

MASTER'S THESIS 2020

Posner Cueing Paradigm in Augmented Reality Environment

Andreas Englesson

Elektroteknik
Datateknik

ISSN 1650-2884
LU-DS-EX 2019-XX
DEPARTMENT OF DESIGN SCIENCES
LTH | LUND UNIVERSITY



EXAMENSARBETE
Designvetenskap

LU-CS-EX: 2019-XX

**Posner Cueing Paradigm in Augmented
Reality Environment**

Andreas Englesson

Posner Cueing Paradigm in Augmented Reality Environment

Andreas Englesson
dat14aen@student.lu.se

June 24, 2020

Master's thesis work carried out at the Department of Design Sciences, Lund
University.

Supervisors: Günter Alce, gunter.alce@design.lth.se
Diederick C Niehorster, diederick_c.niehorster@humlab.lu.se

Examiner: Johanna Persson, johanna.persson@design.lth.se

Abstract

Posner Cueing Task is an experiment to investigate the effect different type of stimulus could have on the visual attention. This master thesis aimed to implement a Posner Cueing Task in an Augmented Reality using an integrated eye tracker. The thesis was divided into two distinct parts. One part to write code for synchronisation and communication between Microsoft HoloLens and Tobii eye tracker. One part to use this framework and devices to implement and run a Posner cueing task with subtle cues. The synchronisation was carried out using the UDP protocol sending data packages over the Tobii network. As the Tobii glasses were specially modified to fit onto the HoloLens the camera had been removed. This caused the need to be able to calibrate the eye tracker without the normal use of the front camera. A task that showed to be possible to solve by providing a digital calibration marker to the eye tracker.

The communication between the two devices were implemented and worked as expected with a fairly low latency. The Posner Cueing Task were then implemented and deployed to the HoloLens glasses using Unity and Visual Studio. The eye tracker broke just before the experiment were to be carried out. The code had to be adapted to not use the eye tracker. The experiment were then carried out with only the HoloLens as an input device.

The test result showed that a subtle cue had a similar effect on the participants' attention as a high contrast cue. The experiment also showed that a valid cue, one that showed at the same side as the target, yielded a lower RT than an invalid cue, one that showed at the opposite side of the target. Cue-target onset asynchronies is the time between that the cue disappeared and the target appears. This varied from 50ms to 650ms, where the lower ones had a positive effect on the RT. The effect decreased with time and yielded a negative effect for 650ms CTOA.

Keywords: Augmented Reality, HoloLens Generation 1, Tobii Pro Glasses 2, Posner Cueing task, eye tracking

Acknowledgements

A special thanks to Günter Alce and Diederick C Niehorster that helped me during the tiresome and prolonged process of writing this thesis. A lot of extra effort had to be put in by the two throughout the project due to never ending changes in conditions. Their ability to always come up with new solutions and possibilities when the previous one didn't work out were essential to the success of the project. Without this commitment and guidance the thesis would never been finished.

Contents

1	Introduction	7
1.1	Motivation	7
1.1.1	A real life scenario	7
1.1.2	The past and the future	8
1.2	Report Outline	9
2	Problem Statement	11
2.1	Goals	11
2.2	Proposed Solution	11
2.3	Scope of the project and its Limitations	12
3	Theoretical Background	13
3.1	Augmented Reality(AR)	13
3.1.1	Microsoft HoloLens	14
3.1.2	Development Environment for AR	14
3.2	Eye-Tracking	15
3.2.1	Tobii Pro Glasses 2	16
3.3	Human-Computer Interaction	17
3.3.1	Eye tracking based HCI	18
3.3.2	Visual Cues	18
3.3.3	Saccades	18
3.3.4	Microsaccades	18
3.4	Posner Cueing Paradigm	19
3.4.1	Posner paradigm	19
3.4.2	Subliminal perception	20
3.4.3	Exogenous/Endogenous Cueing	20
3.4.4	Inhibition of return	20
3.5	Related Work	21
3.5.1	Posner Paradigm	21
3.5.2	Eye-Tracking & Augmented Reality	21

4	Methodologies	23
4.1	Development process	23
4.1.1	Iterative Design process	23
4.2	Controlled Experiment	24
5	System Design	27
5.1	Architectural Overview	27
5.2	Implementation	29
5.2.1	Communication between hardware	29
5.2.2	Calibration without Hololens camera	31
5.2.3	Syncing environment	32
5.2.4	Round Trip Time	33
5.3	Posner cueuing Paradigm	33
5.3.1	Setting up the test environment	33
5.3.2	Participants	34
5.3.3	Experimental Design	35
5.3.4	Dependent Measures	36
5.4	Procedure	36
5.4.1	Participant Consent Form	36
5.4.2	Creation of record and participant	36
5.4.3	Calibration of the eye-tracker	37
5.4.4	Start Experiment	37
5.4.5	Experiment execution	37
6	Results	41
6.1	Description of the result	41
6.2	Data Analysis	41
6.3	Experimental results	42
6.3.1	Valid & Invalid cued trials	42
6.3.2	High contrast & Low Contrast Cues	43
6.3.3	RT depending on CTOA	44
7	Discussion & Analysis	49
7.1	Discussion & Analysis of the result	49
7.1.1	Posner paradigm	49
7.1.2	Threat to validity	50
8	Conclusions and Future Work	53
8.1	Conclusion	53
8.2	Contribution	53
8.3	Future Work	54
	References	55
	Appendix A Experiment Results	61
	Appendix B Form of consent	69

Chapter 1

Introduction

In the first chapter a background to the problem is given, as well as a proper problem formulation with goals and limitations.

1.1 Motivation

1.1.1 A real life scenario

Imagine encountering a for you totally unknown device with only a touch screen available for use. You click your way forward on the best of your ability only to find yourself back on the same starting screen over and over again. You start to realise that you are stuck and need help. Just so happen to be, a women behind you has used this device many times before and steps forward to guide you through each step. A scenario like this just took place when Skånetrafiken, the company in charge of the public transportation in southern Sweden, released their new mobile application during the year of 2019. The only problem to be that since no one else had used it before no one where around to help. The queues to the help desk seemed endless when frustrated passengers couldn't figure out how to buy their well needed tickets. Now imagine the same scenario as the first one, but you where using augmented reality glasses. Even if the room where full with people who had used the device endless number of times, no one could easily guide you since no one but you see the screen. You are left to be guided solely by the application in use. A combination of more and more people getting forced to use digital screens and the rapid evolvment of Augmented Reality(AR)/Virtual reality(VR) technologies raised an interest in how to help and guide people through during the use.

1.1.2 The past and the future

The idea of controlling a computer, untethered, and only using hand gestures has been a concept explored by sci-fi for a long time. Ranging from the movie *The Minority Report*, where a detective is controlling a set of monitors using gloves, to newer movies like in the *Marvel Cinematic Universe*, where Tony Stark (Iron Man) controls holograms using both audible and bodily gestures. Furthermore, the thought of Augmented Reality and holograms can be seen in early episodes of *Star Trek: The Next Generation*, *Star Wars*, and the likes. All of these systems, except partially for Tony Stark's, have one thing in common, bodily gestures for control. That common thread birthed the idea for this thesis.

Systems such as these offer alternative solutions to interacting with a computer, instead of the classical keyboard and mouse setup. The term for this, and all other information technology, interaction is called human-computer interaction, or HCI. The focus is to make the application usable by the users in an easy and understandable way. This sounds easy but could, if not implemented properly, just cause irritation and feel unnecessary complicated to the user. Examples for when you might want the no-contact system mentioned above might be in a sterilized environment, or a dirty one.

The technique is however in its infancy and hence has a long way to go before it is mature enough to be used by a broader audience. Concepts that normally applies in many areas in computer science becomes much more complex, or at least very different, when using AR/VR.

Head-worn devices(HWD) for AR have been used for some time, by and large for research purposes. But it's slowly making its way into the everyday user's home and offices. By implementing binocular eye-tracking technology to a HWD one could control an entire system using only the user's eyes. Imagine, if you will, a modern day living room. The TV is most likely smart, there is probably some type of smart device able to play music, there might be an Alexa standing in the corner. If we consider a really top of the line modern day home, even the lights are smart. With implemented eye-tracking technology in your HWD, it is now become a multi-remote for every single smart device in your vicinity, and perhaps even distant ones. What's to say you can't control your computer in the other room by virtual means? With HWD, eye-tracking and Internet of Things(IoT), the only limiting factor of your control is the hardware.

1.2 Report Outline

The report is divided into the following chapters:

1. **Introduction.** A background to the problem is given, as well as a proper problem formulation with goals and limitations.
2. **Problem statement.** The problem is presented with the proposed solution.
3. **Theoretical Background.** The common terminology and technology are introduced.
4. **System design.** The design of the coding pipeline and system is described. Arguments are brought forwards as to why choices regarding coding and methodology are what they are.
5. **Collected system implementation.** The system design, theory and techniques from previous chapters are put together into a complete system.
6. **Results.** The experiment is presented in detail as well as the result from the same.
7. **Analysis.** The results analysis and evaluations are presented. The analysis is mapped to the previous chapters to simplify reading.
8. **Conclusions and future work.** The work from this thesis is summarized. The main results, evaluations and analysis are presented. Thoughts on future work is discussed.

Chapter 2

Problem Statement

In this section we will state the goals, problems and proposed solution, as well as the project scope

2.1 Goals

This master thesis is divided into two distinct parts and hence has two different goals. The two parts will be implemented separately meaning that the second part will be implemented without the result from the first part of the thesis.

1. Design and implement communication between the Augmented Reality device Microsoft Hololens and a eye-tracking device Tobii Pro Glasses 2.
2. Implement and perform the Posner Cueing task in AR.

The implementation should be able to:

- Perform the Posner queuing task
- Communicate with as low latency as possible
- Produce an output with test data that can be analyzed externally

2.2 Proposed Solution

A variant of the Posner queuing task will be implemented with virtual objects through eye tracking techniques in AR. This is achieved by connecting the AR-capabilities of a Microsoft Hololens with the eye tracking of a pair of Tobii Pro Glasses 2.

By live-streaming the eye trackers gaze coordinates to the Hololens network, a gaze-guided cursor able to perform the Posner Queuing Task is to be implemented. When a user puts on

the Hololens, the user will be prompted to perform a quick calibration of the eye-tracker to be able to continue. This will be followed by a simple menu where the user can choose to start the Posner queuing task.

The task will be implemented using two boxes, one placed to the left and one placed to the right in the user's field of view. One of the boxes will after a random (in a closed interval) time be lightened up in a bright color, meaning that the user should look at that box as fast as possible. To implement the Posner queuing task the lightning up part could possibly be preceded by a subliminal cue, either of a valid kind or not. The time it takes for the user to look at the box would be measured and compared depending on if it were preceded by a subliminal cue or not. With the gathered data from the experiments, the theory about subliminal cues gaze-guiding capabilities can hopefully be confirmed.

2.3 Scope of the project and its Limitations

Due to technical issues with the eye-tracker this were not able to be used during the experiment. Therefore, all the the trials had to be carried out with only the Hololens. The effects on the trial is that the participant has to turn it's head and look at the target instead of gazing at it. The time it takes from the cue is showing until the participant looks at the target will be longer, but the difference between the cued and none-cued rounds will remain unaffected.

Chapter 3

Theoretical Background

This chapter serves to introduce common terminology and technologies that are the basis for this thesis.

3.1 Augmented Reality(AR)

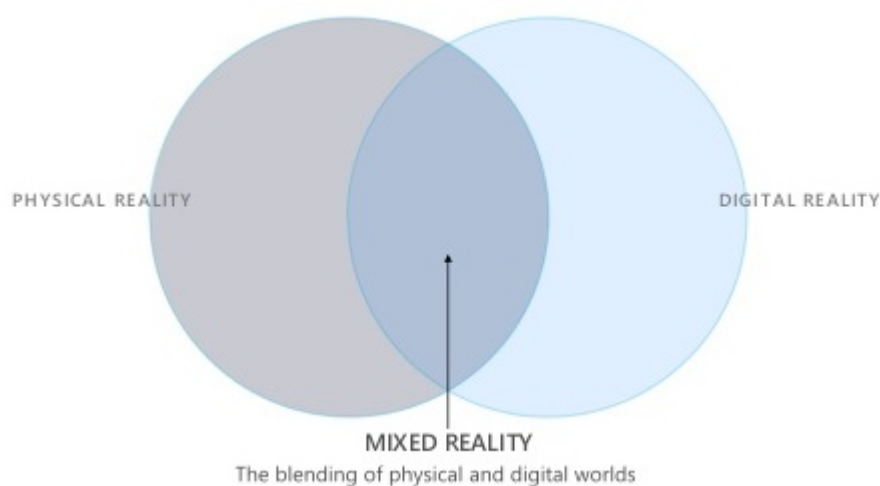


Figure 3.1: The mixed reality spectrum with AR and VR marked at either end. *Source: Microsoft*

Augmented Reality is a part of the Mixed reality spectrum, as seen in figure 3.1. It combines the physical world that humans live in, with the ability to add digital objects through holographic lenses or cameras. There are differing definitions of AR going around, with one being from Azuma et. al. stating that to be an AR system it must have the following properties:

- It must combine real and virtual objects in a real environment;
- It must run interactively and in real time; and
- It must register (align) real and virtual objects with each other. [1]

If one chooses to follow this definition it would mean that a system would have to geographically (for larger, outdoor systems) and geometrically align created digital objects with the real world, in real time. Any device capable of creating AR can do so by enhancing the natural world, be it with sound, video or GPS data. One could argue that, by a broader definition, listening to music on your phone is AR, as it augments the sounds you hear from your environment. The generated objects by a system can then be manipulated in real time by a user, making it very interesting for applying new technologies. This reality manipulation can be seen in the popular mobile app Snapchat ¹, and even in the Swedish post office app, PostNord ², that lets you view the size of incoming packages in AR if you have an apple device.

3.1.1 Microsoft HoloLens

The Microsoft HoloLens is a head-worn display that's also untethered. It has a semi-transparent lens, allowing the user to see both the physical world and the digitally created objects. The HoloLens allows for input using speech, gestures, and "gaze-tracking" by use of a head-direction vector, and output using visual and audio-based objects. The HoloLens field of view has been estimated at a field of view of $30^\circ \times 17.5^\circ$ [2], giving a diagonal of 34.73° using the Pythagorean theorem and it's 16x9 resolution. A mixture of visual studios and Unity is used to place digital objects inside the HoloLens viewport. The full documentation regarding mixed reality in Microsoft and the HoloLens can be reviewed at their own documentation website [3].



Figure 3.2: A user with a head-worn display attached to his head.

Source: Microsoft

3.1.2 Development Environment for AR

The development process for Augmented Reality is a bit special since it relays on two different programs in the process. Testing is therefor a bit different then "normal" programming where most of the test can be carried out in the integrated development environment(IDE). Since the application runs on the HoloLens some functionality can't be simulated properly in the unity engine, but needs to be tested out on an actual pair of glasses(or a virtual HoloLens simulator).



Figure 3.3: Workflow for AR development

¹Snapchat - <https://www.snapchat.com/>

²App Store - <https://apps.apple.com/se/app/id396871673>

Unity

Unity engine is the environment where all virtual objects is modified. It ties it all together so that virtual objects can trigger certain events, can run code or instantiate a whole new scenery. Basicly everything that the user sees through the glasses is modified using Unity Engine.

Unity uses something called scenes for holding a set of virtual objects. A scene is built up by a number of virtual objects that is places in that scene. Each object get assigned a position in the virtual space. It could either be a fixed position or relative position that can be changed or moved by the user. A user can be allowed to interact with objects in this virtual space leading to endless possibilities to build up scenery. In this thesis the only intractable objects accept for the menu will be the boxes that is used as targets for the eye gaze.

Visual Studios

When working in Unity one can chose between Visual Studios or JetBrains as the IDE. The most common one used today, and the one chosen for this thesis, is Visual Studios. The code is written in C# using .net core. Visual studio is well integrated in the workflow and changes made in visual studios is directly reflected in the unity engines environment. Visual Studios is used for two different parts, one is to write the code that is used and one is to deploy the code to the hololens.

Mixed Reality Toolkit v2

Mixed Reality Toolkit v2(MRTK) is a project that is driven by Microsoft to help and accelerate the development process for Mixed Reality applications. As an extensible framework it makes it easy for developers to changes and modified core components of the application. But most importantly it provides all the basic building blocks needed and an in-editor simulation to allow for a fast and seamless development process. In this thesis a number of building blocks from this tool-kit is used i.e. pre-built menus, different boxes and materials.

3.2 Eye-Tracking

Eye-tracking in its simplest form is the tracking of a subjects eye movements to extrapolate where that person is looking, the time they spend looking in that direction and how their eyes travel over objects. There are a couple of companies today that offer different kinds of products and solutions for this type of tracking, each company using its own special features to try and make their own product more precis than competitors. Metrics that are often used in the products are, to name a few, gaze direction, time to first fixation and number of fixations, blink rate and duration of blinking, and the diameter of the pupil.

Eye-tracking has immense possibilities of use, and there are already plenty of areas where it's being used to great effect, such as helping someone with a handicap control their computer using only their eyes, or spelling out words on an eye-gaze controlled speech pad, for instance with the EyeMobile Plus³. Bonino et. al. lists several distinguishing features of eye tracking

³EyeMobile Plus - <https://www.tobiidynavox.com/en-GB/devices/eye-gaze-devices/eye-mobile-plus/>

[4]:

- it is *faster* than other input media, as observed by Ware and Mikaelian [5]; when a user is about to operate any mechanical pointing device, she usually looks at the intended destination;
- it is *easy* to operate, as apart from extraordinary circumstances, no particular coordination or training is required to observe objects;
- it show where the *focus of attention* of the user is located; without actually intending to, a users gaze can be interpreted as the current focus of their attention;
- it suffers from *Midas Touch* problem: expecting to be able to observe objects without having to interact with them. By using different interaction techniques, such as dwell time or blink select, this problem can be mostly overcome;
- it is *always on*; there are currently no, in present society, natural ways of indicating that one wants to interact with an input device, as with a mouse;
- it is *noninvasive*; by only using eye tracker input, a user can relax their arms and hands, causing less physical fatigue;
- it is *less accurate* than other pointing devices, such as a mouse.

3.2.1 Tobii Pro Glasses 2

Eye-tracking techniques

Pupil Centre Corneal Reflection (PCCR)

Pupil Centre Corneal Reflection is the most common method among eye-tracking devices. By illuminating the eye with a light source, a camera can take detailed pictures of the eye. Image processing algorithms are then used to locate the center of the corneal reflection, and the center of the pupil. Tobii uses near-infrared illumination to create the reflections in their glasses.

With the help of advanced image-processing algorithms, Tobii is able to combine these reflections with a 3D model of the eye to estimate the eyes position in space as well as the point of gaze with high accuracy [6].

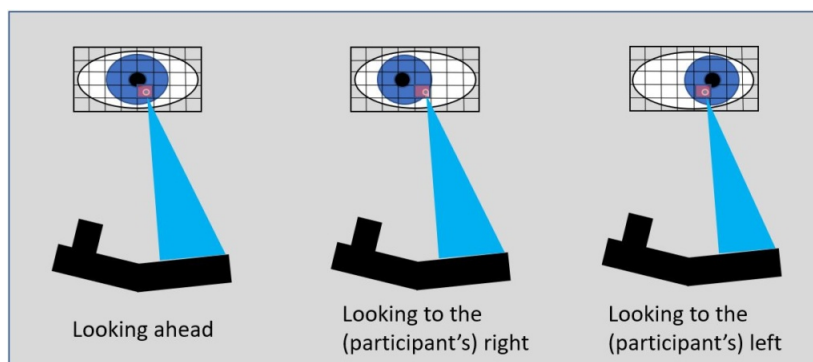


Figure 3.4: Pupil Centre Corneal Reflection

Source: Tobii AB

Dark/Light Pupil Tracking

There are two different ways of illuminating the pupil when using the PCCR technique in eye tracking. In both methods an illuminator is placed near the eye to cause a reflection from it. The difference between the two methods is the placement of this illuminator. When using the bright pupil tracking, the illuminator is placed as close to the optic axis as possible to make the pupil light up, this is also what causes red-eye in a camera using flash. While using the dark pupil tracking, the illuminator is instead placed away from the optic axis, making the pupil appear darkened instead, as seen in figure B.1.

Which technique to use depends on many factors such as age, environmental light, and ethnicity. Many of Tobii's eye trackers have both illumination systems built into them, and during calibration the system decides which of them produce the most accurate result. The Tobii pro glasses 2, however, only utilizes dark pupil tracking [7].

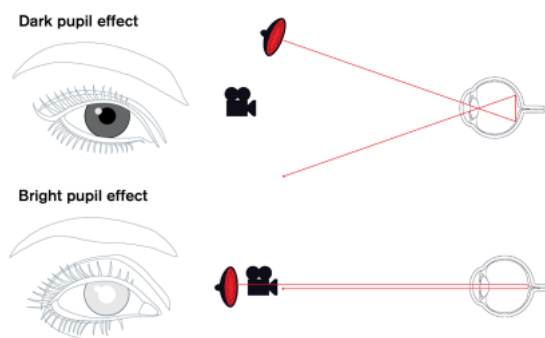


Figure 3.5: Illuminator placement for dark and bright pupil tracking respectively
Source: Tobii AB

3.3 Human-Computer Interaction

Human-Computer Interaction, is a multidisciplinary field of study within computer technology design, and even more so the interaction between humans, or users, and the computer systems. As technology developed from the massive computers filling up entire buildings to fitting in the palm of a hand, HCI has adapted and evolved as well. As stated by John M. Carroll, one of the founding fathers of HCI [8]:

[...] it no longer makes sense to regard HCI as a specialty of computer science; HCI has grown to be broader, larger and much more diverse than computer science itself. HCI expanded from its initial focus on individual and generic user behavior to include social and organizational computing, accessibility for the elderly, the cognitively and physically impaired, and for all people, and for the widest possible spectrum of human experiences and activities.

As humans interact with computers, so can computers interact with humans. Examples of this are when a mobile device tells the user that there are restaurants nearby that would fit

the users previous tastes, this is called *implicit* interaction - i.e. humans interact with non-deliberate input and receive an output from the system. If, on the other hand, the user inputs the question “Are there any restaurants for me nearby?” to the system and receives an output, that’s *explicit* interaction.

More than a quarter-century ago, Mark Weiser claimed that technology demands active attention, and that we will never truly experience the potential of technology until it moves into the peripheral [9]. That statement, and the amount of attention required to perform HCI has been looked into since then, for example by Saskia Bakker who researched the use of peripheral attention in technology devices [10].

3.3.1 Eye tracking based HCI

When using a eye-tracker and implementing an "eye mouse", controlling the mouse with the eyes, new factors related to eye movement measures play a vital role. For this experiment where the focus is to measure how the attention could be shifted using stimulus saccades is the most important one. There could be different stimulus to trigger the attention shift for the participant. Those are often called Visual Cues when related to the pointer cueing task.

3.3.2 Visual Cues

Visual cues are ways of actively interacting with a user to shift their attention. They can be explicit, subliminal or not there at all, where the range for what classifies as explicit or subliminal might vary between users. These cues can be used to tell a user where there are interesting points in a system, as with bottom-up/top-down processing, one can actively shift attention to something by implementing a more explicit cue [11].

3.3.3 Saccades

A saccade is a rapid movement of the eye between point A and B. Voluntary saccades occurs in many situations such as orientation and reading. Saccades can be divided into two main categories, stimulus-elicited and voluntary saccades. A stimulus-elicited saccade is triggered by a stimulus of some sort, i.e. a movement to a point outside outside the point of fixation that draws the attention of the eye. A voluntary saccade is one that is deliberately performed by the person i.e. someone is asking the person to look left. This concept is tightly tied together with exogenous- and endogenous cues that is used in the Posner paradigm.

3.3.4 Microsaccades

To prevent the vision to fade during a period of fixation our eyes is constantly moving which is called fixational eye movement. There are three different types of this depending on the size of the movement, Microsaccade(Largest), drifts and tremor(smallest).[12] A microsaccade is a fast and small involuntary saccade that occurs during a prolonged period of fixation. To differ between a microsaccade and a saccade it is not enough to look at the magnitude of the saccade alone, since a voluntary or exploratory saccade can be as small as a microsaccade. In fact, it is not possible to differ the two type of saccades by physical characteristics alone.

But microsaccades cannot be differentiated from saccades according to their magnitude alone, as exploratory or voluntary saccades can be the same size as microsaccades. Indeed, it is not possible to differentiate saccades from microsaccades according to any physical characteristic. For this reason, one cannot know whether a small-sized saccade constitutes a fixational microsaccade (and thus it is part of a fixation period), or an exploratory, non-fixational saccade.

3.4 Posner Cueing Paradigm

3.4.1 Posner paradigm

In 1980 Michael Posner published an article in the Quarterly Journal of Experimental Psychology named Orienting of attention[13]. He there presented the Posner cueing paradigm that since has been one of the most important paradigms in the studies of visual attention. The paradigm contains a task where a target is presented that then is to be located and reported by the participant. The target can appear in two different locations and could possibly be preceded by a cue stating. The cue marks one of the two location where the target can appear and tells the participant that the target probably (with some percentage of validity) is to be presented at this location. The result can be measured in different ways where the most normal way being to measure response times of the participant.[14] By comparing the response time for locating the target between trials that were preceded by a visual cue and those that were not, conclusions that visual cues enhance the performance in detecting a target can be drawn.

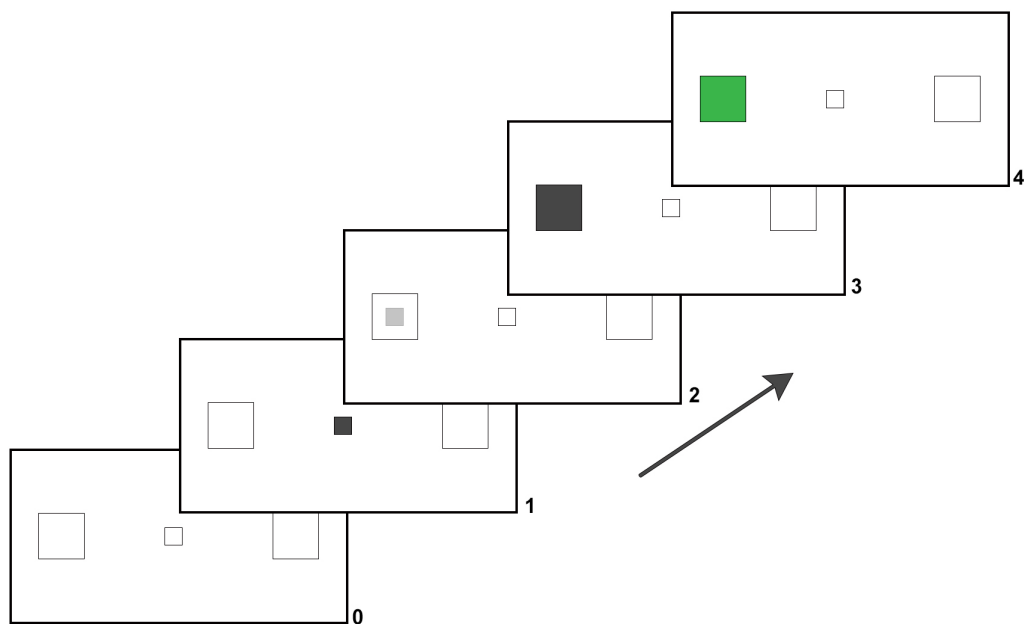


Figure 3.6: Sequence of Events

3.4.2 Subliminal perception

In Detection Theory an important field of research is that of Threshold Models and Choice Theory. Thresholds can have different meanings depending on the context of use. Here we use the term a threshold stimulus according to the following definition : *A threshold stimulus is one that can barely be discriminated from the background or another stimulus.*[15]

The implementation of the Posner paradigm in this thesis is a variant of a subliminal perception experiment. In a subliminal perception experiment a figure of some form is presented x times of the trials but not presented at all in y trials. [15]

Stimuli Types

The Posner paradigm uses two different type of stimulus that is derived from this theory, e.g. High- and Low Cues. A high cue is a stimuli that is just above the threshold while a low cue is one that is just below the same.

3.4.3 Exogenous/Endogenous Cueing

Visual attention and the ability to shift and keep focus at specific areas is a vital part of human cognition. The direction of the focus in visual space can be driven either by external stimuli or voluntarily directed to a location. [16]. The two type of triggers that causes the direction of the attention has many different names depending on context and taste. Goal-driven attention is often refereed to as endogenous attention or top-down. External stimulus-driven attention is often refereed to as exogenous attention or bottom-up[17][18].

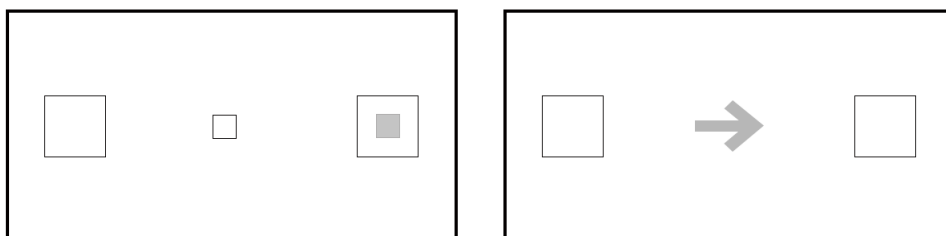


Figure 3.7: Exogenous Cue & Endogenous Cue

The two types could both used in the implementation of a Posner paradigm depending on the desired goal of the experiment. A endogenous type of stimuli typically consists of an arrow head in the center of attention pointing in the direction of the likely position of the target. A exogenous type of stimuli would be a stimuli placed directly at the likely position of the target.

3.4.4 Inhibition of return

Inhibition of return is the were in the early stages called inhibitory aftereffect and where first described in 1984[19]. This is the foundation for Posners work. If a location is validly cued the reaction time for a person to detect a new object on that location within a certain time period is shorter than that for a object in a non cued location. A common experiment to prove the Inhibition of return is the Posner cueing task which this thesis is centred around.

3.5 Related Work

3.5.1 Posner Paradigm

Posner Paradigm being one of the most well used experiments for shift of visual attention the number of different implementations are uncountable. The implementation details varies greatly between different experiments. Kullberg and Lindkvist [20] wrote a bachelor's thesis with regard to gaze guidance in augmented reality systems, using the same set-up of a Hololens with Tobii Pro Glasses 2 as will be used in this thesis. The communication between the devices varies as well as the implementation of the Posner paradigm. Kullberg and Lindkvists implementation used two independent variables as the subtle cue and the user. The experiment however seems to have a few flaws that makes the validity of the experiment questionable. How the transparency of a subtle cue is set is not mentioned, no invalid cues is used as references to the valid cues nor is the number of each trial equally distributed.

MacInnes, W.J. & Bhatnagar, R[21] performed a experiment using Posner cueing paradigm with the same Blocked Cue-target onset asynchronies(CTOA) condition will be used in this thesis meaning that the CTOA is varied in an interval using some different fixed numbers, e.g. 50ms, 250ms, 450ms, 650ms in this implementation. The big difference in this paper are the use of four different possible locations of the cue and target instead of the two used in this implementation. Their hardware varied in that they used a standalone eye-tracker and presented the cues on a LCD monitor instead of using a Head-worn display. The result from that experiment is interesting since we expect a result similar to their conclusion that states(IOR meaning inhibition of return):

As predicted, we found significant facilitations at early (50ms) CTOA. This SRT advantage was absent by 250ms CTOA and became a cost at later CTOAs (450+) reflecting IOR at the cued location.

3.5.2 Eye-Tracking & Augmented Reality

Using eye gaze systems to control, or interact, with your environment is a large field of study in medical technology. For instance in Shi et. al. [22] where an eye gaze system was implemented to control ICT, Information and Communication Technology, objects in the users home. Or the DOGEye system [4], developed by Bonino et. al., an eye tracking based home control application designed with the COGAIN guidelines [23] in mind. COGAIN(Communication by Gaze Interaction) being an association aiming to promote and development and research in the field and has released design guidelines for gaze interaction systems.

In a broader sense, work with smart home environments has been consistently growing for the last couple of years. Controlling these with augmented reality has been proposed by Marques et. al. [24]. Other suggestions for controlling a smart home include motion matching and smart watches [25] or ZigBee [26] for users with special needs.

Additionally, Majaranta et. al. [27] researched a gesture-based gaze interface by hinting gaze patterns to a user, that in turn prompted a command. Moreover, Köpsel et. al. researched gaze versus gesture interaction combined with different feedback modalities [28].

Chapter 4

Methodologies

4.1 Development process

4.1.1 Iterative Design process

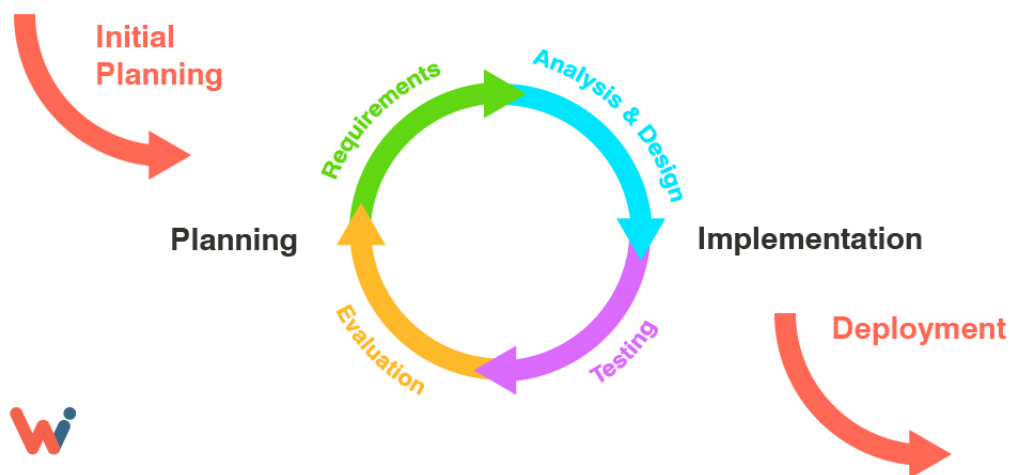


Figure 4.1: Iterative design process source: <https://wishdesk.com>

Initial Planning

All projects start with a overall planing process, both agile iterative projects as well as waterfall like projects. During the planing process requirements of the product to be developed is needed to be gathered if not already provided by a customer. The initial planning process during the master thesis included theory studies of subjects related to the

topic. This included studies of previously made implementations of the Posner paradigm as well as Tobii eye tracker and Hologens coding environment and coding principles.

Requirements

The requirements were deducted from how a Posner paradigm is implemented[13]. This were combined with real time programming principles to minimize latency and other factor which could possibly affect the validity of the experiment. This step included both brainstorming sessions as well as literature studies.

Analysis & Design

After the planning is done the projects business logic needs to be described and the different tasks needs to be clarified and time estimated. This task made sure that the time plan set for the master thesis were reasonable and that there were enough time deducted to every phase of the project.

Implementation

This is the development part of the process. The development tasks from previous step in the iterations is to be implemented to fulfil the requirements. This is tightly connected to the other phases, but most tightly connected to the testing phase. Written code is deployed and tested in very short iterations.

Testing

When the development of the code is done, it is necessary to perform testing to find and fix all the bugs and errors. As previously stated this where done in really short iterations so the testing and coding where in reality not clearly divided into two steps. However a more clearly divided testing step often occurred in the end of each iteration. This where usually a more integration type of test to evaluate the functionality of the different components run together.

Evaluation

The last step of the iteration is the evaluation. The application where evaluated and the functionality where thoroughly tested to see if all the requirements where met. A new iteration where then started if more development where needed.

4.2 Controlled Experiment

Easterbrook et.al describes a controlled experiment in the following way:*A controlled experiment is an investigation of a testable hypothesis where one or more independent variables are manipulated to measure their effect on one or more dependent variables.*[29] This is used to determine if there exists a cause-effect relationship exists between variables and how they are related to each other. A controlled experiment consists of at least two variables, where

each combination of those are called Treatments. To conduct an experiment each treatment is tested on a person, called *subject*, and the effect of the treatments is measured.

Chapter 5

System Design

In this chapter, the high level design of the system is described. Arguments of the various design choices are brought forwards as well as reflections around the chosen methodologies.

5.1 Architectural Overview

The System used for performing the whole Posner paradigm and associated analysis is built up by three main components, Microsoft Hololens, Tobii Pro Glasses and a computer. The two main components to perform the experiment is the Hololens and the Tobii Pro Glasses. Those two units are also physically connected to each other so that the glasses is attached to the Hololens. The computer is used to analyse the result of the experiment, and hence don't play any active roll in the experiment.

Head-Worn Display (Microsoft Hololens)

The Hololens is the central part of the system as it both runs all the necessary code and acts as a server for the UDP communication. This includes running the experiment as well as running the code for receiving and merging gaze-data from the eye-tracker. The Hololens is responsible for initiating the communication between the units by sending POST requests to the API endpoints exposed by the pro glasses. The Hololens also controls the virtual objects in the augmented reality.

Eye Tracker (Tobii Pro Glasses 2)

The Pro Glasses has two divided functions in the implementation. It acts as an eye-tracker and it provides a WiFi-connection through which all communication between the systems are carried out. The eye-trackers data is streamed to the hololens using UDP communication. The Tobii Pro glasses used in this implementation are modified to fit our needs e.g. to fit the Hololens nose attachment. This is achieved by 3D-printing a frame for the eye tracker that can be attached to the Hololens, in the process removing the front camera.

Computer

The computer no longer has any important role as a server, as in previous set up. This leaves the computer with the sole purpose of evaluating the validity of the test result and analyze the result of the same. This makes it debatable if its really a part of the system or not. The fact that test result can't be open in a good way, nor so being analysed at all, on the Hololens alone, the conclusion where that the computer still is a vital part of the set up.

Baseline System

The work in this thesis is based around Kullberg and Lindkvist [20]. In this implementation, as well as in theirs, the system is built with three different units communicating over wifi using different protocols. In their thesis they managed to connect the Tobii Pro glasses 2 with a Hololens and send gaze data from the glasses to the Hololens. This was achieved by connecting the Hololens to the glasses via wifi, allowing for *http* communication between the two. They examined the conjunction between reaction time and subtle cues, giving them the need of a computer as a server to store and analyse the data. This was achieved by connecting the Hololens to the computers network, and send the test results through *TCP*. The Tobii glasses were connected to the network through a *UDP* protocol. The different protocols are visualised in figure 5.1. However, this previous work did not involve any live feed of gaze data from the eye tracker and the AR glasses. Neither did it involve any working calibration of the eye-tracker. This implementation of the system includes the lacked futures from their implementation

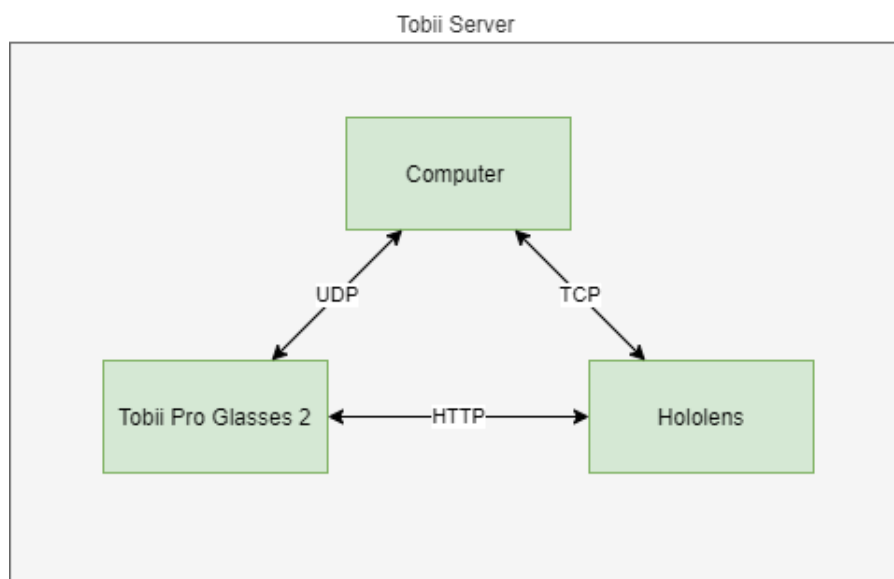


Figure 5.1: The communication protocols

New System

The new implementation is using the same hardware as the Baseline system. However, the communication is performed differently between the components. As the new system is based on direct communication between the Hololens and Tobii Glasses, it need to be able to read and merge the coordinates live from both systems and use them to perform the Posner paradigm. The structure and communication protocols used can be viewed below and red

about in the succeeding chapter.

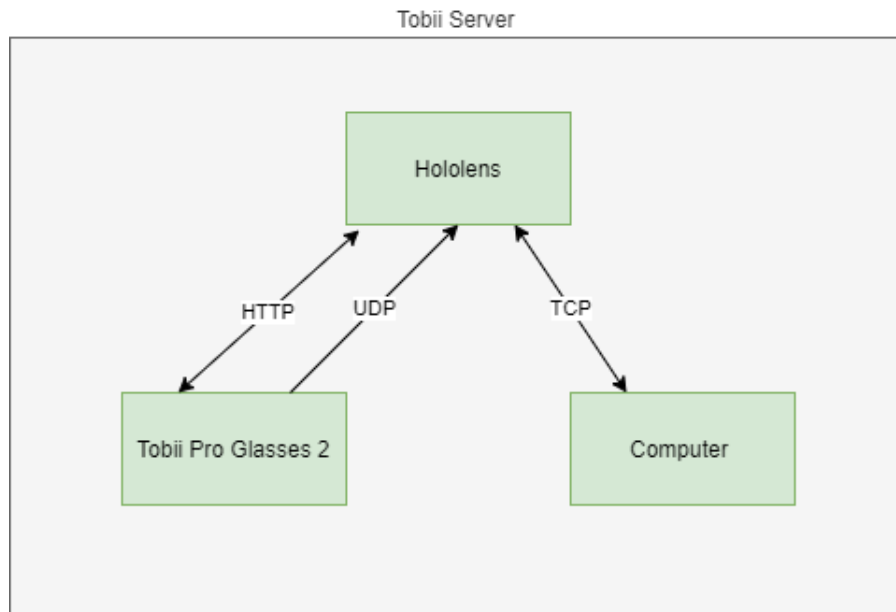


Figure 5.2: The communication protocols

5.2 Implementation

5.2.1 Communication between hardware

This chapter will describe how the system was implemented to fulfil the requirements and motivate the choices that were made. Tobii Pro Glasses 2 has four API endpoints including POST API, REST API, Livestream API and Discovery API for different purposes. The two used in this implementation is the REST API and the Discovery API.

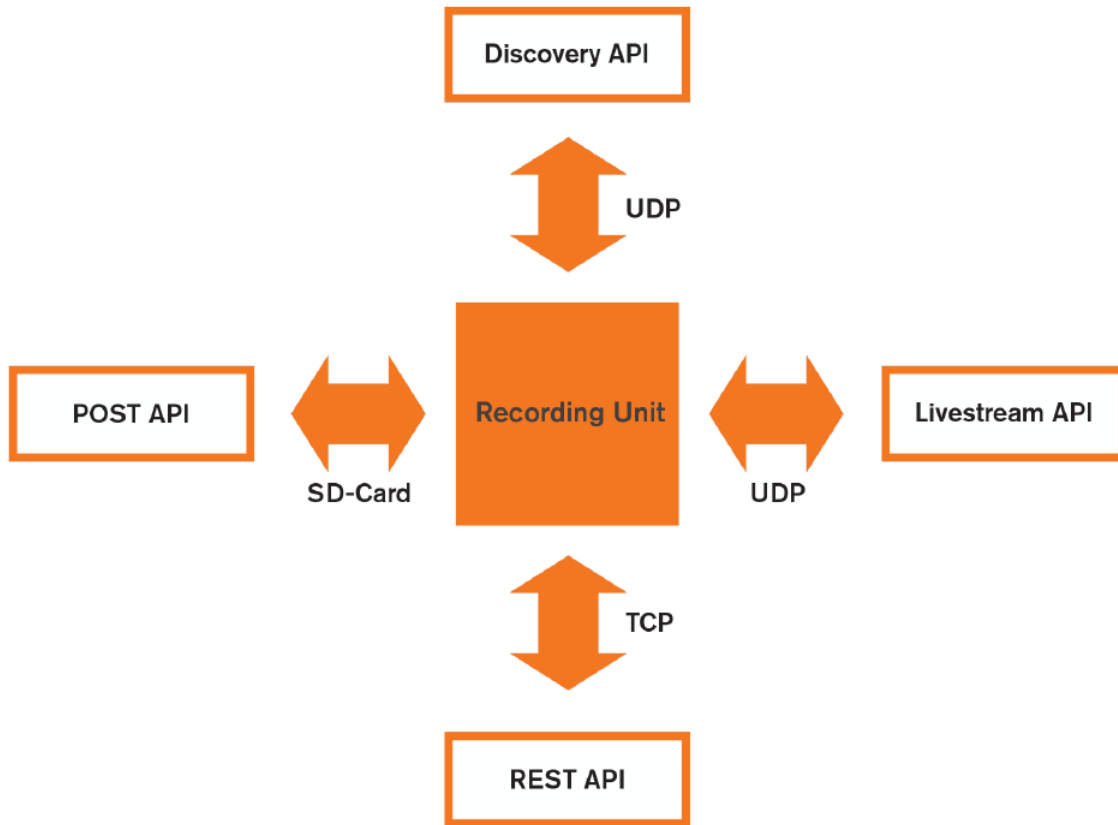


Figure 5.3: Tobii Communication Endpoints *Source: Tobii Pro Glasses 2 API 1.3*

Tobii Pro Glasses 2 & Hololens

UDP Communication

The direct, near real-time, communication is achieved by using a client server architecture between the two devices. In the implementation the Hololens acts as a server and the tobii pro glasses as a client. This differs from the baseline architecture where the computer acted as the server and the Hololens as the client. To get the information needed for setting up a connection to the Tobii Pro Glasses a Discovery-ping is sent on the UDP port 13006. The response contains all needed information as a json object:

```
{
  "type": "identity",
  "class": "glasses2",
  "name": "MyTobiiG2",
  "version": "1.0.4-246-elefantora-g35a0b15",
  "id": "TG02B-080104031171",
  "port": 80,
  "interface": "eth0",
  "ipv4": "10.46.16.4"
}
```

Figure 5.4: Doiscovery response *Source: Tobii Pro Glasses 2 API 1.3*

Stream sockets are created in both ends which allows for *UDP* communication between the devices instead of between the computer and the glasses. Since the eye-gazing data is used in near real-time by the Hololens and never stored anywhere, the *UDP* protocol without any form of error handling is sufficient. Lost packages will only lead to a small delay in the cursor movement before the next package is sent from the eye-tracker. Due to the use of *WiFi* communication, and in some extent the low performance of the Hololens, some delays will be encountered. The impact of this delay is estimated to be kept low and not affect the system to the extent that it could jeopardize the validity of the experiment.

HTTP Communication

The communication between the two devices is carried out over *TCP* using *HTTP* protocol. To control the Tobii Pro Glasses 2 the *REST API* endpoint are used. As the name implies it is a *RESTful* web service using *json* objects for the data transfer. This allows for an easy communication using *HTTP POST* and *HTTP GET* requests. This endpoint allows for operations such as create projects, start and stop calibrations and recordings, but it can also be used to retrieve the status of the glasses and general information about the unit.

5.2.2 Calibration without Hololens camera

When calibrating a standard version of an Tobii eye-tracker it is using the built in camera. A calibration card is held up approximately a meter in front of the user that is asked to look at the center of the same. The camera locate and calculate the coordinates of the calibration card. At the same time the eye tracker is measuring your gaze coordinates (and a various of other needed measurements) and performs the actual calibration.[6] This procedure is not possible since the camera of the eye tracker has been removed to fit into the Hololens. Instead an alternative way is used which takes advantage of the endpoints that are exposed by the Tobii pro glasses.

```

{
  "ts":1113243866,
  "marker2d":[0.5678, 0.2311],
  "marker3d":[31.322,27.654,22.442],
  "s":0
}

```

```
POST /api/calibrations/<ca_id>/marker/<x>/<y>/<z>
```

Figure 5.5: CalibrationEvent

Source: Tobii Pro Glasses 2 API 1.3

Using REST API we can use a POST request containing the coordinates of a virtual point in space. The virtual point is the representing the calibration card. Instead of Tobii calculating the position of the calibration card using the built in camera it gets the x,y,z coordinates as an input through the request. The procedure then continues following the same steps as a "normal calibration". A normal calibration means that the Tobii pro glasses continuously receives images from the front camera and recalculates its distance from the glasses. To simulate this the POST request continuously sends a stream of Calibration events to the Tobii Pro glasses. Normally this process, receiving images from the from camera and analyze the distance, continues for 7s or until the calibration process returns a state of failed or success. Since Hololens doesn't know about the state of this process it will continue to send Calibration events until told otherwise. A return message from the Tobii Pro glasses is therefor analysed between each step to see if the calibration state has changed or not. This particular implementation of the Calibration Event places a virtual "calibration card" at the coordinates 0, 0, 1100 in space, where a virtual box also is places,. This equals to the center of the view and 1.1m in front of the hololens.

5.2.3 Syncing environment

Synchronization between the eye-tracker and the Hololens is vital since it is two separate units communicating over a connection where a delay is present. A synchronized communication is achieved between the two systems by using synchronisation events that gets sent. A synchronization event is simply a type of json object that can be sent via a POST request to the eye tracker. The json object contains three fields containing information about the type of event and where it is used. The most important field is the ets field where an external timestamp can be stored. Here the ets contains the internal timestamp from the hololens. The tobii pro glasses then assigns its own internal timestamp to this message, this is stored on the tobii pro glasses internally and can later be used for synchronization purposes.

```
POST /api/events HTTP 1.1
```

Sample Json data that can be sent:

```

{
  'ets': 67967988,
  'type': 'BigEvent',
  'tag': 'some data which could be one-line formatted json, xml or whatever'
}

```

Figure 5.6: SyncEvent

Source: Tobii Pro Glasses 2 API 1.3

5.2.4 Round Trip Time

The Round Trip Time(RTT) for a message sent in the system is calculated. This calculation is implemented using the sync event available in the Tobii API. Ten sync messages is sent to the tobii in the system to estimate the delay between the two units. The tobii pro glasses returns a message to the hololens acknowledging the event. When the message gets received from the tobii pro glasses the time difference between the send event and the receive event is calculated and corresponds to the RTT for a message between the two units. Since the setup is relying on messaging using wifi and http protocol there are possible variations in the delay. To minimize the possible effects of this the RTT is calculated ten times in a row, where the lowest value is used as the RTT through out the test. The RTT is used as an estimate, or a best guess, and not as a value for exact calculations. This is the reason that an RTT value calculated once and then used throughout the test can be accepted instead of more advances calculations off RTT during run-time.

5.3 Posner cueing Paradigm

5.3.1 Setting up the test environment

Find the detection threshold

One variant of the Posner paradigm uses subliminal cues as stimuli to examine the relations to IOR. To make any claims and make conclusions about the effect of subliminal stimuli its important to find the correct threshold. This can't be a fixed hard coded value in the test used for every person, even though most persons would end up with approximately the same value. The reasons for this are many. Each person has variations in their eye sight, meaning that one test person could find a cure clearly visible while another test person couldn't see it at all. This would lead to a test environment where no conclusions could be made at all. Secondly the lightning conditions between two tests could change. Even though a test environment should be as static as possible, changes could occur making the circumstances different for two test persons.

A variation of "the method of adjustment" [14] is used to find each participants detection threshold. In this variation each participant is asked to perform eight rounds of "calibration". Half of the rounds starts with a clearly visible cue and half of the rounds start with a cue that is not visible at all. The participant is then asked to change the transparency in either direction so that it is just barely visible, meaning that if they increase the transparency just one step, the cue is invisible to them. The average of these eight trials is calculated and used as the participants detection threshold.

Test Sequence

Each test participant is performing a test that is build up by a few different parts. This is stored in a class called PosnerTrial which contains the foundation elements of a test. A test contains of 192 of trials divided into two rounds with 96 trials each, one round with high contrast cues and one round with low contrast cues. A trial is a round of the test and contains of a stimuli, a marker and corresponding directions of the two. To make the test

valid the number of each type has to be equally divided but in a random order so that the outcome of next round can't be predicted. This is achieved by creating a list with the correct amount of each stimuli type and the correct amount of each marker direction. This is then shuffled in a random order.

Test Data Model

To store and analyze the test data this has to be stored somewhere. This is done in a class called named `TrialData`. This class contains all the needed variables of the test and is instantiated once for every new round of trial.

```
[SerializeField]
public int ParticipantID { get; set; }
[SerializeField]
public int TrialNumber { get; set; }
[SerializeField]
public DateTime trialStartTime { get; set; }
[SerializeField]
public DateTime trialStopTime { get; set; }
[SerializeField]
public DateTime cueShowTime { get; set; }
[SerializeField]
public DateTime cueDisappearTime { get; set; }
[SerializeField]
public DateTime boxShowTime { get; set; }
[SerializeField]
public DateTime boxDisappearTime { get; set; }
[SerializeField]
public DateTime fixationShowTime { get; set; }
[SerializeField]
public DateTime fixationDisappearTime { get; set; }
[SerializeField]
public int orderOfTest { get; set; }
[SerializeField]
public long cueDurationTime { get; set; }
[SerializeField]
public long fromFixationToTarget { get; set; }
[SerializeField]
public long reactionTime { get; set; }
[SerializeField]
public long timeForGazeHitEvent { get; set; }
[SerializeField]
public string boxDirection { get; set; }
[SerializeField]
public string typeOfCue { get; set; }
[SerializeField]
public bool validCue { get; set; }
```

Figure 5.7: Data fields to be stored

5.3.2 Participants

Thirteen participants took part of the experiment.

Number of men: 8, number of women: 5, age range: 24-65, mean: 34.7 years.

Three participants had impaired vision but wore glasses during the trials and hence should not affect the result.

No one of the participants had a prior knowledge to the experiment neither any experiment using a hololens. However they all had to conduct a small test experiment prior to the real experiment to get used to the surroundings and in return minimize errors during the first rounds of the experiment.

5.3.3 Experimental Design

Independent Variables

In the implementation four independent variables during trials were used - cue location, target location, cue validity and CTOA.

Cue Location: Cue location can vary between two pre-determined location marked with placeholders. One location to the right and one equally separated to the left of the center in the users fov. **Cue transparency:** Cue transparency is varied between two levels, high contrast and low contrast. A high contrast cue is fully visible to the user. A low contrast cue has the transparency set to the level earlier calibrated by the user and hence should be very close to the users detection threshold. **CTOA:** CTOA can take one out of four possible values, 50ms, 250ms, 450ms or 650ms. The different values is equally distributed between the combinations of cue location and cue transparency.

Dependent Variables

The dependent variables in this test were all the variables that directly depend on the variables factors mentioned above. This were only the reaction time.

Possible disturbance

The experiment is using AR in combination with an eye-tracker to create a virtual test environment where the experiment is carried out. This set up is sensitive to certain changes in the environment, e.g. changes in lighting and changes in background.

Fixed external variables

It is important that the external variables mentioned above to be kept as constant as possible to minimize any differentiation between test rounds. The test is therefor carried out in the basement of the IKDC building at LTH. The lighting intake in these premises are more easy to control than in other parts of the building and hence changes in lighting is minimal.

Fixed background

The participants are placed in front of a whiteboard at a fixed distance of 3m. The all white background of the whiteboard will make it easier for the participant to find the right threshold to be used for the stimulus. It also makes it more reliable for the execution of the test since small variations in any direction wont affect the background.

Cue & Target

Cue- & Target location

The location of the cue is equally distributed to show up either in the right placeholder or in the left placeholder. In the same way is the location of the Target equally distributed to show up in one of the two placeholders. The order of the sequence is however randomly shuffled while maintaining the correct distribution of valid and invalid cues in a test. Since the distribution between valid and invalid cues is to be kept equally shuffled, the summation of all rounds will add up to the following distribution:

24 rounds - Cue left, Target left (Valid Cue)

24 rounds - Cue left, Target right (Invalid Cue)

24 rounds - Cue right, Target right (Valid Cue)

24 rounds - Cue right, Target left (Invalid Cue)

Cue Validity

There are two type of cues used in this test. The Cue type is determined by its placement in relation to the target marker. This can be seen in the list above. A cue that is presented on the same side as the target is considered a valid cue, while a cue presented on the opposite side as the target is considered to be invalid.

CTOA

A fixed blocked CTOA is used throughout the test. This means that the time between that the stimuli disappears and that the target is presented varies between a preset and fixed number of options. In this thesis the four blocked times 50ms, 250ms, 450ms, 650ms is chosen as CTOA and is randomly varied between trials. Each participant is however presented with the same summed amount of each trial type, the order between trials is randomly selected for each test round.

5.3.4 Dependent Measures

The dependent variables in the experiment are solely the reaction time.

5.4 Procedure

5.4.1 Participant Consent Form

Each participant of the experiment fills out a consent form. This ensures that the participant knows what the experiment is about and how the result of the experiment will be used.

5.4.2 Creation of record and participant

When a participant opens up the test for the first time the main menu is opened up. Here four different options is presented to the user. However the process of setting up the trial

has already started. A connection to the Tobii Pro glasses is already set up and a participant with a, for the participant to remain anonymous, fixed pre-set user name is created in the eye-tracker.

5.4.3 Calibration of the eye-tracker

The user is now asked to calibrate the eye-tracker. A process that is initiated by pressing a menu button "start calibrating". The calibration process normally takes less than a second and is simply carried out by looking at a box in the center of the field of view. The process is intuitive to the user since the menu disappears during the calibrations, so the only thing present is the calibration box.

5.4.4 Start Experiment

When calibration is performed successfully the next step is to enter the test menu. Here the participant is presented with three different buttons. A new calibrations is to be done before the test can be performed that is the calibration of the cue to be used. The process of this calibration is described in 3.1.5.

This calibration finishes the preparations and the actual test can be performed. There are two type of tests where one is using the low cue type and one that is using high cue types. To keep possible unwanted effects of variables to a minimum half of the participants are performing one type of test first while the other half is starting with the other type.

5.4.5 Experiment execution

A trial is built up by five different steps(step 0 - step 4) each further described below. The five steps show the example sequence of how a trial round with a invalid cue could look.

Step 0

When a new trial round is initiated a blank test board is shown. This is shown during a two second period to let the participant prepare for the new round. This is "pre-step" and not related to the actual test in any way more than that the placeholders is showing as a background.

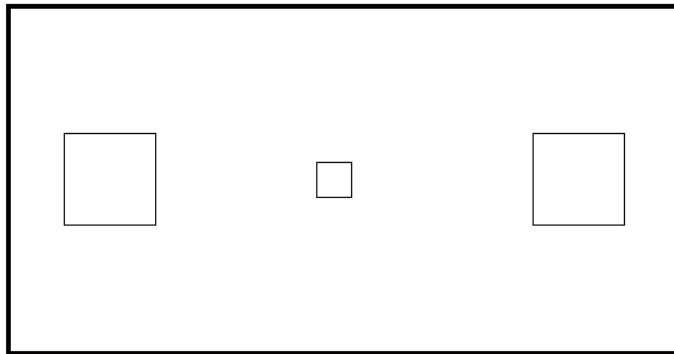


Figure 5.8: step 0, Duration: 2s

Step 1

This step is present in the test to make it clear to the participant that a new trial round has began. In this step a central fixation point is displayed to the participant. This is shown 1s, fixed time, to direct the participants focus to the center of the board. The participant is on beforehand asked to focus on this fixation point when shown which makes sure that the participant is susceptible to stimulus from cues.

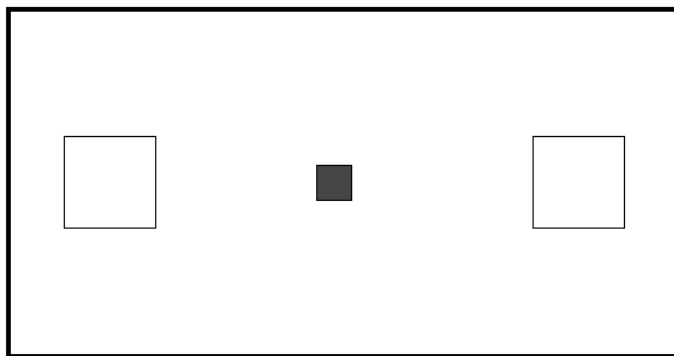


Figure 5.9: Step 1, Duration: 1s

Step 2

In step 2 the cue is presented to the participant for a duration of 100ms. The cues location is varied according to the description in previous section.

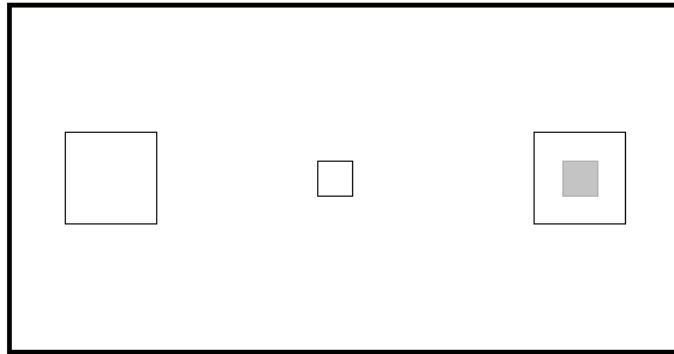


Figure 5.10: Step 2, Duration: 100ms

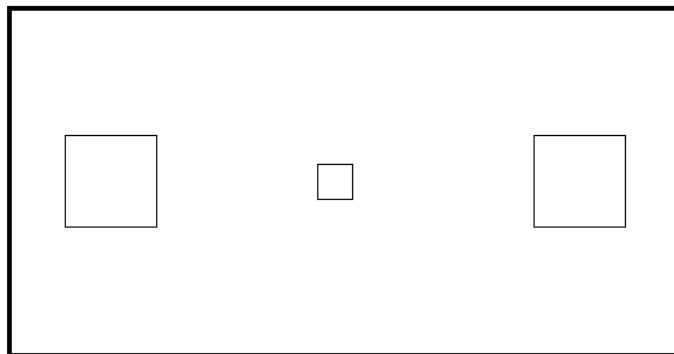


Figure 5.11: step 2.5, Duration: 50-650ms

Step 3

Step 3 is enabled after a certain amount of time has passed since the cue has disappeared. This time is varied between four values(50ms, 250ms, 450ms, 650ms) according to the description in previous section. The target is shown for a maximum of three seconds, or until the participant has performed a saccade that makes the gaze ray activate the target resulting in a valid round.

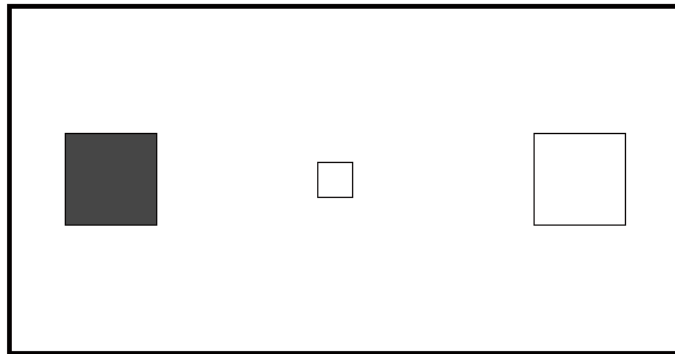


Figure 5.12: Step 3, Duration: <3000ms

Step 4

Step 4 is used as feedback to the user. If the trial round is successful the target color changes to green before restarting at step 0. If the trial isn't successful, e.g. the participant fails to saccade at the box within the 4000ms, the box changes color to red instead.

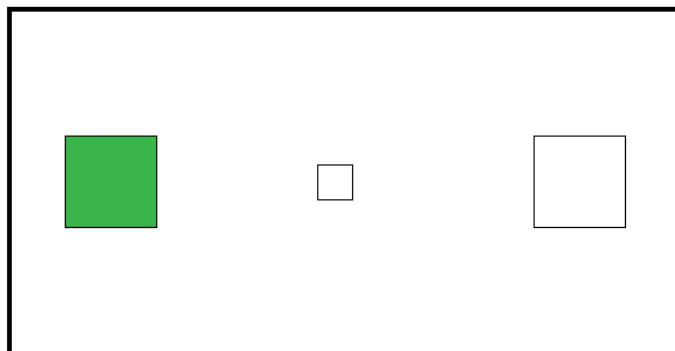


Figure 5.13: Step 4, Feedback

Chapter 6

Results

6.1 Description of the result

The result from the experiment is a number of different RT depending on the dependent variables. It is divided into three different sections. One for the overall results from the posner cueing task that aims to show how the overall RT was affected of the cue. One section that shows the resulting differences between a high contrast cue and a low contrast cue. The last section shows how the RT varied depending on the CTOA time.

6.2 Data Analysis

In total 13 participants conducted 2496 trials. Erroneous trials were removed during the data import. This included trials with a reaction time (RT) over 1500ms and trials with a reaction time less than 100ms.

Total number of erroneous trials: 69, 2,7%

Outliers with a reaction time > 1500ms: 17, 0,68%

Outliers with a reaction time < 100ms: 52, 2,08%

This gave 2427 valid trials divided between high visible cues and low visible cues.

High visible cues: 1204, 49,61%

Valid cues: 587, 24,18%

Invalid cues: 617, 25,42%

Low visible cues: 1228, 50,39%

Valid cues: 608, 25,05%

Invalid cues: 618, 55,46%

Analysis of the gathered data is conducted using the Scipy package for python. Plotting graphs is conducted using the included packages Seaborn, Pandas and matplotlib.pyplot. To determine that the test result is statistically significant an Three-way ANOVA(Analysis of Variance) were conducted. The F-ratio obtained from the calculations were then used to determine if the null hypothesis holds true. A value close to or under 1.0 would imply that its true. A large F value implies that the null hypothesis is not true and hence the measured difference is larger than the difference you would expect to obtain by chance. The p-value(probability value) is used to prove that the data is reliable enough. a calculated p where $P < 0.05$ is considered acceptable here.

6.3 Experimental results

The results from the experiment is described below in tables and with box plots. The tables contain average reaction times both per participant and summarized.

Source	SS	DF	MS	F	p-unc	np2
typeOfCue	10206.005	1.0	10206.005	0.414	0.521	0.002
validCue	165488.889	1.0	165488.889	6.709	0.010	0.034
fromFixationToTarget	82579.899	3.0	27526.633	1.116	0.344	0.017
typeOfCue * validCue	98.312	1.0	98.312	0.004	0.950	0.000
typeOfCue * fromFixationToTarget	2689.553	3.0	896.518	0.036	0.991	0.001
validCue * fromFixationToTarget	125325.822	3.0	41775.274	1.694	0.170	0.026
typeOfCue * validCue * fromFixationToTarget	31582.476	3.0	10527.492	0.427	0.734	0.007
Residual	4735708.462	192.0	24665.148	NaN	NaN	NaN

As shown in the ANOVA above there is only one significant effect, that is the validCue. As interpreted from the table above ValidCue had the following values

$$F(1, 192) = 6.71, p = 0.01, \eta_p^2 = 0.034$$

None of the other interactions showed a significant impact on the test. This implies that a subtle cue can yield the same effect on the participants attention that a high contrast cue.

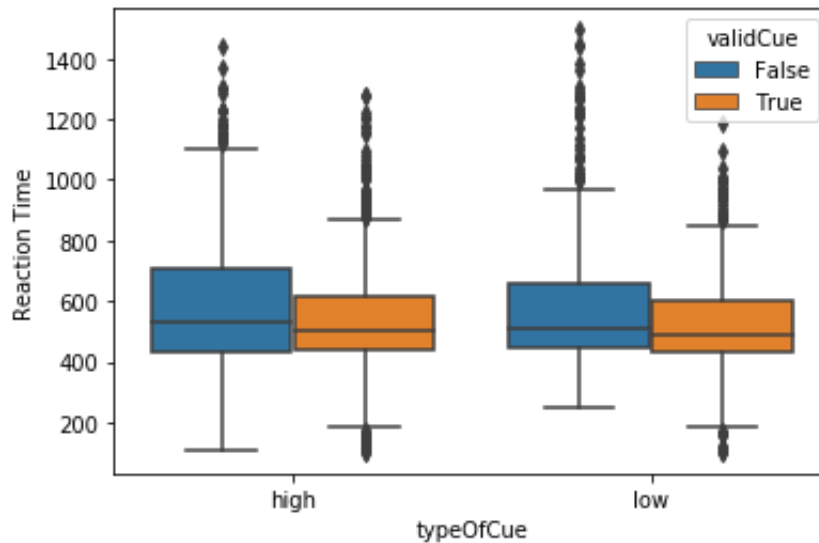
6.3.1 Valid & Invalid cued trials

Here the resulting RT is shown for the two type of cues. A valid cue is a trial where the cue is shown in the same box as the target, an invalid trial is when the cue is showed in the opposite box as the target.

As seen in table 6.1 the difference between a valid and an invalid cue where in average 54ms. The ANOVA table below shows that the F value is 43.2 strengthening the conclusion that the cue has a significant impact on RTs.

Table 6.1: Average RT, cue dependent

Participant	Valid Cue	Invalid Cue	Difference
412	446	446	0
1147	389	473	83
3578	467	453	-14
4466	817	807	-9
4967	606	615	8
5823	481	585	104
6071	491	498	6
6640	458	452	-6
8363	743	805	61
8450	491	489	-2
8841	555	623	67
9541	480	450	-29
9613	524	956	431
Summary	536	588	52

**Figure 6.1:** Average RT

6.3.2 High contrast & Low Contrast Cues

Here the resulting RT is shown for the two type of cues divided further into the two different contrasts of a cue. A Low cue is a cue that has a contrast set by the participant and hence is barely visible, A high cue is one that is fully visible. The table shows that the average RT of the participants were 52ms longer when an invalid cue was presented compared to a valid cue.

Table 6.2: RT depending on cue contrast and cue validity

Participant	High Valid Cue	Low Valid Cue	Difference	High Invalid Cue	Low Invalid Cue	Difference
412	437	455	-18	442	451	-9
1147	391	388	3	426	521	-95
3578	452	482	-30	422	483	-61
4466	880	754	126	850	765	85
4967	593	619	-26	614	615	-1
5823	464	498	-34	612	559	53
6071	554	428	126	534	462	72
6640	472	445	27	444	460	-16
8363	797	689	108	865	745	120
8450	513	469	44	508	469	39
8841	560	550	10	620	626	-6
9541	466	494	-28	436	465	-29
9613	501	547	-46	971	941	30
Summary	545	524	20	596	582	14

6.3.3 RT depending on CTOA

Here the resulting RT is shown individually for the four different combinations of CTOA and Cue Type. That gives four tables as follows:

- High Contrast Valid Cue
- High Contrast Invalid Cue
- Low Contrast Valid Cue
- Low Contrast Invalid Cue

Two Boxplots is provided that compares the transparency affect depending on the cue type. The tables show to important patterns. When presenting valid cues the RT is increasing as the CTOA is increasing. The opposite is the case when the cue is invalid. A longer CTOA is instead decreasing the RT.

Table 6.3: High contrast valid Cue per CTOA

Participant	High Valid 50ms	High Valid 250ms	High Valid 450ms	High Valid 650ms
412	419	460	459	416
1147	290	442	465	405
3578	452	456	453	443
4466	886	906	869	871
4967	604	595	597	576
5823	394	372	516	615
6071	615	534	574	508
6640	468	478	481	468
8363	764	775	630	932
8450	480	490	537	544
8841	558	541	551	613
9541	448	485	471	460
9613	590	292	447	648
Summary	536	525	542	577

Table 6.4: Low contrast valid Cue per CTOA

Participant	Low Valid 50ms	Low Valid 250ms	Low Valid 450ms	Low Valid 650ms
412	480	449	454	438
1147	430	365	387	369
3578	511	482	486	459
4466	773	762	725	747
4967	635	611	587	644
5823	439	458	543	562
6071	430	456	397	429
6640	456	435	452	440
8363	633	685	722	725
8450	502	471	452	451
8841	558	519	558	561
9541	559	508	472	458
9613	636	445	502	604
Summary	542	511	518	530

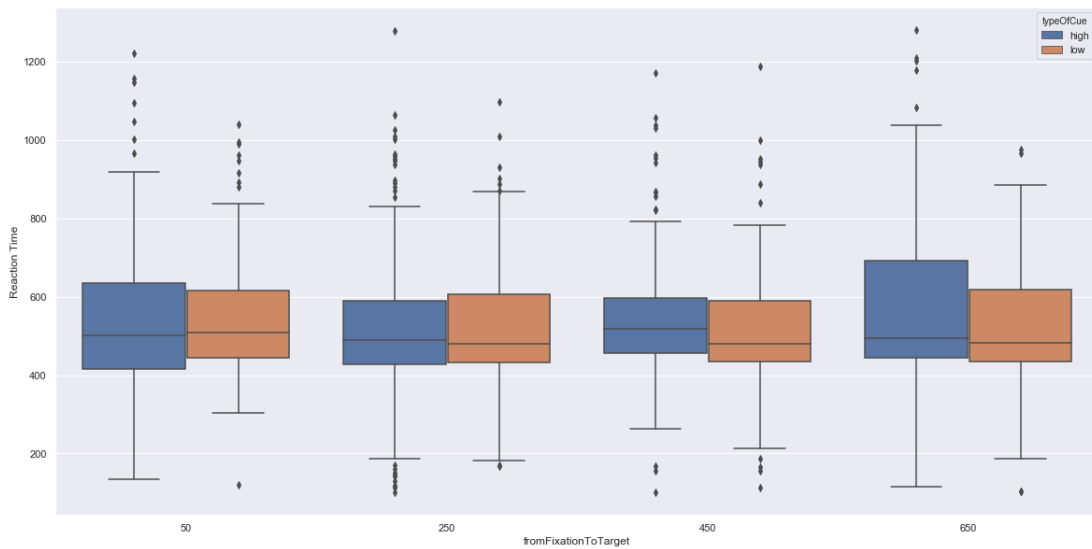
**Figure 6.2:** Average RT for Valid cues depending on CTOA**Low contrast cues**

Table 6.5: High contrast invalid Cue per CTOA

Participant	High Invalid 50ms	High Invalid 250ms	High Invalid 450ms	High Invalid 650ms
412	515	418	416	404
1147	539	443	382	352
3578	443	419	401	419
4466	985	842	812	768
4967	634	587	637	604
5823	702	678	561	532
6071	600	538	475	486
6640	479	433	455	395
8363	1010	859	881	658
8450	588	543	485	397
8841	703	713	620	516
9541	464	427	441	411
9613	1151	1027	947	749
Summary	678	610	578	515

Table 6.6: Low contrast invalid Cue per CTOA

Participant	Low Invalid 50ms	Low Invalid 250ms	Low Invalid 450ms	Low Invalid 650ms
412	442	452	460	450
1147	635	507	516	425
3578	514	511	453	458
4466	799	717	769	767
4967	673	676	554	584
5823	587	646	547	449
6071	482	477	434	455
6640	473	456	448	462
8363	794	781	738	685
8450	447	452	524	464
8841	651	641	668	541
9541	532	438	441	433
9613	953	1022	1008	794
Summary	614	598	582	536

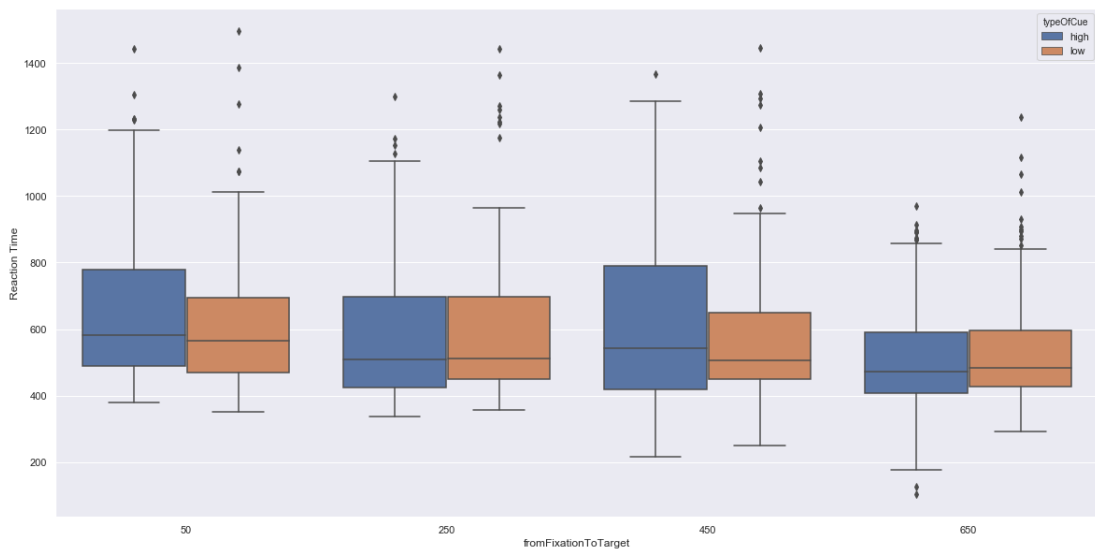


Figure 6.3: Average RT for Invalid cues depending on CTOA

Chapter 7

Discussion & Analysis

7.1 Discussion & Analysis of the result

The experiment is conducted using only the HoloLens as an input. This gives a longer measured reaction time compared to if the eye tracker would have worked during the experiment. This also makes it hard to validate the system by comparing it to similar works since they often use an eye tracker or other type of devices for input. [20] reported an average reaction time for uncued targets of 267ms, while [30] reported 250ms and [31] reported 310ms. The expected difference between the difference cue types and contrasts was approximately the same as the results in the experiments previously mentioned. It is not feasible to compare the RT differences in relation to the overall RT since the overall RT is expected to be longer since a head movement is more time consuming than a saccade with the eyes.

7.1.1 Posner paradigm

The experimental result where in many ways as what could be expected for a Posner cueing task. The result, seen in Figure 6.1 as a box plot, shows that a subtle cue can give a similar affect on the visual attention as a fully visible cue. This is shown when comparing the difference in reaction time between a valid and invalid cued target for the two cue types (high contrast and low contrast). The average difference between the RT for a high visual cue and the RT for a cue with the transparency is very low. For valid cues the difference is 20ms while it was only 14ms for invalid cues. This proves, as expected, that even when the cue is low contrasted it get perceived by the participant [31] and triggers a similar reaction as the high contrast cue.

CTOA difference

In experiments using an eye tracker the shortest CTOA for valid cues yield the lowest RT.

The affect then lowered as the CTOA increased and for the longest CTOA instead returned an negative result. The result from the trials showed that a 50ms valid cue actually had a higher RT in average compared to a CTOA of 250ms. The average RT of both high and low cues for CTOA of 50ms was 539 while it was 518 for 250ms. The expected pattern showed for CTOA of 450ms and 650ms where it successively increased.

As seen in Figure 6.2 and in Figure 6.3 that shows the difference in reaction time for cues divided by CTOA the affect on the reaction time is heavily dependent on this. As predicted, the reaction time for a valid cued target was overall lower than that of a invalid cued target. The result is varying with the CTOA which is as expected.

Invalid cues

For Invalid cues the same pattern occurs clearly for both the high and the low cues. An Invalid cue, both high contrast and low contrast, with a 50ms CTOA has in in average a reaction time of 646ms. For the same type of cue with a CTOA of 650ms the average RT is 525ms. The affect of the CTOA on the average RT for invalid cues is decreasing as the CTOA is prolonged, which is a similar result as shown by [21]. For CTOA of 250ms and 450ms approximately equivalent result is expected to show for each cue type. The test result confirms this thesis and shows that for a valid cue a CTOA of 250ms had an average RT of 518 while a CTOA of 450ms had an average RT of 530. The box plot confirms this result and shows similar values for both the high contrast and the low contrast cues. Invalid Low cues in general showed a greater amount of upper outliers compared to the high contrast cues. This pattern could however not be seen when comparing valid low- and valid high cues.

Valid cues

For valid cues the shortest RT showed for CTOA of 250ms. For a low contrast cue the longest RT showed for the shortest 50ms CTOA which was a bit surprising. It was expected that this would be noticed for a CTOA of 650ms. However, for CTOA of 250ms-650ms the pattern followed the expected curve and the RT became successively longer as the CTOA became longer. [31] The effect varied a lot more between high contrast cues and low contrast cues than for invalid cues. For CTOA of 650ms the difference between a high and a low contrast cue were as big as 47ms. The difference however decreased as the CTOA became shorter.

7.1.2 Threat to validity

Inconsistent transparency in the whole FOV

After the implementation of the test a limitation of the hardware showed to pose a possible threat to the validity of the test. Transparency in Hololens glasses doesn't seem to be consistent in the whole fov. Meaning that a properly calibrated cue could be interpreted differently when placed in one of the target boxes.

When a participant calibrated the cue he were instructed to look directly at the cue and then decide if it were to be more or less transparent. This was done iterative because the cue appeared more transparent when looking at the buttons than directly at the cue. This was considered a valid way to overcome the problem with the unwanted difference in transparency depending on where in the FOV it was placed. However the problem arose again when the cue where showed in one of the boxes. The cues showed less of a

transparency than when calibrated since it was place near the edges of the FOV in the hololens glasses.

A proper way to overcome this issue would be to show the cues in its boxes were it later would be showed instead of in the center.

No working eye tracker

The test had to be performed without a working eye tracker. This forced the participant to turn the head so that the pointer in the center of the view hit the target, This caused some problems for the test and could potentially affected the result of the test. The biggest affect had the fact that when the participant turned his head a bit "to far" and past the one of the targets, the other target fell out of the FOV and hence couldn't be noticed by the participant. This could affect a participant reacting on a cue not seeing that the actual target showed in the opposite box. This potential problems would prolong the RT for that trial and give a false result.

Chapter 8

Conclusions and Future Work

8.1 Conclusion

The goal of this thesis was to implement a low latency communication between Microsoft HoloLens and Tobii eye tracker. The implemented system should then be used to perform a Posner cueing task in an AR environment. The project followed an iterative process where each part of the system were built in different sprints. The first sprints were mostly aimed at developing the communication between the hardware and get that working seamless. When this crucial part of the system were done the implementation of a Posner cueing task took form. The central piece of this process was to determine the data model for a trial. What would need to be stored to be able to analyse and make proper conclusions and how could it be stored and transferred efficient.

With the data model in place the posner cueing task was implemented so that a valid and reliable test could be conducted. The eye-tracker broke just as the test were to be carried out and hence the test had to be done without this crucial part of the system. The conclusion were still that the test results were reliable enough. The analysis of the experiment data confirmed this with only minor differences from what was expected to be seen from the test.

The test showed that valid subtle cues is resulting in a shorter RT while an invalid subtle cue prolongs the participants RT.

8.2 Contribution

As not much work has been done earlier for Augmented Reality and subtle gaze guidance the thesis brings some insights in how this can be implemented. The subtle cueing definitely seems like a promising technique to be used for as a technique for gaze guidance in a Augmented reality environment.

8.3 Future Work

With the findings of possible threat to validity the natural next step would be to adapt the experiment to avoid these problems. The transparency of the cues should probably be calibrated on its proper positions to avoid problems with the varying transparency levels throughout the FOV.

Many participants are new to the hololens environment and the validity of the test could possibly be increased with a longer test experiment. Also a clicker could be used between the trials to ensure that the participant is in position and ready for a new target to appear.

The test environment could be placed more suitably in the space if the participants always could be placed at the same location. This was not possible since the only students were allowed to enter the VR-lab and hence the test had to be carried out on different locations. This could gain the test to lock the boxes "on" a whiteboard or similar background a few meters away so that the boxes never left the FOV even if the participant turned the head at one of the boxes. This should however not be a big problem when using a eye-tracker.

It should be taken into consideration that hololens gen.2 is to be released to the public in a nearby time. This version has a lot of improvements as well as an integrated eye tracker. So it is possible that not all the encountered problems would be the same if the test were run on that instead. Since the field is growing rapidly it is important that the amount of research carried out is following the same path. When it comes for gaze guidance in AR/VR environments the importance will grow at least as fast as other parts of the field. Especially when the technology gets used by more non-technical people without proper experience of either computers or AR/VR.

References

- [1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. Macintyre, “Recent advances in augmented reality,” *Computer Graphics and Applications, IEEE*, vol. 21, pp. 34 – 47, 12 2001.
- [2] Okreylos, “On the road for VR: Microsoft HoloLens at Build 2015, San Francisco,” <http://doc-ok.org/?p=1223>, added: 2015-05-01, accessed: 2019-10-11.
- [3] Microsoft, “Microsoft Mixed Reality Toolkit documentation,” <https://docs.microsoft.com/en-us/windows/mixed-reality/>, accessed: 2019-10-31 as pdf download.
- [4] D. Bonino, E. Castellina, F. Corno, and L. De Russis, “Dogeye: Controlling your home with eye interaction,” *Interacting with Computers*, vol. 23, no. 5, pp. 484–498, 2011.
- [5] C. Ware and H. H. Mikaelian, “An evaluation of an eye tracker as a device for computer input2,” *Acm sigchi bulletin*, vol. 18, no. 4, pp. 183–188, 1987.
- [6] Tobii AB, “How do Tobii Eye Trackers work?” <https://www.tobii.com/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/>, accessed: 2019-10-11.
- [7] —, “Dark and bright pupil tracking,” <https://www.tobii.com/learn-and-support/learn/eye-tracking-essentials/what-is-dark-and-bright-pupil-tracking/>, accessed: 2019-10-17.
- [8] J. M. Carroll and J. Kjeldskov, “The encyclopedia of human-computer interaction,” 2013.
- [9] M. Weiser, “The computer for the 21st century,” *IEEE pervasive computing*, vol. 1, no. 1, pp. 19–25, 2002.
- [10] S. Bakker, “Design for peripheral interaction,” Ph.D. dissertation, Department of Industrial Design, 2013.

- [11] W. Lu, D. Feng, S. Feiner, Q. Zhao, and H. B.-L. Duh, "Subtle cueing for visual search in head-tracked head worn displays," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2013, pp. 271–272.
- [12] S. L. M. . D. H. H. Susana Martinez-Conde, "The role of fixational eye movements in visual perception," *Nature Reviews Neuroscience volume*, vol. 5, pp. 229—240, 2004.
- [13] M. I. Posner, "Orienting of attention," *Quarterly Journal of Experimental Psychology*, vol. 32, no. 1, pp. 3–25, 1980.
- [14] C. K. A. Miguel P. Eckstein, Steven S. Shimozaki, "The footprints of visual attention in the posner cueing paradigm revealed by classification images," *Journal of Vision*, vol. 2, no. 1, 2002.
- [15] C. D. C. Neel A. Macmillan, *Detection Theory : A User's Guide (2nd edition)*. Lawrence Erlbaum Associates, Inc., 2005.
- [16] D. A. B. G. R. M. E. W. Katherine A. MacLean, Stephen R. Aichele and C. D. Saron3, "Interactions between endogenous and exogenous attention during vigilance," *Atten Percept Psychophys*, vol. 71, no. 5, pp. 1042—1058, 2009.
- [17] J. Theeuwes, "Exogenous and endogenous control of attention: The effect of visual onsets and offsets," *Perception & Psychophysics*, vol. 49, pp. 83–90, 1991.
- [18] C. E. Connor, H. E. Egeth, and S. Yantis, "Visual attention: bottom-up versus top-down," *Current biology*, vol. 14, no. 19, pp. R850–R852, 2004.
- [19] R. M. Klein, "Inhibition of return," *Trends in Cognitive Sciences*, vol. 4, no. 4, pp. 138—147, 2000.
- [20] V. Kullberg and E. Lindqvist, "Gaze guidance through head-mounted augmented reality display," Master's thesis, Malmö Universitet, 2019.
- [21] W. M. . R. Bhatnagar, "No supplementary evidence of attention to a spatial cue when saccadic facilitation is absent," *Scientific Reports volume*, vol. 8, 2018.
- [22] F. Shi, A. Gale, and K. Purdy, "Helping people with ict device control by eye gaze," in *International Conference on Computers for Handicapped Persons*. Springer, 2006, pp. 480–487.
- [23] F. Corno, E. Castellina, R. Bates, P. Majaranta, H. Istance, and M. Donegan, "D2. 5 draft standards for gaze based environmental control," *Communication by gaze interaction (COGAIN)*, 2007.
- [24] B. Marques, P. Dias, J. Alves, and B. S. Santos, "Adaptive augmented reality user interfaces using face recognition for smart home control," in *International Conference on Human Systems Engineering and Design: Future Trends and Applications*. Springer, 2019, pp. 15–19.
- [25] D. Verweij, A. Esteves, V.-J. Khan, and S. Bakker, "Smart home control using motion matching and smart watches," in *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. ACM, 2017, pp. 466–468.

- [26] R. Aburukba, A. Al-Ali, N. Kandil, and D. AbuDamis, “Configurable zigbee-based control system for people with multiple disabilities in smart homes,” in *2016 International Conference on Industrial Informatics and Computer Systems (CIICS)*. IEEE, 2016, pp. 1–5.
- [27] P. Majaranta, J. Laitinen, J. Kangas, and P. Isokoski, “Inducing gaze gestures by static illustrations,” in *11th ACM Symposium on Eye Tracking Research and Applications, ETRA 2019*. ACM, 2019.
- [28] A. Köpsel, P. Majaranta, P. Isokoski, and A. Huckauf, “Effects of auditory, haptic and visual feedback on performing gestures by gaze or by hand,” *Behaviour & Information Technology*, vol. 35, no. 12, pp. 1044–1062, 2016.
- [29] D. I. K. S. Forrest Shull, Janice Singer, “Selecting empirical methods for software engineering research.” in *Guide to Advanced Empirical Software Engineering*. ACM, 2008, pp. 285—311.
- [30] S. S. et al., “Voluntary spatial attention has different effects on voluntary and reflexive saccades,” *The Scientific World Journal*, vol. 3, pp. 881—902, 2003.
- [31] B. P. et al., “Influence of subliminal cueing on visual search tasks,” *Extended Abstracts on Human Factors in Computing Systems*, pp. 1269—1274, 2013.

Appendices

Appendix A

Experiment Results

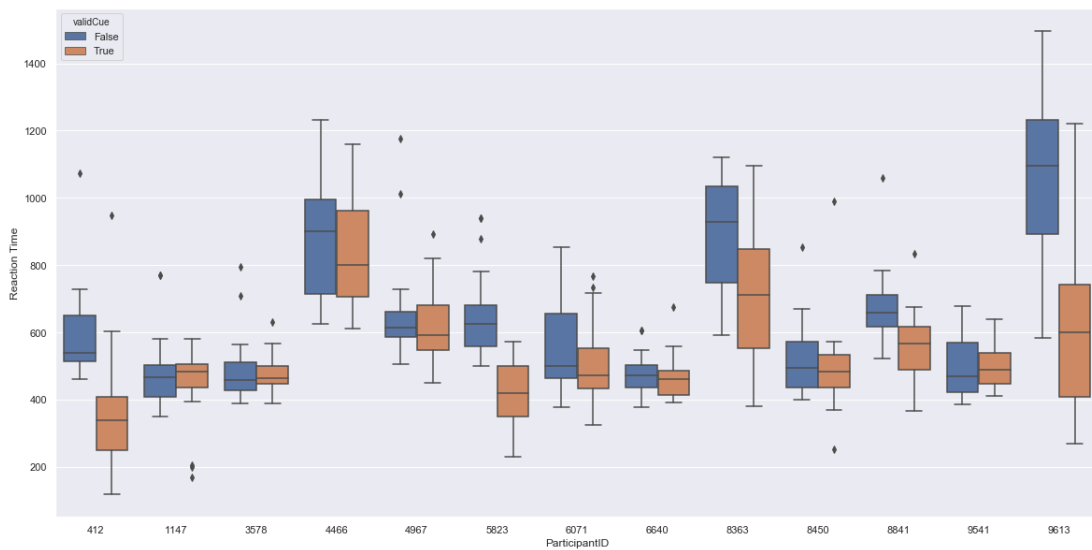


Figure A.1: Average RT depending on cue type

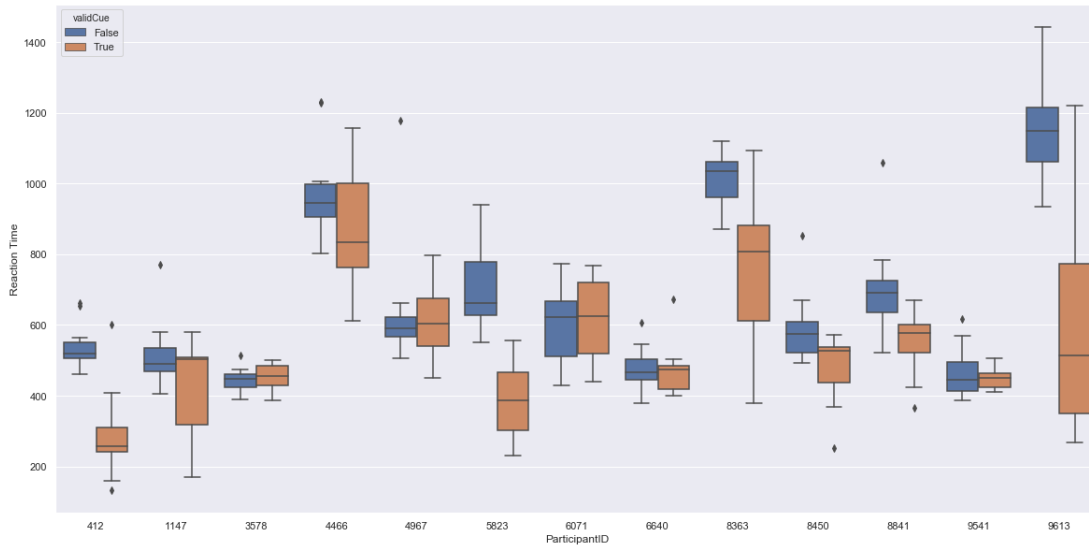


Figure A.2: Average RT per participant,high contrast cues

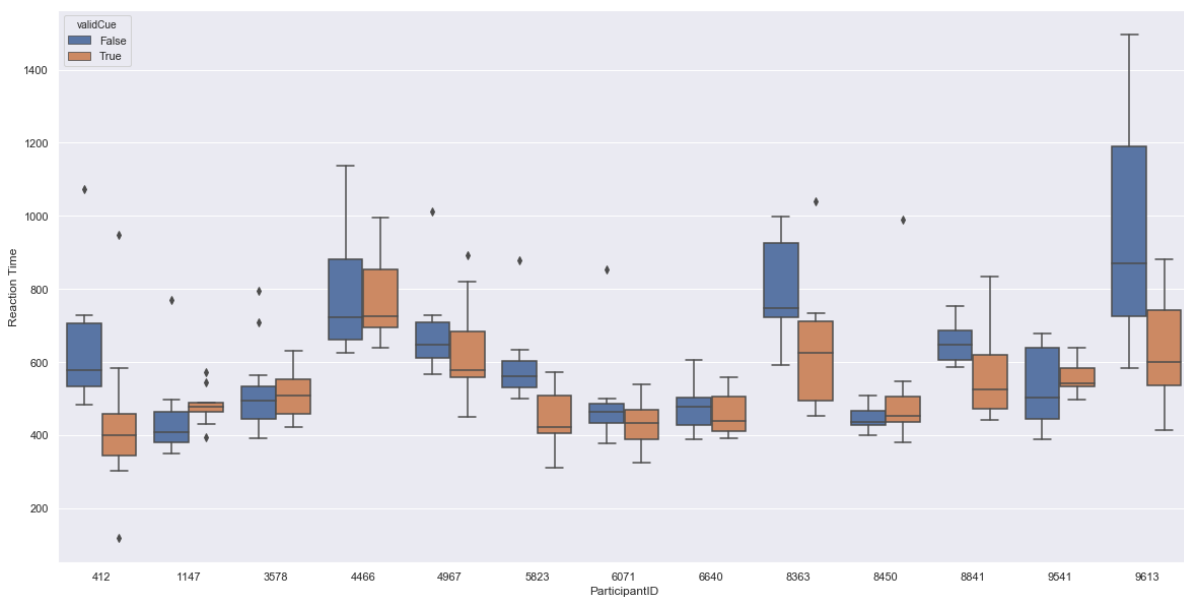


Figure A.3: Average RT per participant,low contrast cues

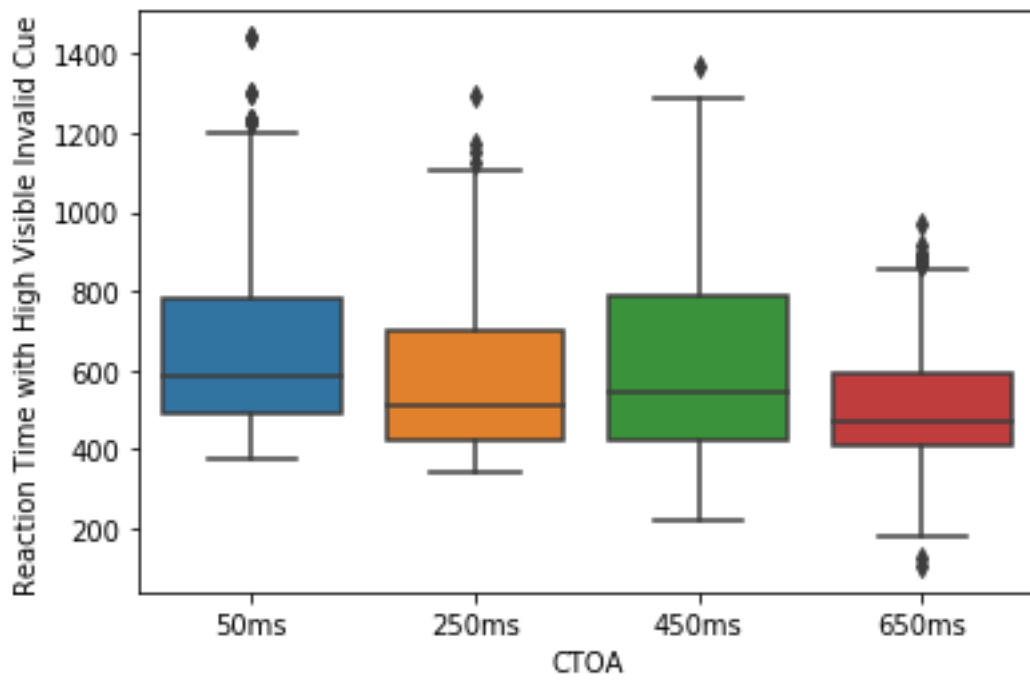


Figure A.4: Average RT, divided per CTOA

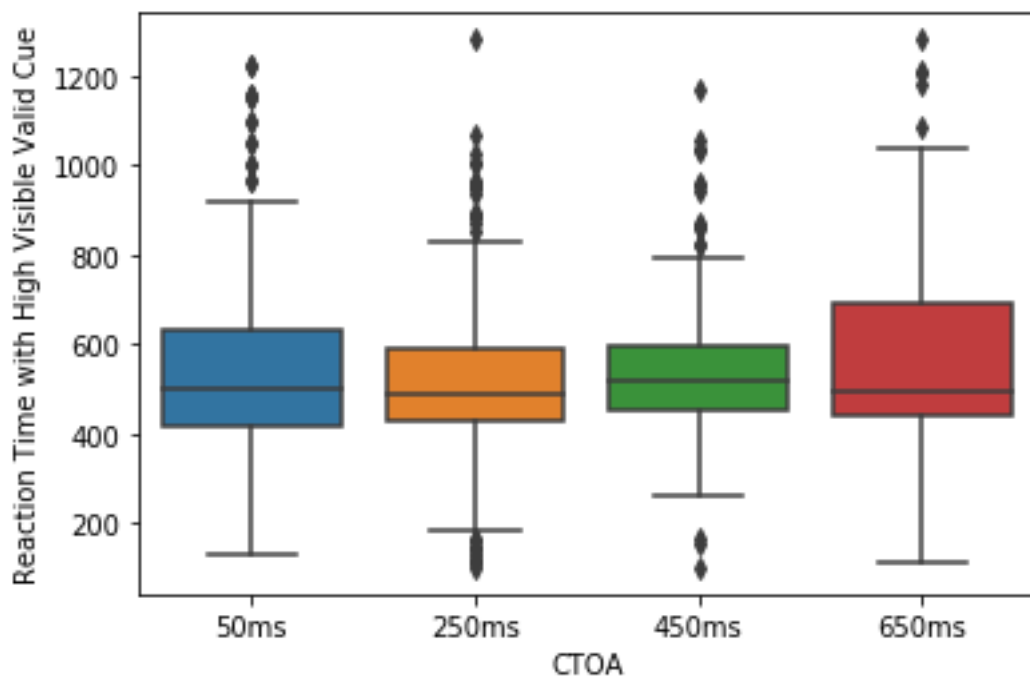


Figure A.5: Average RT, divided per CTOA

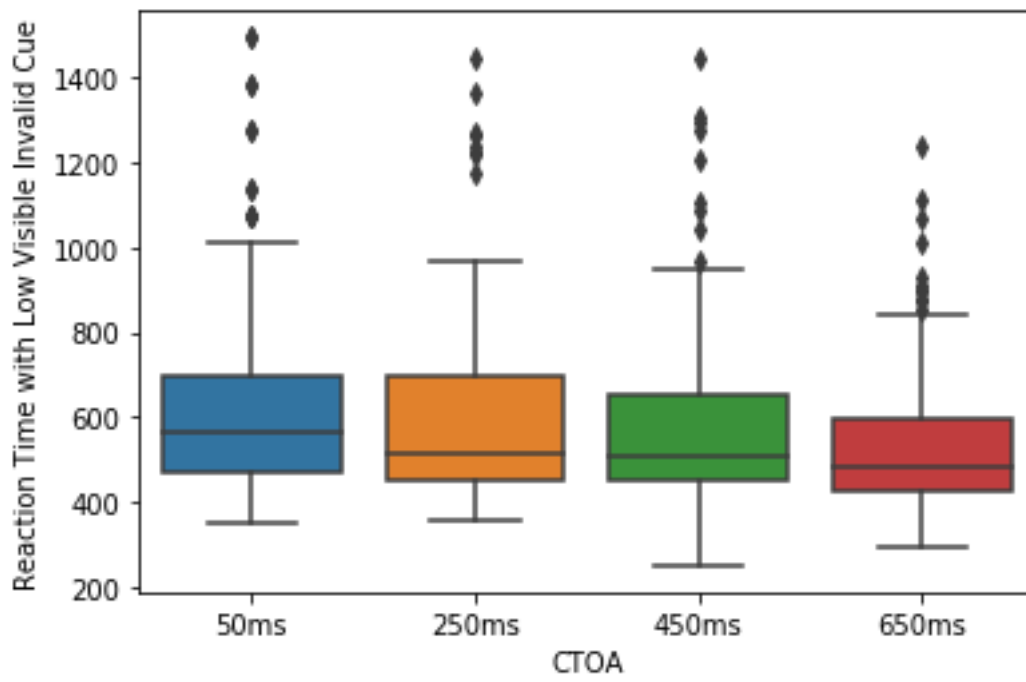


Figure A.6: Average RT, divided per CTOA

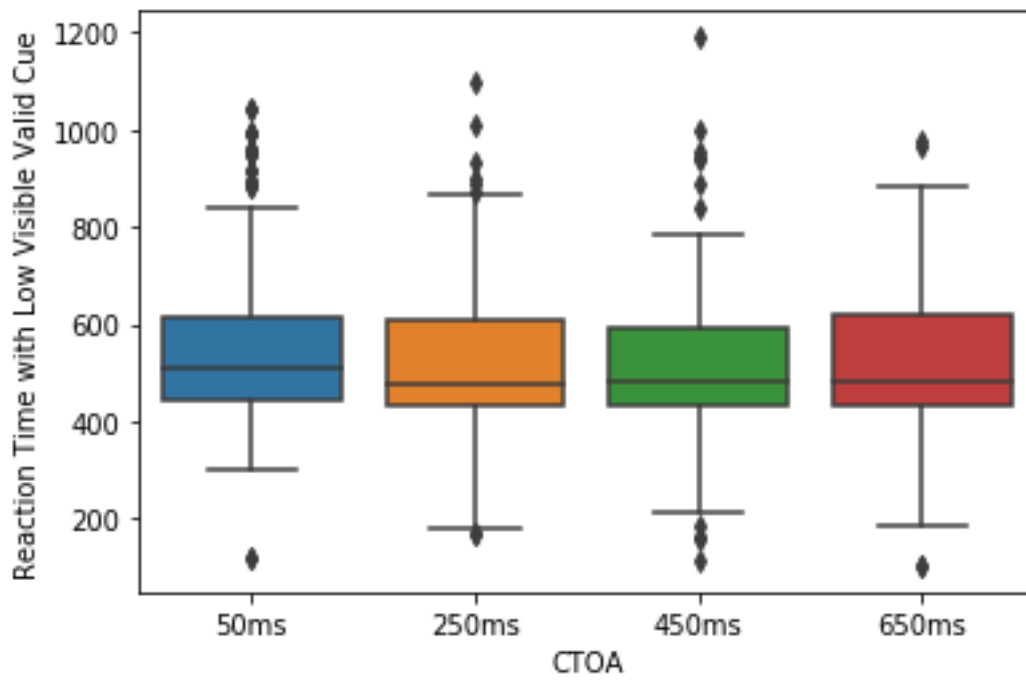


Figure A.7: Average RT, divided per CTOA

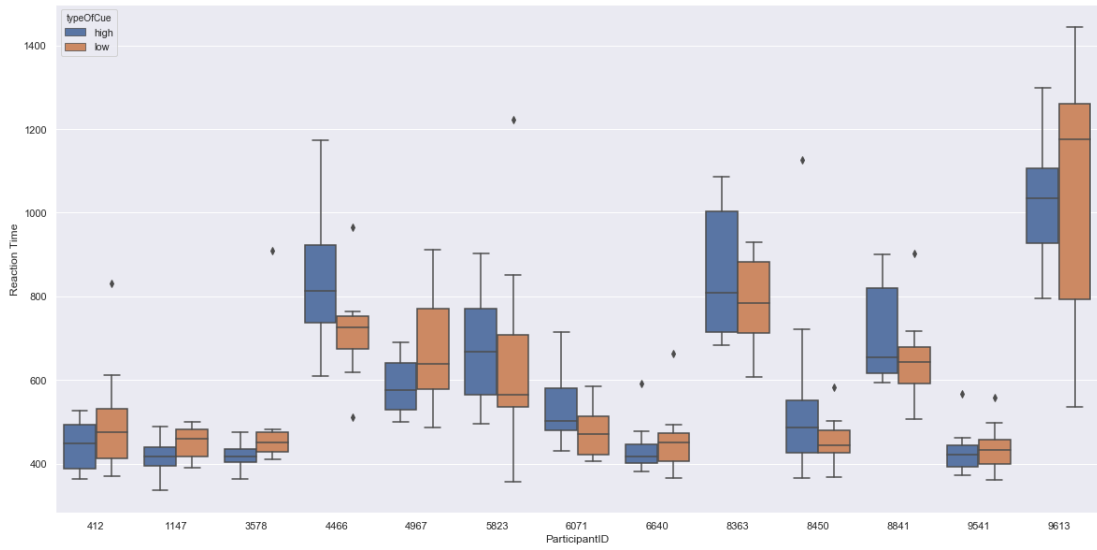


Figure A.8: Average RT, divided per CTOA

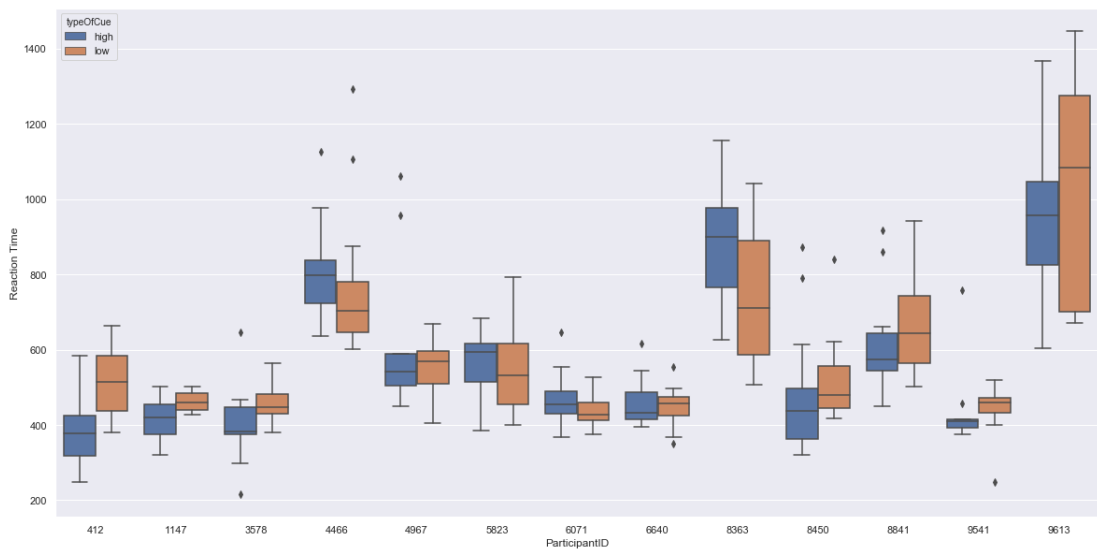


Figure A.9: Average RT, divided per CTOA

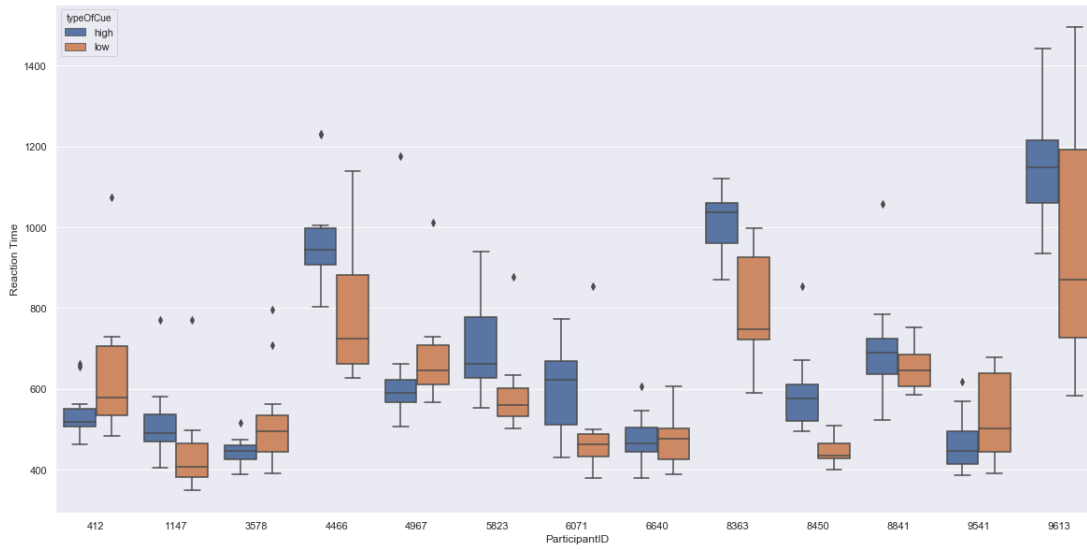


Figure A.10: Average RT, divided per CTOA

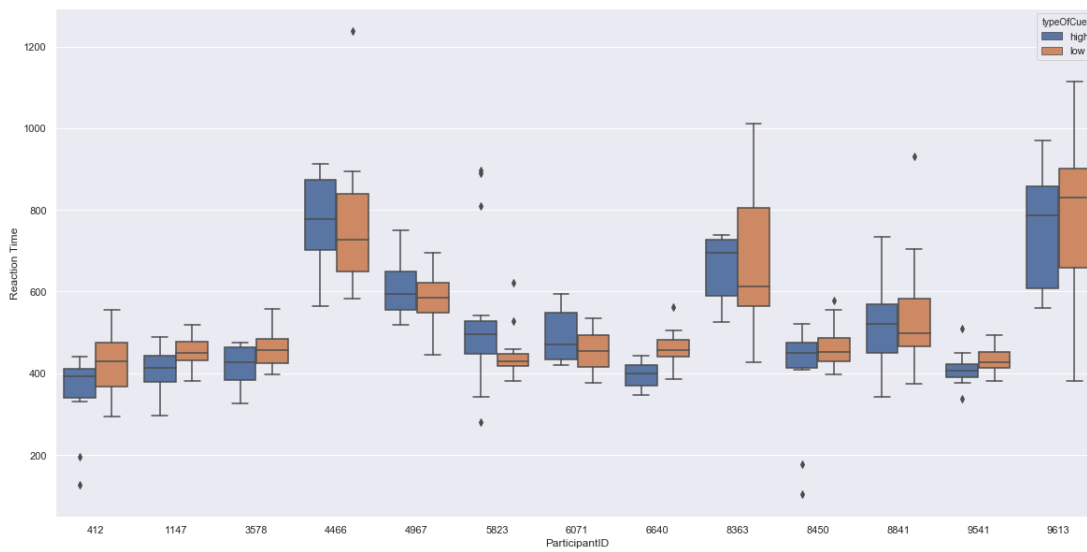


Figure A.11: Average RT, divided per CTOA

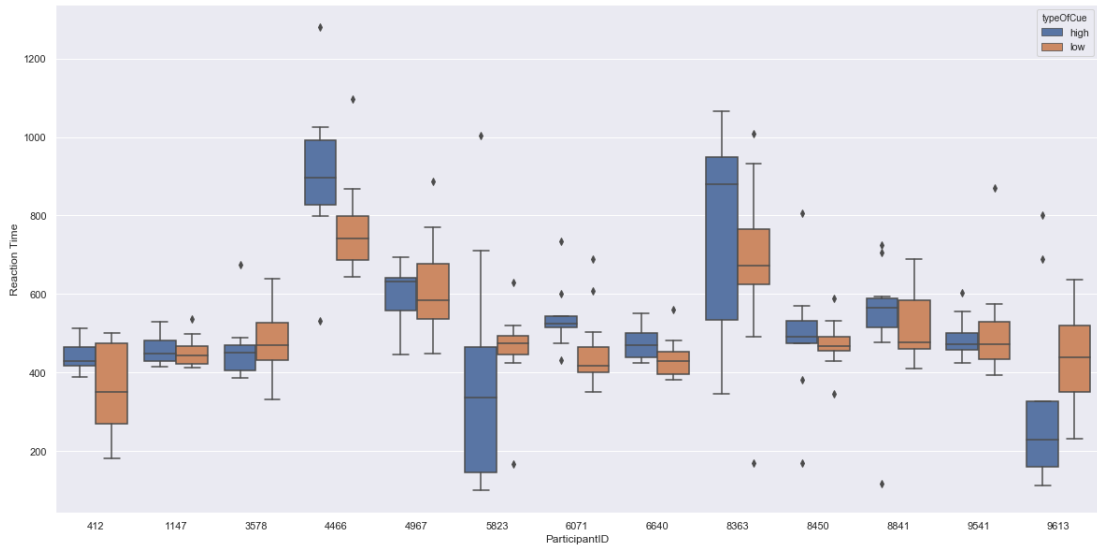


Figure A.12: Average RT, divided per CTOA

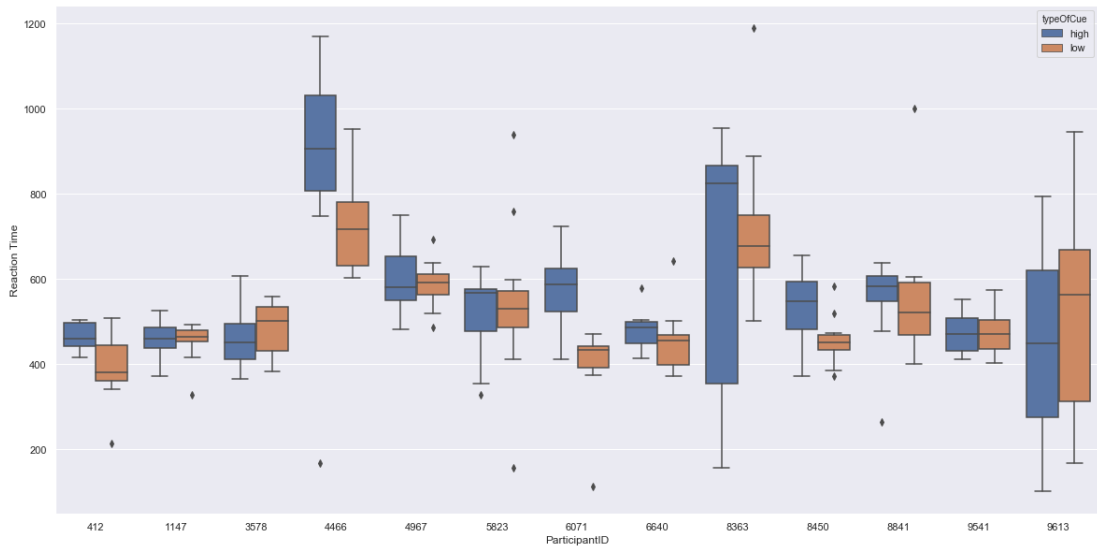


Figure A.13: Average RT, divided per CTOA

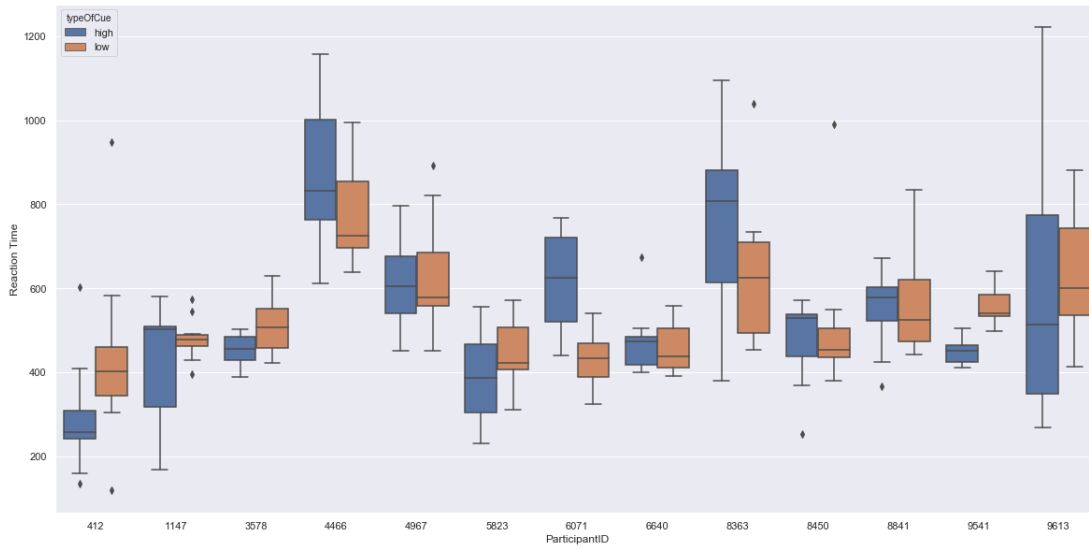


Figure A.14: Average RT, divided per CTOA

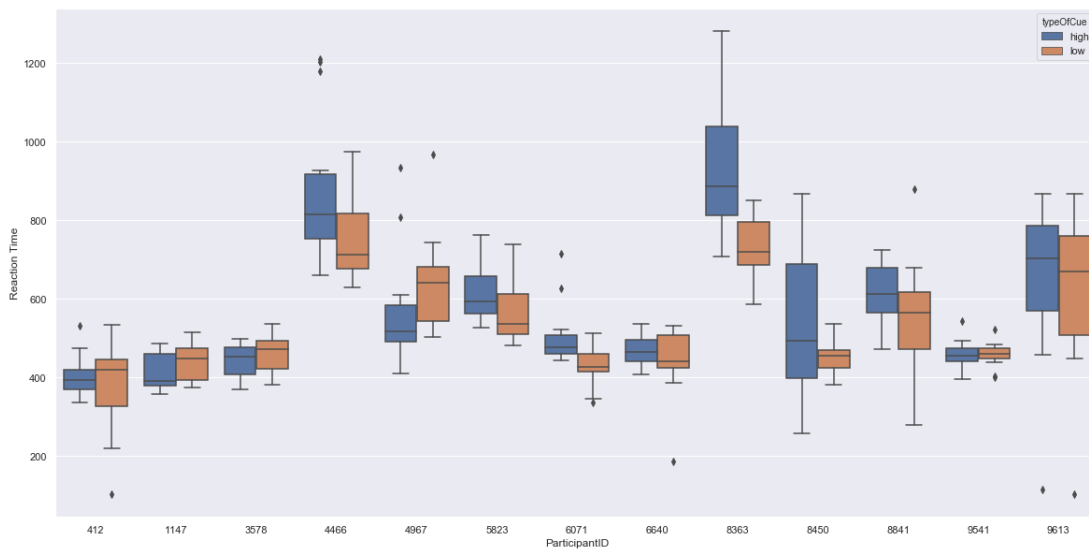


Figure A.15: Average RT, divided per CTOA

Appendix B

Form of concent

Informed consent

I hereby affirm that I have been given the following information:

- I understand that all participation is voluntary and that I have the right to withdraw from the experiment at any time.
- I know that I can regret my participation and get all data destroyed as long as it has not yet been used in a presentation and or publication.
- I approve that the team can use data recorded, in presentation, publications and in other academic contexts.
- I have taken part in information about the study and have gotten satisfying answers to my questions.
- I know that all data will be anonymized and will be treated confidentially.

Name: _____

Signature: _____

Date & place: _____

If you regret your consent or have any further questions,
send an email to Andreas Englesson via dat14aen@student.lu.se