is processed and evaluated by the algorithm and prepares the project for further point of regard finding.

The "Eye Tracking Process" software solution algorithm is displayed in Figure 2.18. When the function "getXYradfromCamera" gets called, it first calls the function "getCameraFrame" which loads the current frame from the camera and returns it in Mat format. The "getXYradfromCamera" then calls the function "getEgdeImage" which mirrors the input Mat, splits it into red, blue and green channels, blurs the red channel to smooth the edges and returns the modified Mat. Next, the "getEyeCircle" function gets called which first loads the HoughConstants structure with five float values and calls the OpenCV function "HoughCircles" with loaded parameters from "Settings" tab from "EyeTrackApp". The *X* and *Y* coordinates of the found circle as well as the radius of the circle are returned in EyeStruct structure. The "getXYradfromCamera" then returns the EyeStruct to the calling function. Because the algorithm works with one set of values at a time, it is necessary to call this algorithm in a loop. It is also necessary to initialize the camera before starting the loop to avoid constant turning on and off of the camera.

After the eye tracking algorithm returns the circle coordinates, calling the function "drawCircleInImage" may be called in the same loop. The function will flip the camera frame and draw the found image in it. It will show a window with the camera capture with the circle drawn inside.

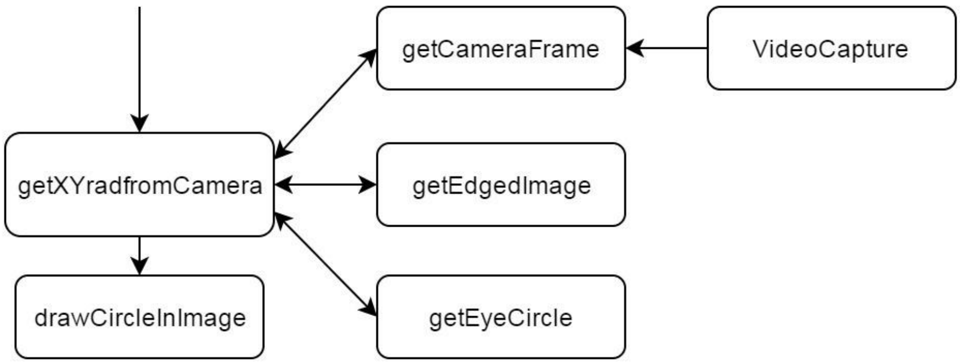


Figure 2.18 Eye Tracking programme algorithm

The "Calibration Process" (see Figure 2.19) is created to connect the coordinates of the eye centre from the "Eye Tracking Process" with the coordinates on the testing screen. After it gets called from the "EyeTrackApp", the new CLR Windows Form object initializes. As it initializes, the loop with eye tracking algorithm and circle drawing starts, while simultaneously saving the *X* and *Y* coordinates of the found circle, rewriting them with each round of the loop. In the meantime, nine buttons appear distributed on the testing screen (see Figure 2.20). The subject must focus his sight while simultaneously clicking the watched buttons one by one. As the subject clicks each button, the current coordinates of the *X* and *Y* of the circle are saved into the calibration file along with the clicked button coordinates on the screen. This enables the calibration to function on any screen resolution. Before the calibration, the camera needs to be attached in horizontal position and aimed at the eye of the tested subject. It is recommended to check this by using the “START” button in “Settings” tab of the “EyeTrackApp”.

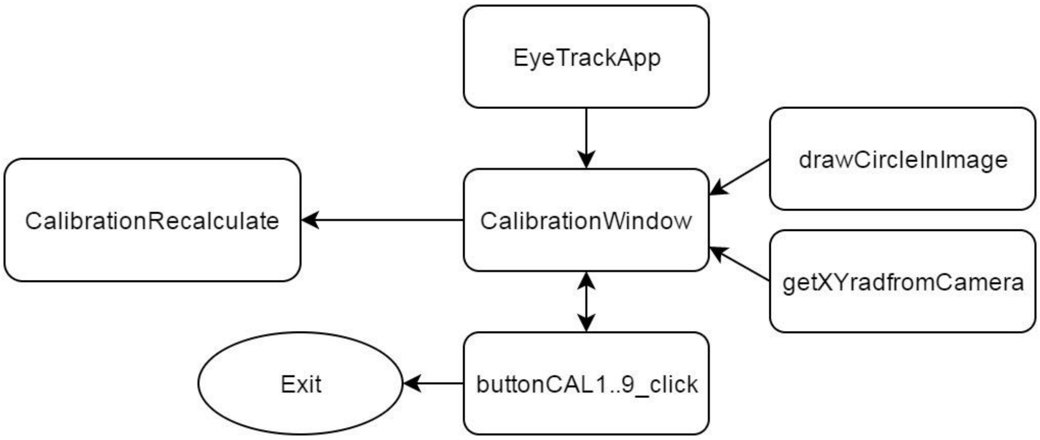


Figure 2.19 Calibration programme algorithm

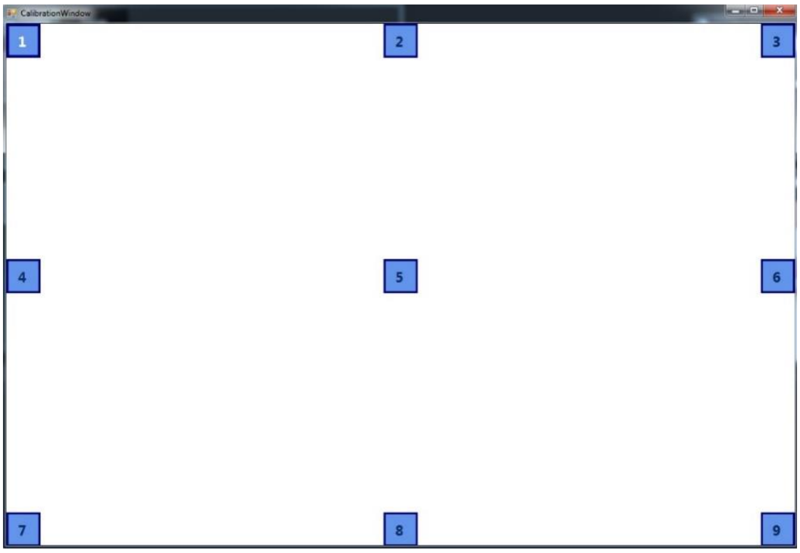


Figure 2.20 Calibration window with nine buttons

After the calibration process is finished, the “CalibrationCalc” (CalibrationRecalculate) function can be called to calculate the constants for point of regard finding. The function calculates four constants, two (calX and calY) to connect the [0;0] coordinates of the screen with the coordinates of the same point from the camera frame. To calculate the calX constant, the six values are calculated, each being a horizontal distance of the two horizontally neighbouring buttons divided by the distance of centres found by clicking the same buttons. The calX constant is then calculated as the average of those values. The calY constant is calculated in the same fashion, only the distances are of every two vertically neighbouring buttons. The other two constants (x and y) carry the information of how many pixels on the screen in each direction equals one pixel in the same direction in the camera frame. The *x* constant is calculated as a sum of three X values of most-left circle centres gained by clicking the left column during the calibration divided by 3. The y constant is calculated as a sum of three Y values of most-top circle centres gained by clicking the top row of the buttons divided by 3. With those four constants, the application is able to reliably assign each measured eye circle to the point on the screen, gaining point of regard.

After the calibration is completed, it is recommended to start the “Calibration Check”, which is the new CLR Windows Form object, where the real-time tracking algorithm is initialized. The object contains a rectangle drawn on the screen with “EXIT” button (see Figure 2.21). The tested subject watches the rectangle while simultaneously the calculated point of regard shows on the other screen in real-time.

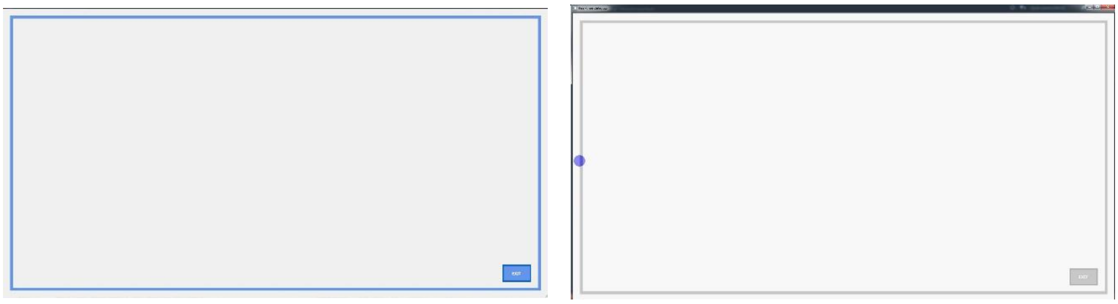


Figure 2.21 Calibration Check window (left) and its evaluation window with detected point of regard (purple dot in right window)

The form calls “RealTime” function which loads the calibration constants along with the testing screen background. It then resizes the loaded background image to match the size of the image on the testing screen. The algorithm then initializes the loop with eye tracking algorithm. It recalculates the detected eye centre coordinates with calibration constants and with each loop saves eye centre coordinates and recalculated coordinates to the text file. The algorithm then draws the recalculated circle to the empty matrix of the same size as the image and adds the two matrixes (= images) together. At the end of each loop, the window with the viewed picture with drawn point (circle) of regard is shown (see Figure 2.21). This way the test supervisor is able to tell if the calculation was accurate or whether the calibration process must be repeated.

2.5 TESTING PROCESS

This chapter describes the testing part of the application, the testing methods and possibilities. The tests are designed not only to measure the test subject, but also test the tracking system capabilities and precision. There are six tests in total, each focuses on a different aspect of the application and test subject’s qualities.

The first test consists of a picture of two dogs, which the measured subjects are watching. The test shows, which part of the picture was the most attractive to the viewer. The test number two is the alternative version of the test one. It is an image consisting of several pictures. The test evaluates which of the individual pictures attracts the eye the most. Testing images num. 1 and 2 may be found in Figure 2.22.

Figure 2.22 Testing images of test 1 (left) and test 2 (right) [21]

The third test is a reading test, in which the subject is supposed to read a short text. The test evaluation shows the path, in which the subject was looking at the text while reading. Reading example is to be found in Attachment num. 1.

The fourth test is the application called Find the Panda. It is supposed to evaluate the perception of the tested subject. The subject must find the panda hidden in the picture, which consist of the field filled with different objects. The test consists of four rounds, each of them measured by the timer. The tested images are captured in Figure 2.23.

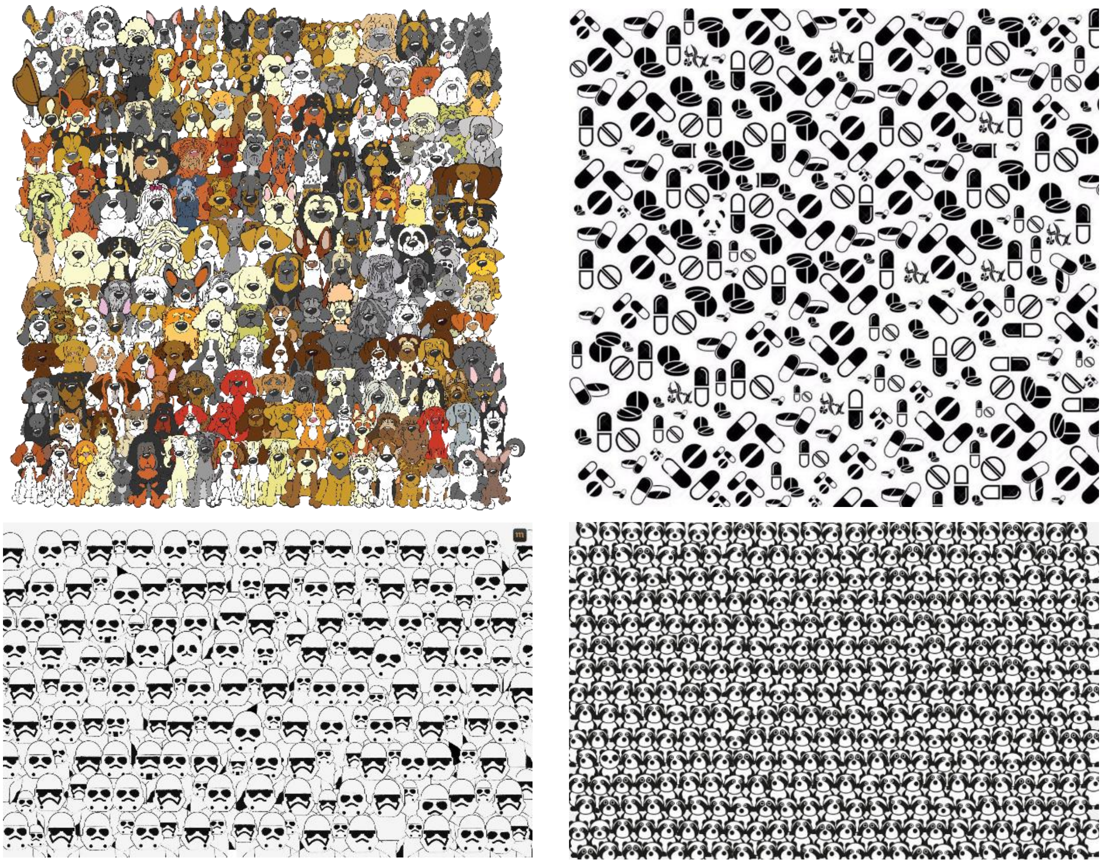


Figure 2.23 Testing images of test 4 [22]

Test number five is designed to evaluate the inaccuracies by showing the static dot in the middle of the testing screen. The subject focuses on the dot, the measured result is evaluated by heat map and average and maximum deviation from the dot centre are measured by the testing application.

In the last test called “Watch the Dot” the moving dot appears on the screen while the subject must follow it with his eyes. The calculated point of regard appears on the screen in real-time. The path of the dot movement is saved in included text file and recorded in image used for evaluation. It can be found in Attachment num. 1.

The algorithm of tests 1, 2, 3 and 5 is almost identical. By selecting "TEST 1", "TEST 2", "TEST 3" or "TEST 5" in the testing combo box and clicking the “START TEST” button, the “CoordinationGetter” function starts with the number input accordingly to the selected test. The function then loads the calibration constants, opens the file to save the measured results to and loads the appropriate testing image by calling the “ImageGetter” function. The algorithm then initializes the loop with eye tracking algorithm circle drawing function and saving the eye and screen recalculated coordinates each loop. The test is ended by hitting the “q” (quit) key on the keyboard. It is possible to start the tests with real-time drawing function by clicking the “Real Time Detection” button. Real-time testing using the “RealTime” function works in the same way as “Calibration Chceck”, loading different pictures and opening different files to save the measured results according to the chosen test (see Figure 2.24).

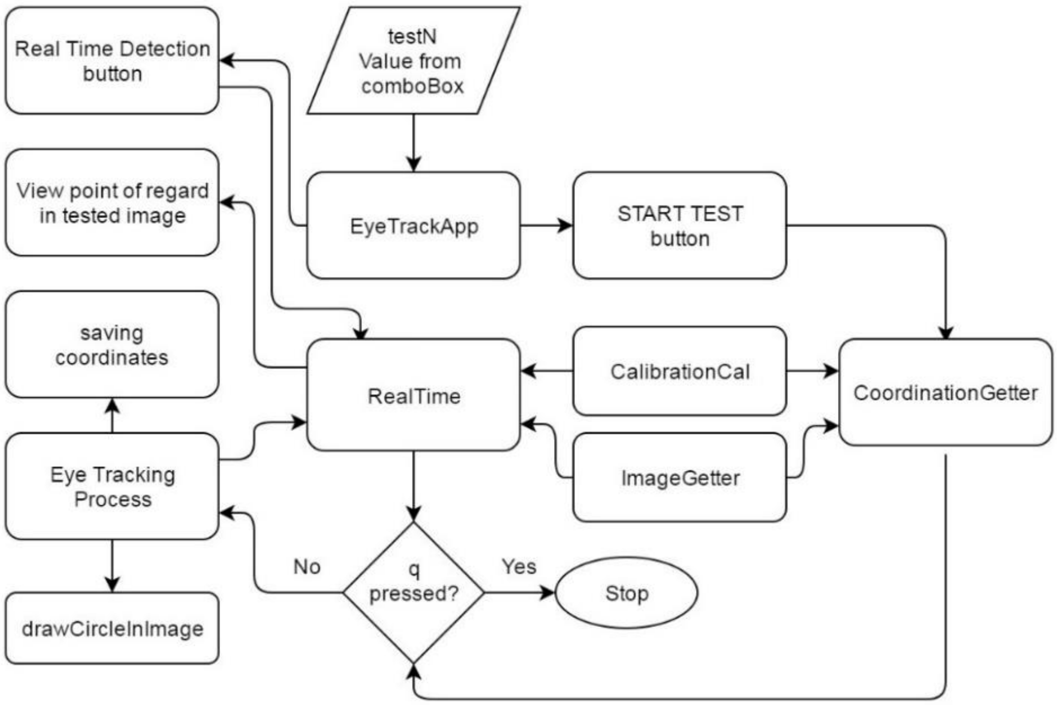


Figure 2.24 Testing algorithm for test 1, 2, 3 and 5

The algorithm for test 4 is created as a new CLR Windows Form object. By selecting “TEST 4” and clicking the “START TEST” button, the new window appears. The window contains the “START” and “EXIT” buttons, four labels, and the testing field. After clicking the “START” button, the timer starts and the first image loads. The objective is to click on the invisible button on top of the panda. This also starts the

“CoordinationGetter” function with proper input value. After the test subject finds the panda and clicks on the transparent button on its head, the timer resets and the “CoordinationGetter” function stop. The image and the transparent button become invisible and the next testing picture with another transparent button appears. The value of first test’s time appears in the first label. The timer and the newly called “CoordinationGetter” function starts. The test consists of four pictures with hidden pandas and after finding all of them the measured times are displayed.

The “Watch the Dot” test initializes by clicking the “Watch the Dot” button. The algorithm opens the text file with saved coordinates of the moving dot and another text file to save the point of regard’s coordinates into. Then the calibration constants are loaded. The “for” loop reads the first file with prepared dot coordinates until the file ends. In the loop, the algorithm draws the circles of the prepared dot to the empty matrix, calls “getXYradfromCamera” function, calculates the point of regard and saves it to the second text file. It also draws the detected circle to another empty matrix and adds the two matrixes together. With every loop, it shows the image with two dots, one which the subject is watching and the other one with the calculated point of regard.

The testing application was originally designed for two monitors measuring. It is possible to use it even with one, but the measuring is more accurate with two since the test evaluator can control the eye capturing at all times. It also allows the evaluator to assess the testing in real time. While measuring the testing that is described in attachment num. 1 of the bachelor thesis’s, two identical 24 inch monitors were used. They were both EIZO EV2450 monitors with Full-HD resolution, connected to a single notebook via HDMI. The used notebook runs on Windows 7 and runs testing application.

It was also ensured that the testing subject could not see the screen of evaluator, since it could attract his eyes and cause measurement inaccuracy. The position described and used is captured in figure 2.25 and figure 2.26. They capture the testing of subject num. 1 while taking test 1 during the daylight conditions while measurement is evaluated by heat map replay (figure 2.25) and real-time tracking of test 1 under artificial light conditions (figure 2.26).

Figure 2.26 also includes additional light source (top right corner), which was used while measuring of testing subject num. 3, who wears glasses with reflective glass surface (see attachment num. 1, chapter 2.1.4). For the rest of measurement only one light source was used. It was a ceiling light fixed to the ceiling so the artificial light was coming from behind of the subject. During the day light conditions the only source of light was a window on the left of the subject (see figure 2.25). In spite of that fact, no major difference between the measuring results while testing under different light types was recorded, it is possible to use any of the setups.

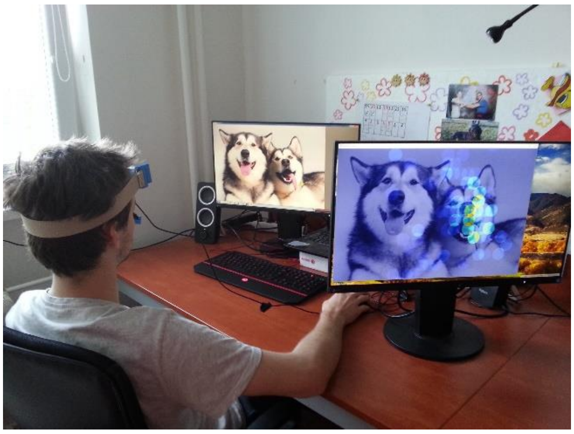


Figure 2.25 Scene setting - daylight conditions

Figure 2.26 Scene setting - artificial light conditions

To run this application, the computer needs Visual Studio software installed. It was created on Visual Studio 2013 version, because the newer versions of the software do not support toolbox function in default. Furthermore, the OpenCV 3.1 libraries must be added to the Windows environment variables and added to the Visual Studio project. The OpenCV libraries can be downloaded on the website www.opencv.org, which contains the manuals of how to set them up as well. There is also a guide of how to implement OpenCV into the Visual Studio project.

After Visual Studio 2013 and OpenCV is successfully installed, the project can be opened. To successfully run the script, go to Project Properties -> Linker -> System, under the subsystem option select Windows(/SUBSYSTEM/WINDOWS). Go to Linker -> Advanced -> and choose “EyeTrackApp” as the Entry point. Choose the Debug compiler. The script is executable now and ready for testing or modifying.

In case that there is another camera built in the computer that is not supposed to be used, it is convenient to shut it down before starting the application. In case of starting the application with another camera connected to the computer it is possible, that the program will choose the other camera as a primary one and use it instead of the camera attached to the camera holder. In this case, the user should press exit button in the application, disconnect or shut down the other camera and load the application again.

2.6 EVALUATION PROCESS

The process of evaluation is the part of the software, which presents the test outputs of the application. The application has several test output methods, each of them is described in this chapter. They are created to thoroughly evaluate the testing process, focusing on different aspects of the measuring.

The first evaluation option is Connect the Dots evaluation. The output of this evaluation is the original picture with calculated points of regards connected with drawn lines between two following points. This evaluation shows the track, which the eyes of the subject travelled on the screen. The example test output is shown in Figure 2.27.

The evaluating algorithm, depending on the function input value, opens according text file with saved data from measuring to read the point of regard's coordinates from and loads the image of the according test. In case the text file or the image was not found, the error message appears. The application then resizes the image to match the testing dimensions, converts the image to greyscale and then back to RGB format. This makes the tested image greyscale while having three colour channels. This enables the software to add it with other three colour channel images, as well as drawing in them in colour. This part of the algorithm is the same for every evaluation type (marked by yellow in the following algorithms).

Evaluation by Connect the Dots then reads the first line of the text file and saves the coordinates of the point of regard's positions to "lastX1" and "lastY1". The algorithm then starts the "for" loop by reading the first saved value of the new row (which represents the *x* value of the detected eye circle) until the EOF (end of file). The algorithm then loads the remaining values from that row, saving the values of the detected point of regard to "lastX2" and "lastY2". The programme then draws the line between [lastX1; lastY1] and [lastX2; lastY2] coordinates to the previously loaded and adjusted image. Then the "lastX2" and "lastY2" coordinate values are saved to "lastX1" and "lastY1" and the whole loop repeats. After the application draws the last line, the result image with lines drawn in it is shown. This evaluation's algorithm is presented in Figure 2.28.

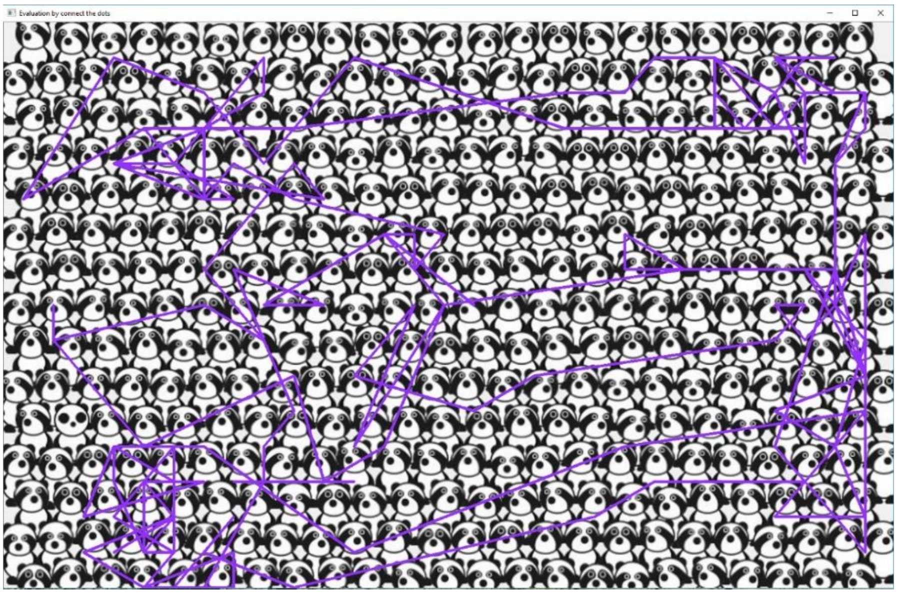


Figure 2.27 Evaluation by Connect the Dots example

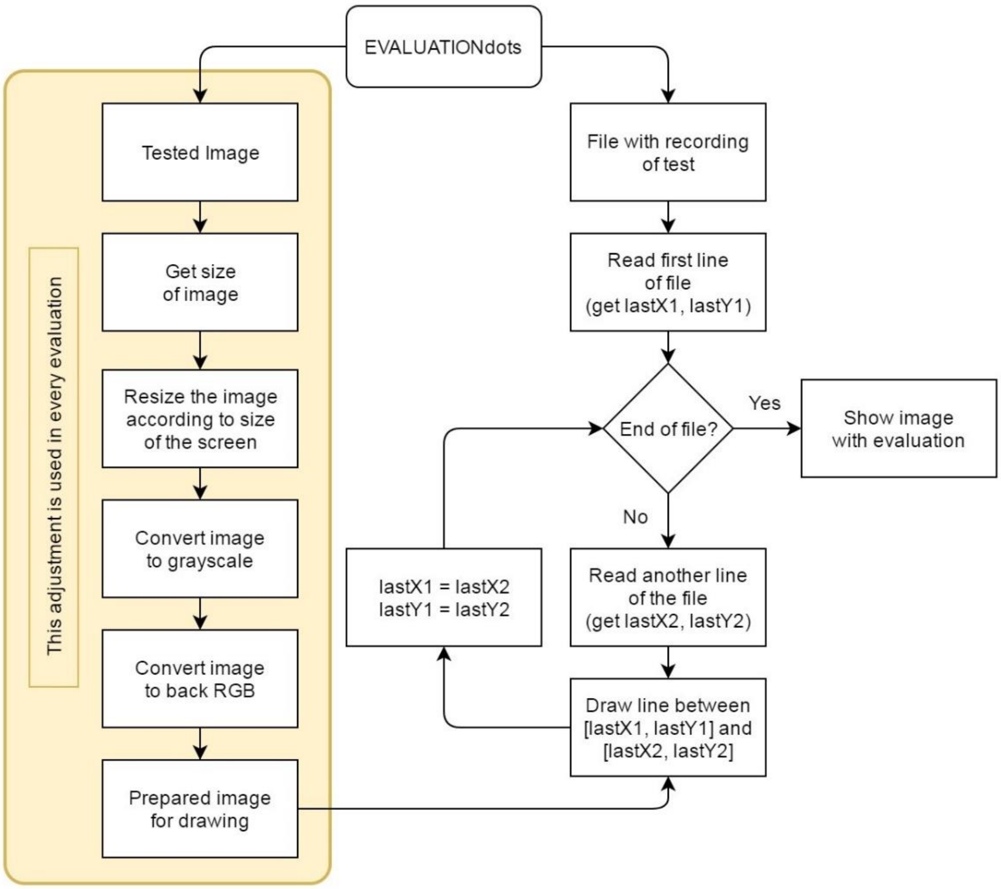


Figure 2.28 Evaluation by Connect the Dots algorithm

The second evaluation option is called Heat Map. It highlights the areas in the image with varying intensity, depending on the number of views by the subject. The algorithm and the example are shown in the Figure 2.29 and Figure 2.30, while evaluating test 2.

The algorithm of this evaluation also depends on the function input value, which opens according text file with saved data from measuring to read the point of regard's coordinates from and loads the image of the according test. The application then adjusts the loaded image the same way as in "Connect the Dots" evaluation (marked by yellow in the algorithm). Next it creates a ”zero” matrix of the same size as the adjusted image named "map". The algorithm then starts the "for" loop by reading the first saved value of the new row (which represents the *x* value of the detected eye circle) until the EOF (end of file). The algorithm then loads the remaining values from that row, saving the values of the detected point of regard to "lastX1" and "lastY1". It draws the circle with the centre at those coordinates to mark the measured point of regard into the new empty matrix called "iter". The algorithm then adds "iter" to the "map" matrix. This process repeats with all measures points of regard while clearing the "iter" matrix in every round of the loop. After the loop ends, the "map" values are normalised to occur the whole range of colours after the colour map is applied. The normalised "map" is then added with the previously adjusted image in 1:1 scale. The output of the algorithm is a single image with the colour map on the tested image background.

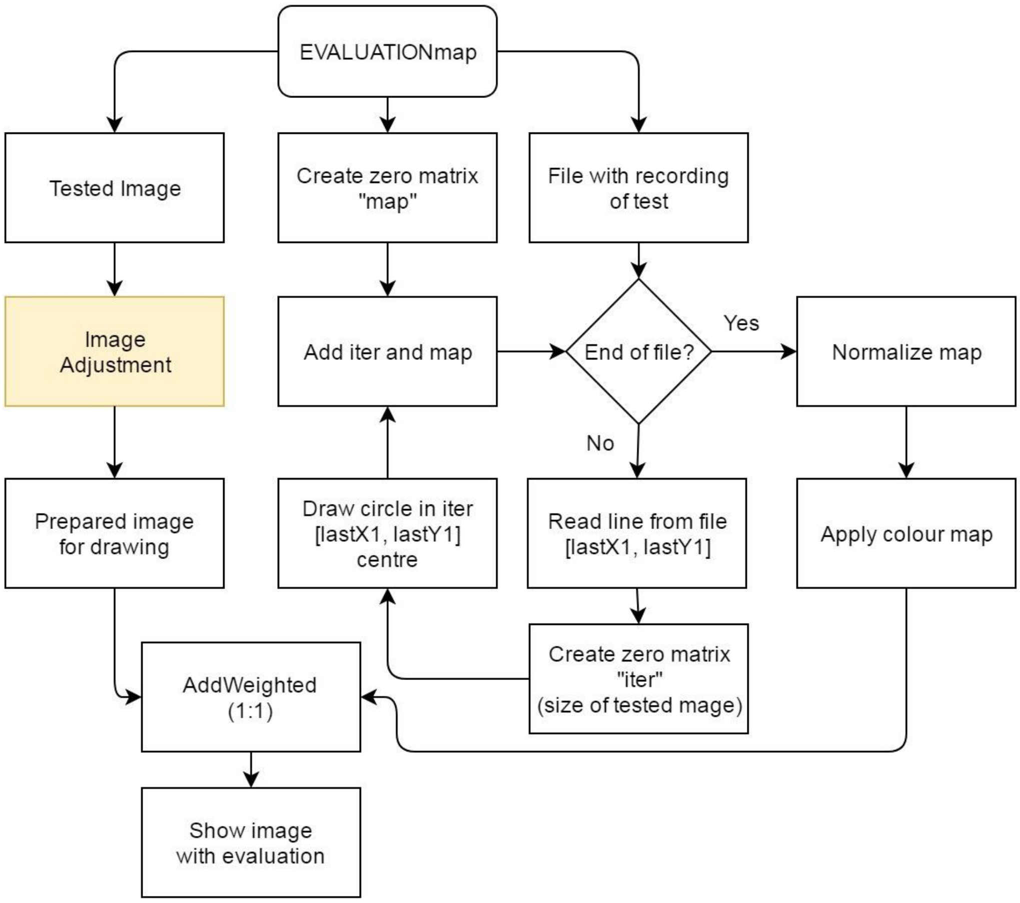


Figure 2.29 Evaluation by Heat Map algorithm

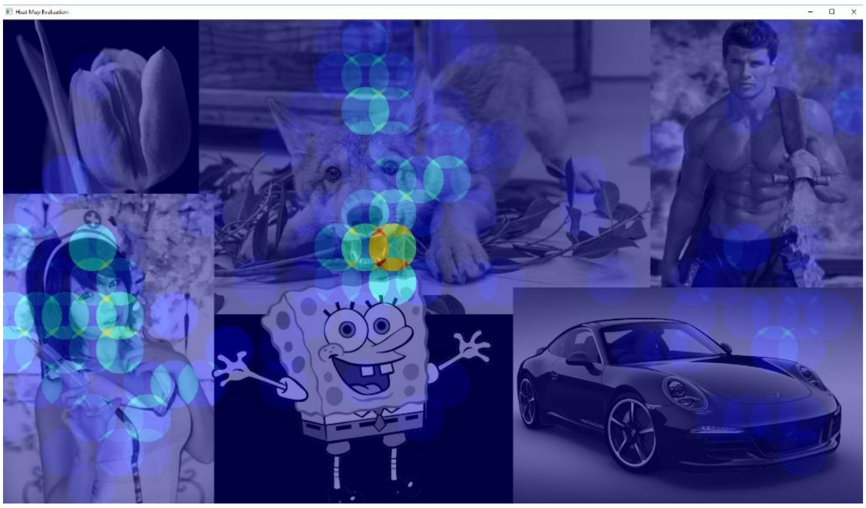


Figure 2.30 Evaluation by Heat Map example

The next evaluation method is called Dot Replay. It replays the tested image with dot of the measured points of regard shown over time. The algorithm of the evaluation is shown in Figure 2.31. The example of Dot Replay may be found in Figure 2.32, while evaluating test 1, while the subject was looking at the dog's left ear. The algorithm is very similar to the algorithm of previous evaluation. After the same adjustment of loaded image, it opens the according file with saved test's recording and begins similar loop. In this loop, it also reads "lastX1" and "lastY1" coordinates from file, draws small dot with centre in these coordinates, which is also saved into "iter" matrix. The "iter" matrix is then added with previously adjusted image in 1:1 ratio and shown on the screen. This process repeats until the end of file, which stops the animation.

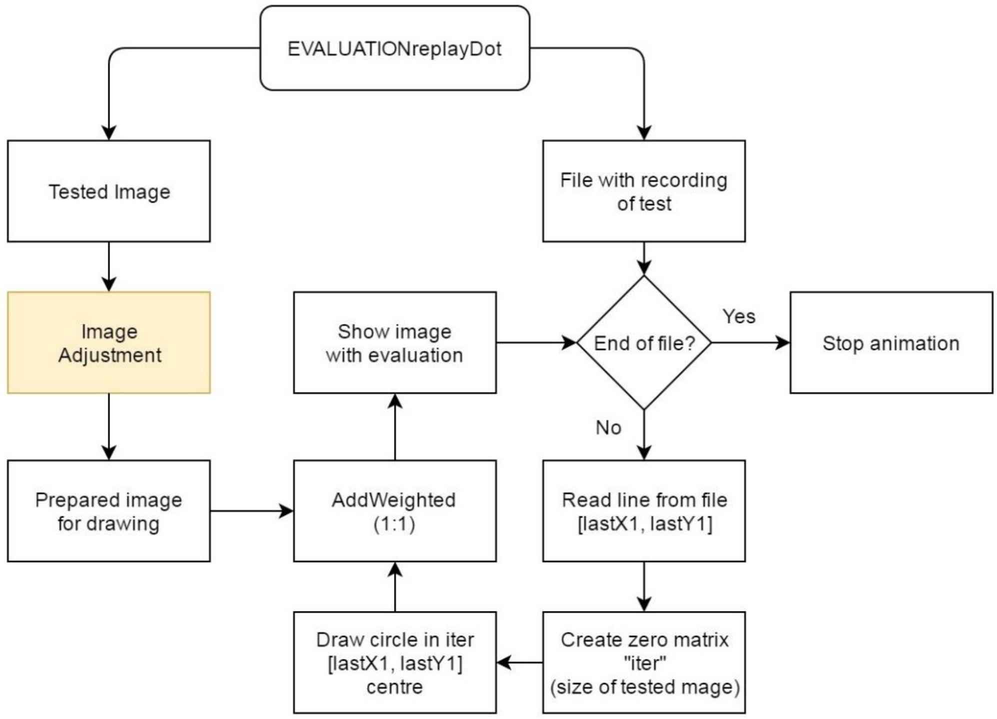


Figure 2.31 Evaluation by Dot Replay algorithm

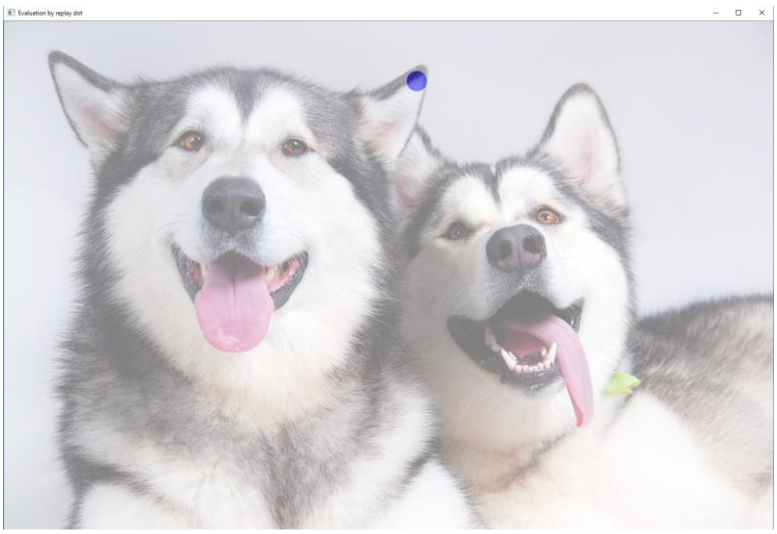


Figure 2.32 Evaluation by Dot Replay with detected example (dot on the ear)

The last type of evaluation is called Map Replay evaluation. It generates the heat map from measured points of regard and shows it over time, as the new points were measured. The algorithm is described in Figure 2.33. The example of replayed evaluation may be found in Figure 2.30, since its replayed evaluation is the same as Heat map evaluation.

Also, this algorithm is very similar to the algorithm of "Heat map evaluation". After the same adjustment of loaded image, it opens the according file with saved test's recording and begins similar loop. In this loop, it also reads "lastX1" and "lastY1" coordinates from file, draws dot with centre in these coordinates, which is also saved into "iter" matrix. The "iter" matrix is then added with previously created "zero" matrix called "map", which is then colour mapped and added to previously adjusted image in 1:1 ration and shown on the screen. This process repeats until the end of file, which stops the animation.

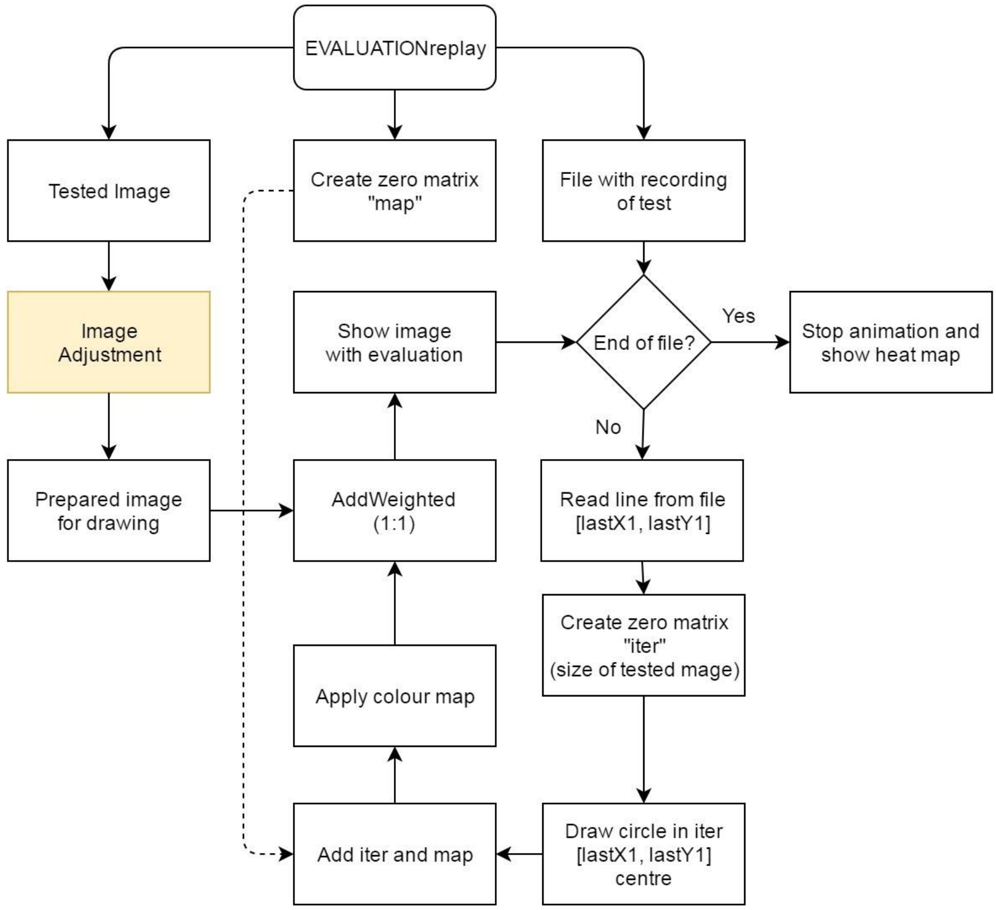


Figure 2.33 Evaluation by Map Replay algorithm

While evaluating the test 5 using Heat Map evaluation, the deviation error calculation is included. The first measurement error calculated gives an average deviation in pixels (see Equation 2.4). It equals sum of the deviations of each measuring from the screen centre divided by number of measurements in both *x* and *y* directions. The average deviations in *x* and *y* directions are then recalculated by Pythagoras sentence.

$$acc_{\mathrm{PX}} = \sqrt{\left(\frac{\sum_1^n |x_0 - x_i|}{n}\right)^2 + \left(\frac{\sum_1^n |y_0 - y_i|}{n}\right)^2}$$

(Equation 2.4)

where x_0 and y_0 are coordinates of the centre of the drawn point in test 5, x_i and y_i are coordinates of calculated point of regard of the i measuring and n is the total number of measured points.

The second measurement error gives an average deviation in millimetres (see Equation 2.5). The calculation is similar to the previous one, adding the x and y values of pixel to millimetre ratios.

$$acc_{\mathrm{MM}} = \sqrt{\left(\frac{\sum_1^n |x_0 - x_i|}{n} \cdot \frac{sx}{px}\right)^2 + \left(\frac{\sum_1^n |y_0 - y_i|}{n} \cdot \frac{sy}{py}\right)^2}$$

(Equation 2.5)

where sx and sy are the dimensions of the screen in millimetres and px and py are the dimensions of the screen in pixels (resolution).

The third and fourth error give maximum deviation value from the measurement in pixels (see Equation 2.6). First, the measuring furthest from the centre is found by comparing all the individual measurement's deviations gained by Pythagoras sentence of the x and y values. The maximum distance values are then used in the equation below.

$$error_{\mathrm{max-PX}} = \sqrt{(x_0 - x_{\mathrm{max}})^2 + (y_0 - y_{\mathrm{max}})^2}$$

(Equation 2.6)

where x_{max} and y_{max} are the coordinates of the furthest point from the centre coordinates.

The last deviation is the value of the maximum deviation in millimetres (see Equation 2.7). It combines the previous calculations with the ratio from Equation 2.5.

$$error_{\mathrm{max-MM}} = \sqrt{((x_0 - x_{\mathrm{max}}) \cdot \frac{sx}{px})^2 + ((y_0 - y_{\mathrm{max}}) \cdot \frac{sy}{py})^2}$$

(Equation 2.7)

The calculated deviations is to be displayed in text boxes in "EyeTrackApp" main window under the "EVALUATE HEAT MAP" button (see Figure 2.34).

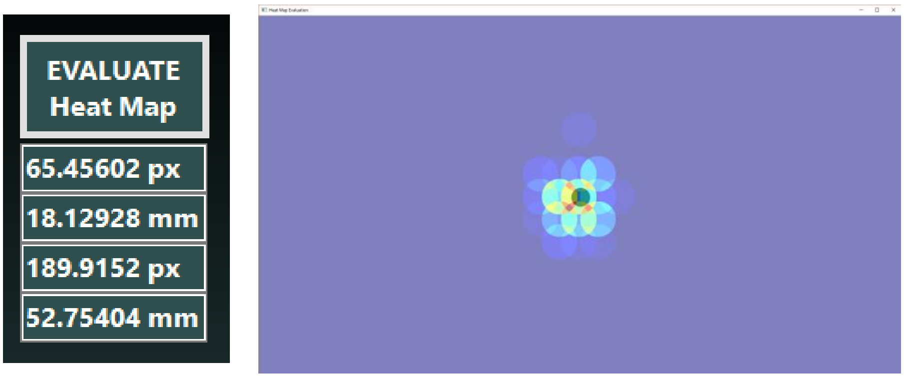


Figure 2.34 Example of evaluated inaccuracy (left) for its according measurement (right)

3 MEASUREMENT RESULTS

In this chapter, the measuring that was performed to evaluate the eye tracker's accuracy and functionality, will be assess. It will also give an overview of advantages and disadvantages of the used method, including its limitation. The detailed report on the mentioned measurement is attached to this thesis as an attachment num. 1. It is called "Measurement and Its Results".

The first part of reports included in the attachment gives five examples of full measurement taken under different circumstances. Full measurement consists of all tests available in the testing application (see Attachment num. 1, chapter 2.1). Number one and two measurements were performed on same testing subject, the first taken during the daylight conditions and the other during artificial lighting. The next two examples present measurements with glasses. The first kind of glasses are non-reflexive, the others are highly reflexive. The last measurement is performed on test subject wearing contact lenses. Each measurement of full test was performed on 9 subjects, on some of them repeatedly. Records of remaining tests are summed up in shortened reports (see Attachment num. 1, chapter 2.2). To evaluated system's accuracy, multiple measuring of test 5 was performed. From that, the average inaccuracy and maximum error was calculated.

The average inaccuracy of testing was calculated to be 26.8 mm (which matches 95 px on the testing screen). The maximum error of testing is 60.5 mm (218 px) in average. The average inaccuracy of testing of subject with no special conditions (no glasses or lenses) was measured as 22.3 mm (80 px) and maximum error as 54.0 mm (194 px). Average inaccuracy of testing with nonreflexive glasses was 30.6 mm (104 px), maximum error was 61.1 mm (220 px). For the testing of lenses was determined as 26.5 mm (96 px) for average inaccuracy and 57.4 mm (207 px) for maximum error.

Although the inaccuracies may seem relatively high, the human eye does not focus on a single point but it detects large area which leads to the conclusion, that the evaluating system evaluates the point of regard correctly.

This is a valid conclusion only if the testing was proceeded correctly and none of the testing requirements were broken. As proved by testing measurement (see Attachment num. 1), the testing process meets several requirements and limitations.

The lighting of the room in which the measuring takes place is an important factor. The human eye reflects light very easily, making every light source a fault cause. In case of unsuitably placed light source, light would reflect in the eye of the subject and could cause the program to detect reduction of circularity of the eye, making it impossible to detect even for Hough transform that requires neither full nor perfect circle. This problem was solved during the measurement by placing light sources into certain positions (see Chapter 2.5). This way no light source faced the eye directly.

This factor is even more limiting when the tested subject wears glasses. The testing performed to compare influence of glasses is assessed in Attachment num. 1, chapter 2.1. As a result of glasses with reflective glass surface, it was hard to complete the calibration process. The light from monitor reflecting in glasses caused many false circles. The situation was solved by reducing the brightness of the screen and retaking the calibration part. Nevertheless, this measuring was least precise of all recorded measuring. Testing of the subject wearing glasses with non-reflexive surface was more accurate and did not require retaking the calibration. The measurements were also acquired with subjects using contact lenses. Also in this case, no major complication occurred. For the example of detection of the iris on the eye with no eye condition and the detection with non-reflexive glasses see Figure 3.1.

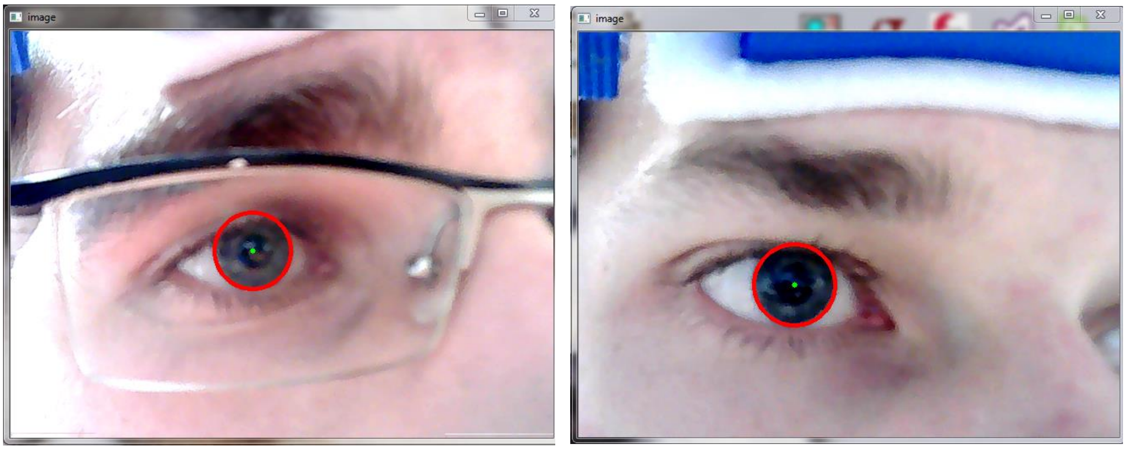


Figure 3.1 Example of iris detection with non-reflexive glasses (left) and no eye condition (right)

The geometry of the eye together with some eye movements prevent the system from detecting the eye. If the subject looks excessively toward the nose, a significant area of the iris becomes less visible for the camera located at the outer side of the face causing the algorithm to have no circle to look for. Looking down hides a part of the eye behind the eyelid, confusing the software to imprecisely locate the centre point of the eye. This problem also occurs while measuring the subject with strongly faded eyelids (see Figure 3.2). The significant part of the iris remains hidden even if the subject looks straight in front of him. This makes the detection in the fringe positions impossible, which disables the calibration process and therefore the testing.

Figure 3.2 Example of unsuitable eye shapes

The camera holder is not perfectly stabilized on the head of the subject, it can slightly slide down the forehead generating the minor error when computing the point of regard. Any head movement shakes the whole construction resulting in slight trembling of the captured image.

The most significant error occurs, when the tested subject moves his head (see Figure 3.3). The camera capturing the eye does not recognise that movement, because it moves with the head and does not capture the surroundings. This movement lessens the value of the calibration and therefore after the significant head movement it is recommended to repeat the calibration process.

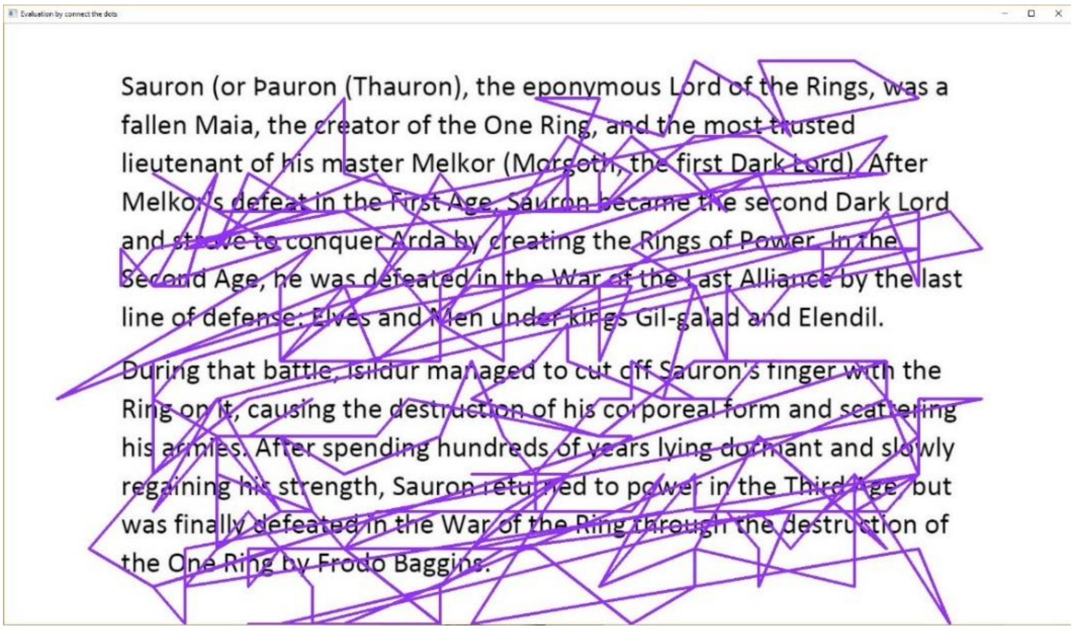


Figure 3.3 Detection error caused by head movement

Another limiting factor is the camera used in the realization of this project. While it has satisfying results for functionality verification of this method, the common web camera is not constructed for capturing the image at close distances. Its resolution and frame rate also reduce the precision of this method significantly. The camera is the source of noise and blur as well, both are being suppressed in our algorithm, but no filter can get rid of 100% of these factors.

The camera's inability to focus on the close iris generates sudden errors. While "HoughCircles" function of the OpenCV draws the circle around the most significant edges of the image, the imperfectly focused picture does not contain the sharp edge around the iris. This may result in detecting the circle imprecisely, which later generates the error in point of regard finding process. The even greater error occurs, when the error happens during the calibration process, which results in imprecise calibration. The error is cumulative with other image imperfections, for example light reflections or hiding the iris behind the eyelid. The example of faultily detected circle inside the eye is shown in Figure 3.4 together with responding inaccuracy caused by it.

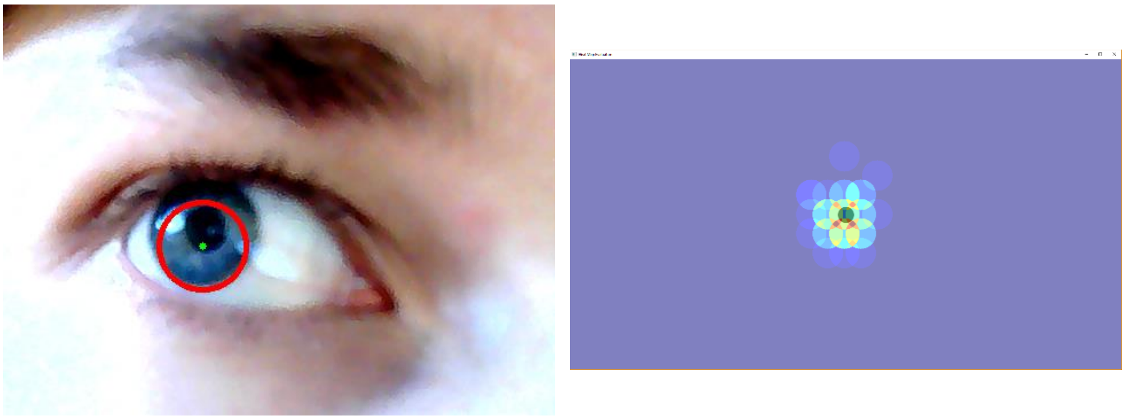


Figure 3.4 Faultily detected circle due to the imperfect focus of the camera and light reflections (left) and inaccuracy caused by it (right)

To sum up all the evaluated measurements acquired from the tests, the comparison and analyses were made in Attachment num. 1. The test num. 1 and 2 was evaluated by heat map as it gives the best information about the most attractive areas of the picture. For the test num. 1, the most attractive areas were the faces of the dogs. In test num. 2 the most attractive areas were the puppy followed by the nurse. The evaluation of the test num. 3 was done by the connect the dots. The results show, that tested subjects read the whole text, but the eye movement tracking system is not optimised for reading evaluations, as the point of regard calculation is not optimised for micro movements of the eye. The system works reliably for area marking, such as in test 1 and 2 or for real-time tracking evaluated in watch the dot test as well as in test 4. The advantage of this system is also the possibility of evaluation by replay. In test 4 the solving of the first part (panda between dogs) took on average 20.1 seconds, which is the least of all tests. The most challenging was the fourth test (panda between raccoons), which took on average 35.6 seconds. The test results also show, that the realised eye tracking system is blind to the eye colour factor, which means that is works equally well with any iris pigmentation.

The advantages of this eye tracking system lie in its non-invasiveness, very low price and light and durable construction of the camera holder. The system is easily adjustable for different cameras thanks to the option of changing the Hough parameters and universal camera attachment place on the holder. Although, the main advantage of used camera is its adjustable base, which allows the testing coordinator to precisely aim the camera in the horizontal position. Another plus side of the system is the application. It contains multiple testing methods with several different testing outputs, optional settings and intuitive controls. It is possible to extend the application in the future.

4 FUTURE EXTENTIONS OPTIONS

There are several possible improvements and ideas that should be mentioned in this thesis as a thinkable adjustment to the currently used solution. They are summed up in this chapter together with some descriptions of currently limiting issues.

4.1 ALTERNATIVE CAMERA OPTION

Although the camera that is being used in current solution is functional and works fine as the proof of the concept with satisfying results for functionality verification of this method, it has several characteristics the change of which could improve the whole solution.

The camera chosen for the testing has sufficient resolution and frame rate for the method functionality verification, but for more precise measuring a specialized camera would be more suitable. The camera currently used by this system, type A4Tech PK910H, was available, so there was no need to purchase a different one. Because it is a web camera, it is not able to sharply focus on the eye at such a short distance (9 cm from the eye), the video capturing causes significant noise interference. While the camera supports FullHD resolution with 30 frames per second according to its datasheet from its manufacturer, it is not able to work on those two parameters simultaneously. This fact forces the solution to choose between higher resolution and higher frame rate. In used solution, the higher frame rate was chosen as the lower frame rate (only 15 frames per second) resulted in high time delays between measurements. Unfortunately, lower resolution caused higher inaccuracy and deviation of the measurement. The fewer pixels the video contains in every processed frame, the higher the deviation of slightly faulty detected circle. As the detection of circle is moderately trembling, mostly because the camera is not able to focus on the iris and because it has low camera resolution, the recalculated point of regard on the screen is trembling significantly more. This is because of the significantly higher resolution of the screen in comparison to the resolution of the camera. One last disadvantage of PK910H is that the LED light inside the camera could not be turned off, so the transparent parts of the camera excluding the capturing zone were covered with black tape, so that it did not bother the testing subject and did not cause any light reflection in the eye, making it harder to detect the circle around the eye.

Replacing the currently used camera with a camera with micro focus would enable the focused capture of the eye. The camera with higher resolution and framerate would ensure higher accuracy in point of regard calculating and higher smoothness of the testing recording. What is more, a smaller size of the camera would reduce the blocking of vision of the tested subject while wearing the camera holder.

Purchasing the other kind of camera would be convenient only if the measurement precision requirements were higher than those in the testing that was performed as a part of this theses, where the testing output in the way of heatmap, showing that the subject was observing dog’s muzzle, is sufficient.

Additionally, the main advantage of used camera is its adjustable base, which allows the testing coordinator to aim the camera precisely in the horizontal position. Another advantage is, even though we did not pay for the camera since it was already available at school, that the price of PK910H is especially low, which makes the solution very cheap, providing it would need to be redone.

4.2 ADDITIONAL CAMERAS OPTIONS

The choice of the second camera scanning the other eye was a possibility for the produced system, but the advantages like more precise calculation possibilities in some eye positions were outweighed by the acquisition price, calculation complexity, asymmetric eye capturing problems, the additional weight for the camera holder and parallel video input issues. Despite those complications it is an option, that could improve the created eye tracking system.

Using more than one camera would be a great advantage and possibly or even a must in case of a need of detecting the point of regard while observing something else than a computer screen. In this case, another camera would be added on the top of the camera holder to capture the scene in front of the subject and the measurement output would be a video capture from the frontal camera with point of regard detected in it. This would also simplify solving the head movement problem by capturing the view in front of the subject which would tilt together with the head, meaning it would keep its inner reference frame by which it could detect the roll, yaw and pitch of the head and lead to recalculation of head's orientation in space. For example, the usage of currently used PK910H for this purpose would be very suitable while using different camera for eye capturing could lead to more precise solution, usable even outside, not only while sitting in front of computer.

The other camera would need to be significantly more precise in order to occur detection detailed and accurate enough to detect every change in pupil's size. Otherwise this solution would be useless in exterior measuring, since the pupil changes its size while focusing on an object closer to the viewer in comparison to the size it has while looking in long distance. In this case, the system would not recognise at what distance the measured subject is focusing. Also, the cameras would require to be fixed in steady position to each other, so the base of PK910H would need to be anchored permanently to the same holder as the other non-movable camera to prevent its reference frame from shifting apart. In case of any shift of this kind, the obtained results would be inaccurate.

Another option of two or more cameras usage would be placing all of them to capture the same eye. In such case, every capture of the eye could be processed separately the same way as the current system processes it and then they would combine to one single output. This kind of solution is frequently used in commercially accessible eye trackers, such as Tobii glasses, which use four extra small cameras [10]. In case of solution of this project which is using the web camera, this would lead to adding a lot of extra weight on the holder, that would be complicated to balance and would not lead to any simplification of the measuring.

4.3 INFRARED LIGHTING OPTION

Another issue to address are light reflections from the eye captured on the camera causing inaccuracy of the measuring. Homogeneous light in the measuring room would significantly reduce those reflections. During the measurement, the light source was placed behind or next to the subject (see Chapter 2.5). This reduced almost all the reflection caused by light source and the only reminding reflections were from the testing screen. Another option would be using an infrared or near infrared camera or infrared additional lighting.

The first of the two other possible solutions, would be to use a different type of camera, for example the one with macro focus, an infrared or near-infrared camera that is able to focus on close ranges. Using the infrared or the near infrared camera would help the measuring precision significantly. The pupil seems clearly black while observing it in infrared light and brightly white while observing it in near infrared light. This would enhance the pupil detection capabilities, because the edges around would be more significant.

This would also enable us not to detect the whole iris but only the pupil, which is smaller and placed in the middle of iris. For that reason, it would allow the detection to be more precise, since while the eye is in the fringe position, the pupil is still fully visible. These cameras, mostly with short distance focus, are also usually smaller and lighter, which would reduce the load on the frontal expansion of the camera holder or in case of extra small camera, the holder could be eliminated to the usage of simple glasses frame. The currently used camera could then be used for capturing the scene in front of the subject which is one of the possible ways to pinpoint the point of regard. Using the infrared light source could also reduce this fault factor significantly, as it can light the eye equally without the subject seeing it while camera still does. This would improve the quality of the captured image without increasing the invasiveness of the method.

4.4 SOFTWARE MODIFICATION OPTION

Currently used solution involving the camera with lower resolution (see Chapter 4.1) is based on iris detection. Since the iris in comparison to the white of the eye is significantly darker, the edge between the iris and white is the most visible one (see Figure 2.15 in Chapter 2.4). Although it is possible to detect pupil in edged image, it can be easily confused with false circles that occur in the image while image processing. In case of pupil detection, the software could faultily assess the small circle and cause minor inaccuracy in detection. In case of more suitable resolution of the input capture, the software could detect both the pupil and the iris, using “HoughCircles” function implemented in OpenCV, as two concentric circles, smaller for the pupil and bigger for the iris. That would make the circle detection more accurate as both pupil and iris move simultaneously and have mutual centre. It would also reduce the trembling of the circle (see Figure 3.4 (left) in Chapter 3).

4.5 GYROSCOPE AND ACCELEROMETER OPTION

By adding a gyroscope and an accelerometer to the current eye tracking system, the information about the head orientation would be gained. This information could improve the measuring accuracy significantly, since it could detect even the slightest head movement. The gyroscope gives information about rotation in *x* axis (pitch – “yes” movement), *y* axis (yaw – “no” movement) and z axis (roll - “maybe movement”) of the head. It calculates the head orientation according to gravity acceleration and Earth’s magnetic field, which is received from a built-in magnetometer. The accelerometer detects the position change by sensing the force affecting the module caused by the acceleration [17].

The accelerometer and the gyroscope are often included in a single module with microprocessor and communication port, for example I2C. To connect the module to the eye tracking system a development kit with a microprocessor, which would able to receive the data from the communication interface from the module and to proceed the data to the computer, is required. The suitable development kits are Raspberry Pi or Arduino Uno. The advantage of using those kits is that they are able to process the data from the module and send them to the computer in desired format. The information received from the module contains pitch, yaw and roll values from the gyroscope and x, *y* and z acceleration values from the accelerometer [18].

While the gyroscope and accelerometer are able to make the system more accurate, there are several problems with the data gained from the module that have to be addressed first. Since they only give spatial orientation data, there are several unknown quantities that need to be specified like their initial orientation to the screen or the distance of the module from the screen. Without precise information about those quantities, any calculation only estimates the point of regard recalculation when even the slightest head movement occurs. This results in continuous recalculations with cumulative error, which after several iterations are capable of causing critical inaccuracy. Another imprecision of the gyroscope occurs due to non-linear distance of the module from the screen. Assuming the gyroscope module is in front of the screen, its distance from the centre of the screen is different from the distance from the corner. The error cannot be fixed since it is impossible to tell the exact position of the module related to the screen. The module itself works with several percent error toleration which adds to the error total with previous values [19] [20].

To ensure the gyroscope and accelerometer work properly, several steps must be taken. The distance and orientation of the screen must be precisely specified in relation to the module position. The screen dimensions must be known to fix the translation errors. The accelerometer and the gyroscope must be set to give true and precise information about the translation and rotation movements. The recalibration software must be modified to take all the information into account and work with them exactly and precisely, since the measured head movements add with each other and every inaccuracy may lead to the critical calibration loss.

Some eye tracking devices use the accelerometers and the gyroscopes as assistance and as power saving option in combination with additional cameras. The cameras detect the user’s surroundings, calculate point of regard and the object positions, while the accelerometer and the gyroscope give later approximated information about the position differences. Before the error cumulates, the new camera measuring is realised.

Due to the lack of additional cameras, calculation complexity, memory requirements and complex calibration, the accelerometer and gyroscope are not suited for the created eye tracking system.

4.6 OTHER OPTIONS

There are many more aspects that are possible to modify or improve the realised solution. For example, the problem with the camera holder stability is possible to solve by adding one more stripe of elastic band to fixate the construction to the head more reliably. Different design of the camera holder could improve the camera system stability as well, for example a cap-like design, which on the other hand would require more material to produce and the plastic helmet may not fit every testing subject. Stable measuring conditions, for example in the testing room with optimised lighting, would increase the method’s accuracy and reliability. Alternatively, to diminish the head movement error, it is possible to build a head-fixating device, that would disable the tested subject’s head movement. The huge downside of this method would be the discomfort caused to the subject, which would increase the invasiveness of the method.

SUMMARY

This Bachelor's thesis deals with the topic of eye movement tracking. The goal of the thesis was to create an eye tracking system and to carry out its testing to evaluate both precision and functionality of the whole project. The required output of the testing application was two-dimensional information about eye movement over time. From a wide range of possibilities, a solution for construction and programming of application for detection of the point of regard, was chosen.

The thesis first introduces the reader to the eye tracking topic including human eye description, the history of the topic and possible eye tracking techniques, applications and solutions.

Then the chosen solution is described. It consists of a custom-made camera holder, one camera aimed at the eye of the tested subject and a testing application called "EyeTrackApp". The testing application was created on Windows 7 operating system, written in C/C++ via Visual Studio 2013, it is using OpenCV 3.1 libraries for image processing and is based on CLR Windows form. The realised eye tracking system is designed to be able to comply with all requirements specified in the thesis assignment.

The foundation of this design is one camera fastened to a camera holder and aimed on the left eye of the tested subject. This solution provides a cheap and functional eye tracking system. The camera holder is designed in SolidWorks and printed on 3D printer, attached to the tested subject's head by an elastic band. The camera then passes the captured video to a computer via a one-meter-long cable with a USB 2.0 port.

The process of eye tracking is described from the viewpoint of the image processing as well as from the viewpoint of the created code´s algorithm. The software solution consists of four main processes including eye tracking, calibration, testing and evaluation. The eye tracking process detects the circle around the iris and its centre. Together with calibration process it gives recalculated point of regard which is recorded during the testing process with the possibility of real-time detection. The testing process features six different kinds of tests, each focuses on a different aspect of the application and on different qualities of the tested subject. The tests consist of a reading part, watching the picture part and an easy brain teaser. The evaluation process then gives the output in the chosen form. Available forms of evaluation are "Connect the Dots" which connects all points of regard, showing the track, which the eyes of the subject travelled on the screen, "Heat Map" which draws the heat map over the tested image and two "Replay" evaluations which show the detected point of regard over time.

A number of tests was carried out on 9 different subjects and under different conditions to prove the eye tracker's accuracy and functionality. The detailed test outcomes are included in the attachment. The first part of attachment's reports gives five examples of full measurement taken under different circumstances. Full measurement consists of all tests available in the testing application. Number one and two measurements were performed on the same testing subject, the first taken during the daylight conditions and the other during artificial lighting. Next two examples present measurements with glasses. First kind of glasses are nonreflexive, the others are highly reflexive. The last full measurement is performed on test subject wearing contact lenses. Next, the shortened reports are included in the attachment to record the measurement's

completions. Some subjects were given the full test to complete repeatedly. Some measurements were carried out only to assess system's accuracy repeatedly on the same subjects. This way the system inaccuracy was calculated as an average of all measurement results.

The calculated average inaccuracy during the measuring was calculated to be 26.8 mm (which matches 95 px on the testing screen). The maximum error of testing is 60.5 mm (218 px) on average. Although the inaccuracies may seem relatively high, the human eye does not focus on a single point but it detects larger area which leads to the conclusion, that the system evaluates the point of regard correctly and detects the part of the screen that was being watched. The thesis also includes the possible limitations of this system.

In the last chapter of the thesis the possible extensions of the eye tracking system are discussed. The conclusion indicates that some of them might be included to the current system to increase the accuracy, but they would significantly increase the price or the invasiveness of the system.

REFERENCES

[1] Anatomy of the eye, *EyeRis Vision* [online], 2016 [ref. 2016-12-2], Available at: http://www.eyerisvision.com/anatomy-of-the-eye.html

[2] KOLB, H., *Gross Anatomy of the Eye by Helga Kolb, Webvision* [online], 2015 [ref. 2016-12-2], Available at: http://webvision.med.utah.edu/book/part-i-foundations/gross-anatomy-of-the-ey/

[3] Anatomy of the Eye, in: *American Association for Pediatric Ophthalmology and Strabismus* [online], 2015 [ref.2016-12-2], Available at: https://www.aapos.org/terms/conditions/22

[4] YARBUS, A. L., *Eye movements and vision*. New York: Plenum Press, 1967.

[5] BENSON, P. J. et al., *Simple Viewing Tests Can Detect Eye Movement Abnormalities That Distinguish Schizophrenia Cases from Controls with Exceptional Accuracy*, Biological Psychiatry [online], Volume 72, Issue 9 , 716 - 724, 2012 [ref. 2016-12-1]. Available at: http://www.biologicalpsychiatryjournal.com/article/S0006-3223(12)00396-4/fulltext.

[6] Portal UXFocus.cz [online], 2016 [ref.2016-12-1]. Available at http://www.uxfocus.cz.

[7] DUCHOWSKI, Andrew T. *Eye tracking methodology: theory and practice*. 2nd ed. London: Springer, c2007. ISBN 1846286085.

[8] ROBINSON, D. A., A *Method of Measuring Eye Movement Using a Scleral Search Coil in a Magnetic Field*, in IEEE Transactions on Bio-medical Electronics [online], vol. 10, no. 4, pp. 137 - 145, Oct. 1963, Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4322822&isnumber=4322817

[9] Medicton.com [online], 2016 [ref. 2016-12-1]. Available at: http://medicton.com/

[10] Products, Tobii Pro [online], Stockholm, 2016 [ref. 2016-12-1], Available at: http://www.tobiipro.com/product-listing/.

[11] SensoMotoric Instruments [Online], 2016 [ref 2016-12-1], Available at: http://www.smivision.com/.

[12] SR Research [Online], 2016 [ref 2016-12-1], Available at: http://www.srresearch.com/.

[13] The Eye Tribe [Online], 2016 [ref 2016-12-1], Available at: https://theeyetribe.com/.

[14] Neurotechnology [online], 2016 [ref 2016-12-1], Available at: http://www.neurotechnology.com/sentigaze.html.

[15] Open CV documentation. OpenCV. [online]. 09-12-2016 [ref. 2016-12-08]. Available at: http://docs.opencv.org/2.4/index.html

[16] Learning OpenCV: computer vision in C++ with the OpenCV library. ISBN 9781449314651.

[17] MPU-6050 Accelerometer + Gyro, Arduino. [online]. 2017 [ref. 2017-05-17]. Available at: https://playground.arduino.cc/Main/MPU-6050

[18] Accelerometer & Gyro Tutorial by Gadget Gangster in sensors, instructables [online], 2017 [ref 2017-05-17]. Available at: http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/

[19] ARNOLD, R. N., MAUNDER, L., *Gyrodynamics and its Engineering Applications,* 1st

edition, Academic Press 1961. ISBN: 9781483216140.

[20] NAGHSHINEH, Sam; AMERI, Golafsoun; ZERESHKI, Mazdak. Human Motion Capture using Tri-Axial accelerometers [online]. 2009 [ref 2017-05-17]. Available at: http://edge.rit.edu/edge/P10010/public/PDF/HME.pdf

[21] Pictures downloaded from https://pixabay.com

[22] Pictures downloaded from http://www.boredpanda.com/find-the-panda-illustrated-puzzles-openlist/?page_numb=1

LIST OF ATTACHMENTS

\ Attachment1-measuring

\ Attachment2-cameraHolder \

\ celo-obroucky.SLDPRT

\ drzak kamery.SLDPRT

\ uchyt-gumicky.SLDPRT

\ Attachment3-videos \

\ Calibration-Test2.avi

\ DotReplayEvaluation.avi

\ Settings.avi

\ Test4.avi

\ Attachment4-Binary \

\ Visual_CPP_DLL \

\ msvcp120.dll

\ msvcr120.dll

\ vcomp120.dll

\ Visual_CPP_Redistributable \

\ vcredist_x64.exe

\ EyeTracking.exe

\ EyeTracking.exe.metagen

\ EyeTracking.pdb

\ Metro.dll

\ opencv_world310.dll

\ ReadMe.txr

\ Attachment5-Source \

\ EyeTracking.sln

\ EyeTracking \

\ CalibrationCheck.cpp

\ CalibrationCheck.h

\ CalibrationCheck.jpg

\ CalibrationCheck.resx

\ CalibrationRecalculate.cpp

\ CalibrationRecalculate.h

\ CalibrationWindow.cpp

\ CalibrationWindow.h

\ CalibrationWindow.resx

\ CamInput.cpp

\ CamInput.h

\ CoordGetter.cpp

\ CoordGetter.h

\ dot.jpg

\ Drawer.cpp

\ Drawer.h

\ dvojka.jpg

\ EVALUATIONdots.cpp

\ EVALUATIONdots.h

\ EVALUATIONmap.cpp

\ EVALUATIONmap.h

\ EVALUATIONreplay.cpp

\ EVALUATIONreplay.h

\ EVALUATIONreplayDot.cpp

\ EVALUATIONreplayDot.h

\ EyeTrackApp.cpp

\ EyeTrackApp.h

\ EyeTrackApp.resx

\ EyeTracker.cpp

\ EyeTracker.h

\ EyeTracking.aps

\ EyeTracking.rc

\ EyeTracking.vcxproj

\ EyeTracking.vcxproj.filters

\ GetEyeXY.cpp

\ GetEyeXY.h

\ Hough.h

\ ImageProcesser.cpp

\ ImageProcesser.h

\ pandaInDogs.jpg

\ pandaInDogs.png
\ pandaInPills.jpg
\ pandaInRacoons.jpg
\ pandaInTroops.jpg
\ reading.JPG
\ RealTimeTracking.cpp
\ RealTimeTracking.h
\ resource.h
\ Resource.resx
\ TEST2.jpg
\ Test4.cpp
\ Test4.h
\ Test4.resx
\ TestImageGetter.cpp
\ TestImageGetter.h
\ WatchTheDot.JPG
\ WATCH-THE-DOT-input.txt

\ Attachment6-exampleOfTxtFiles \
\ calibration.txt
\ real.txt
\ TEST1.txt
\ TEST2.txt
\ TEST3.txt
\ TEST4-panda1.txt
\ TEST4-panda2.txt
\ TEST4-panda3.txt
\ TEST4-panda4.txt
\ TEST5.txt
\ times.txt
\ watchDot-SAVED.txt
\ WATCH-THE-DOT-input.txt

\ ReadMe

ATTACHEMENT NUM. 1

This text is an attachment to the bachelor thesis on topic of the eye movement tracking. It is created to give a measurement outcome of testing that was performed as a part of the mentioned thesis as a way of evaluating its precision and documents its functionality.

It consists of a brief description of the testing method and tests outcomes, measurement itself and its assess. Each measurement of a full test was performed on 9 subjects. Some subjects were given the full test to complete repeatedly. Some measurements were carried out only to assess system's accuracy repeatedly on the same subjects. This way the system inaccuracy was calculated as an average of certain measurement results.

In the text the reader may find the full reports of five measurements taken under different circumstances like daylight in comparison to artificial lighting, glasses with reflexive or non-reflexive glass surface and contact lenses. Next, the shortened reports are given to record the measurement's completions, although the full reports are not the part of the text as the number of full reports in this text is already proving the concept and functionality.

Later in the text there are tables with recorded data including average inaccuracy and maximum deviation for each measuring. From those the average measuring precision is calculated.

1 TESTING METHOD

To analyse the precision and reliability of the eye tracker, which was designed as a part of bachelor thesis on the topic of the eye movement tracking, the measuring on a number of subjects under different conditions were performed. This eye tracking method is non-invasive.

Each measurement starts with calibration process, in which the subject’s eye position is given to the calibration part of the application, which recalculates the position of the eye according to the position on the screen. It is necessary to complete this process for the eye tracking to be reliable and precise. From this point on, the subject must try not to move its head, so it is recommended to take the comfortable position before starting the calibration process.

After the calibration is complete, the calibration check is recommended. It consists of a blue rectangle on the screen, on which the subject is looking. The test evaluator checks, whether the calibration is precise by watching the real-time calculated point of regard on the other screen. In case of imprecise calibration, it is recommended to repeat the calibration process. Another way of evaluating the calibration is the watch the dot test, in which the measured subject watches the moving dot on the screen and the calculated point of regard is showing in real-time in the same window as the testing dot. This way the measured subject is able to confirm the accuracy of the calibration itself. The results of the watch the dot test are a part of the full reports, as seen below.

The first test consists of a picture of two dogs, which the measured subjects are watching. The test evaluation shows, which part of the picture was the most attractive to the viewer. The test number two is the alternative version of the test one. It is the image consisting of several pictures. The test evaluates which of the individual pictures attracts the eye the most.

The third test is a reading test, in which the subject is supposed to read a short text. The test evaluation shows the path, in which the subject was looking at the text while reading.

The fourth test is an application called Find the Panda. It is supposed to evaluate the perception of the tested subject. The subject has to find a panda hidden in the picture, which consist of the field filled with other elements. The test consists of four rounds, each of them is measured by the timer.

Test number five is designed to evaluate the inaccuracies by showing the static dot in the middle of the testing screen. The subject focuses on the dot, the measured result is evaluated by heat map, average and maximum deviation from the dot centre measured by the testing application.

Each test can be evaluated in four different ways, by heat map, showing the most attractive areas, heat map replay, showing the areas watched over time, connect the dots, which tracks the path of the eye movement and the replay by dot, showing the calculated point of regard over time. The reader may find the examples of heat maps and connect the dot evaluations in this text. The evaluation of heat map’s replay and the dot replay may be found in the video attachment of the bachelor thesis.

2 MEASUREMENT

In this chapter, the measuring reports are presented. First, the measurement through the whole testing range is shown in five examples followed by five shortened reports. After brief discussion and comparison, the deeper analysis of method accuracy is given. This includes more measurement of test 5 repeated on subjects and deeper analysis of method's drawbacks.

2.1 FULL MEASURING REPORTS

This part of attachment gives five examples of full measurement taken under different circumstances. Full measurement consists of all tests available in the testing application. Number one and two measurements were performed on same testing subject, first taken during the daylight conditions and the other during artificial lighting. Next two examples present measurement with glasses. First kind of glasses are nonreflexive, the others are highly reflexive. Last measurement is performed on test subject wearing contact lenses.

All reports include figures with evaluation by heat map for tests 1 (figures 1, 11, 21, 31 and 41), 2 (figures 2, 12, 22, 32 and 42), 5 (figures 9, 19, 29, 39 and 49) and watch the dot (figures 8, 18, 28, 38 and 48) and evaluation by connect the dot for test 3 (figures 3, 13, 23, 33 and 43), 4-1 (figures 4, 14, 24, 34 and 44), 4-2 (figures 5, 15, 25, 35 and 45), 4-3 (figures 6, 16, 26, 36 and 46) and 4-4 (figures 7, 17, 27, 37 and 47). They also include image of test subject's eye (see figures 10, 20, 30, 40 and 50) and table of times of test 4 completion (see tables 2, 4, 6, 8 and 10).

Every report also includes table with average inaccuracy and maximum error detected while measuring test 5 (see tables 3, 5, 7, 9 and 11). The comparison of these measurements inaccuracies provides table 1. Here you can see that the most accurate was testing of subject num. 1 under daylight condition and the most inaccurate was testing of subject num. 3 with reflexive glasses.

Table 1 Average inaccuracy and maximum error comparison

Testing subject	Condition	Average inaccuracy	Average inaccuracy	Max error	Max error
Num. 1	Daylight	68 px	19.1 mm	189 px	52.7 mm
Num. 1	Artificial light	94 px	26.2 mm	201 px	56.0 mm
Num. 2	Non-reflexive glasses	106 px	29.5 mm	212 px	58.9 mm
Num. 3	Reflexive glasses	159 px	44.2 mm	375 px	104.1 mm
Num. 4	Contact lenses	92 px	25.7 mm	212 px	58.9 mm

2.1.1 Measuring Report num. 1

Measurement number	1
Tested subject identification	1
Age, sex	24, male
Eye status	none
Light conditions, time	**Daylight**, 4 PM

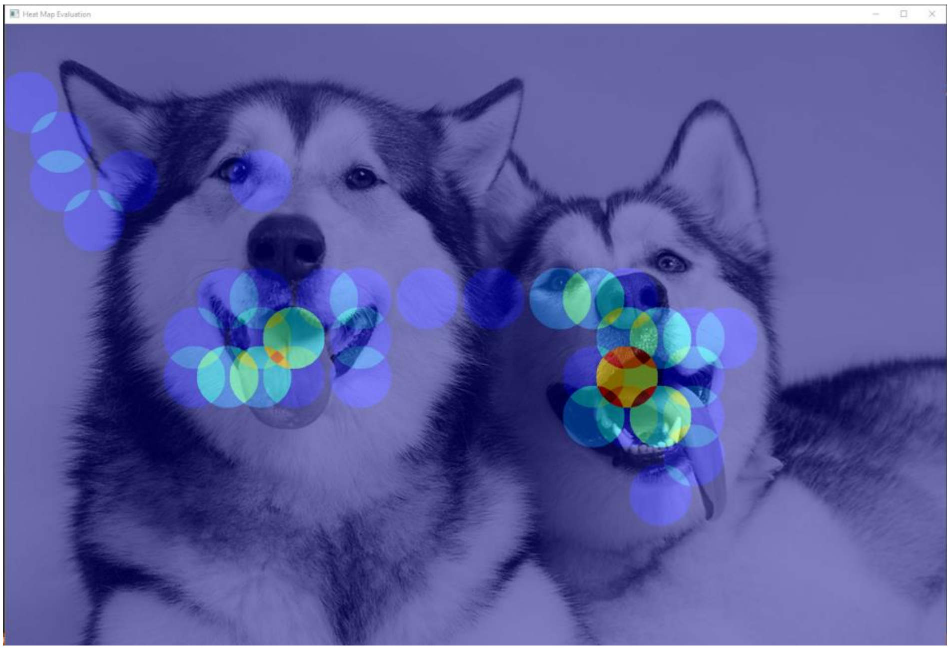


Figure 1 Test 1 evaluated by heat-map, testing subject num. 1 (1st)

Figure 2 Test 2 evaluated by heat-map, testing subject num. 1 (1st)

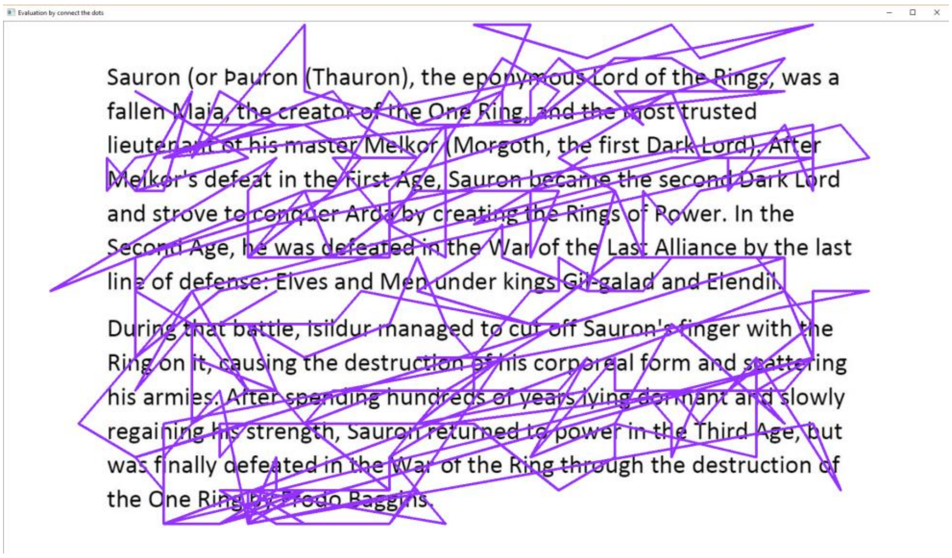


Figure 3 Test 3 evaluated by connect the dot, testing subject num. 1 (1st)

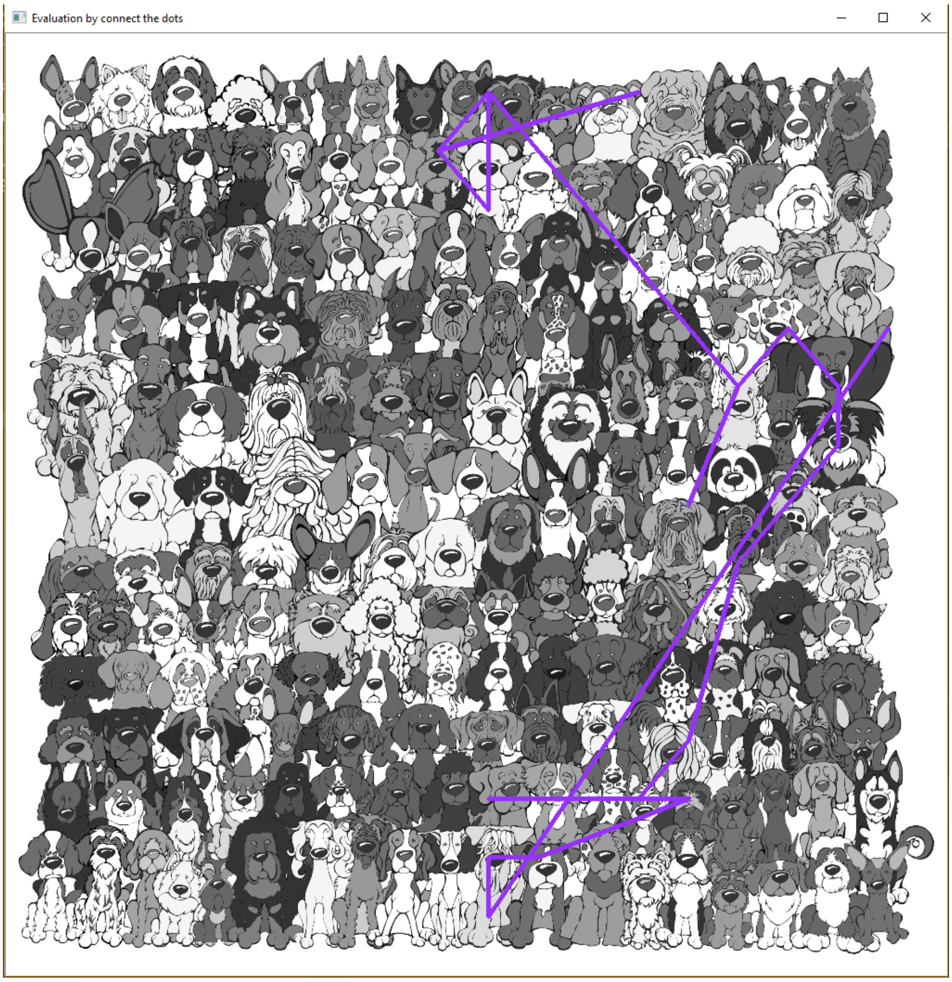


Figure 4 Test 4-1 evaluated by connect the dot, testing subject num. 1 (1st)

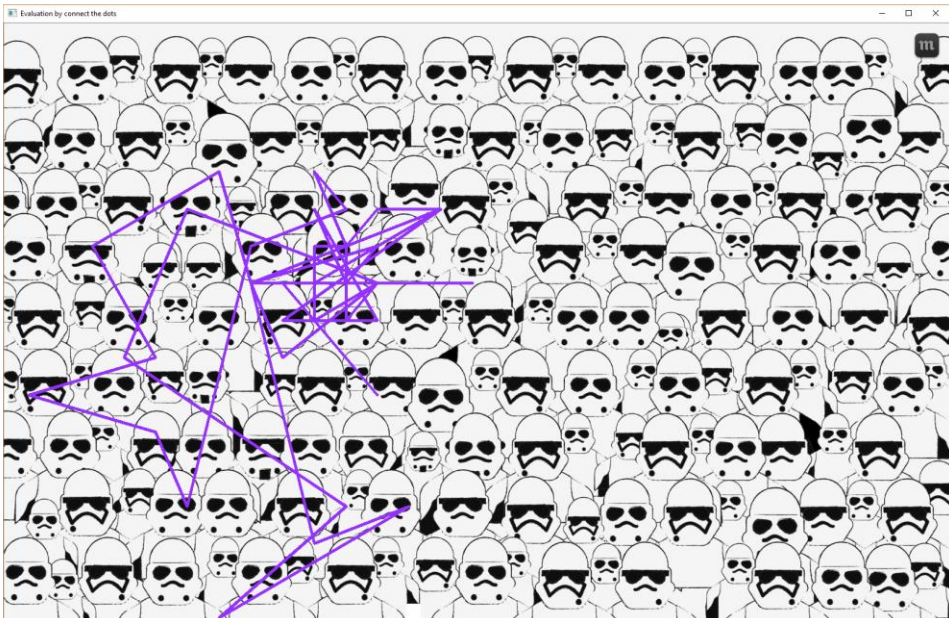


Figure 5   Test 4-2 evaluated by connect the dot, testing subject num. 1 (1st)

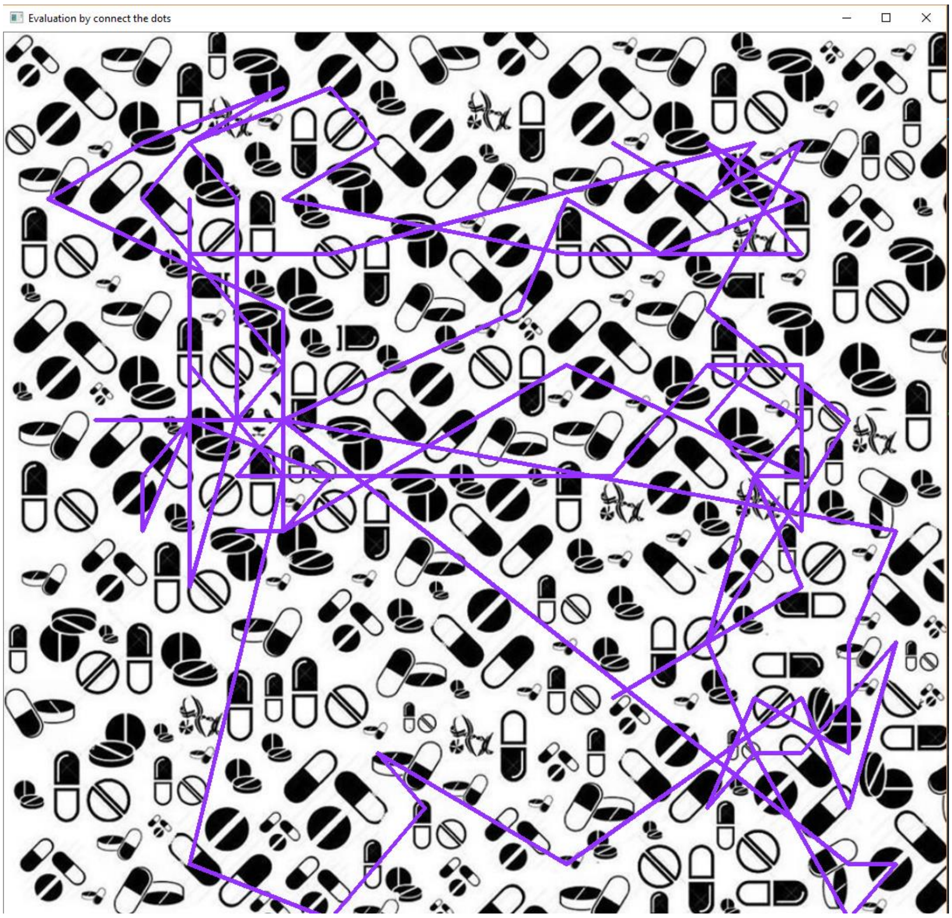


Figure 6   Test 4-3 evaluated by connect the dot, testing subject num. 1 (1st)

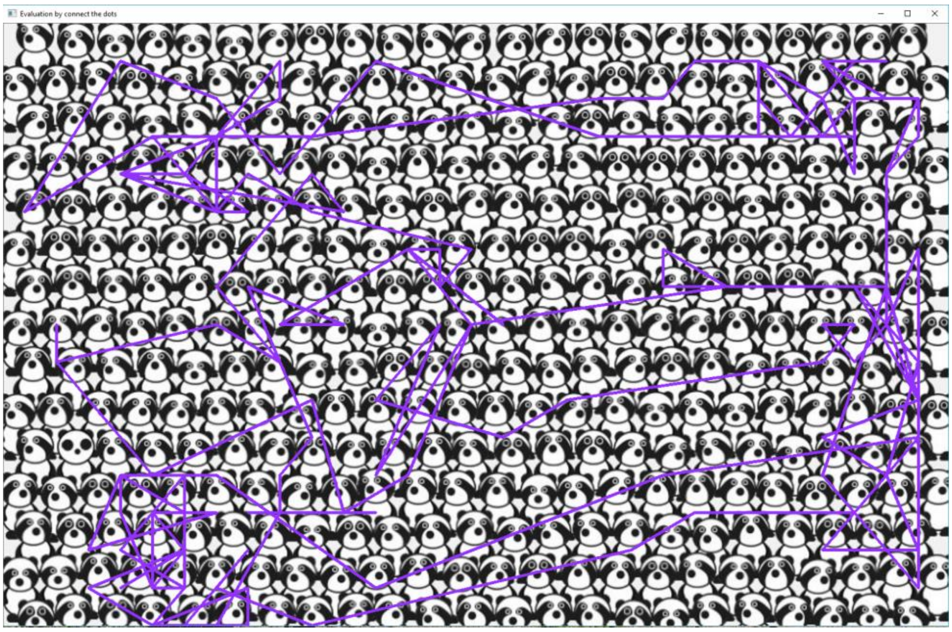


Figure 7 Test 4-3 evaluated by connect the dot, testing subject num. 1 (1st)

Table 2 Times of Test 4 completion of testing subject num. 1

Test	Time [s]
4-1	7
4-2	7
4-3	16
4-4	36

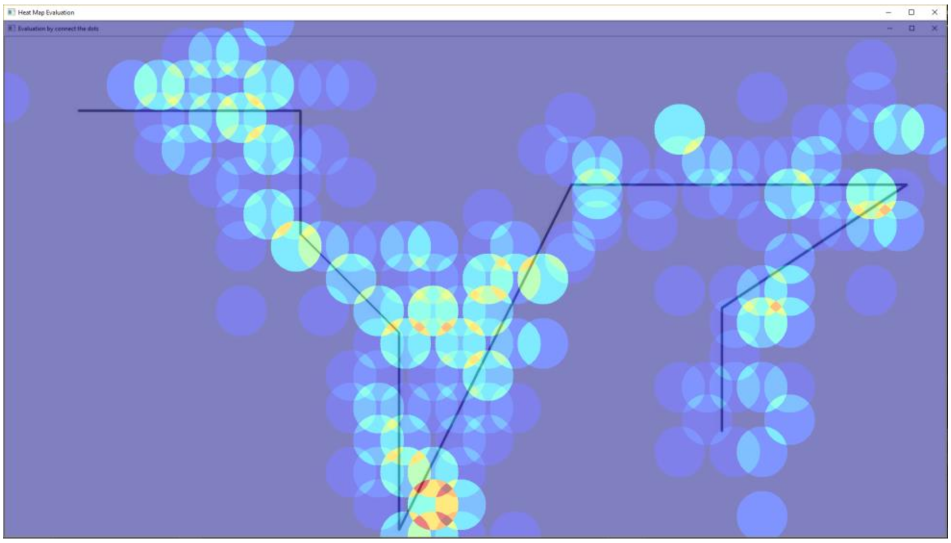


Figure 8 Watch the dot evaluated by heat-map, testing subject num. 1 (1st)

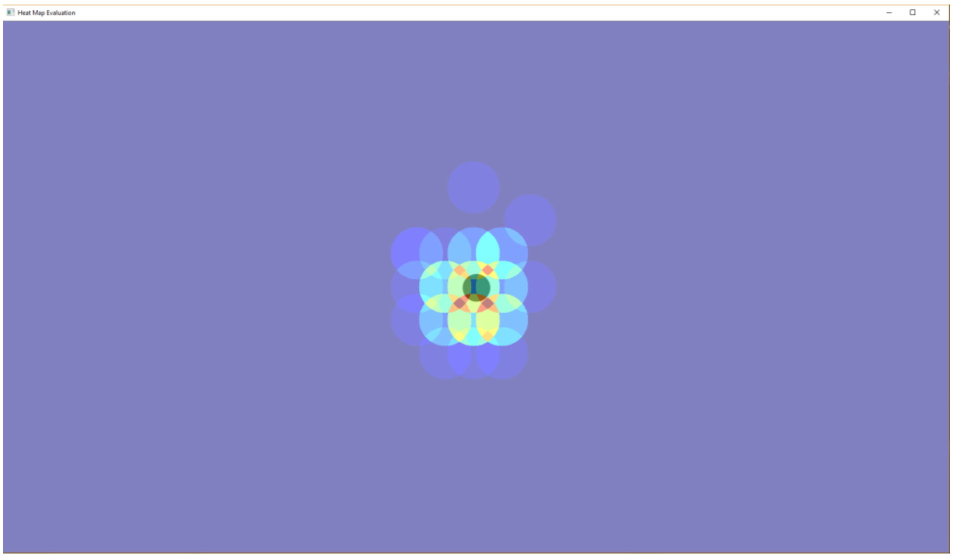


Figure 9 Test 5 evaluated by heat-map, testing subject num. 1 (1st)

Table 3 Average inaccuracy and max error of testing subject num. 1 (1st)

Average inaccuracy [px]	Average inaccuracy [mm]	Max error [px]	Max error [mm]
68	19.1	189	52.7

Figure 10 Left (tested) eye of test subject num. 1

Note:

Testing of subject num. 1 under daylight conditions was performed without any complications. Its average inaccuracy detected in test 5 was small (only 19.1 mm) as well as maximum error (only 52 mm). The tested subject agreed with measurement's evaluation and confirmed that he was looking at detected places.

2.1.2 Measuring Report num. 2

Measurement number	2
Tested subject identification	1
Age, sex	24, male
Eye status	none
Light conditions, time	**Artificial lighting**, 10 PM

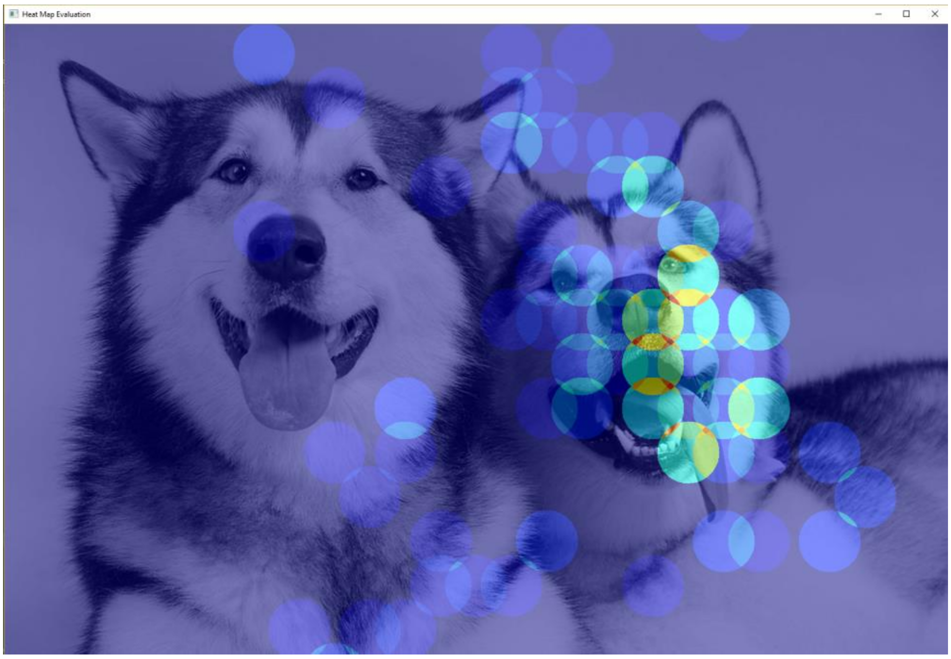


Figure 11 Test 1 evaluated by heat-map, testing subject num. 1 (2nd)

Figure 12 Test 2 evaluated by heat-map, testing subject num. 1 (2nd)

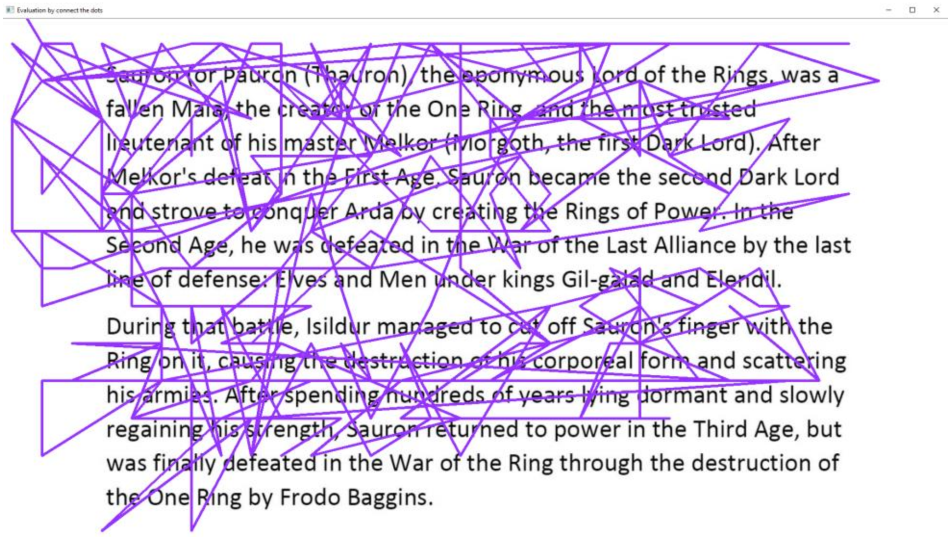


Figure 13 Test 3 evaluated by connect the dot, testing subject num. 1 (2nd)

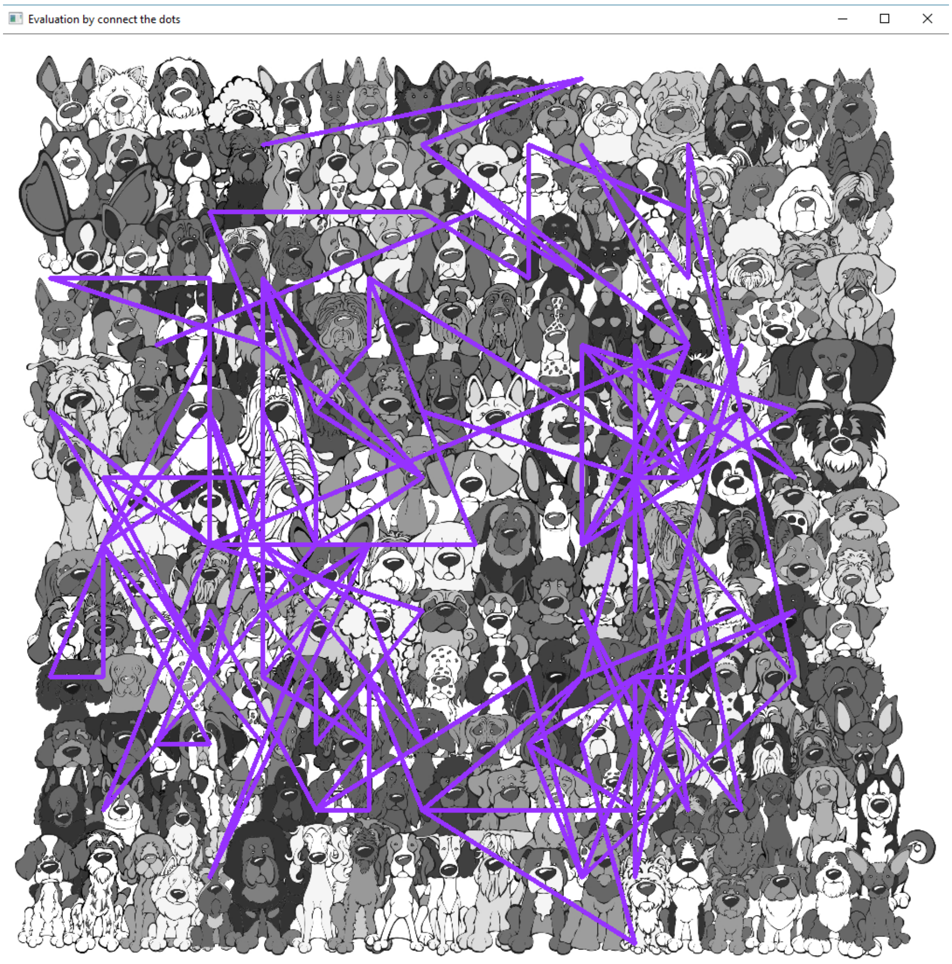


Figure 14 Test 4-1 evaluated by connect the dot, testing subject num. 1 (2nd)

Figure 15 Test 4-2 evaluated by connect the dot, testing subject num. 1 (2nd)

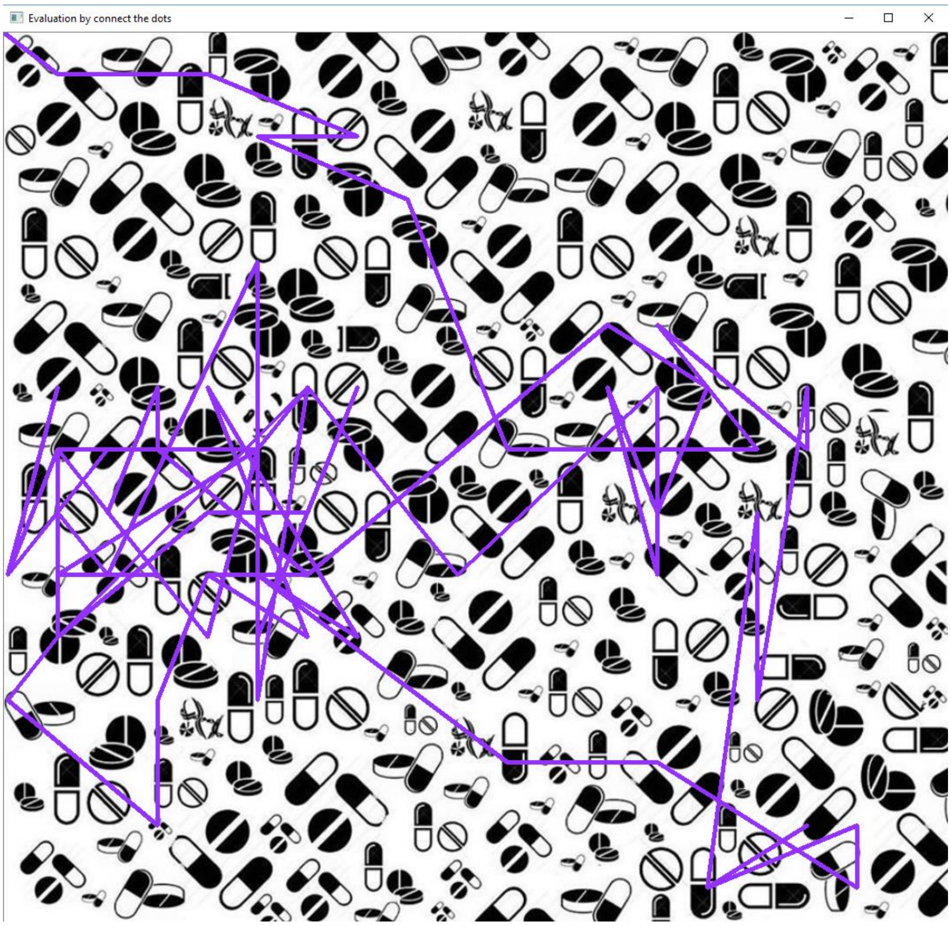


Figure 16 Test 4-3 evaluated by connect the dot, testing subject num. 1 (2nd)

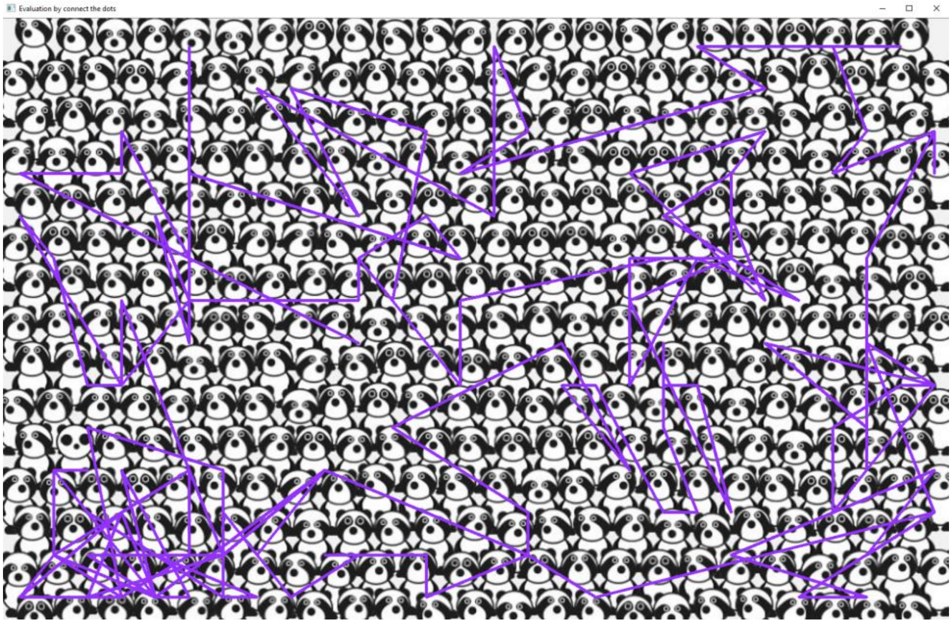


Figure 17 Test 4-4 evaluated by connect the dot, testing subject num. 1 (2nd)

Table 4 Times of Test 4 completion of testing subject num. 1

Test	Time [s]
4-1	21
4-2	14
4-3	11
4-4	22

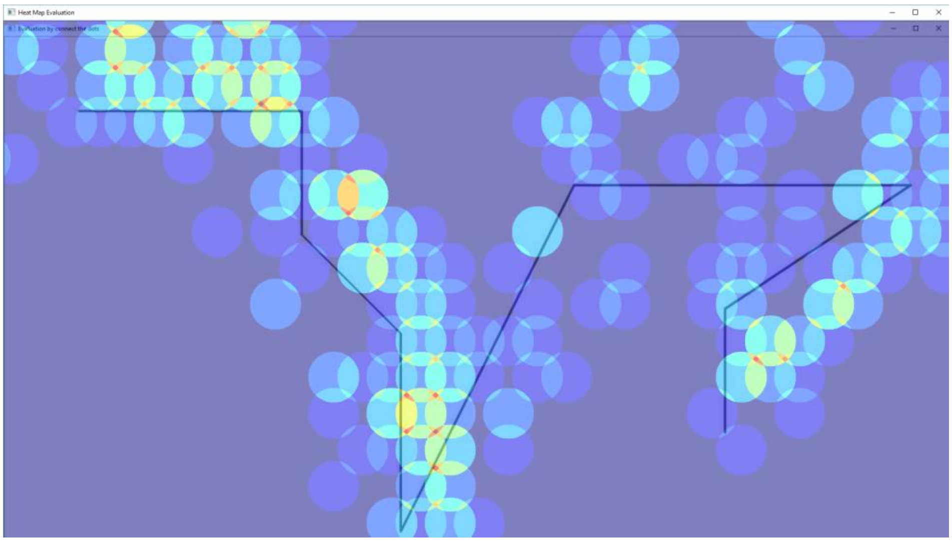


Figure 18 Watch the dot evaluated by heat-map, testing subject num. 1 (2nd)

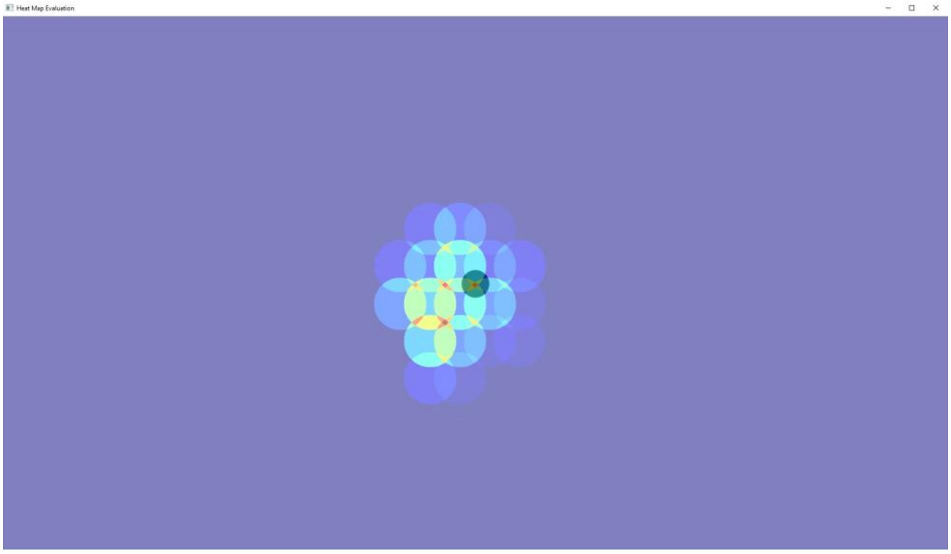


Figure 19 Test 5 evaluated by heat-map, testing subject num. 1 (2nd)

Table 5 Average inaccuracy and max error of testing subject num. 1 (2nd)

Average inaccuracy [px]	Average inaccuracy [mm]	Max error [px]	Max error [mm]
94	26.2	201	56.0

Figure 20 Left (tested) eye of test subject num. 1

Note:

Testing of the subject num. 1 under artificial lighting conditions was performed without any complications. Its average inaccuracy detected in test 5 was quite small (26 mm) as well as maximum error (56.0 mm). The tested subject agreed with measurement's evaluation and confirmed that he was looking at detected places.

2.1.3 Measuring Report num. 3

Measurement number	3
Tested subject identification	2
Age, sex	22, male
Eye status	**Glasses, nonreflexive**
Light conditions, time	Daylight, 2 PM

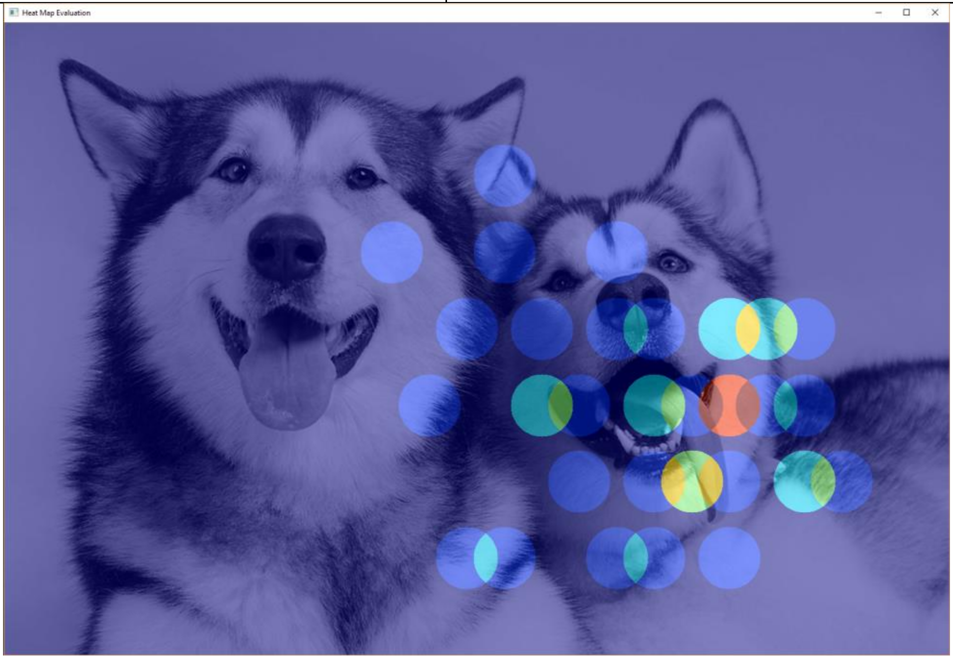


Figure 21 Test 1 evaluated by heat-map, testing subject num. 2

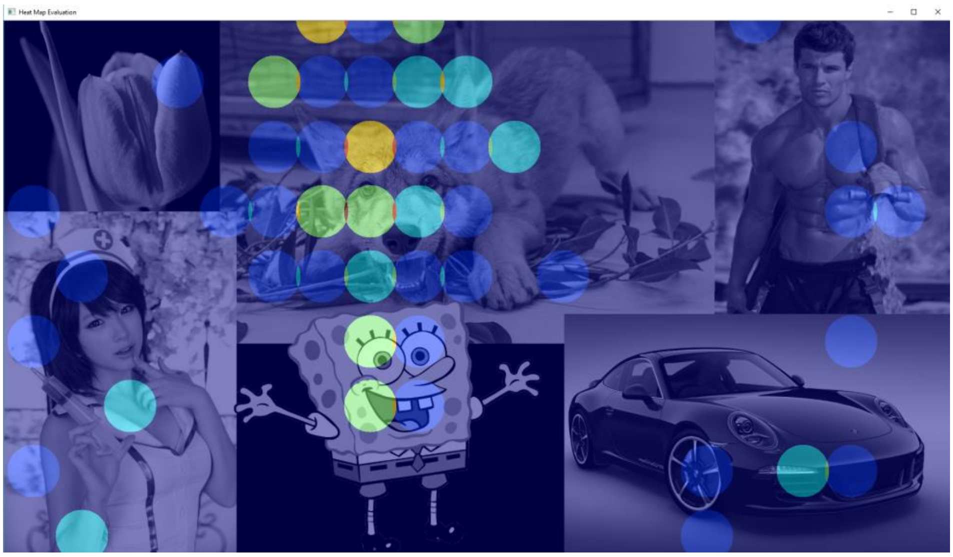


Figure 22 Test 2 evaluated by heat-map, testing subject num. 2

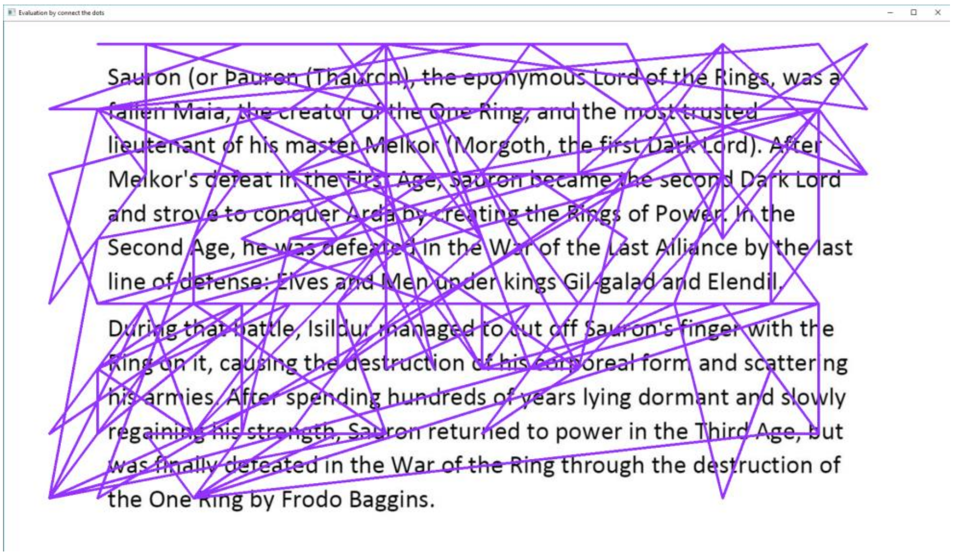


Figure 23 Test 3 evaluated by connect the dot, testing subject num. 2

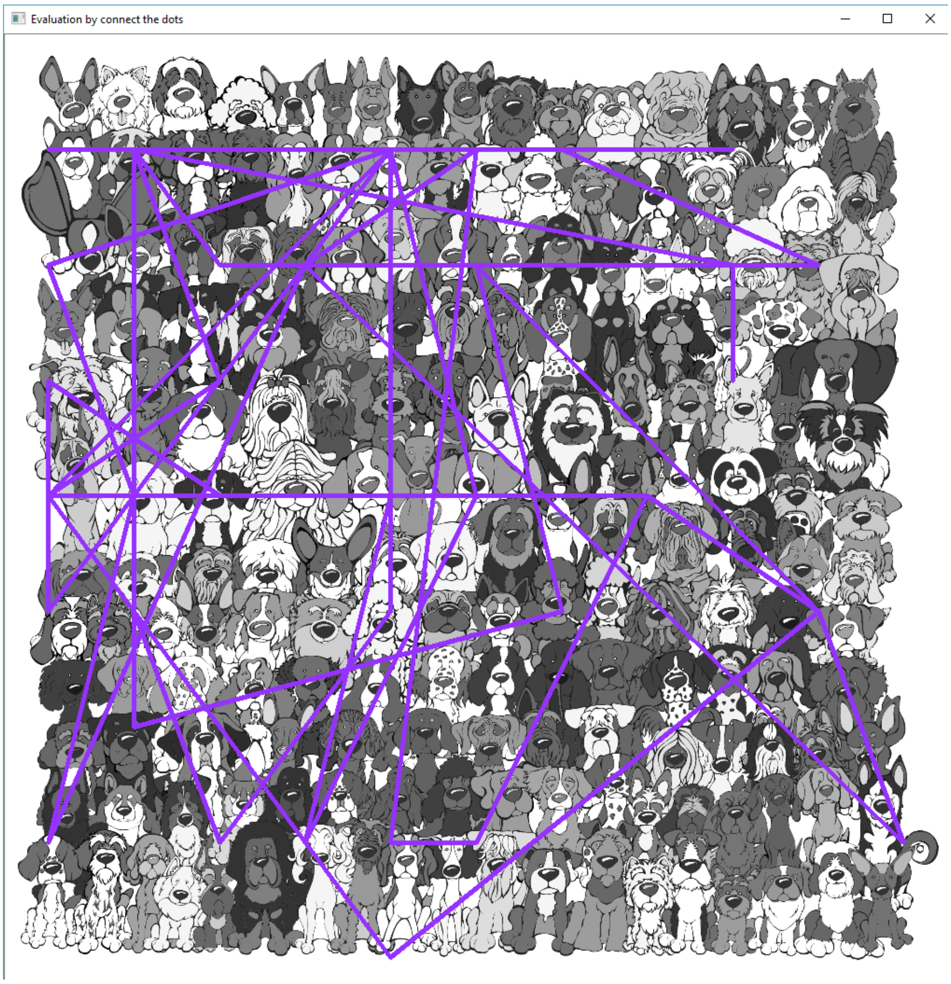


Figure 24 Test 4-1 evaluated by connect the dot, testing subject num. 2

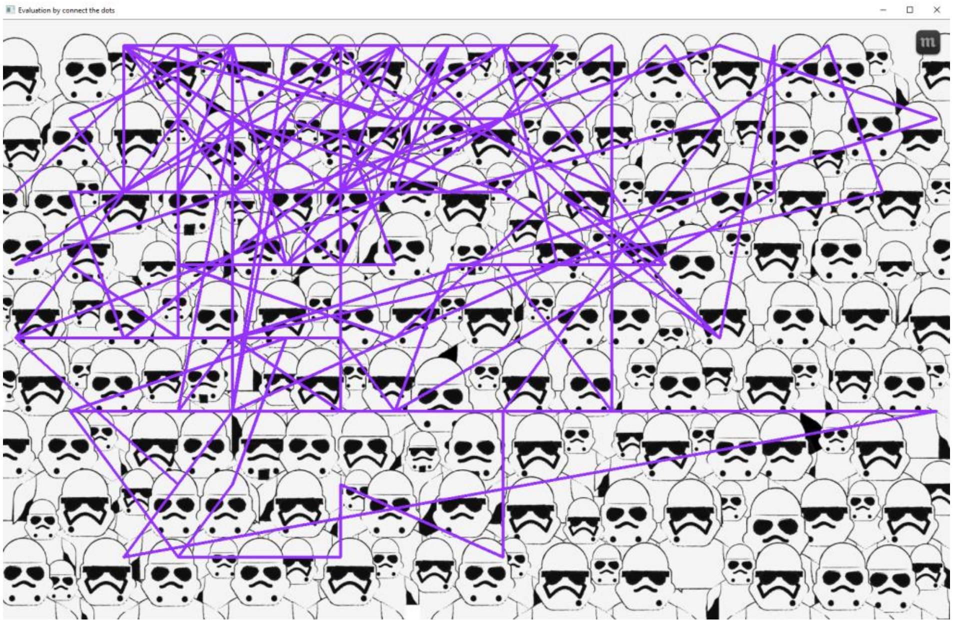


Figure 25 Test 4-2 evaluated by connect the dot, testing subject num. 2

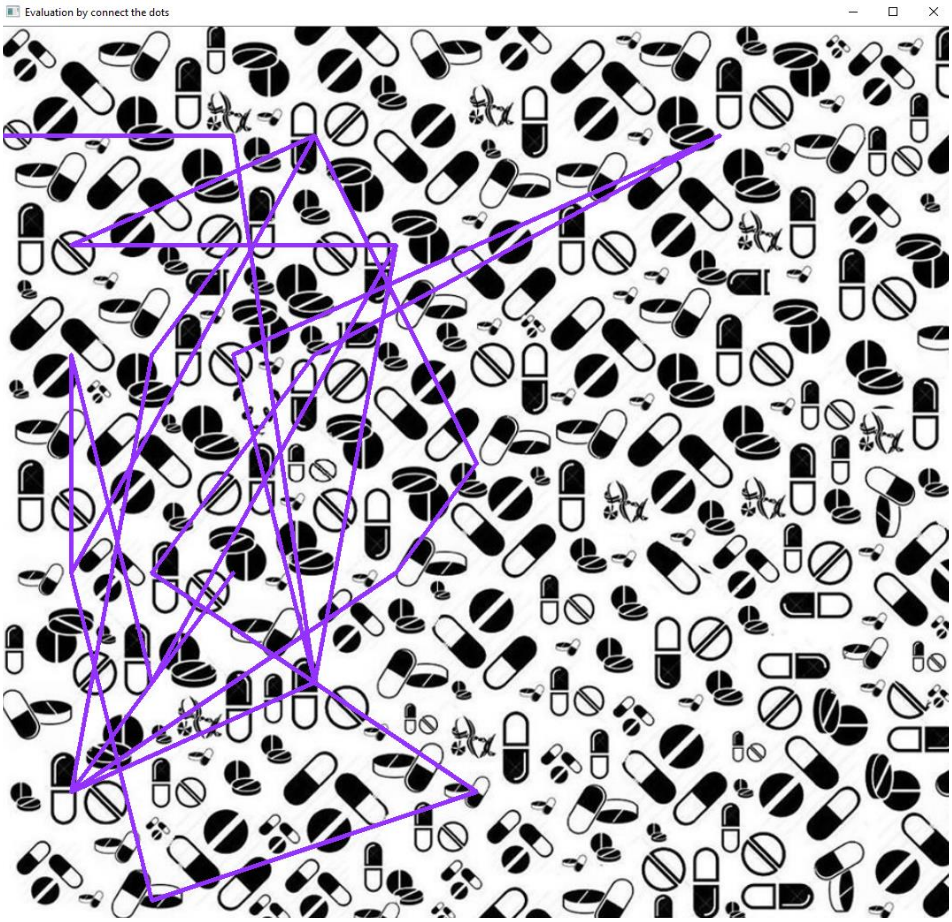


Figure 26 Test 4-3 evaluated by connect the dot, testing subject num. 2

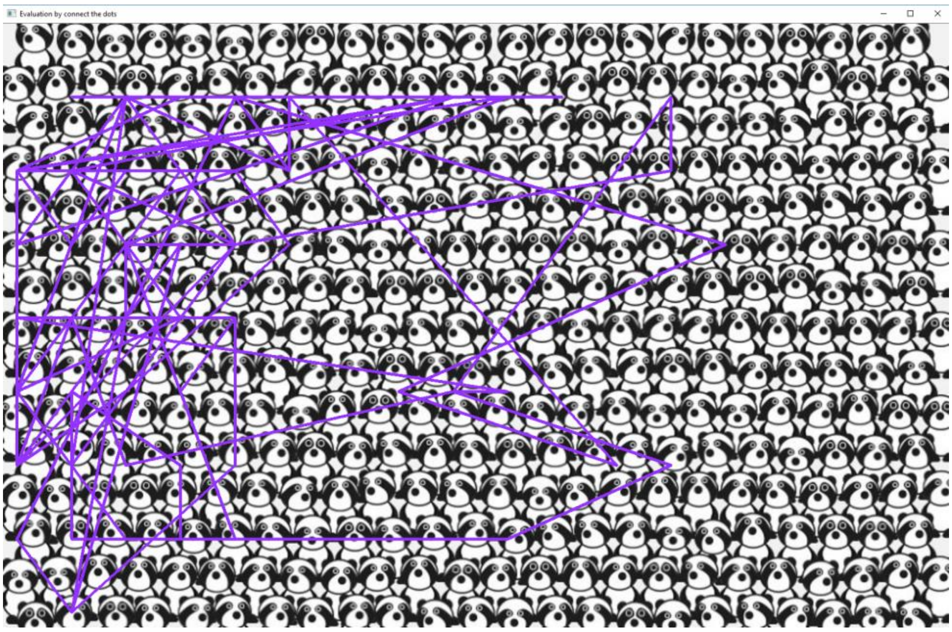


Figure 27 Test 4-4 evaluated by connect the dot, testing subject num. 2

Table 6 Times of Test 4 completion of testing subject num. 2

Test	Time [s]
4-1	16
4-2	53
4-3	12
4-4	44

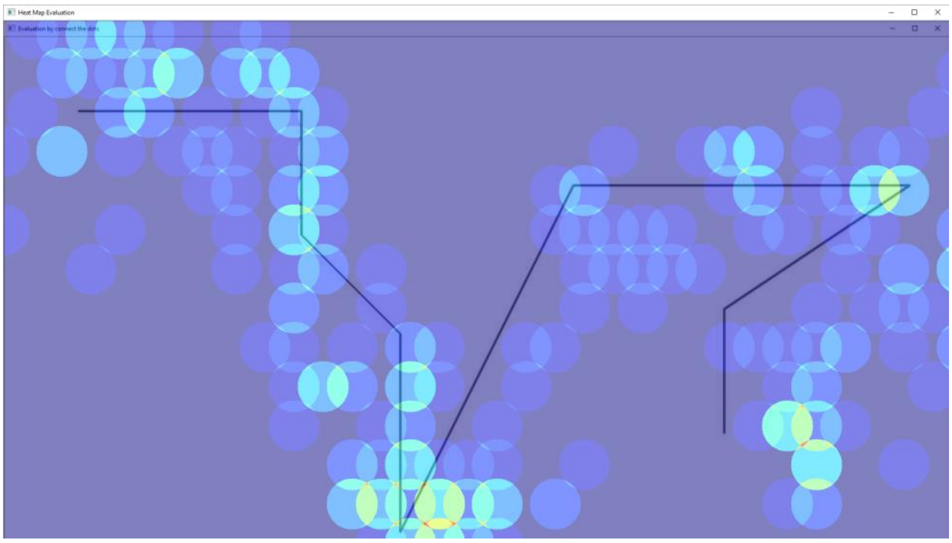


Figure 28 Watch the dot evaluated by heat-map, testing subject num. 2

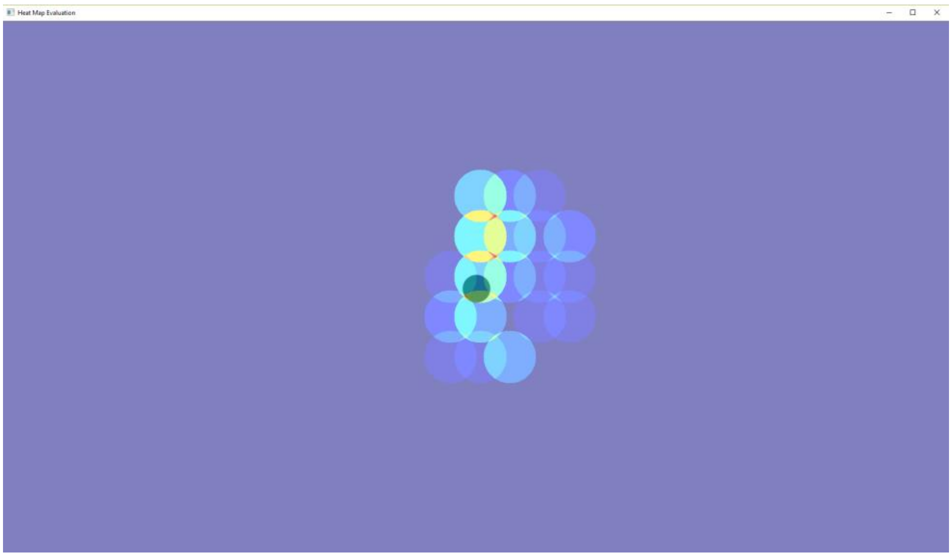


Figure 29 Test 5 evaluated by heat-map, testing subject num. 2

Table 7 Average inaccuracy and max error of testing subject num. 2

Average inaccuracy [px]	Average inaccuracy [mm]	Max error [px]	Max error [mm]
106	29.5	212	58.9

Figure 30 Left (tested) eye of test subject num. 2

Note:

Testing of subject num. 2 with nonreflexive glasses was performed without any complications. Its average inaccuracy detected in test 5 was on medium level (29.5 mm) as well as maximum error (58.9 mm). The tested subject agreed with measurement's evaluation and confirmed that he was looking at detected places. Nonreflexive glasses did not cause any major complications.

2.1.4 Measuring Report num. 4

Measurement number	4
Tested subject identification	3
Age, sex	21, male
Eye status	**Glasses, reflexive**
Light conditions, time	Daylight, 6 PM

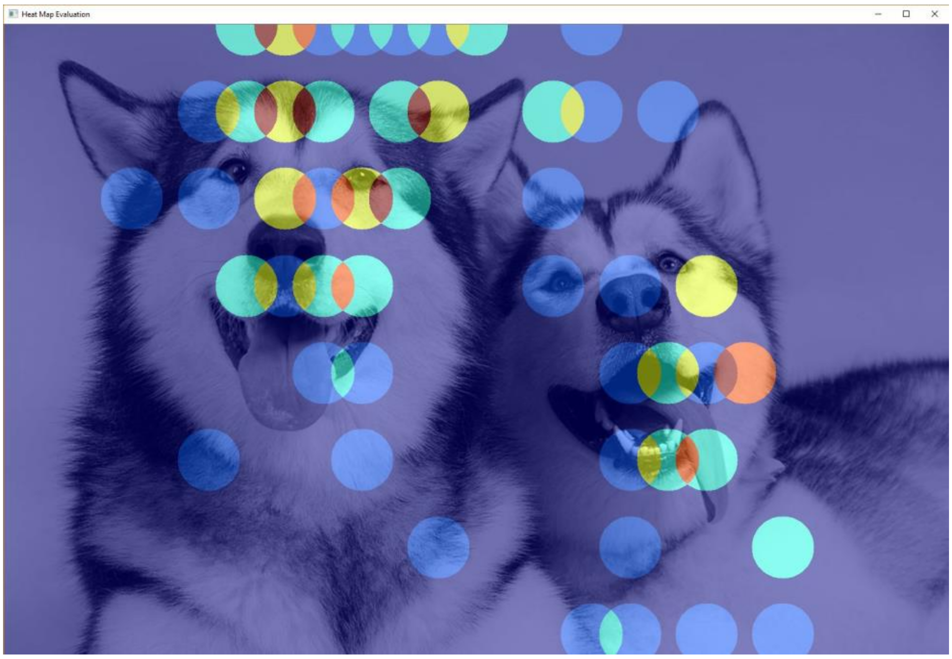


Figure 31 Test 1 evaluated by heat-map, testing subject num. 3

Figure 32 Test 2 evaluated by heat-map, testing subject num. 3

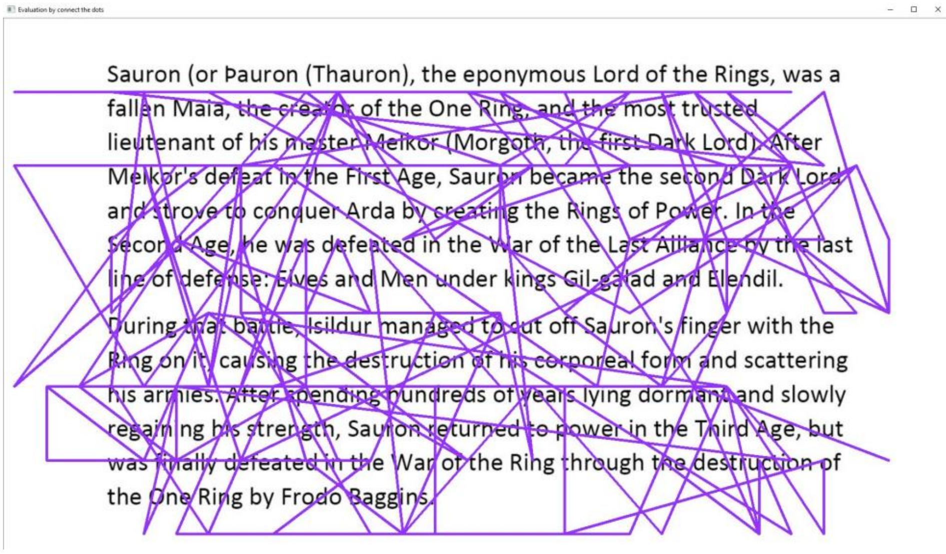


Figure 33 Test 3 evaluated by connect the dot, testing subject num. 3

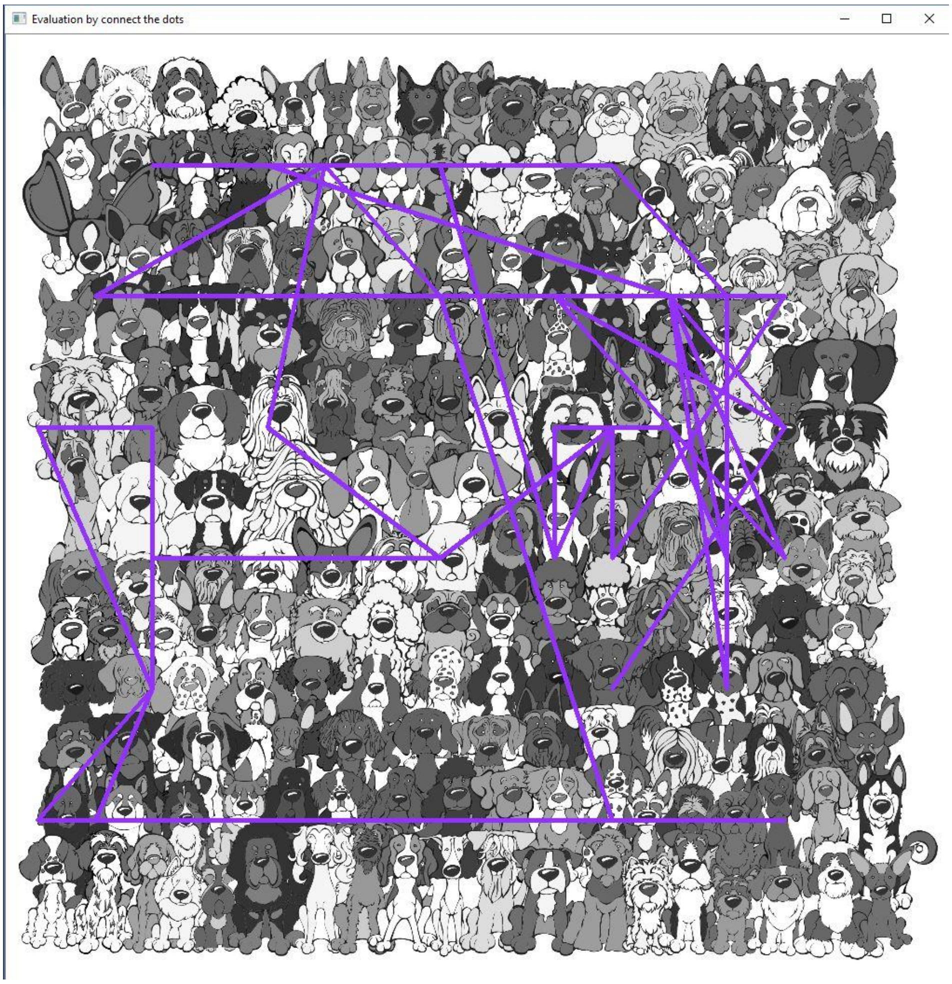


Figure 34 Test 4-1 evaluated by connect the dot, testing subject num. 3

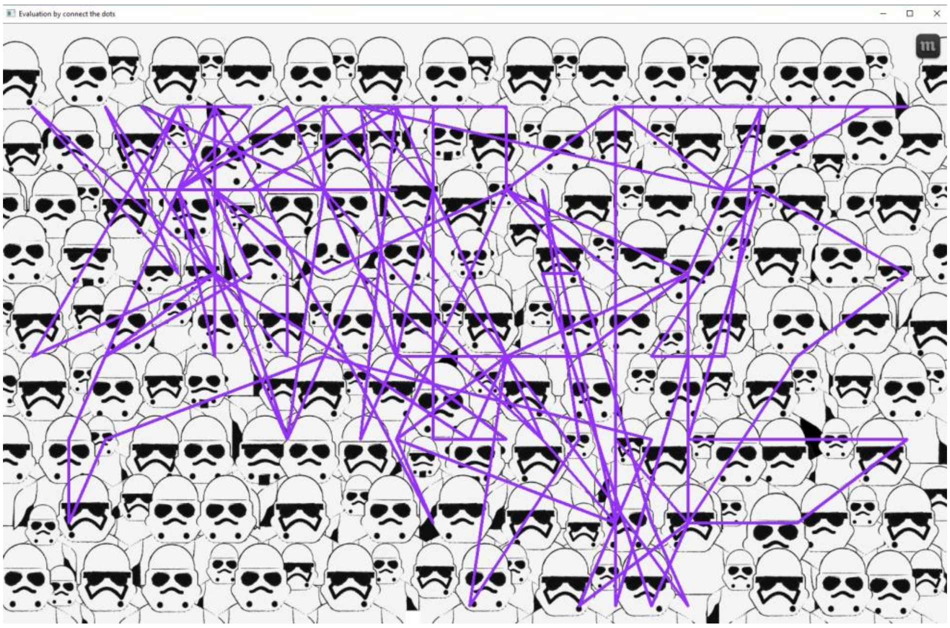


Figure 35 Test 4-2 evaluated by connect the dot, testing subject num. 3

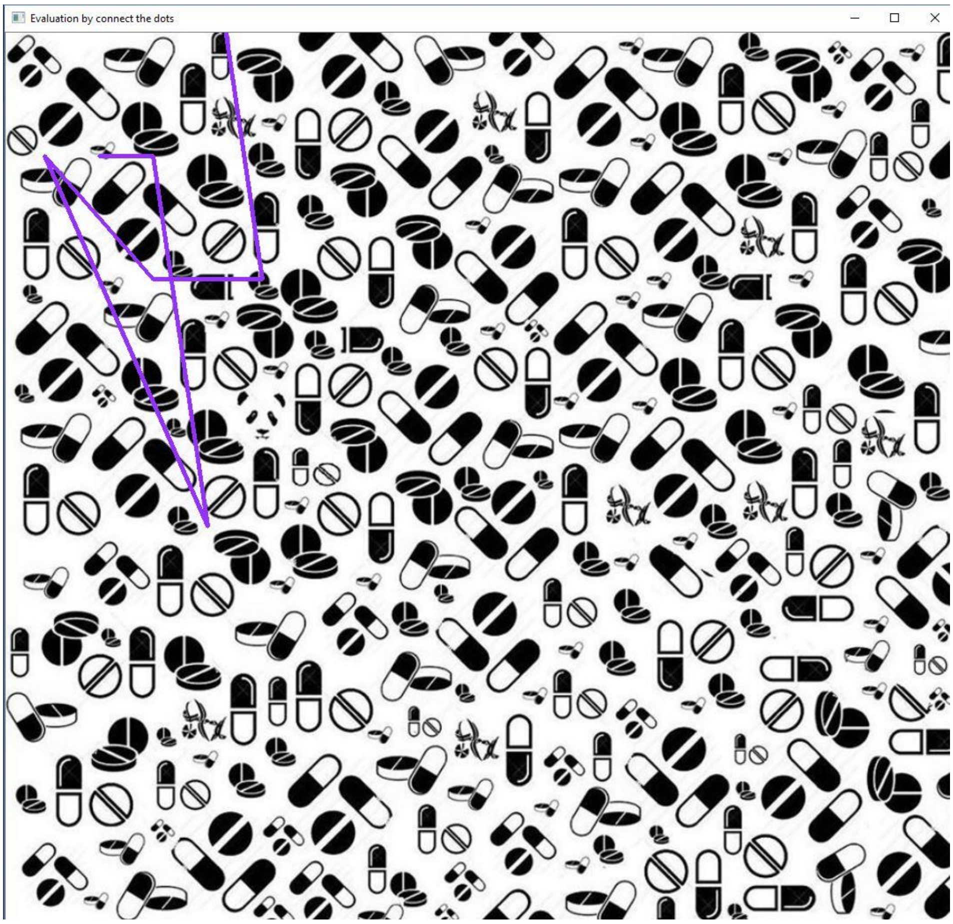


Figure 36 Test 4-3 evaluated by connect the dot, testing subject num. 3

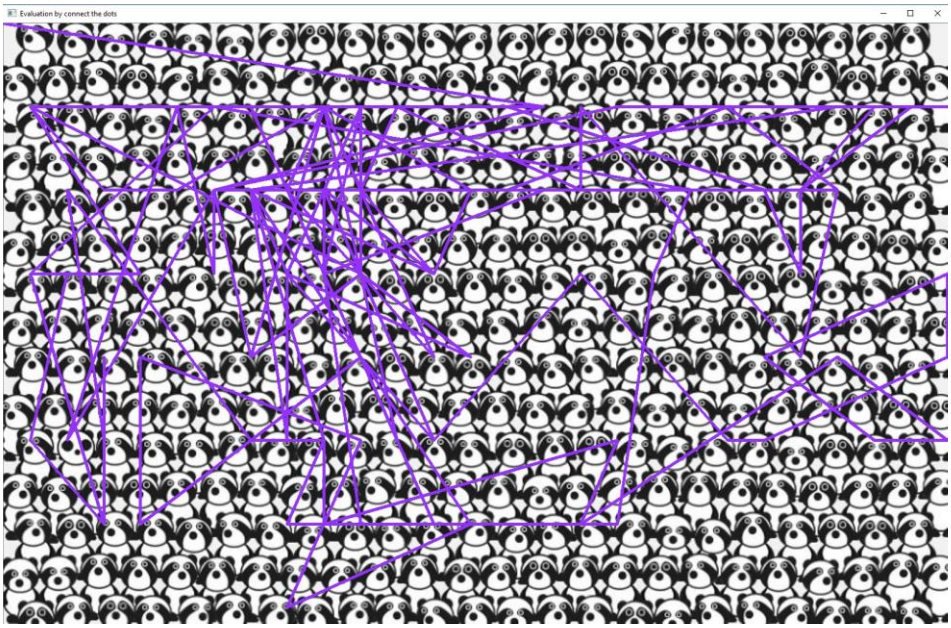


Figure 37 Test 4-4 evaluated by connect the dot, testing subject num. 3

Table 8 Times of Test 4 completion of testing subject num. 3

Test	Time [s]
4-1	20
4-2	20
4-3	2
4-4	32

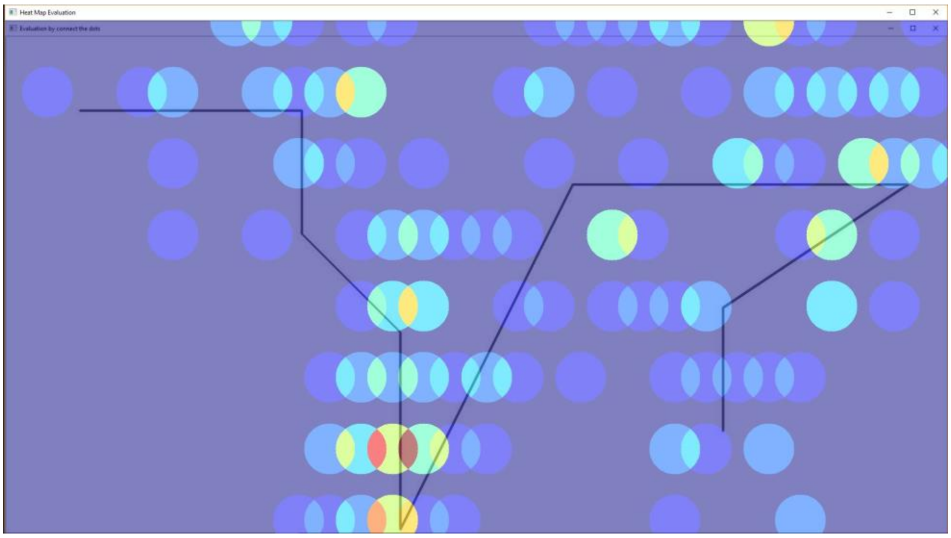


Figure 38 Watch the dot evaluated by heat-map, testing subject num. 3

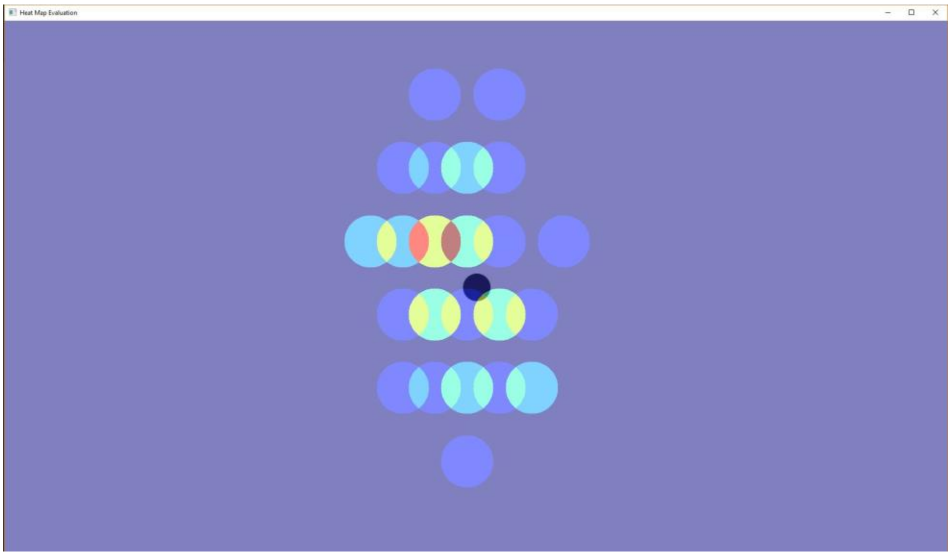


Figure 39 Test 5 evaluated by heat-map, testing subject num. 3

Table 9 Average inaccuracy and max error of testing subject num. 3

Average inaccuracy [px]	Average inaccuracy [mm]	Max error [px]	Max error [mm]
159	44.2	375	104.1

Figure 40 Left (tested) eye of test subject num. 3

Note:

Testing of subject num. 3 with reflexive glasses was performed with some complications. Its average inaccuracy detected in test 5 was high (44.2 mm) as well as maximum error (104.1 mm). Tested subject agreed with measurement's evaluation and confirmed that he was looking at detected places, but during calibration there were some major problems with subject's glasses. The calibration process was completed successfully at the third take. The first two takes were inaccurate because of screen reflection in glass of glasses, which was partially solved by reducing the screen brightness. Anyway, the inaccuracy of measurement stayed high.

2.1.5 Measuring Report num. 5

Measurement number	5
Tested subject identification	4
Age, sex	24, male
Eye status	**Contact lenses**
Light conditions, time	Daylight, 4 PM

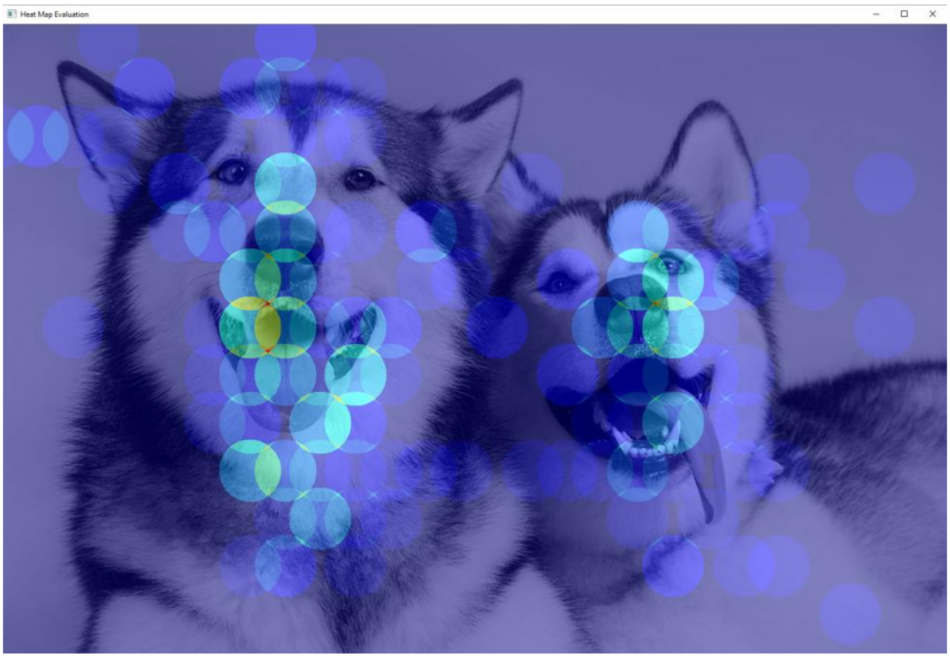


Figure 41 Test 1 evaluated by heat-map, testing subject num. 4

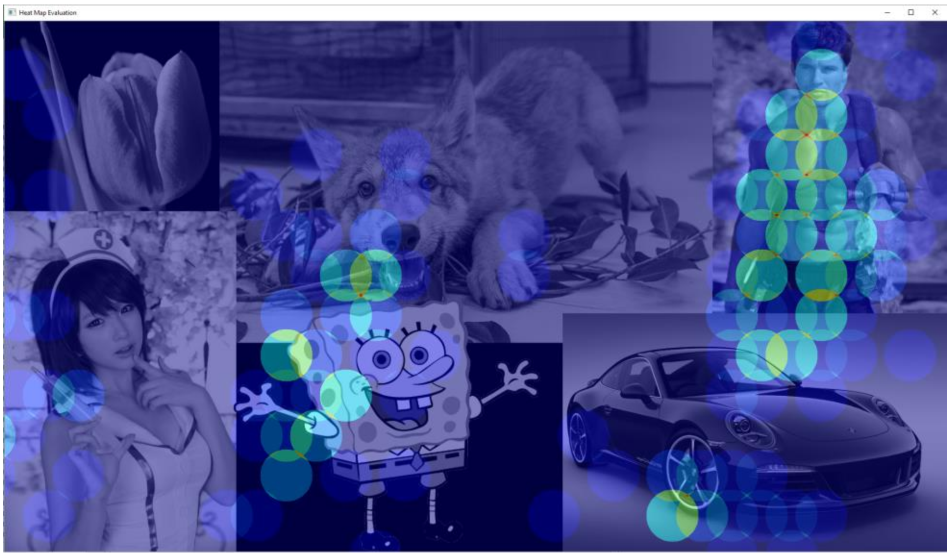


Figure 42 Test 2 evaluated by heat-map, testing subject num. 4

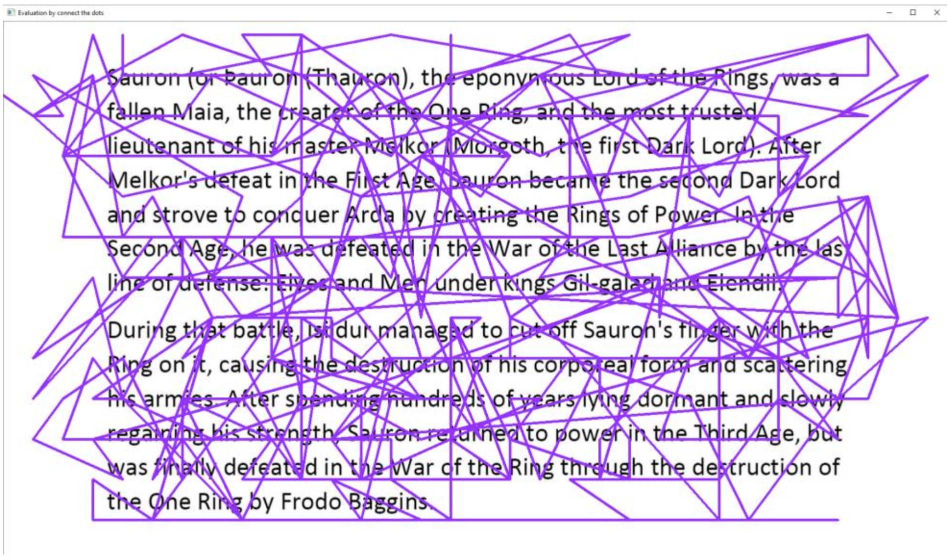


Figure 43 Test 4-3 evaluated by connect the dot, testing subject num. 4

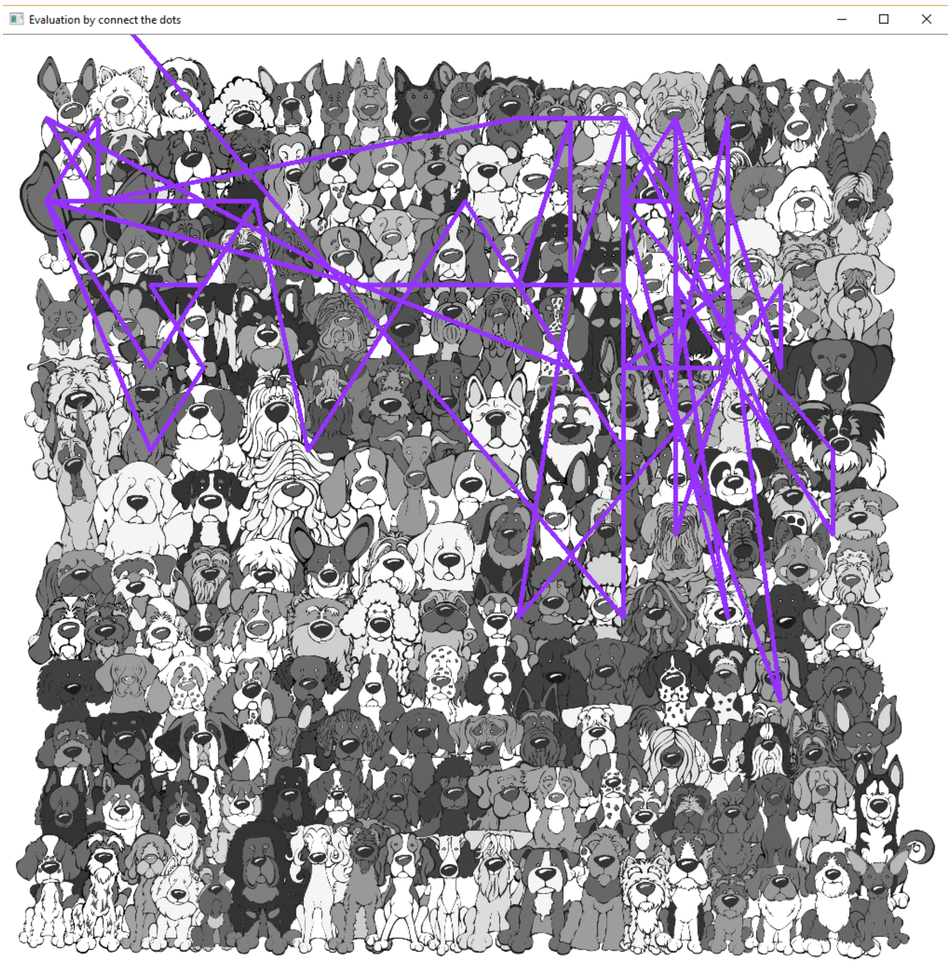


Figure 44 Test 4-1 evaluated by connect the dot, testing subject num. 4

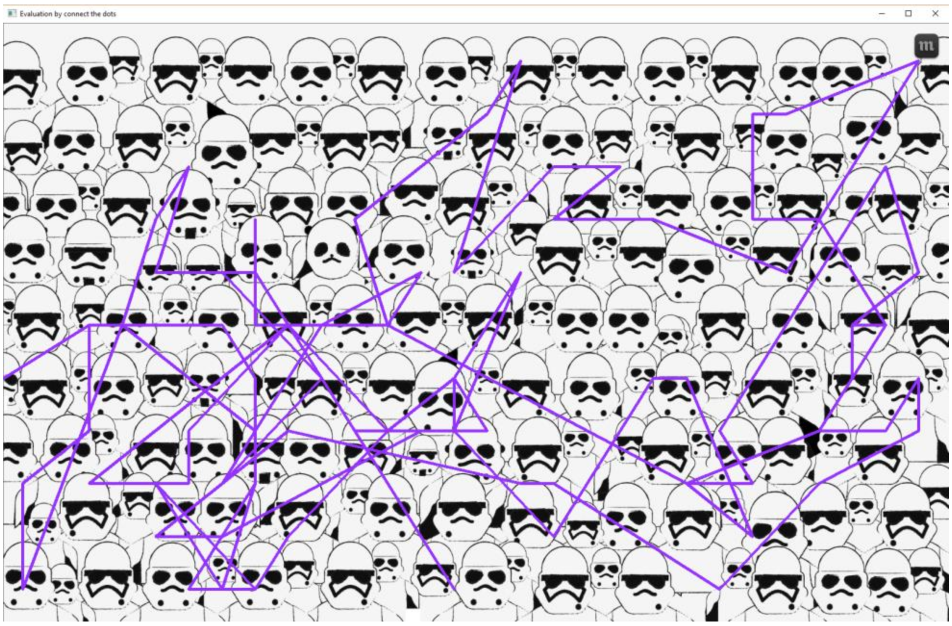


Figure 45 Test 4-2 evaluated by connect the dot, testing subject num. 4

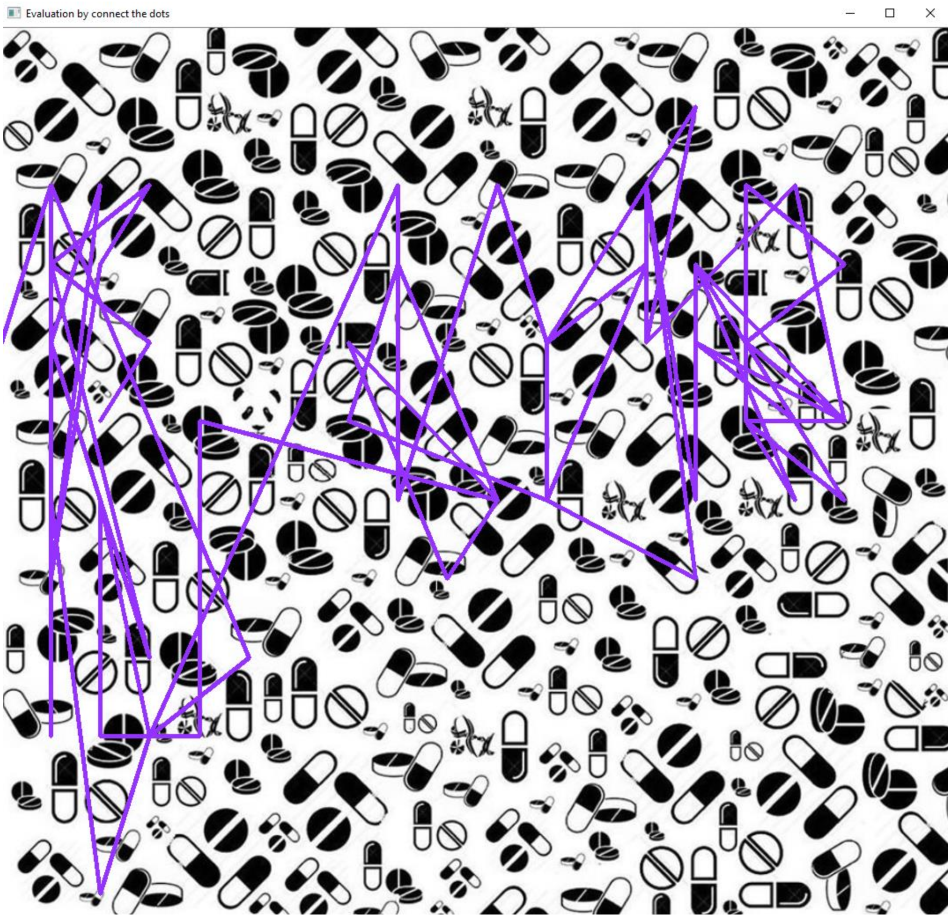


Figure 46 Test 4-3 evaluated by connect the dot, testing subject num. 4

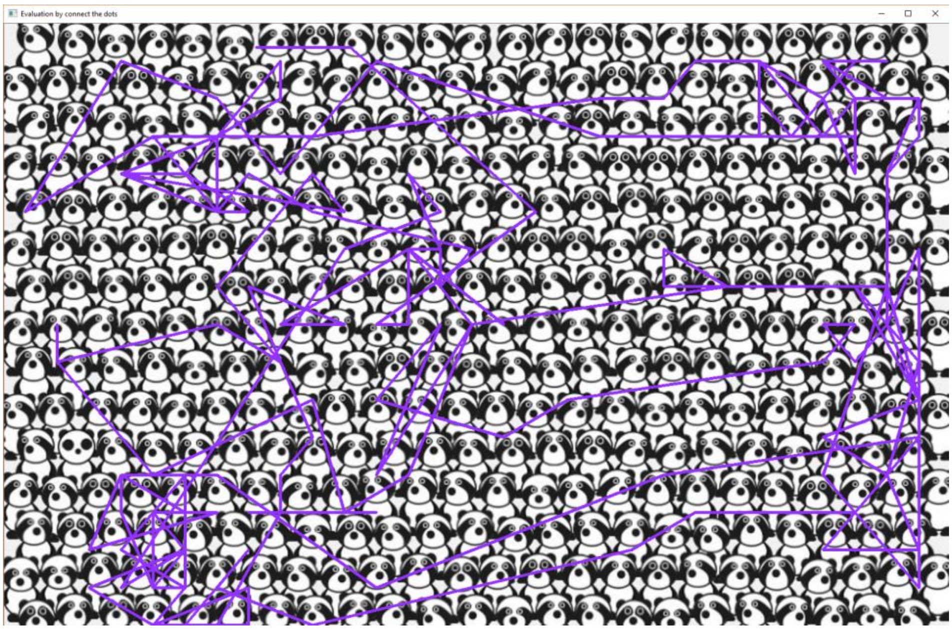


Figure 47 Test 4-4 evaluated by connect the dot, testing subject num. 4

Table 10 Times of Test 4 completion of testing subject num. 4

Test	Time [s]
4-1	15
4-2	17
4-3	9
4-4	11

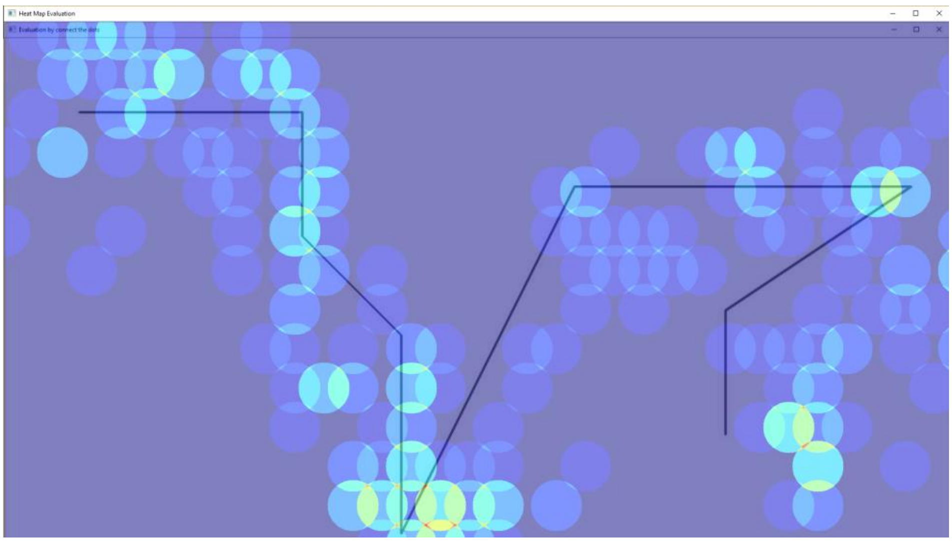


Figure 48 Watch the dot evaluated by heat-map, testing subject num. 4

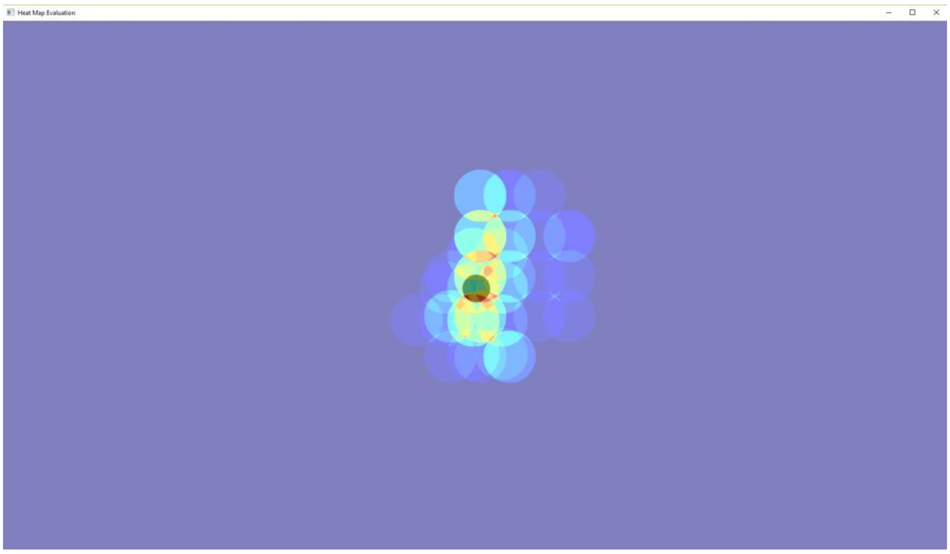


Figure 49 Test 5 evaluated by heat-map, testing subject num. 4

Table 11 Average inaccuracy and max error of testing subject num. 4

Average inaccuracy [px]	Average inaccuracy [mm]	Max error [px]	Max error [mm]
92	25.7	212	58.9

Figure 50 Left (tested) eye of test subject num. 4

Note:

Testing of subject num. 4 with contact lenses was performed without any complications. Its average inaccuracy detected in test 5 was small (25.7 mm) as well as maximum error (58.9 mm). The tested subject agreed with measurement's evaluation and confirmed that he was looking at detected places. Contact lenses did not cause any complications.

2.2 SHORTENED MEASURING REPORTS

In this part of attachment num. 1, instead of full reports, five shortened reports are given. Every shortened report includes evaluation of a random test, heat-map evaluation of test 5 and an image of tested subject's eye (see figures 51-55). The inaccuracy comparison and times of finishing test 4 are given in tables below (see tables 12 and 13).

Measurement number	6
Tested subject identification	5
Age, sex	32, male
Eye status	**Contact lenses**
Light conditions, time	Daylight, 1 PM

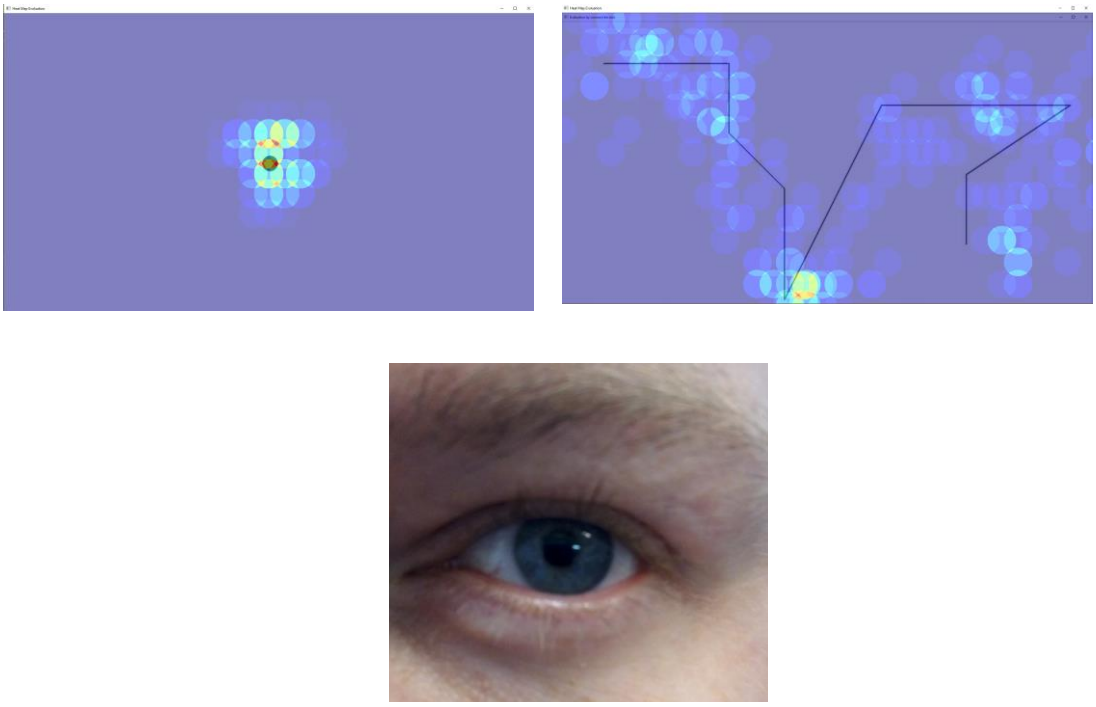


Figure 51 Measurement num. 6 results (top-left is test 5, top right is Watch the dot test, bottom is the eye of the tested subject – with contact lenses)

Measurement number	7
Tested subject identification	6
Age, sex	15, male
Eye status	None
Light conditions, time	Daylight, 6 PM

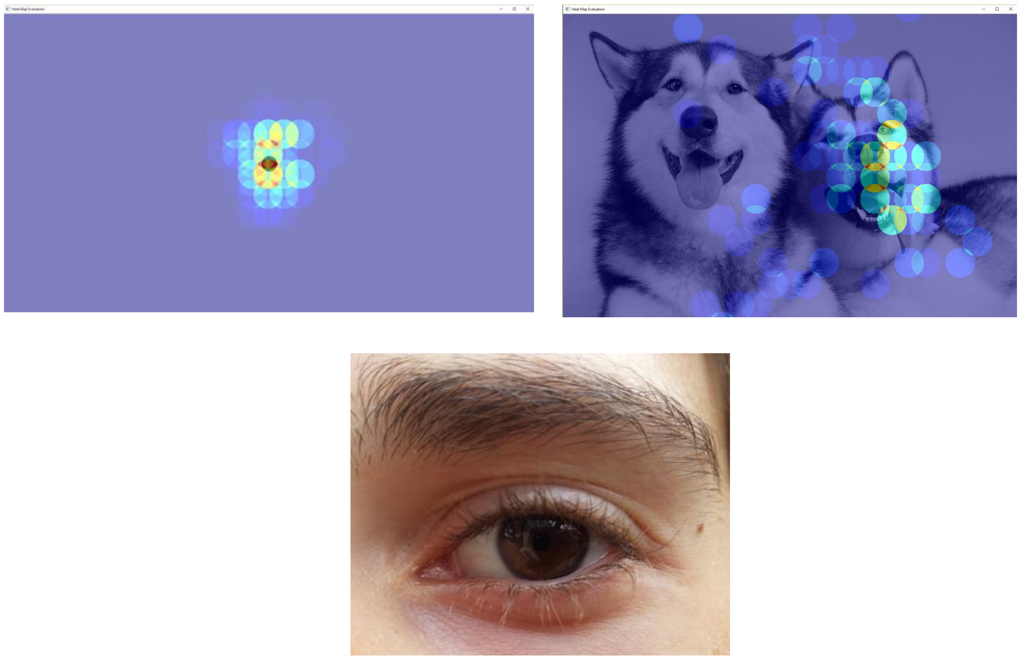


Figure 52 Measurement num. 7 results (top-left is test 5, top right is test 1, bottom is the eye of the tested subject)

Measurement number	8
Tested subject identification	7
Age, sex	48, female
Eye status	Nonreflexive glasses
Light conditions, time	Daylight, 11 AM

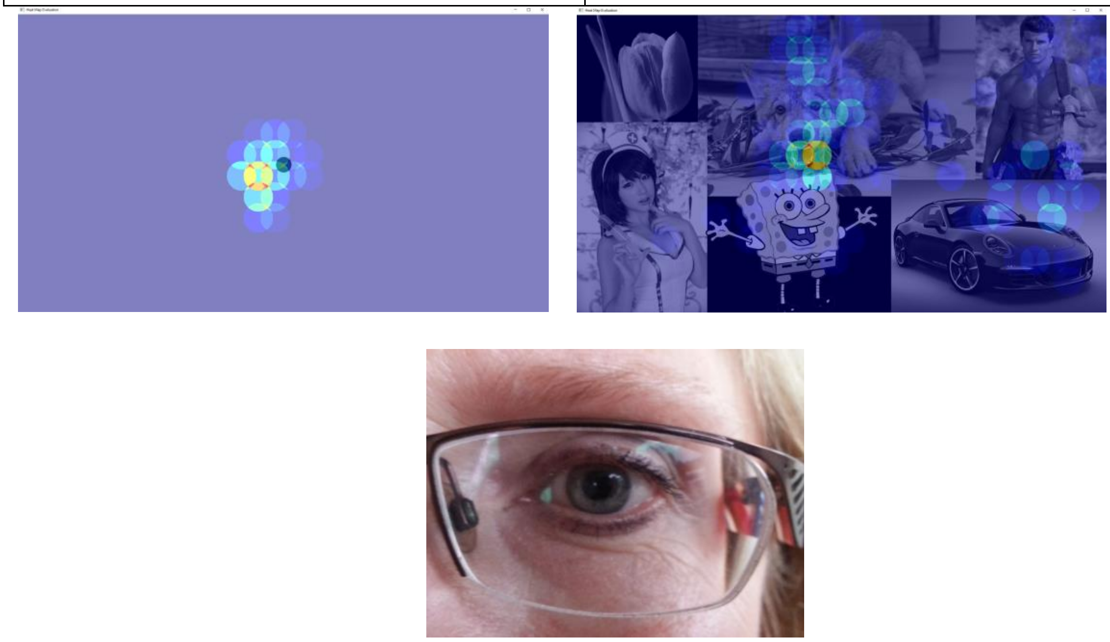
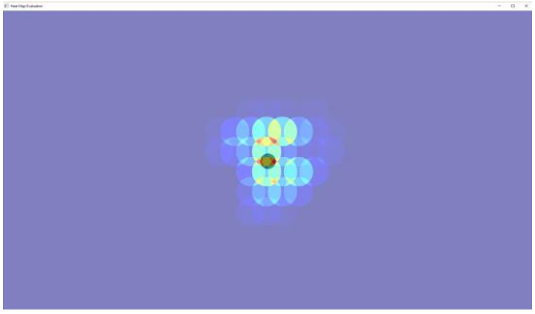


Figure 53 Measurement num. 8 results (top-left is test 5, top right is test 2, bottom is the eye of the tested subject – with nonreflexive glasses)

Measurement number	9
Tested subject identification	8
Age, sex	51, female
Eye status	Nonreflexive glasses
Light conditions, time	Daylight, 12 AM



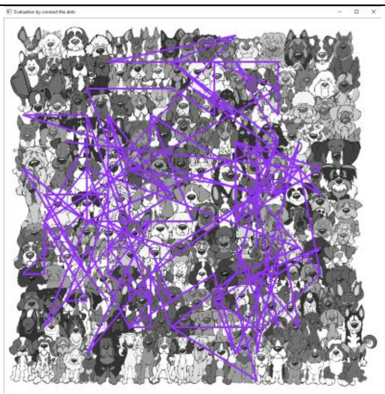
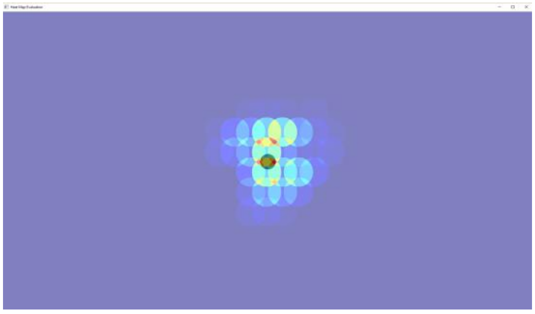


Figure 54 Measurement num. 9 results (top-left is test 5, top right is test 4-1, bottom is the eye of the tested subject – with nonreflexive glasses)

Measurement number	10
Tested subject identification	9
Age, sex	64, male
Eye status	None
Light conditions, time	Daylight, 10 AM



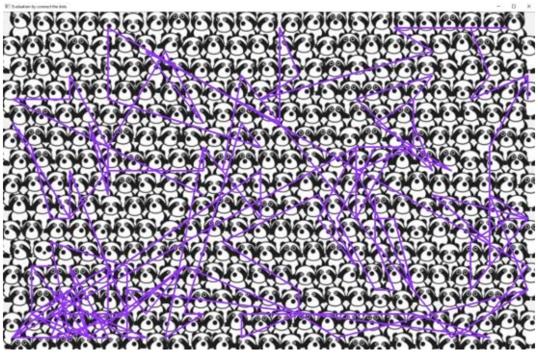


Figure 55 Measurement num. 10 results (top-left is test 5, top right is test 4-4, bottom is the eye of the tested subject)

Table 12 Times of Test 4 completion of testing subjects num. 5-9

Testing subject	Test 4-1	Test 4-2	Test 4-3	Test 4-4
Num. 5	8	42	31	77
Num. 6	4	12	28	27
Num. 7	25	16	16	84
Num. 8	66	58	95	7
Num. 9	19	68	73	16

Table 13 Average inaccuracy and maximum error comparison

Testing subject	Condition	Average inaccuracy	Average inaccuracy	Max error	Max error
Num. 5	Contact lenses	99 px	27.4 mm	201 px	56.0 mm
Num. 6	None	92 px	25.7 mm	233 px	64.5 mm
Num. 7	Nonreflexive glasses	89 px	24.8 mm	233 px	64.5 mm
Num. 8	Nonreflexive glasses	91 px	25.2 mm	233 px	64.5 mm
Num. 9	None	82 px	22.9 mm	192 px	53.2 mm

2.3 MULTIPLE MEASURING AND ACCURACY

To evaluate accuracy of the used method, multiple measurements of test 5 were performed while changing the conditions of the test surroundings.

For testing with no special conditions, the testing subject num. 1 was chosen. Five more measurements were performed to evaluate average accuracy and maximum error. From those average values were calculated (see table 14).

Table 14 Average inaccuracy and maximum error – condition "None"

Measurement identification	Condition	Average inaccuracy	Average inaccuracy	Max error	Max error
11	None	88 px	24.5 mm	192 px	53.2 mm
12	None	70 px	19.5 mm	185 px	51.3 mm
13	None	68 px	19.1 mm	189 px	52.7 mm
14	None	89 px	24.8 mm	210 px	58.4 mm
15	None	85 px	23.7 mm	196 px	54.5 mm
Average	-	80 px	22.3 mm	194 px	54.0 mm

For testing with nonreflexive glasses, the testing subject num. 2 was chosen. Five more measurements were performed to evaluate average accuracy and maximum error. From those average values were calculated (see table 15).

Table 15 Average inaccuracy and maximum error – condition "Nonreflexive glasses"

Measurement identification	Condition	Average inaccuracy	Average inaccuracy	Max error	Max error
16	Nonreflexive glasses	124 px	34.5 mm	272 px	75.6 mm
17	Nonreflexive glasses	95 px	35.5 mm	198 px	55.0 mm
18	Nonreflexive glasses	106 px	29.5 mm	212 px	58.9 mm
19	Nonreflexive glasses	90 px	25.0 mm	201 px	55.9 mm
20	Nonreflexive glasses	103 px	28.7 mm	217 px	60.3 mm
Average	-	104 px	30.6 mm	220 px	61.1 mm

Since measurement with reflexive glasses was imprecise and with high inaccuracy, no more measurements were performed.

To evaluate influence of contact lenses, the measurements of subjects num. 4 and 5 were used. From those average values were calculated (see table 16).

Table 16 Average inaccuracy and maximum error – condition “Contact lenses”

Measurement identification	Condition	Average inaccuracy	Average inaccuracy	Max error	Max error
5	Contact lenses	92 px	25.7 mm	212 px	58.9 mm
6	Contact lenses	99 px	27.4 mm	201 px	56.0 mm
Average	-	96 px	26.5 mm	207 px	57.4 mm

To sum up average inaccuracy and maximum error, the table 17 was constructed. It shows all twenty measurements, their conditions and results. The average values are calculated with and without measurement num. 4 (reflexive glasses).

Table 17 Average inaccuracy and maximum error comparison

Measurement identification	Condition	Average inaccuracy	Average inaccuracy	Max error	Max error
1	None	68 px	19.1 mm	189 px	52.7 mm
2	None	94 px	26.2 mm	201 px	56.0 mm
3	Nonreflexive glasses	106 px	29.5 mm	212 px	58.9 mm
4	Reflexive glasses	159 px	44.2 mm	375 px	104.1 mm
5	Contact lenses	92 px	25.7 mm	212 px	58.9 mm
6	Contact lenses	99 px	27.4 mm	201 px	56.0 mm
7	None	92 px	25.7 mm	233 px	64.5 mm
8	Nonreflexive glasses	89 px	24.8 mm	233 px	64.5 mm
9	Nonreflexive glasses	91 px	25.2 mm	233 px	64.5 mm
10	None	82 px	22.9 mm	192 px	53.2 mm
11	None	88 px	24.5 mm	192 px	53.2 mm
12	None	70 px	19.5 mm	185 px	51.3 mm
13	None	68 px	19.1 mm	189 px	52.7 mm
14	None	89 px	24.8 mm	210 px	58.4 mm

Measurement identification	Condition	Average inaccuracy	Average inaccuracy	Max error	Max error
15	None	85 px	23.7 mm	196 px	54.5 mm
16	Nonreflexive glasses	124 px	34.5 mm	272 px	75.6 mm
17	Nonreflexive glasses	95 px	35.5 mm	198 px	55.0 mm
18	Nonreflexive glasses	106 px	29.5 mm	212 px	58.9 mm
19	Nonreflexive glasses	90 px	25.0 mm	201 px	55.9 mm
20	Nonreflexive glasses	103 px	28.7 mm	217 px	60.3 mm
Average	-	95 px	26.8 mm	218 px	60.5 mm
Average without 4	-	913 px	25.8 mm	209 px	58.2 mm

The following table 18 consists of the times of completion of test 4. Each part of test 4 was evaluated separately and then the average time of each test was calculated.

Table 18 Times of Test 4 completion of measurements 1-10

Measurement identification	Time of Test 4-1 [s]	Time of Test 4-2 [s]	Time of Test 4-3 [s]	Time of Test 4-4 [s]
1	7	7	16	36
2	21	14	11	22
3	16	53	12	44
4	20	20	2	32
5	15	17	9	11
6	8	42	31	77
7	4	12	28	27
8	25	16	16	84
9	66	58	95	7
10	19	68	73	16
Average	20.1	30.7	29.3	35.6

Each average value was calculated as a sum of measured values divided by the number of tests. The average inaccuracy of testing was calculated to be 26.8 mm (which matches 95 px on the testing screen). The maximum error of testing is 60.5 mm (218 px) in average. The average inaccuracy of testing of subject with no special conditions (no glasses or lenses) was measured as 22.3 mm (80 px) and maximum error as 54.0 mm (194 px). Average inaccuracy of testing with nonreflexive glasses was 30.6 mm (104 px), maximum error was 61.1 mm (220 px). For the testing of lenses was determined as 26.5 mm (96 px) for average inaccuracy and 57.4 mm (207 px) for maximum error.