Gaze and Peripheral Vision Analysis for Human-Environment Interaction: Applications in Automotive and Mixed-Reality Scenarios

Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

vorgelegt von
Mohammad Mehdi Moniri

Saarbrücken
2. Januar 2018

Dekan:
Prof. Dr. rer. nat. Frank-Olaf Schreyer
Vorsitzender des Prüfungsausschusses:
Prof. Dr. Jörg Hoffmann
Berichterstatter:
Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster
Prof. Dr. Antonio Krüger
Promovierter akademischer Mitarbeiter der Fakultät:
Dr. Tim Schwartz
Tag des Kolloquiums:
14. Februar 2018

In memory of my father, Yadollah Moniri.

Short Abstract

This thesis studies eye-based user interfaces which integrate information about the user's perceptual focus-of-attention into multimodal systems to enrich the interaction with the surrounding environment. We examine two new modalities: gaze input and output in the peripheral field of view. All modalities are considered in the whole spectrum of the mixed-reality continuum. We show the added value of these new forms of multimodal interaction in two important application domains: Automotive User Interfaces and Human-Robot Collaboration. We present experiments that analyze gaze under various conditions and help to design a 3D model for peripheral vision. Furthermore, this work presents several new algorithms for eye-based interaction, like deictic reference in mobile scenarios, for non-intrusive user identification, or exploiting the peripheral field view for advanced multimodal presentations. These algorithms have been integrated into a number of software tools for eye-based interaction, which are used to implement 15 use cases for intelligent environment applications. These use cases cover a wide spectrum of applications, from spatial interactions with a rapidly changing environment from within a moving vehicle, to mixed-reality interaction between teams of human and robots.

Kurzzusammenfassung

In dieser Arbeit werden blickbasierte Benutzerschnittstellen untersucht, die Informationen über das Blickfeld des Benutzers in multimodale Systeme integrieren, um neuartige Interaktionen mit der Umgebung zu ermöglichen. Wir untersuchen zwei neue Modalitäten: Blickeingabe und Ausgaben im peripheren Sichtfeld. Alle Modalitäten werden im gesamten Spektrum des Mixed-Reality-Kontinuums betrachtet. Wir zeigen die Anwendung dieser neuen Formen der multimodalen Interaktion in zwei wichtigen Domänen auf: Fahrerassistenzsysteme und Werkerassistenz bei Mensch-Roboter-Kollaboration. Wir präsentieren Experimente, die blickbasierte Benutzereingaben unter verschiedenen Bedingungen analysieren und helfen, ein 3D-Modell für das periphere Sehen zu entwerfen. Darüber hinaus stellt diese Arbeit mehrere neue Algorithmen für die blickbasierte Interaktion vor, wie die deiktische Referenz in mobilen Szenarien, die nicht-intrusive Benutzeridentifikation, oder die Nutzung des peripheren Sichtfeldes für neuartige multimodale Präsentationen. Diese Algorithmen sind in eine Reihe von Software-Werkzeuge integriert, mit denen 15 Anwendungsfälle für intelligente Umgebungen implementiert wurden. Diese Demonstratoren decken ein breites Anwendungsspektrum ab: von der räumlichen Interaktionen aus einem fahrenden Auto heraus bis hin zu Mixed-Reality-Interaktionen zwischen Mensch-Roboter-Teams.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to Prof. Wahlster, not only for giving me the opportunity and supporting me to work on my PhD thesis, but also for encouraging me in my ideas and believing in me since my early days at DFKI. Throughout these years, I truly enjoyed each of our conversations. In each discussion, either short or long, I always gained valuable insights and learned a lot. It was a pleasure and also a big honor to work with him closely.

I also want to thank Prof. Krüger who, without any hesitation and with great enthusiasm, agreed to review this thesis. His research has always inspired me, and I am very happy to have him on the reviewers board. I also thank Dr. Christian Müller for his encouragement and support. He always believed in me and gave me valuable insights. With his unconditional help, I had the great opportunity to work on interesting projects in the field of Automotive User Interface. Likewise, I want to thank Dr. Tim Schwartz and Dr. Michael Feld who had always listened carefully to my concerns and were excellent companions throughout my PhD studies and research. They always provided me a supportive and encouraging working atmosphere. Thank you again for the "Progress Chart"! Also, I want to thank Dr. Dietmar Dengler, who always supported my ideas and helped me realize them in many small and big projects. I also want to express my thanks to Andreas Luxenburger, who helped me with his talent and effort to realize various experiments and projects. During my research, I was also very lucky to work together with Dr. Daniel Sonntag. I thank him for his support and insights. This work would have not been possible without a great and inspiring working environment with supportive and creative colleagues at DFKI. Particularly, I thank the members of the Intelligent User Interface group, DFKI Corporate Communication in Saarbruecken and also the office of Prof. Wahlster: Dr. Anselm Blocher, Fabio Espinosa, Rafael Math, Christian Bürckert, Tanja Schneeberger, Winfried Schuffert, Yannick Körber, Guillermo Reyes, Reinhard Karger, Christof Burgard, Armindo da Silva Ribeiro, Heike Leonhard, Sabine Langlet and Andrea Nawrath-Herz. I also had the great opportunity to work with very talented and motivated students. I thank them for their out-of-the-box ideas and also enthusiasm for our research. I particularly thank Dieter Merkel, Caspar Jacob, Jonas Mohr, Felix Scherzinger, Fabian Spaniol, Nicolas Erbach, Bach-Thi Dinh, Oliver Jank, Roy Chiranjit and Sina Zand Vakili.

Last but definitely not least, I would like to thank my family and friends. In particular, I want to thank my wife Shiva for her continuous patience, encouragement and moral support

throughout my PhD studies. She inspires me very much. I am very grateful to have her in my life. I also thank "Ghahremane Baba", Artemis! She is the joy of my life. She accompanied Shiva and me during the last phase of my PhD and gave us strength. I am also very thankful to my mother-in-law, Shirin Poorjam, who helped me organize my time for writing the thesis when Artemis was born. I am especially grateful to my mother, Parvaneh Elahi, for her continuous support during my whole life. Without her I could not have gone to university. Without her this PhD thesis would not have been possible.

Contents

1 INTRODUCTION

Human beings have always been interested in interacting with the surrounding environment. Signs of this interest appear in thousand-year-old handicraft and literature. As the environment around humans changed, the type of interaction changed as well. However, what did not change were the senses through which we perceive the environment. Our perception is a result of a complex information flow between the different sensory receptors and the brain. These sensors produce neural signals which are then processed in the brain. This process forms a part of our cognition of the environment. Perception and cognition are tightly related, and it seems that there is no distinct border between them so that one can find out where perception ends and cognition begins [Tacca, 2011]. Our actions are successive to these two stages and can also influence them. Generally, one can think of perception, cognition and action as a connected sensory-motor loop which forms the human-environment interaction.

Considering the discussed sensory-motor loop for human-environment interaction, the research presented here concentrates mainly on interactions based on our visual perception. It studies eye-based interfaces which help humans to enrich their interaction with the surrounding environment. Integrating information about users' focus-of-attention into multimodal systems can help them better interpret users' intentions and provide personalized interaction which is adapted to the user's needs [Qvarfordt, 2017]. Each movement of the eye gaze plays an important role in humans' interactions and collaborations [Yu et al., 2012]. However, gaze includes only the vision in the fovea region of our eyes, which has ca. 2 deg of arc diameter [Strasburger et al., 2011]. Our total visual perception is far greater than that. Due to Barfield et al. in [Barfield et al., 1995], the binocular field of view of the human being is 120 degrees horizontally and 135 degrees vertically[1]. Comparing this large view field with the 2 degrees of foveal vision (eye gaze) shows the amount of visual information which we are receiving but not focusing on. According to [Duchowski, 2007]

[1] Both the monocular and binocular fields of view add up to 190 degrees horizontally [Barfield et al., 1995].

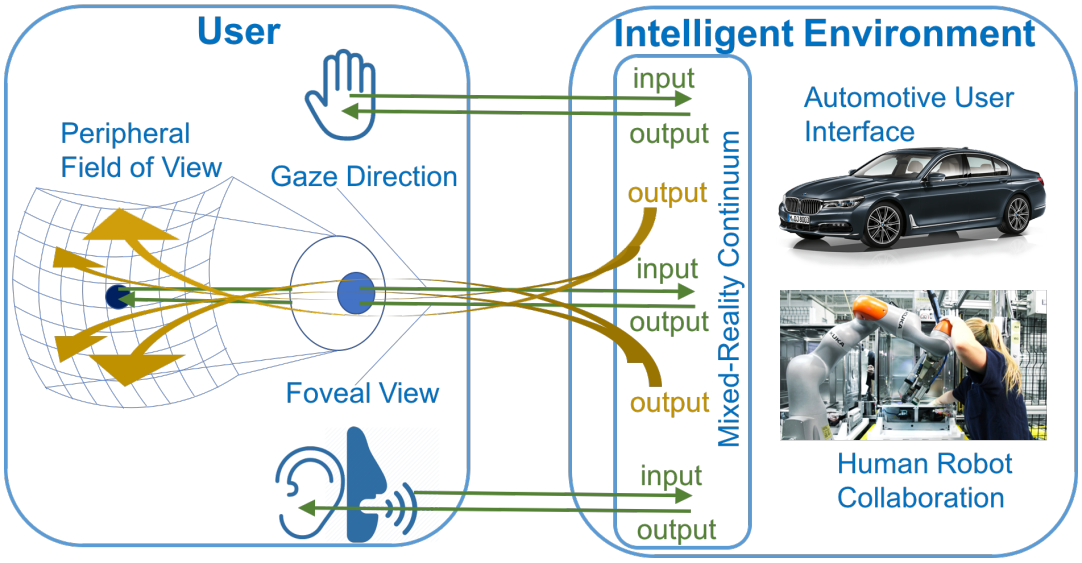

Figure 1.1: Two new modalities for Human-Environment Interaction: Eye-Gaze and output in the peripheral field of view.

the "useful" visual field extends to about 30 degrees; beyond this region human vision has very poor resolvable power and it is mainly used for the perception of motion. A comprehensive eye-based application should take both these regions (visual focus and peripheral vision up to 30 degrees and even beyond) into account. An eye-based[2] application should know what is currently in the visual focus of the user and which objects lay in different eccentricities on the peripheral vision. This research covers all of these fields. Figure 1.1 depicts the modalities which are addressed in this study. In addition to the haptic and speech-based interfaces, this thesis examines two new modalities for Human-Environment Interaction: eye-gaze[3] and output in the peripheral field of view. All of these modalities are considered in the whole spectrum of the mixed-reality continuum. For example, prototypes have been developed which enable interactions in the virtual environment. At the same time, these prototypes have counterparts which act merely in the real environment. Furthermore, both of these systems are connected to an augmented reality application which enables users to interact with physical or virtual objects at the same time.

In this research, scenarios from two main application fields are in focus: Automotive User Interfaces and Human-Robot Collaboration (see Figure 1.1). The reason for choosing these fields is the recent developments in these areas which make studying user interaction in the corresponding scenarios even more important than before. In the Automotive field, commercial autonomous driving seems to be just a few years

[2] In the presented study the term eye-based refers both to the gaze and also the peripheral vision.
[3] Regarding eye-gaze both input and output are considered (see Figure 1.1).

away considering the recent developments in Machine Learning and GPU computing. Many car manufacturers and software companies are investing more and more in this field. Volvo for example, intends to sell fully autonomous vehicles by 2021 and has already begun rolling out its semi-autonomous vehicles to test families[4]. In such autonomous vehicles, user-environment interaction will complement the current user-vehicle interaction. The passengers of these vehicles will have more possibilities to explore the environment around the car. The following is an example scenario. Either as a tourist driving through a new city or as a resident of a large city, one comes across interesting urban objects about which one wants to know more. These objects can be a building with historical or architectural significance, an artistic statue, a normal shopping store, or even a small house. If the vehicle identifies where the passenger is looking and can deliver information related to the objects in the visual focus, this will pave the way for a whole new level of passenger-environment interaction in the vehicle. The vehicle can provide this feature by considering the contextual information of the vehicle and the passenger while he is referring to the target building.

Recent years have also witnessed many advancements in the fields of mixed-reality and lightweight robotics. Regarding mixed reality, many new advanced devices have been introduced by various companies. These devices vary from fully virtual reality headsets to augmented reality glasses. Many of these also include internal positioning modules which are very promising relative to the past models. In the field of lightweight robotics, in the past few years several manufacturers have introduced robots which are so safe, they do not need a cage for human safety. It is possible for a worker to share her workplace with these robots and collaborate and interact with them in a way similar to a human co-worker. Combining these two technologies (mixed-reality and lightweight robotics) is very promising for building advanced user interfaces for Human-Robot Collaboration. This is particularly true when the human's focus-of-attention is considered while designing the user interface for this collaboration [Admoni and Scassellati, 2017]. Therefore, this field was also selected as one of the scenarios to study humans' visual attention in advanced user-environment interaction.

The main research question answered in this thesis is:

- **How can eye gaze and peripheral vision support humans' interaction with the environment?** As already mentioned, integrating humans' focus of attention plays an important role in the interaction of the human with the environment in the whole Mixed-Reality continuum. The applications which can benefit from this information range from interactive applications in modern

[4]https://www.cnet.com/roadshow/news/volvo-kick-starts-self-driving-program-with-civilian-riders/ (last visited on 12.12.2017).

vehicles to advanced collaboration methods between humans and robots.

In order to answer this main question, the following related questions must be answered:

- **What are the characteristics of humans' visual perception in the fovea and peripheral vision and how are these characteristics used by previous eye-based interaction paradigms? What are the drawbacks for these systems and how can we overcome them?** This analysis leads to improvement of the state of the art in the eye-based applications.

- **What are the limits for visual reference resolution when using gaze to refer to objects in an inside or outside environment?** Reference resolution plays an important role in the eye-based environment interaction. This question examines how the reference resolution differs in an outside environment from a controlled indoor lab environment.

- **How can general peripheral vision be modeled for gaze-aware intelligent user interfaces?** Building on previous studies in the field of human visual perception, a model is developed that offers a visibility measure which can be used in the eye-based interactive applications.

- **How can we design new algorithms to enrich eye-based interactions in Automotive User Interfaces and Human-Robot Collaboration scenarios?** This question studies how it is possible to design algorithms that use gaze, head direction, or peripheral vision to enrich eye-based interfaces. In particular, applications in the fields of Automotive User Interfaces and Human-Robot Collaboration are considered.

- **How can we expand the spectrum of human-environment interaction by integrating additional eye-based input and output channels?** This question specifically examines how various input indicators of the visual focus of attention (such as eye gaze and head pose) assist the human-environment interaction. Regarding output, it investigates how various applications can benefit from the presented peripheral visibility model both in augmented and virtual environments.

The rest of this thesis is organized as follows. Chapter 2 gives an overview of the foundations of visual perception and eye-based applications. In this chapter, first the basics of visual perceptions are described. These are topics like light, the anatomy of the human eye, the different types of eye movements and other important topics

regarding human visual perception. This chapter also gives an overview of the available hardware and software solutions which are very useful for building eye-based applications. These include, for example, different types of eye-trackers or different types of mixed-reality applications. Finally, this chapter discusses the role of eye-based interfaces in human-environment interaction. Chapter 3 gives an overview of the current state of the art in eye-based human-environment interaction. It is divided into two sections: gaze-based human-environment interaction, and peripheral vision in human-environment interaction. In each of these categories, the scenarios selected are the applications which have their focus on the fields described above. Chapter 4 gives details on two analyses which are conducted in this thesis. In the first one, the imprecision of the 3D gaze using an off-the-shelf eye-tracker is measured. This measurement is performed indoors in a controlled lab environment and also outdoors in the vehicle. Furthermore, in this analysis the effect of eye tracking imprecision on reference resolution for small urban objects is studied. The second part of Chapter 4 is concerned with a 3D peripheral analysis model. This model, which is one of the contributions of this thesis, defines the visibility of objects in our peripheral vision. It is developed and evaluated through an experiment with 68 participants. Chapter 5 goes through a series of algorithms which are developed in this thesis for using gaze to refer to objects in highly mobile scenarios or for peripheral vision analysis based on the solid angle of the objects. Chapter 6 describes the developed toolkit for eye-based interaction and analysis (TIFoA). This toolkit implements the algorithms developed in Chapter 5 and provides the possibility to be used for developing different applications. Chapter 7 introduces a set of prototypes which are developed using TIFoA. These prototypes range from a 3D-based reference resolution application for vehicle infotainment scenarios to mixed-reality human robot collaboration and applications for peripheral view analysis. Finally, Chapter 8 revisits the research questions stated before and provides answers to these questions on the basis of the research performed in this thesis. Furthermore, this chapter lists the various contributions of this thesis and also gives an outlook for the future of the discussed topics.

2 FOUNDATIONS OF VISUAL PERCEPTION AND EYE-BASED APPLICATIONS

Our perception of the environment is a result of a complex communication between the different sensory receptors and the brain, which in turn processes the received signals. As it is mentioned by [Schwartz, 2012], the number of senses that we possess varies throughout the literature as there is no agreement about the definition of the sense. Figures 2.1 and 2.2 depict two different categorizations of the major senses presented by [Snowden et al., 2006] and [Mather, 2006], respectively. For each of these senses, the human body has special sensory receptor cells. The process by which the sensory receptor cells convert environmental energy (e.g. light, sound) into electrical neural signals is called Transduction [Mather, 2006]. This Transduction is achieved by a variety of methods, one example is light absorption which triggers molecular changes in photoreceptors (in the case of Vision), another example is mechanical deflection of tiny hairs by fluid currents in the inner ear (in the case of Hearing). The resulting neural signals are then processed in the brain. This process forms a part of our cognition[1]. Perception and cognition are tightly related, and it seems that there is no distinct border between them so that one can find out where perception ends and cognition begins [Tacca, 2011]. Our actions are successive to these two stages and can also influence them. Generally, one can think of perception, cognition and action as a connected sensory-motor loop which forms the human-environment interaction.

The research presented here does not concentrate on cognition but rather on perception and action. Regarding perception, the focus is on visual perception in the form of a geometrical analysis. This analysis considers the position of an observer and his gaze direction in a modeled environment. These analyses may include for

[1] Definition of cognition in Oxford dictionary: The mental action or process of acquiring knowledge and understanding through thought, experience, and the senses.

	Vision	Hearing	Touch	Smell	Taste
Physical messenger – what carries the information?	Light	Sound	Surface shape	Chemicals	Chemicals
Distance – how far away does the sense tell us about stuff?	Close + far	Close + far	Close	Close + far	Close
Spatial detail – how well can we resolve fine detail in the scene?	High detail	Poor detail	High detail	Poor detail	Poor detail
Time detail – how well can we resolve whether things are changing fast in scene?	High detail	High detail	Medium detail	Poor detail	Poor detail
Does the sense start and stop rapidly?	Yes	Yes	Yes	No – smell lingers	No – have aftertaste

Figure 2.1: Categorizations of the major senses presented in [Snowden et al., 2006].

Sense	Stimulus	Receptor	Sensory structure	Cortex
Vision	Electromagnetic energy	Photoreceptors	Eye	Primary visual cortex
Hearing	Air pressure waves	Mechanoreceptors	Ear	Auditory cortex
Touch	Tissue distortion	Mechanoreceptors, thermoreceptors	Skin, muscle, etc.	Somatosensory cortex
Balance	Gravity, acceleration	Mechanoreceptors	Vestibular organs	Temporal cortex
Taste/smell	Chemical composition	Chemoreceptors	Nose, mouth	Primary taste cortex, olfactory cortex

Figure 2.2: Categorizations of the major senses presented in [Mather, 2006].

example a reference resolution for finding the object in the user's focus or finding a visibility measure for objects in the user's peripheral vision. The considered use cases are selected from the applications in automotive and Mixed-Reality scenarios. In this chapter the required background for this research is presented. This includes theoretical concepts as well as software and hardware tools. Section 2.1 presents the whole process of vision including the structure of light, the physiology of the human eye and other relevant topics. It also describes the concept of foveal and peripheral vision including a 3D model of the human visual field. Section 2.1 also provides details on visibility measure including visual acuity and solid angle based visibility measurement. Section 2.2 discusses the different forms of the currently available Eye-Tracking technologies. Section 2.3 gives background on the whole Mixed-Reality spectrum, beginning from reality without any augmentation up to fully immersive virtual reality. Section 2.4 discusses the different techniques for environment reconstruction in indoor and outdoor environments. Finally, Section 2.5 provides an overview on the available technologies for position and orientation

assessment in indoor and outdoor environments.

2.1 Visual Perception

Despite the differences between the researchers for categorizations of the different human senses, the importance of vision and visual perception is indisputable. Our eyes are the sensors that receive the light which is emitted or reflected by different objects in the environment. The human visual system has a complex procedure of processing the received light. In this section the physiology of human vision is discussed. For this purpose, first the structure of light and then the anatomy of the eye are described. In addition, some important aspects of human vision like spatial vision or effects of eye movements in visual perception are described in more detail.

2.1.1 Light

Light is a form of radiant energy that is capable of stimulating receptors in the eye and evoking a visual sensation [Mather, 2006]. The behavior of light can be described as rays, as particles, or as waves. We experience light in our everyday life as rays. These rays travel normally in a straight line at a speed of approximately $3 \times 10^8 m/s$. From a light source, rays are emitted from the source point in all directions. These rays are deflected when they pass from one transmitting medium into another. For example, the rays that hit the water from air change their traveling direction in the water. This behavior is very important for understanding how images are produced by lenses or the human eye.

Isaac Newton (1642-1727) believed that light rays were composed of a stream of particles that traveled in straight lines. Christian Huygens (1629-1695), proposed that light travels from a source in the form of waves similar to water waves. His theory drew more attention when Thomas Young published the results of his experiment in 1801. In this experiment, light was passed through two adjacent slits in an opaque screen. Then the pattern created on a second screen was observed. The image formed on the second screen consisted of alternating light and dark lines. In the same way that water waves can cancel each other out or create bigger waves, the two light wavefronts canceled each other out (dark lines) or augmented each other (bright lines). According to James Clerk Maxwell (1831-1879), light waves can be described as transversely oscillating electrical and magnetic fields that propagate at finite speed. The wavelengths of electromagnetic radiation can be as small as $10^{-13} m$ or as big as several kilometers. The receptors in the human eye can be stimulated by only a very narrow band of wavelengths in the whole electromagnetic spectrum (be-

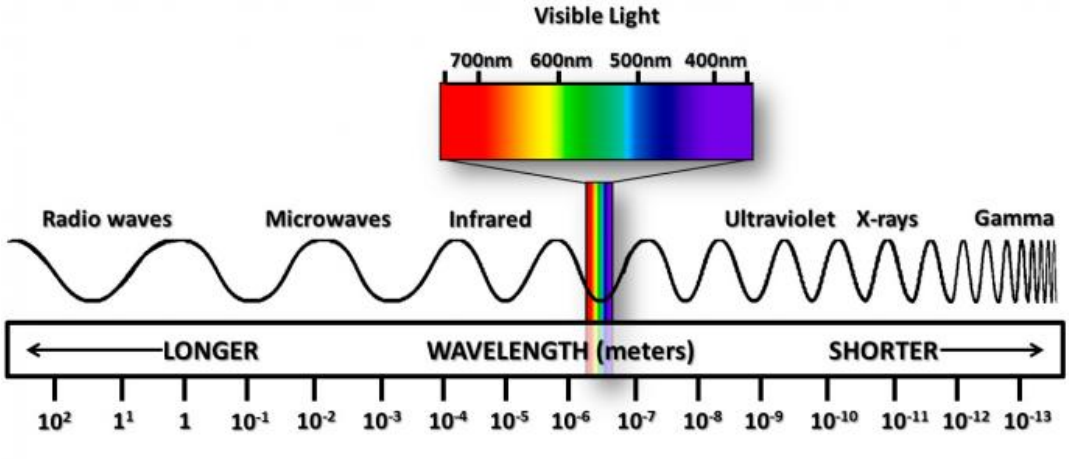

Figure 2.3: The receptors in the human eye can be stimulated by only a very narrow band of wavelengths in the whole electromagnetic spectrum[2].

tween 400 and 700 nm) [Mather, 2006] (see Figure 2.3).

Max Planck (1858-1947) proposed that light is emitted as a stream of discrete packets (quanta) of energy. He described the wave-like property of light to be occurring from vibration of each quantum at a specific frequency. Generally, the behavior of light is consistent with rays, waves, and streams of particles. All these three behaviors are important for understanding the visual perception [Mather, 2006]:

- Ray properties are useful for understanding how images are formed by optical devices such as eyes.
- Wave properties are important when considering the behavior of light at fine scales such as passing through small apertures (e.g. pupil).
- The quantal nature of light is useful when light intensity is so low that quantum absorptions can be counted individually.

As the behavior of light in the fine and quantum scale are not very relevant for the questions posed in this research, for the further investigations only the ray property of the light is considered.

[2]Source: http://www.ces.fau.edu/nasa/

2.1.2 The Human Eye

The function of the human eye is to catch light photons and direct them onto photoreceptors in order to begin the process of vision [Mather, 2006]. Figure 2.4 shows the main structures of the human eye. As a light ray reaches the eye, first it penetrates a transparent window called the cornea. The cornea is curved and acts as a lens. While entering the cornea, the light ray deflects. This is due to the fact that light travels faster through the air than through the cornea. Behind the cornea is the anterior chamber filled with aqueous humour, which is a watery fluid that regulates the pressure. Next to the anterior chamber are the iris and pupil. These two organs react to light intensity. With high levels of light the iris constricts and the pupil becomes smaller, which limits the amount of light passing through. In low lights the iris relaxes and the pupil allows more light to pass through. The lens is located behind the iris and pupil. The lens has less power than the cornea, however it has the advantage that it is adjustable. For example, when looking at a close object, the lens becomes more rounded to deflect the light ray accordingly. The vitreous chamber is located behind the lens and in front of the optic nerve. The vitreous chamber is filled with vitreous humour which keeps the eyeball in shape and the retina pinned to the back of the eye (see Figure 2.4).

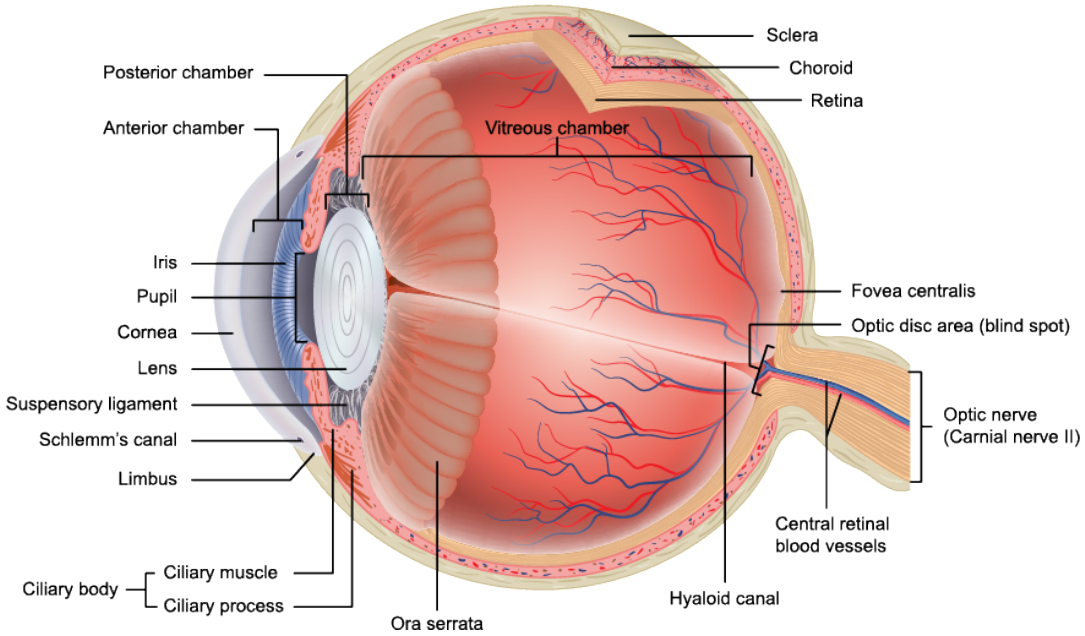

Figure 2.4: The main structures of the human eye[3].

[3]Source: https://www.genes-vision.ch/retinalearn/eye-anatomy/

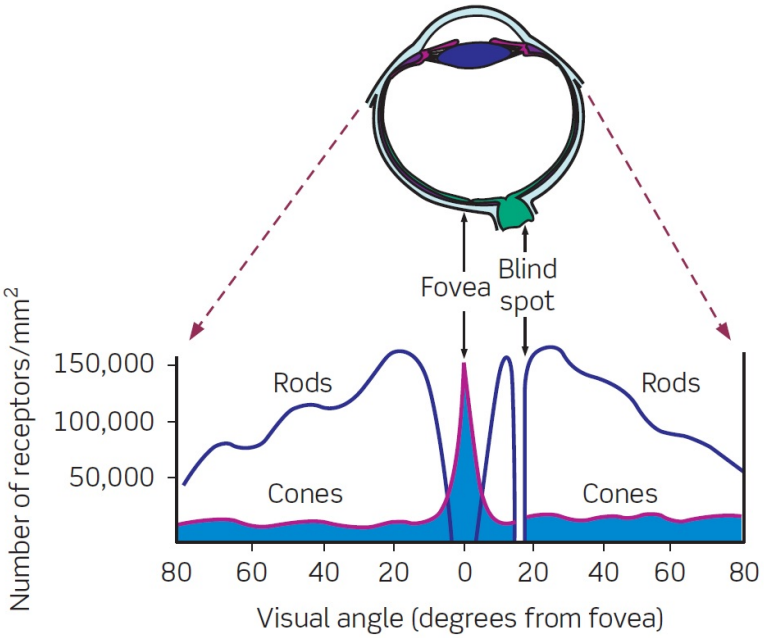

Figure 2.5: The distribution of rods and cones across the retina [Snowden et al., 2012].

The retina of each eye contains over 100 million photoreceptor cells that are responsible for transduction, which is converting light energy into neural activity [Mather, 2006]. Based on the shape of their outer segments, the human photoreceptors fall into two categories: rods and cones. The two types of photoreceptor cells are not evenly distributed across the retina. The central area of the retina known as the fovea is heavily populated by the cones. While rods do not exist in the fovea, they can be found heavily $12 - 15°$ into the periphery. Figure 2.5 depicts the distribution of rods and cones across the retina. The different properties of the cones and rods can be summarized as follows:

- Rods and cones contain different light-sensitive pigments. Cone pigments are 30-100 times less sensitive to light, and they only function when high levels of lights are available. On the contrary, rods are sensitive to low light.

- The number of rods are approximately 20 times more than the cones (120 million vs. 6 million).

- While all rods are basically the same, cones come in three main sorts which are sensitive to long-wave, middle-wave, and short-waves of light. These three types are responsible for our color vision (red, green, and blue).

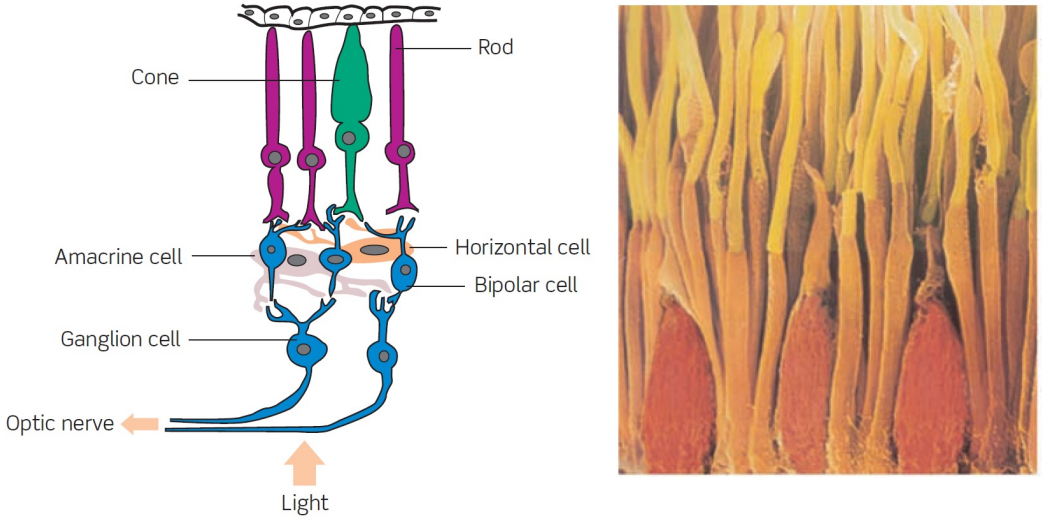

Figure 2.6: Rods and cones are connected to ganglion cells through the bipolar cells [Snowden et al., 2012].

Rods and cones are connected to the bipolar cells which in turn synapse with the last layer of cells in the retina which are the ganglion cells (see Figure 2.6). These ganglion cells are absolutely vital as they carry information from the eye via the optic nerve to the visual cortex. The optic nerve leaves the eye through the optic disc area, or the so-called blind spot (see Figure 2.4). As there are no rods or cones in the blind spot, the projected image on this area cannot be seen by the observer.

2.1.3 Spatial Vision

Although the term spatial vision includes all things related to seeing the space around us, in visual perception it is usually restricted to non-moving two-dimensional luminance patterns [Varadharajan, 2012]. The ability of a human to perceive luminance variation across space is called contrast sensitivity function (CSF). In order to obtain the CSF, sine wave gratings are generally used as the stimuli. The FACT (functional acuity contrast check) chart which is used to clinically test the CSF contains five rows of sine wave gratings. While each row has gratings of fixed spatial frequency[4], the contrast decreases from left to right. There are several factors which can affect the CFS, such as optical factors, retinal and neural factors, or stimulus factors.

[4]Spatial frequency is a measure of how closely spaced light and dark areas are [Block, 2015]. Its unit is c/deg (cycle per degree).

2.1.4 Eye Movement

The human eye has six degrees of freedom for its movement: three translations and three rotations [Duchowski, 2007]. As a result of eye movement, the fovea will be repositioned. This repositioning can be a combination of the following five basic movement types: saccadic, smooth pursuit, vergence, vestibular, and physiological nystagmus (miniature movements associated with fixations) [Duchowski, 2007]. Vergence movements are used for depth perception. In this kind of movement both eyes move in opposite directions to be able to focus on objects at different distances. Vergence movements have two types: movements for far-to-near focus and movements for near-to-far focus. The vestibular system detects brief, transient changes in head position and produces rapid corrective eye movements [Purves et al., 2001]. These eye movements are the vestibular movements. In this case the eyes move in the opposite direction of the head. The speed of the head and the eyes are identical in the vestibular movements. Pursuit movements are involved when a person is tracking a moving target object. Depending on the velocity of the target object, human eyes are able to match its speed and move accordingly [Duchowski, 2007]. There are reports that eye velocity is most often less than 30 deg/sec. If the object speed exceeds this limit, the human eye starts to employ catch up saccades to bring the fovea back to the target object[5].

While looking at a static object, the human eye performs mainly saccades and fixations. However, in more dynamic situations in which the object, the observer or both are in motion, other eye movements occur. These eye movements are responsible for keeping the fovea aligned with the target object. So as the depth between the observer and the target object is changed, vergence movements are triggered. As the position of the object (in the same depth) changes, smooth pursuit movement is triggered. Finally, if the position of the object is static and the observer's head and body moves, the vestibular movement is triggered. In a highly mobile situation, for example in a driving scenario, all of these eye movements will occur. This makes it more challenging to design and implement an interaction algorithm which involves the human visual focus-of-attention. In the following, saccades and fixations are discussed in more detail.

Saccades are rapid eye movements which reposition the fovea in the visual environment [Duchowski, 2007]. Saccades can be voluntary or reflexive. The duration of a saccade can be between 10 ms and 100 ms. There are also reports that the average saccade duration is between 20ms and 40ms[6]. Saccades are ballistic, which refers

[5] Source:https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/types-of-eye-movements/

[6] Source:https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/types-of-eye-movements/

to the presumption that saccade destinations are preprogrammed and as soon as the saccade movement is calculated it cannot be changed. The movement direction of both eyes in the saccades are similar.

Fixations are eye movements that stabilize the retina over a stationary object of interest [Duchowski, 2007]. During fixations, only 3 or 4 square degrees (out of 25,000 available square degrees) are seen most clearly with the highest resolution [Irwin, 1992]. This area corresponds to the part of the world that falls on the fovea. Fixation is composed of miniature eye movements: tremor, drift, and microsaccades [Duchowski, 2007]. The duration range of a fixation is usually between 150 ms and 600 ms, however longer fixations have been also reported.

2.1.5 Visual Acuity

Visual acuity is the degree to which fine detail can be resolved by our visual system [Snowden et al., 2012]. The visual acuity is higher where the concentrations of cones is higher in the retina and it is at its peak in the fovea (see Figure 2.7). Besides the high density of cones in this region, the higher spatial resolution in the fovea is due to the differences in the neural connections in the retina. As mentioned, the photoreceptors (rods and cones) are connected to the bipolar cells which are in turn connected to ganglion cells. In fovea these interconnections are more exclusive, meaning a single ganglion cell is connected only to few cones or even a single cone [Hitzel, 2015]. This exclusivity results in high spatial resolution, as the brain can deduce exactly which photoreceptor in the retina has observed the light photon and in turn it is possible for the brain to compute the exact position of the light source. Other regions in the retina do not have this specification; if they had, the blind spot (though which the optic nerves travel) would be so big that it would take most place in the eye [Snowden et al., 2012]. As a result, in non-foveal regions of the retina numerous photoreceptors are connected to a single ganglion cell. As the optic nerve brings the information from this ganglion cell to the brain, it is not possible for the brain to deduce from which photoreceptor the information is coming. Hence the visual acuity is reduced.

Besides the mentioned density of ganglion cells, there are several other factors that affect visual acuity. Campbell and Green argue in [Campbell and Green, 1965] that in a scene viewed at constant high photopic luminance, there are two main factors which affect fine spatial acuity: The quality of the optics of the eye forming the image on the retina and the ability of the retina (coupled with the brain) to resolve the detail of the image. While the latter factor is caused amongst other things by the

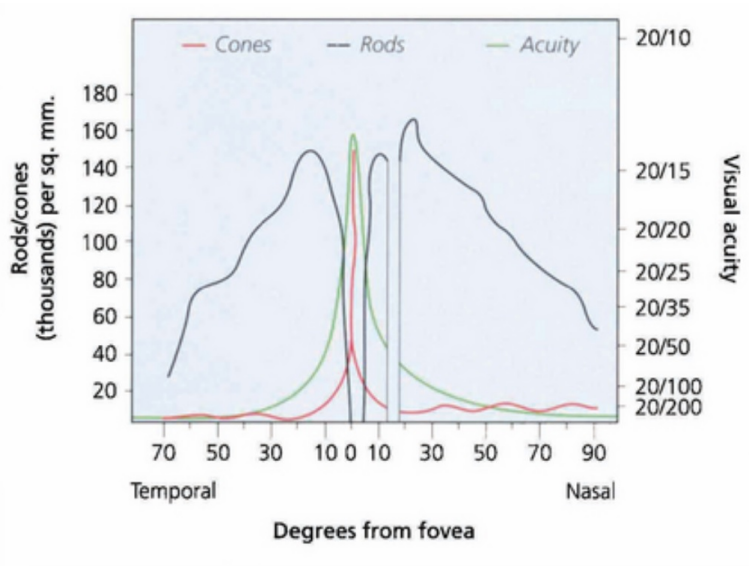

Figure 2.7: Visual acuity is higher where the concentrations of cones is higher in the retina [Spalton et al., 2013].

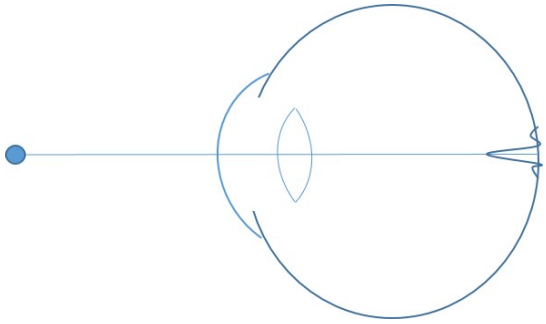

Figure 2.8: Due to distortions created by the optics of the eye, the image of a point source will be distributed on the retina as a Point Spread Function (PSF).

density of ganglion cells, the former factor is affected by aberration [7] and diffraction [8] [Smith and Atchison, 1997]. Due to these distortions, the image projected from a point source will be distributed on the retina as a Point Spread Function (PSF) (See Figure 2.8). This function describes the light distribution of a point source in visual space on the retina. The Rayleigh criterion states that two points or lines can be resolved if these objects are separated by the width of their PSF. If the images of these two objects are within this distance, our visual system cannot resolve them as

[7] The failure of all the light rays from an object to converge to a single image point after passing through an optical system is called optical aberration [Guenther, 2015].

[8] The departure of light from the predictions of the geometrical optics. This name is given by Francesco Maria Grimaldi (1613-1663) [Guenther, 2015].

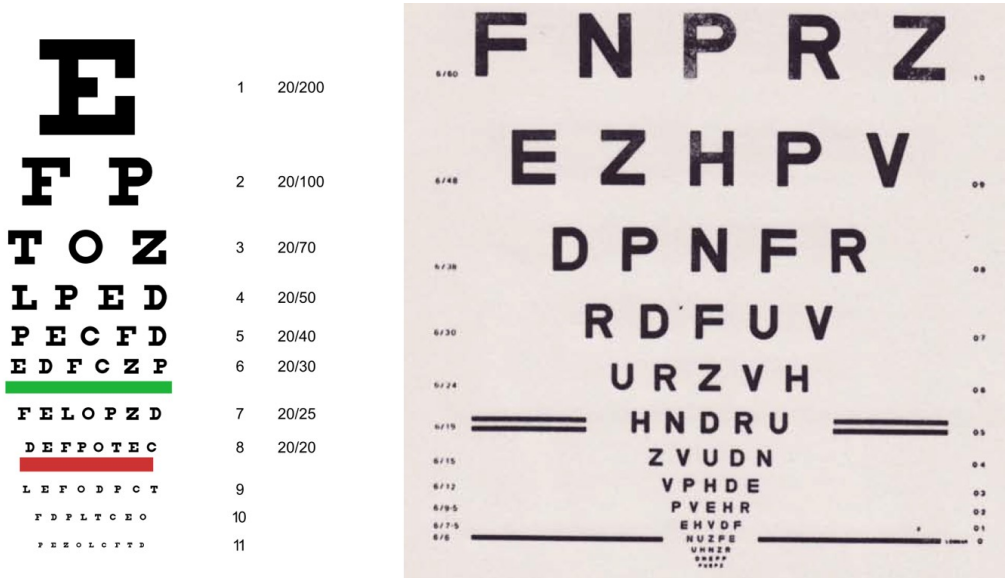

Figure 2.9: Left: A typical Snellen chart. It was originally developed by Herman Snellen in 1862 to estimate visual acuity. Right: A LogMAR chart. It was developed by the National Vision Research Institute of Australia in 1976 to enable a more accurate estimate of visual acuity than the Snellen chart.

two separate objects. Apart from the illumination and the location of the retina being simulated, visual acuity can also be affected by contrast and refractive error[9]. The following are the different kinds of visual acuity.

- Detection Acuity: Here the discrimination of the target details is not important, only the detection of the presence or absence of an aspect of a stimulus is important. For example, detection of the orientation of an Illiterate E.
- Recognition Acuity: Here recognition or naming the target is important, such as the different letters in the Snellen chart (see Figure 2.9).
- Localization Acuity: Here discriminating differences in the spatial position of segments of the test object is important. For example, finding a break or a discontinuity in a contour.
- Resolution Acuity: Here resolving smallest gap between spots or lines is important. In the case of parallel lines, it is called Grating Resolution.

A visual acuity test is one of several clinical vision tests (e.g. refraction test, visual field test, or color vision test). In the visual acuity test, the eye's ability to see details

[9]Refractive errors occur when the shape of the eye prevents light from focusing directly on the retina. Source: https://nei.nih.gov/health/errors/errors

at near and far distances is measured. It should be mentioned that this test mainly covers the acuity in the fovea. The test involves looking at symbols or reading letters of different sizes on an eye chart. Different eye charts are available for different purposes. The Snellen chart, E chart, and Landolt chart are three different charts for distance tests. These charts are positioned 6 meters away from the person. Each chart has several rows each with different sizes for symbols/letters. A person with 6/6 vision (or 20/20) can see at 6 meters what people with normal vision can see at this distance. The LogMAR chart is a more improved chart which is designed to enable a more accurate estimate of acuity [Bailey and Lovie, 1976]. The Jaeger chart is used to test near field acuity. It contains a few short lines or paragraphs of printed text to test the near vision. As with other charts, the text size gets smaller. The card is held about 36 cm from the eye. Figure 2.9 depicts different charts.

2.1.6 Foveal and Peripheral Vision

Foveal and peripheral vision have been subject of research for more than a century in different research disciplines such as ophthalmology, optometry, psychology and engineering sciences [Strasburger et al., 2011]. These studies concentrate on modeling human visual capabilities across different regions of the visual field including its limitations. The transition between these regions is smooth, so that there is no well-defined boundary between fovea and non-fovea regions [Strasburger et al., 2011]. Thus, the terminology is different in various studies. According to some researchers the fovea region has a 1 deg of arc diameter, others consider it to have 5.2° [Strasburger et al., 2011]. The same applies to other regions: parafovea 5-9 degrees, perifovea 9-17 degrees, central visual field 60 degrees etc. According to [Duchowski, 2007] the ”useful” visual field extends to about 30 degrees; beyond this region human vision has very poor resolvable power and it is mainly used for the perception of motion. As it is described in [Strasburger et al., 2011], the presented research considers anything outside 2 degrees as peripheral vision and anything within 2 degrees as foveal vision. The terminology ”different visual field regions” is used for referring to different distances from the foveal region.

The various capabilities of different visual field regions are not only caused by the differences in the structure of the human eye. Other factors such as how the received information (from eye) is processed also play a major role. For example, the number of neurons in the visual cortex responsible for processing the visual stimulus of a given size varies as a function of the location of the stimulus in the visual field [Cohen, 2011]. This fact is referred to as cortical magnification. Due to cortical magnification, a large number of neurons in the primary visual cortex process the data of a very small region of the central visual field and the stimuli detected in the peripheral visual field tend to be processed by a much smaller number of neu-

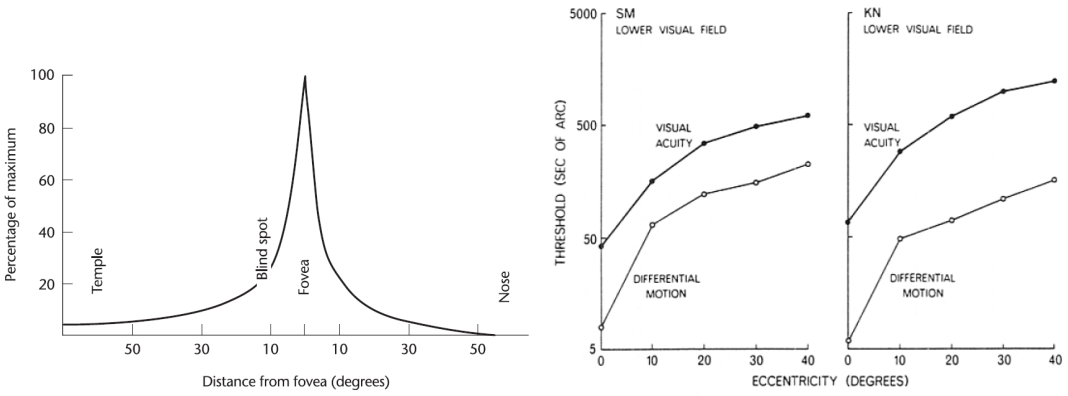

Figure 2.10: Left: The acuity of the eye falls off rapidly with distance from the fovea [Ware, 2013]. Right: Acuity and differential motion as a function of retinal eccentricity [Mckee and Nakayama, 1984].

rons [Cohen, 2011]. The presented research explores the different capabilities of the foveal and peripheral vision, however it does not explain the regarding neuropsychological backgrounds. In the following, some of the different limitations of peripheral vision (relative to foveal vision) are described. Afterwards some models for human visual field are discussed.

One important limitation factor of peripheral vision is spatial resolution, which is the ability to discriminate two nearby points in space [Anton-Erxleben and Carrasco, 2013]. Spatial resolution varies systematically across the visual field; the highest resolution is at the fovea (focus of our gaze) and it declines with increasing distance from the fovea [Anton-Erxleben and Carrasco, 2013]. Figure 2.10 depicts the decline in acuity outside the fovea, the spatial resolution is about one-tenth at the 10 degrees from fovea [Ware, 2013]. This measure is one-dimensional, so that it is used for a distance between two points (considering just a line). If the acuity per unit area is considered, according to the inverse square law, the acuity drops even more. In an area of 10 degrees of eccentricity from the fovea, the acuity will decline to one hundredth [Ware, 2013]. Another limitation factor of humans' vision is the lack of sensitivity of the peripheral vision to motion, relative to that of the fovea [Mckee and Nakayama, 1984]. The ability to detect relative motion between adjacent visual stimuli (differential motion) is less than spatial resolution at different locations of the retina [Mckee and Nakayama, 1984] (See Figure 2.10 right). The target size of the lowest differential motion threshold rages from 1° in the fovea to 20 degrees at 40° eccentricity[10].

[10]Eccentricity: where in the visual field, relative to the locus of fixation, a stimulus appears [Whitney and Levi, 2011].

Figure 2.11: When fixating on the bull's-eye near the construction zone, it is difficult to recognize the child on the left-hand side of the road, because of the presence of the nearby signs. However, it is relatively easy to recognize the child on the right-hand side [Whitney and Levi, 2011]. Visual crowding seriously impacts virtually all everyday tasks, including reading, driving and interacting with the environment. Shot by David Whitney, presented in [Levi, 2011] and [Whitney and Levi, 2011].

Another important limitation of the visual field is crowding, which defines the spatial resolution of a conscious object throughout most of the visual field[Whitney and Levi, 2011]. Crowding is the phenomenon in which objects that can be recognized when viewed in isolation are rendered unrecognizable in clutter [Levi, 2011]. Crowding sets a fundamental limit on conscious visual perception and object recognition throughout most of the visual field [Whitney and Levi, 2011]. Visual crowding seriously impacts virtually all everyday tasks, including reading, driving and interacting with the environment [Whitney and Levi, 2011] (See Figure 2.11). Another feature of crowding is that position information is lost, so observers frequently mistakenly report a flanker instead of a target [Levi, 2011], however it affects identification, not detection[Pelli et al., 2004]. To analyze crowded peripheral information, we make several saccades every second to bring the relevant information into our central vision where visual acuity is optimal [van Koningsbruggen and Buonocore, 2013]. Crowding is affected by factors such as the distance between the flankers and the target, their relative similarities and

global arrangement as well as attention. In the fovea crowding is rare; however, it is very strong at the periphery [Lev et al., 2014]. Crowding may be a natural result of a successful strategy for dealing with a bottleneck in visual processing; representing visual input via summary statistics allows the visual system to simultaneously reduce the information passing through the bottleneck, and yet encode a great deal of useful information about the visual world [Balas et al., 2009].

2.1.7 3D Model of the Human Visual Field

The different limitations of human vision have led researchers to build various models to describe the ability and boundaries of human vision. The paradigm of different Human Visual Field (HVF) models is based on the limitations of human vision and associated areas in the visual field. Figure 2.12 illustrates two different approaches for modeling the HVF. The model of Hatada et al. [Hatada et al., 1980] divides the visual field into four regions with different visual capabilities. For this purpose, they use both the vertical and the horizontal spherical angles to define these four regions. As a result, each of these regions features an elliptic shape in 2D while constituting a cone-like structure in 3D. Similarly, the model of Komatsubara [Komatsubara, 2008] divides the HVF into four regions while discriminating different capabilities of human vision with respect to recognizing text, shape and color. However, angular parametrizations are restricted to the horizontal dimension of the visual field. Barfield et al. [Barfield et al., 1995] propose a field of view model which also considers both spherical angles for defining the blind spot, binocular and monocular fields of view as well as the total vertical degrees of sight. There are also models which do not use very precise measures for defining the position of the limitation. Concerning visual crowding, He et al. [He et al., 1996] mention that it is stronger in upper parts than in lower parts of the visual field.

This research wants to pave the way for bringing 3D peripheral view calculations into gaze-aware intelligent user interfaces. In this context, two important choices must be made. The first one is concerned with an appropriate representation of the HVF in 3D that is applicable to a wide range of applications. The second choice refers to a suitable visibility measure associated with the projection of an observed object from the environment to different regions of the visual field. Concerning the type of the underlying visual perception model, we have to differentiate between the visual models that use just one spherical angle (horizontal or vertical eccentricity) and those methods that use both angles to define their paradigms.

According to our purpose, we opt for 3D while excluding 2D models of the HVF, which basically neglect one viewing dimension. Thus, we use the 3D perceptual model of Hatada et al. Regarding the analysis of an object's appearance, we are

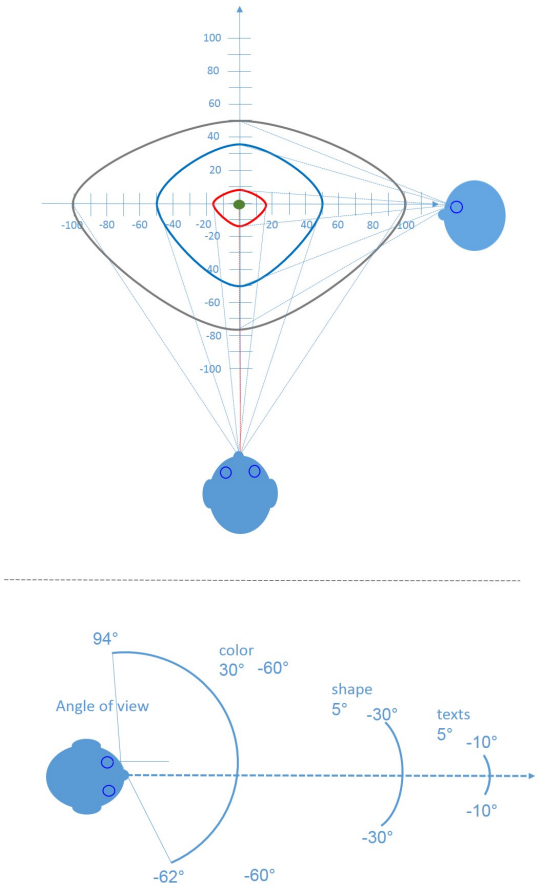

Figure 2.12: Models of the human visual field. Top: Model of Hatada et al. using horizontal and vertical angles for parametrization. Bottom: Model of Komatsubara. Visual regions are defined over one angular dimension.

interested in the area an object occupies in regions of the visual field from the observer's point of view. Hence, a solid angle seems to be an appropriate choice for the basis of our visibility measure. Solid angle together with the 3D perceptual model of Hatada et al., constitute the theoretical foundation of one of our general 3D peripheral view calculation algorithms, for which we give detailed explanations in the following chapters.

Although the concepts of visibility calculation presented in this research can be applied to any model that states a 3D representation of the HVF, the Hatada model is considered for further investigations. The Hatada peripheral view model is very suitable for this purpose because of its detailed descriptions with respect to the characteristics of the defined regions and its exhaustive capture of the HVF following a 2D angular parametrization. The model divides the visual field into the following

four regions with corresponding angular boundaries (see Figure 2.12):

- The discriminatory visual field ($3°$ circular).
- The effective visual field ($3°$ to $15°$ horizontally on each side, $8°$ upwards, and $12°$ downwards).
- The induced visual field ($15°$ to $50°$ horizontally on each side, $8°$ to $35°$ upwards, $12°$ to $50°$ downwards).
- The supplementary visual field ($50°$ to $100°$ horizontally on each side, $35°$ to $50°$ upwards, $50°$ to $75°$ downwards).

As the name states, in the discriminatory visual field, an observer has high precision discriminatory capabilities and perceives detailed information accurately with a visual acuity of over 0.5. Within the effective visual field, the visual acuity falls to about 0.1, while the discrimination of a simple figure can still be accomplished in a short period of time. This is the range within which an observer looks naturally at an object without head movement and is able to effectively process the information perceived. The induced visual field constitutes the range within which an observer has discriminatory capabilities to the extent of being able to recognize the existence of a visual stimulus. Hence, information displayed to the user which falls in this range should feature a reduced level of detail in terms of minimalistic representations. The HVF is complemented in terms of the supplementary visual field which states a range with no direct functional role in the perception of visual information. All it provides is a supplementary function in the sense that a shift of the observer's gaze can be aroused in response to abrupt stimuli.

Having a suitable 3D representation of the HVF at hand, we can now establish the mathematical basis for a peripheral view calculation model. This involves defining solid angle-based visibility measures for both the visual fields as well as discretized target objects that are projected into these fields. In the next section the solid angle-based visibility measure is described.

2.1.8 Solid Angle-based Visibility Measure

As mentioned before, a commonly used measurement unit in different research studies concerned with peripheral vision analysis is angle (e.g. between object's boundaries). As a one-dimensional measurement unit, it appears appropriate for these studies because they mostly use it as a measure of distance between two objects in a spherical coordinate system. However, as a one-dimensional measure, it seems less

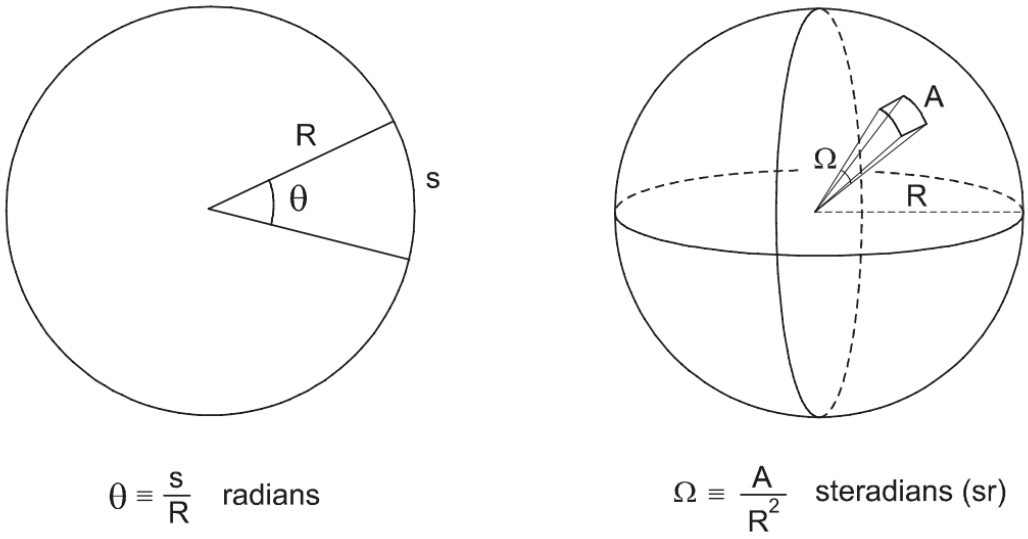

Figure 2.13: The solid angle Ω is defined as 2D equivalent of a conventional angle θ. It determines how large an object appears to an observer.

suited for analyzing and implementing human-environment interaction in 3D. Thus, this section explains the concept of solid angle.

A solid angle Ω [Quimby, 2006] constitutes a two-dimensional angle in 3D space that is subtended by an object from a specific point of view. This way, it provides an intuitive measure for how large an object appears to an observer. Basically, the concept of solid angle is defined analogously to that of a conventional 1D angle θ as illustrated by Figure 2.13. In this respect, the measure states the fraction of a unit sphere's area covered by an observed object rather than the fraction of a circle's circumference. Consequently, the total solid angle of a unit sphere, which is measured in steradian (sr), is $4\pi\ sr$. The general equation for calculating the solid angle of an arbitrary oriented surface subtended at a point [Masket, 1957] is given by:

$$\Omega = \iint \sin(\theta)\ d\theta\, d\phi\ ,$$

where θ and ϕ state the polar and azimuthal angles of a spherical coordinate system, respectively. In Figure 2.13 (right), the solid angle subtended by area A at the origin point of the sphere is measured by the area Ω on the surface of the unit sphere.

2.2 Eye-Tracking

There are four main categories of methodologies for eye movement measurement [Duchowski, 2007]:

- Electro-OculoGraphy (EOG). EOG relies on measuring the electric potential differences of the skin surrounding the ocular cavity[11]. For this purpose, several electrodes must be placed around the user's eyes.
- Scleral contact lens/search coil. This is one of the most precise eye movement measurement methods which involves attaching a mechanical or optical reference object on a contact lens which is worn by the user.
- Photo-OculoGraphy (POG) or Video-OculoGraphy (VOG). VOG is a method of recording eye movement through the use of digital video cameras and infrared light [Gans, 2001]. This method, however, does not provide point of regard measurement [Duchowski, 2007].
- Video-based combined pupil and corneal reflection. This method utilizes cameras and algorithms to compute both the eye movement of the user and the point of regard in real-time.

There are also other sources which consider video-based combined pupil/corneal reflection and VOG as a single category [Lopez, 2010], as they both perform video-based eye-tracking. Because they are able to compute the point of regard in real-time, the eye-tracking systems which use the video-based combined pupil and corneal reflection method are most suitable for use in interactive systems [Duchowski, 2007]. Therefore, this eye-tracking method is used for the investigations of the presented research and its function is explained in more detail.

An Eye-Tracker can determine a user's eye gaze as a 3D vector in space or as a 2D point on a plane [Hansen and Ji, 2010], the former is called the Line of Sight (LoS) and the latter is called the Point of Regard (PoR). Generally, a user's gaze tracking is divided into two subprocesses: eye tracking and gaze estimation [Lopez, 2010]. Eye tracking measures the eye movement by tracking several eye features such as the pupil or iris [Lopez, 2010]. Gaze estimation uses the features extracted in the eye tracking subprocess to determine the LoS or PoR. Figure 2.14 depicts the different modules of a video-based Eye-Tracker [Hansen and Ji, 2010]. If the extracted eye features are inaccurate, the calculated LoS and PoR will be noisy as well. Some applications apply a smoothing on the final results of gaze estimation to reduce the

[11] Eye socket

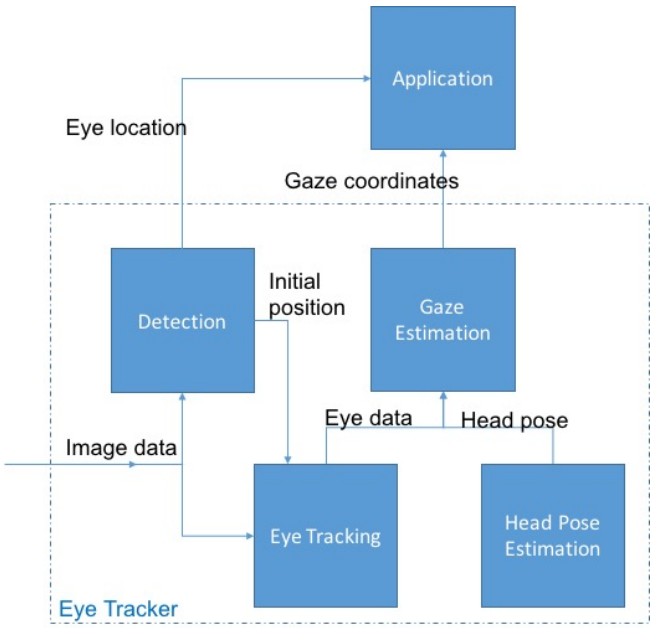

Figure 2.14: The architecture of a gaze-based application with eye and head tracking. Source: [Hansen and Ji, 2010].

jittering. Generally, three steps are required to use eye gaze as an input to control an interface [Lopez, 2010]:

1. Relevant eye features are detected and extracted from the video (considering video-based technique).
2. The data in 1 is used by a gaze estimation algorithm to calculate the LoS or PoR.
3. Considering the type of eye movement, a post-processing algorithm is applied to smooth the signal.

In the first step, in order to track the movement of the eye, three categories of computer vision techniques are used: shape-based, appearance-based, and hybrid methods [Lopez, 2010]. In the shape-based method the image data of the eye are fitted to a predefined eye model. The eye models in the appearance-based method are usually built from a large set of training images with different characteristics like illumination, head pose, eye shape, etc. The hybrid method combines different techniques. In the second step for gaze estimation, either geometry-based methods or interpolation methods are used. The geometry based models extract the gaze data from the eye image by using the geometry of the whole system, that is the eyeballs, the camera

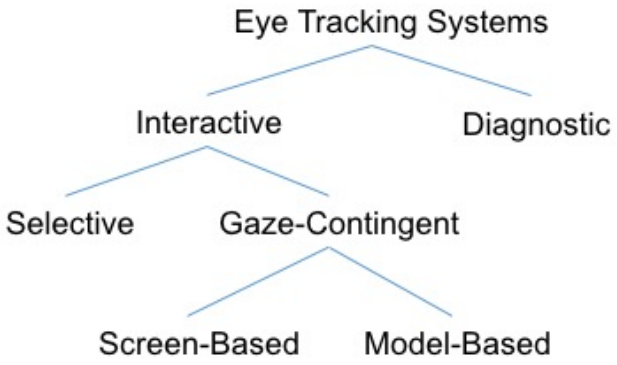

Figure 2.15: Different categories of eye-tracking systems as they are defined in [Duchowski, 2007].

and the screen setup. Here the 3D position of the hardware should be known. The interpolation methods use general purpose equations for this purpose without considering the 3D information of the setup. Both methods require some parameters which will be calculated during the calibration process [Lopez, 2010]. In the calibration process, the geometric characteristics of the user's eye are estimated and used as the basis for a customized gaze point calculation. However, maintaining this calibration for a longer duration is difficult so sometimes drift correction techniques are applied [Sundstedt, 2010].

After the three named steps, the resulting LoS or PoR is sent to the gaze-based application. Generally, the various gaze-based applications can be grouped into two categories: diagnostic and interactive [Duchowski, 2007]. The interactive group of gaze-based applications has different sub groups which are shown in Figure 2.15. In the diagnostic applications, the eye tracker is used to record the different eye movements of a user in order to investigate the user's visual attentional pattern while given a stimulus. In the designed diagnostic eye tracking applications, usually the displayed stimulus does not react to the user's gaze. It is very helpful and even in some cases mandatory that the eye tracking device is installed in an unobtrusive way in such diagnostic applications.

The gaze-based interactive applications are grouped into Selective and Gaze-Contingent subgroups. The selective subgroup contains all the applications which enable the user to perform selection or manipulation with direct line of sight. These actions can be performed in a unimodal or multimodal system architecture. Chapter 3 provides several examples for such systems. The Gaze-Contingent subgroup includes all applications that by using the real-time gaze tracking data and while considering the visual information processing capacity of the observer, attempt to balance the

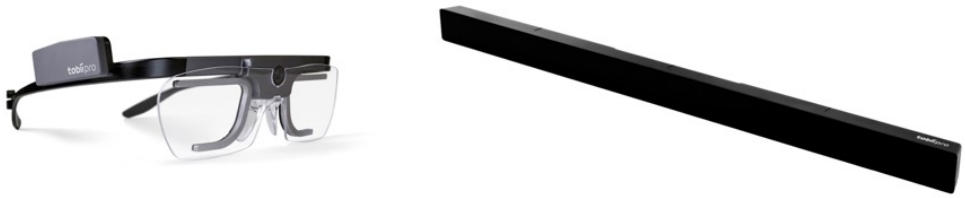

Figure 2.16: Mobile and stationary eye-trackers.[12]

amount of displayed information [Duchowski et al., 2004]. The Gaze-Contingent Displays (GCDs) are either screen-based or model-based. Considering the user's PoR on the display, GCDs degrade the resolution of peripheral image regions. This results in reduction of the required computational performance during image transmission, retrieval or display [Duchowski et al., 2004]. Applications of such a technology vary from flight, medical and driving simulators to generally virtual reality and remote piloting or teleoperation [Reingold et al., 2003]. As GCDs are very useful for reducing performance consumption, they are ideal for applications which run on devices with limited resources like wearables and smartphones. In virtual reality, model-based GCDs reduce the resolution by directly manipulating the model geometry before the rendering. Duchowski et. al mention in [Duchowski et al., 2004] that as of 2004, using this technique in immersive displays is a standard practice. For this purpose, Clark's original criteria of descending the object's hierarchical Level of Detail (LOD) is used [Clark, 1998]. Similar to the model-based GCDs, the screen-based technique alters pixel-level information on the display to match the resolvability of a human retina [Duchowski et al., 2004].

It is possible to divide the video-oculographic eye tracker systems into two main categories: *remote* and *head mounted*. In the remote systems, the camera and the light source are usually installed below the computer screen. The setup is non-intrusive and the user also often has the possibility to move his head in the view box of the camera. It is important however that the user's eyes are visible to the camera (and the light source). The head-mounted systems are installed on glasses or helmets which are then worn by the user. This setup also includes cameras and light sources. The advantage of such systems is that they enable the user to have mobile gaze interaction with the environment. Figure 2.17 shows two examples of remote and head-mounted eye trackers.

Two concepts which help to understand how good the performance of an eye-tracker

[12] Source: www.tobii.com

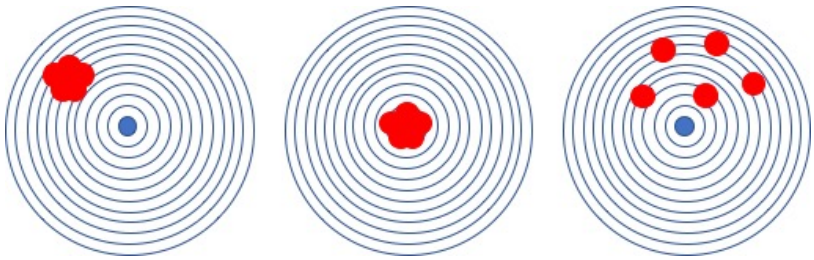

Figure 2.17: Precision and accuracy of an eye-tracker. Left: poor accuracy but good precision. Middle: Good accuracy and good precision. Right: poor accuracy and poor precision.

is are accuracy and precision. Accuracy is defined as the average difference between the real stimuli position and the measured gazed position[13]. Precision is defined as the ability of the device to reliably reproduce the same gaze point measurement. For this purpose, one compares the variation of the Root Mean Square (RMS) of the successive recorded data samples. Figure 2.17 depicts some examples for accuracy and precision. Depending on the type of application, different accuracies and precision are needed. It is also possible to build a useful, interactive application with a not very precise or accurate eye tracker. Accuracy varies for different participants and experiment conditions. Factors such as environment illumination, position of eye in the track box and the quality of calibration affect the accuracy[13].

2.3 Mixed Reality Continuum

The advances in computer graphics technology provide the possibility to totally immerse one into a virtual world and perform interactions there. Figure 2.18 depicts two examples for this purpose. These devices are standalone or mobile. In order to perform interactions, the user has the possibility to use a dedicated controller. The position of the user is also tracked, so that the movements in the real world are translated to the character movements in the virtual world. Regarding HTC Vive, the exact position and orientation of the user's head is measured via infrared sensors in the environment. The high sampling rate of this positioning system, and also the high refresh rate of the display (90 frames per second), provide a very good base for the 3D software engines and 3D applications.

Regardless of virtual reality, our daily life interactions happen in the real-world environment. However, in order to enrich or improve the quality of this interaction it is

[13]https://www.tobiipro.com

Figure 2.18: Left: HTC Vive Virtual-Reality headset. Right: Google Daydream View Virtual-Reality headset.[14]

possible to augment the real world with synthetic objects. The history of augmenting the real-world environment with synthetic objects dates back to the 19th century. John Henry Pepper (1821-1900), a British scientist, used Pepper's ghost illusion technique to make ghostly objects appear or to fade in or out of existence in a room. The audience were able then to see the real-world setup augmented with these objects in the form of an entertainment show. In the last 70 years, augmented reality has been subject of research in several disciplines. In early 1940, the Telecommunications Research Establishment in UK developed the first electronic Head Up Display (HUD) in a de Havilland Mosquito night fighter [Peddie, 2017]. In the late 1950s the first augmented reality helmet was developed by Philco Corporation in USA in the form of a closed-circuit television surveillance system with a helmet-mounted display [Peddie, 2017]. In recent years and with the introduction of mobile phones, augmented reality applications have seen a major boost mainly in the research labs. In 2004 Möhring et al. introduced one of the first AR application based on mobile phones [Mohring et al., 2004].

From the real-world environment into the virtual-world environment there are several stages in which these two realities can be mixed. These stages include augmenting the real world with synthetic objects or augmenting virtual worlds with real elements. Milgram and Kishino define in [Milgram and Kishino, 1994] a Mixed-Reality continuum (Virtuality Continuum) which includes all these stages from real world to virtual environments. Figure 2.19 shows this continuum. One end of the Virtuality Continuum consists of only real objects. As an example, Milgram and Kishino mention conventional video display of a real-world scene or a direct view of the same scene. The other end of the Virtuality Continuum consists of solely virtual objects, for example a computer graphic simulation. Anything between these

[14] Sources: https://www.vive.com/de/ and https://vr.google.com

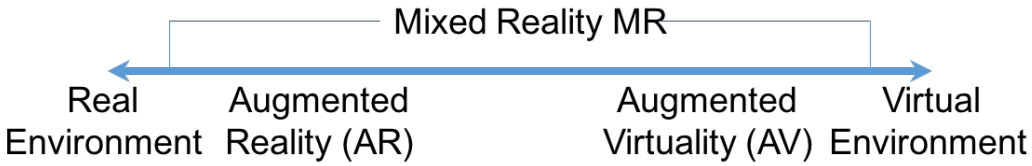

Figure 2.19: Virtuality Continuum as it is defined by [Milgram and Kishino, 1994].

two ends is considered to be a Mixed-Reality environment in which real world and virtual world objects are presented together. Regarding Mixed-Reality interfaces, [Milgram and Kishino, 1994] define following six classes:

1. Monitor based video displays: These are non-immersive windows-on-the-world (WoW) displays which overlay live video of the real world with a real-time computer generated virtual world [Metzger, 1993].
2. Video displays such as the one described in Class 1, however with immersive head-mounted displays (HMDs) instead of WoWs. In this case, the displayed videos should not show the immediate outside real world.
3. Augmenting real world with virtual objects which are optically superimposed using half-silvered mirrors in an HMD.
4. A video see-through similar to Class 2, however in this case the video shows the immediate outside real world.
5. In this Class the reality, for example in the form of a video, is added to a completely immersive virtual environment.
6. Similar to Class 5, however with partial immersion (e.g. large screen displays).

There are also other categories regarding AR displays. For example [Bimber and Raskar, 2006] divide the AR displays into three categories depending on the hardware distance to the human's eyes: head-attached, hand-held, and spatial. Billinghurst et al. divide Augmented Reality displays into four main types depending on how the virtual environment is combined with the real world [Billinghurst et al., 2014]: video based, optical see-through, projection onto physical surface, and eye multiplexed. Video based is a combination of a camera (as a window to the real world) and a display for the augmented reality. This technique uses a digital process to combine virtual information with video of the real-world view. It is possible to use this setup in four different categories [Billinghurst et al., 2014]: Video see-through, Virtual Mirror, Augmented

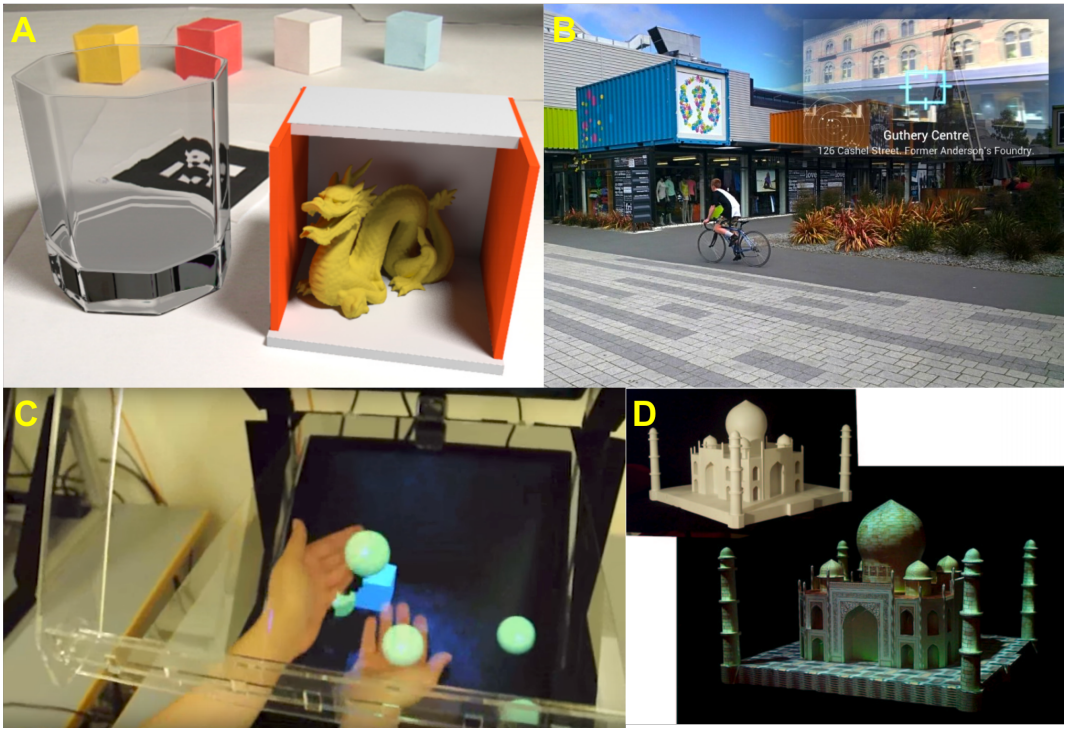

Figure 2.20: Four examples of different AR technologies: the video based (A) [Kán and Kaufmann, 2013], the eye multiplexed AR displays (B) [Billinghurst et al., 2014], optical see-through (C) [Hilliges et al., 2012], and projection (D) [Raskar et al., 2001].

Desk, and Remote AR. In the Video see-through case, the camera captures the immediate reality directly behind the display. In the Virtual Mirror, the camera captures the immediate reality in front of the display. In the Augmented Desk, the camera captures the table on which the display is located. Finally, in the Remote AR case the camera is physically separate from the display and sends the captured information over a network to the display.

The optical see-through augmented reality displays use optical systems to combine the real-world view with the virtual information [Billinghurst et al., 2014]. Head up displays (HUD) in cars or airplanes are examples of this technology. In this technique a beam splitter (e.g. half mirror or combined prisms) combines the observer's view of the real world with virtual information. Transparent projection films are also used in this kind of AR displays. This system is semi transparent so that the observer can see the real world through it, it also diffuses light to show the projected image. High quality transparent displays can enhance the quality of such AR technologies. Another two AR display categories are the Projection based AR Displays and the

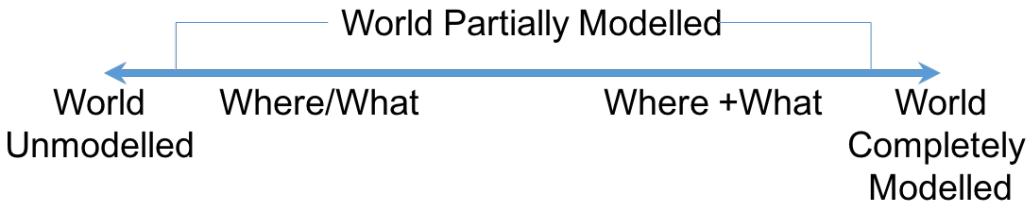

Figure 2.21: Extent of World Knowledge (EKW) dimension as it is presented in [Milgram and Kishino, 1994].

Eye Multiplexed AR Displays. In the former technique, physical objects in the real environment are used as a display medium and no other image plane is involved. For this purpose, the environment is equipped with projectors and the user is not required to wear any extra hardware. A disadvantage of this technique is that the augmented space in the environment is limited to the projection surface. In order to add more augmentation, more projectors are required. Regarding the Eye Multiplexed AR Displays, they do not provide the registration of the real and virtual world to the user. These displays register the virtual world with a physical environment, however with a different view than that of the observer. The user himself should then combine these two worlds in his mind and compensate for the difference between his own view and the one presented by the Multiplexed AR Display. Figure 2.20 shows four examples of different AR technologies: the video based (A) [Kán and Kaufmann, 2013], the eye multiplexed AR displays (B) [Billinghurst et al., 2014], optical see-through (C) [Hilliges et al., 2012], and projection (D) [Raskar et al., 2001].

With the exception of the Multiplexed AR Displays, in all other techniques the precision of the registration while combining the real and virtual environment play a major role in the quality of the final system. Depending on this registration it is possible to place the proper virtual object in the right place in the real environment. For this purpose [Milgram and Kishino, 1994] introduce the Extent of World Knowledge (EWK) as it is shown in Figure 2.21. In this continuum, one end is an unmodeled world, where there is no knowledge about which place in the real world corresponds to the 3D position in the virtual world. On the right end of this continuum is the completely modeled world. Here, the system has full knowledge about both environments just like any position in the 3D virtual world. Anywhere in between these two ends is the extent to which real and virtual objects can be partially merged into the same display. ”Where” refers to the case in which information about the location in the other world is available, for example when an operator can see and manipulate the position of a virtual object in the video stream of a remote place. Here the system itself has knowledge about the registration of the virtual and the real world. The ”What” label refers to the case in which the system has no knowledge about the

Figure 2.22: Top: Microsoft HoloLens and ODG R-9 AR glasses. Bottom: A Projection and a video see-through AR application used by Volkswagen AG for the XL1 model.[15]

spatial registration, however it does have information about objects in the video. In this case, the system may know about the existence of the object, however it may not have any information about its location or orientation. As we move more toward the right end of the displayed arrow in Figure 2.21, there will be more information about ”Where” and ”What” to be able to correctly register the two worlds together.

When the real environment is captured with a camera and registered with the virtual environment in a display, it is not important from which angle the user looks at the display to observe the mixed environment. This applies also when using projection on a flat object. However, when the virtual information is presented as an overlay on the direct line of sight of the observer, for example in a HUD or AR glass, the registration will be much more complicated. In this case the system should also take into account the view angle of the observer and his exact position in the room and render the virtual objects accordingly. Microsoft HoloLens, for example uses an array of sensors consisting of one Integrated Inertial Measuring Unit (IMU), four environment understanding cameras, one depth camera, one 2MP HD video camera, and a mixed reality capture sensor[16]. Together they map the 3D environment and

[15]Sources: www.microsoft.com, osterhoutgroup.com, and www.volkswagenag.com.
[16]Source: https://developer.microsoft.com/en-us/windows/mixed-reality

determine the position of the user's head and its orientation in the space. For this purpose, the ODG R-9 AR glass uses one IMU, one 13MP Autofocus Camera, two HD front-facing cameras for stereo capture and depth sensing, and one ultra wide-angle fisheye camera for enhanced environmental tracking and positioning[17]. These two devices, together with projection and video see-through, are currently used intensively in the pre-production and research labs in industry. Figure 2.22 depicts the Microsoft HoloLens, ODG R-9, and two AR applications with projection and video see-through. The next section covers the different methods for environment reconstruction and Section 2.5 describes more techniques regarding position and orientation assessment.

2.4 Environment Reconstruction and Modeling

Environment modeling or reconstruction is necessary to be able to process the spatial information in the world. This information can then be used in different modules of an application as world knowledge. The level of detail (LOD) of the 3D city models characterizes them and indicates their spatio-semantic complexity [Biljecki et al., 2017]. This concept is originated from computer graphics where it is used to balance the visualization quality and computational complexity. The LOD indicates in geographic information systems (GIS), the resolution of the data, its usability and also the degree of abstraction. However, despite its importance, it is still an ambiguous and undefined term [Biljecki et al., 2014]. This research does not go into detail on different LODs. The described modeling and environment reconstruction techniques are divided into three categories: 2D, 2.5D, and 3D. Regarding 2D, the two dimensional footprint of the objects is considered. In 2.5D this footprint is elevated with the height of the objects. In the 3D case, the 3D details of the objects are also considered. However, the fineness of these details is not considered in the form of LODs.

It is possible to define two general categories for environment modeling and reconstruction: indoor and outdoor. Although there are algorithms and methods which cover both categories [Sun et al., 2002], because of the differences between these two environments the modeling and reconstruction tools for indoor and outdoor environments are not always the same. This section covers the current hardware and software tools available for both these categories. However, the algorithms and methods which these tools use are not reviewed in detail as this topic is not in the focus of the presented research. The topics are presented in one section for environment reconstruction and another section for modeling.

[17] Source: http://osterhoutgroup.com/presskit/R-9-TechSheet.pdf

2.4.1 Environment Reconstruction

The process of environment reconstruction begins with the hardware choice. The utilized hardware can range from a simple camera to sophisticated depth sensors with different technologies. Generally, the method used for the 3D reconstruction can be divided into two categories [Ladikos, 2011]: active methods and passive methods. Depending on the used techniques, a different hardware such as single camera, stereo camera, or various depth sensors can be used. The sensor information of each of these pieces of hardware can also be combined with other data sources such internal sensors (for example IMUs) or projectors which use different visible or invisible patterns. Generally, important characteristics of the systems which perform 3D reconstruction are the real-time feature, the 3D data type of the results, the size of the 3D reconstructable object or environment, and the precision of the approaches. In the following, several examples for such systems are presented in the two categories: passive and active. As the 3D reconstruction is not the core topic of this thesis, these examples only illustrate the different approaches. The different details about the four named characteristics are not presented.

Active 3D reconstruction techniques include methods like laser range scanning, photometric stereo, structured light, and Time of Flight (TOF) cameras [Ladikos, 2011]. Laser range scanners use laser beams for their sampling. Using triangulation and the stereo geometry of the laser camera, the 3D position of the point can be computed using this method. The form of this laser beam can differ from a single dot to a line projection. In each case, in order to sample a scene, multiple scans are performed and then the results are stitched together. The output of the laser range scanning cameras are highly accurate 3D point clouds. In order to reconstruct a 3D environment, these single point clouds are then registered together. The structure light method acts like the laser range scanning. Instead of a point or a line, it projects a 2D light pattern into the environment. This technique then measures the deformation of the projected light pattern and from this deformation it calculates the 3D form of the environment. [Zhang et al., 2015] introduces a single fixed camera system which applies different illumination directions to reconstruct the 3D model of an object from multiple 2D images. This approach is called photometric stereo. They use a low-cost accessory to a commercial digital single-lens reflex (DSLR) camera for the 3D reconstruction. Regarding the TOF method, the setup is much simpler than the other mentioned techniques. This method creates the depth map of a scene by sending a modulated infrared light and measuring the phase shift of the reflected light. All components of this system can be packed in a camera sized housing. Some other variations of the TOF which is used for gesture recognition are even much smaller.

Active 3D reconstruction techniques are used depending on the task, for example if the 3D model of a dynamic scene is required, structured light or TOF are used. These

Figure 2.23: Top Left: Structured light system from Asus. Top Right: Laser range scanner from FARO. Bottom Left: Passive stereo camera from ZED. Bottom Right: Time of Flight camera from PMD.

two methods can deliver the 3D model of the environment with high frame rates. Microsoft Kinect V1 used structured light for analyzing the movement of the users in front of the XBOX while Kinect V2 uses TOF. Both these types of hardware are able to analyze the depth of the environment several times in a second and are appropriate for gaming applications. On the contrary, the comprehensive setup of the photometric stereo or the low sampling rate of a laser range scanner are not suitable for such a dynamic environment. Laser scanners are very good for precise offline modeling of indoor and outdoor environments. The photometric stereo has applications in analyzing the metal-plastic surface of automotive components[18]. Figure 2.23 depicts a laser range scanner from FARO, a structured light system from Asus, a TOF camera from PMD, and a passive stereo camera from ZED.

There are also systems that can simultaneously process several images shot at once for 3D reconstruction purposes [Gava and Stricker, 2015]. Such systems aim to develop a unified Structure from Motion (SfM) framework designed for central projection cameras. While many systems are limited to either partial or static scenes, there are also approaches that aim to reconstruct a dynamic scene in 3D. For example, [Furukawa et al., 2011] introduces a system which performs this 3D recon-

[18]http://www.vision-systems.com/articles/print/volume-19/issue-11/features/choosing-a-3d-vision-system-for-automated-robotics-applications.html

struction using multiple projector and camera systems. Their approach allows for reconstructing the entire shape of an object within a single scan at each frame. Regarding passive 3D object reconstruction methods, [Ladikos, 2011] divides them into three categories: Stereo, Multi-View Stereo, and Shape from Silhouette. Passive stereo 3D reconstruction is one of the oldest methods in environment reconstruction. It uses two cameras on a known baseline. To reconstruct the scene, it finds point correspondences between the images of the left and the right camera. Multi-view stereo method reconstructs a complete 3D object from a collection of images taken from known camera viewpoints [Seitz et al., 2006]. Regarding Shape from Silhouette method, as its name indicates it is a method of estimating the shape of an object from its silhouette images [Cheung et al., 2005]. This method is not concerned with the photometric consistency of the reconstruction, instead the 3D reconstruction problem is posed geometrically to find the shape which is maximally consistent with the observed silhouette images [Ladikos, 2011]. The shape which is recovered is called the visual hull. Of all the passive methods, the stereo 3D reconstruction is a very practical and popular method in research and also in industry. For example, [Geiger et al., 2011] introduces an approach which utilizes a stereo camera rig for a real-time 3D reconstruction. Their system is CPU based and can provide a new depth map of the scene at 3-4 frames per second (FPS). Figure 2.23 shows a stereo reconstruction camera from ZED (Below Left).

It should also be mentioned that for each of the named pieces of hardware, an algorithm and software are necessary to utilize the hardware sensors for the 3D reconstruction. For example, KinectFusion provides a software solution for 3D environment reconstruction using depth cameras based on the structured light method [Izadi et al., 2011]. There are also other software solutions which perform a sensor fusion for 3D reconstruction. For example, [Dewangan et al., 2016] introduces a system for 3D reconstruction of unknown indoor and outdoor environments using a single webcam and the wheel odometry of a robot. Here the data from the IMU sensors and robot kinematics of a low-cost robot platform is used together with the camera images to perform the 3D reconstruction. There are also studies which perform a 3D reconstruction of urban environments using video material together with GPS and internal vehicle measurements (for example [Pollefeys et al., 2008], [Mordohai et al., 2007] and [Akbarzadeh et al., 2006]). The next section gives an overview of how environment modeling is performed. Unlike environment reconstruction which is performed automatically, the environment modeling requires manual work to some extent.

2.4.2 Environment Modeling

In order to perform analysis on the digital representations of the indoor or outdoor environments, we need a model of these spaces. This model can have different dimensions (1D, 2D, 2.5D, 3D). The last section reviewed the different techniques for the 3D environment reconstruction. In order to acquire a digital 3D model of a whole city, it is possible now to use new 3D data collection techniques such as airborne LiDAR[19] or 3D photogrammetry[20][Kaňuk et al., 2013]. As it is not possible to reconstruct the smaller urban objects (such as traffic signs etc.) from high distances, it is possible to use drones to complement these city models. These drones can, for example use photogrammetry in a smaller area to reconstruct a traffic sign [Palummo, 2017]. Depending on the chosen reconstruction method, the resulting data is either a point cloud or a polygon mesh. These data do have basic semantics and allow only for partial analysis. Depending on the focus of the respecting application, a different 3D standard has been developed with a different purpose [Zlatanova et al., 2012]. For example, the standard for visualization provides fast and realistic features, for the data management the efficiency of the storage is important; for modeling, validity and topology are important. Finally for the data exchange the platform independence plays a major role [Zlatanova et al., 2012]. As the focus of this thesis is the human-environment interaction, it is very important that the developed systems can perform a spatial query in the reconstructed models. This section describes the different methods and tools available for querying and analysis on spatial data. For this purpose, the currently available systems for different dimensions are considered.

Regarding 1D digital representation of the environment, there exist many choices, as for example using float numbers. As this case is trivial it is not considered for the further discussions. Generally, for higher dimensions (2D, 2.5D, and 3D), there are two main methods: it is possible to use a dedicated Geographic Information System (GIS) or a game engine. Each of these choices is very comprehensive and has various advantages and disadvantages. This section considers these technologies with respect to the view point of the presented research questions. In the following, first the GIS and then the game engine are considered.

GISs were initially developed as a tool for storage, retrieval, and display of geographic information [Fotheringham and Rogerson, 2013]. One of the main components of every GIS is a spatial database. Spatial database is a database that defines spatial data types for geometric objects and makes it possible to store geometric data in regular database tables [Obe and Hsu, 2011]. Spatial data is data about posi-

[19] Light Detection And Ranging

[20] Photogrammetry is a technique to extract 3D geometric information from two-dimensional images or video.

tions, attributes, and relationships of features in space [Guptill et al., 1995]. Spatial databases also provide functions and indexing methods for querying and manipulating these data. The functionalities of the database management systems were mainly limited to 2D, however over time more functionalities for the third dimension were added to them [Zlatanova, 2006]. In recent years there has been a huge increase in the amount of spatial data produced by various devices such as smartphones and satellites [Eldawy and Mokbel, 2015]. Thus researchers and developers worldwide try different ways to store, analyze and visualize these data. For example, [Eldawy and Mokbel, 2015] present a MapReduce framework with native support for spatial data which includes programming paradigm for distributed processing for efficient large-scale data processing. Regarding a 3D dynamic update of the spatial database, only in recent years had it attracted the focus of the research community to add this feature to the available set of functions [Guo et al., 2016].

A game engine is a software framework for 2D or 3D game development, which helps with several core areas that most of the games have. These core areas are, for example, game physics, Artificial Intelligence (AI), networking, rendering 2D or 3D graphics, audio, memory management and more. The idea of integrating these modules in this framework is to prevent the developers from reinventing the wheel and implementing each of these core modules by themselves. This way, developers can concentrate more on the creative and innovative aspects of their development. Some of the game engines (for example Unity3D[21]) also have a graphical editor and a limited possibility for environment modeling. With more advances in game engine technology, its applications have become far more than entertainment gaming.

Generally, many game genres have profited from recent advances in information technology (hardware and software), for example pervasive games [Schmitz and Moniri, 2009]. A game genre which has profited a lot from the recent development of the game engines is the Serious Games. Serious Games are the kind of games that have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement, however, this does not mean that they are not allowed to be entertaining [Abt, 1987]. Although the roots of Serious Games date back to centuries ago [Wilkinson, 2016], recent advancements have made this game genre a hot topic in business and research. Currently, the Serious Games market includes military, government, corporate, educational, and healthcare games [Michael and Chen, 2005]. One example is the IBM Smartplay, which integrates real processes and real data into problem-solving games[22]. Figure 2.24 depicts two further commercial examples for Serious Games.

One of the functionalities of the game engine which can help with different kinds of spatial query is culling. Culling is the early rejection of any object which will not

[21] https://unity3d.com

[22] Source: https://www-935.ibm.com/services/us/gbs/gaming/

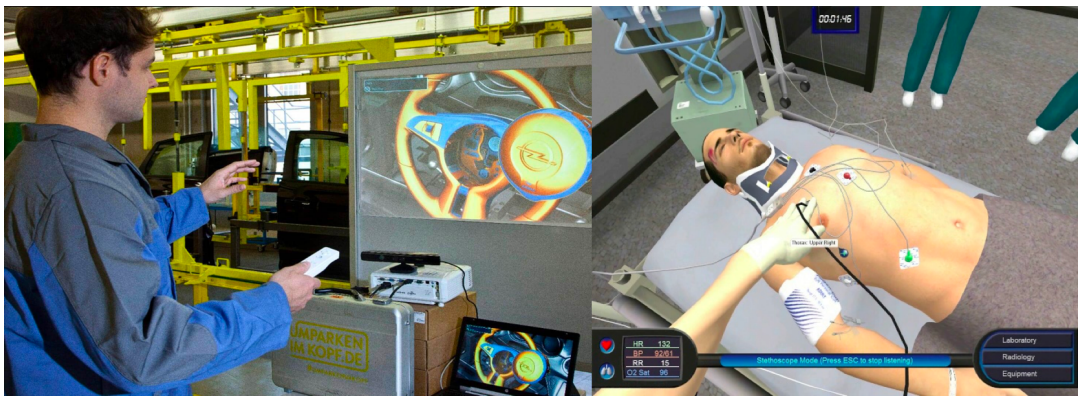

Figure 2.24: Left: Virtual assembly line training for achieving proficiency in assembling Opel vehicles. Right: vHealthcare™ is a virtual environment for supporting discovery-based learning and decision-making skills in medical environments.[23]

appear in the final rendered image. The different methods for culling are backface culling, view frustum culling, portal based culling, and occlusion culling. In the backface culling, the polygons which do not face the user are ignored. In the view frustum culling, any object which lies outside of the viewing frustum is not rendered. In portal based culling, the 3D scene is separated into cells that are joined together by portals. These portals allow objects in one cell to be viewed from a neighbor cell. One of the most important culling methods is occlusion culling. Here the objects that cannot be seen by the camera because of occlusion of other objects are not rendered. When using a model of a city or an indoor environment, culling can be used to find out which objects are visible to the user. This information is very useful for interaction design. Figure 2.25 shows two different culling methods compared with a normal scene without culling.

Another functionality of a game engine which can help with the different kinds of spatial query is ray casting. In ray casting a ray-surface intersection is used to solve different problems in computer graphics. One of these problems is to find the first object in a scene which is intersected by a ray. In many game engines this functionality is implemented through the physics engine. A physics engine is a software that simulates the certain properties of physical systems. There are two kinds of physics engines: high-precision and real-time. As the high-precision physics engine requires many resources, the game engines include the real-time class. In a dynamic scene where objects move in every possible direction at any moment and the shift of the user's gaze happens continuously, the ray casting functionality of a physics engine

[23] Source Left: http://www.seriousgames.net/home/workplace-learning-games/
Source Right: http://www.breakawaygames.com/vhealthcare/

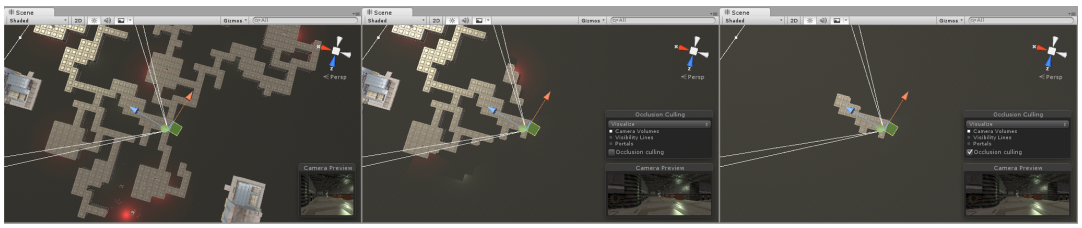

Figure 2.25: Left: Normal scene view with all game objects. Middle: Frustum culling renders objects within the cameras view. Right: Occlusion culling renders parts of the scene which is not occluded by other objects.[24]

can be very helpful to determine the object in the user's focus at any time. For this purpose, the position and orientation of the different dynamic objects in the scene can be updated in each physics/game frame, and then a ray in the direction of the user's gaze can be cast to determine the colliding object which is also the object in the visual focus of attention of the user.

In order to perform a spatial analysis, it is possible to combine the different GIS and game engine methods. For example, in the cultural heritage domain, [Merlo et al., 2012] use the Unity3D game engine to add more degrees of interactivity between users and the objects. [Sharkawi et al., 2008] combine 3D game engine with different GIS elements to develop a 3D navigation system. [Mat et al., 2016] proposes an online 3D oil palm plantation management system based on game engine. For this purpose, they want to incorporate GIS data inside a game to allow easier and more efficient management procedure. [Mat et al., 2014] provide a review of different studies which use game engine for 3D terrain visualization of GIS data. So depending on the target application and the required spatial analysis, it is possible to choose between or to combine different features from available GISs and game engines. For example, in the case of an outdoor 2D application, it is possible to integrate the footprints of the buildings in a 2D spatial data base and perform the analysis there. Similarly, it is possible to do this integration and analysis in the game engine. Regarding 2.5D or 3D outdoor applications, it is theoretically possible to use both game engine and GIS software for the analysis. However, if the environment is dynamic and the position and orientation of the objects or the user is changing permanently, it is better to use the game engine as it is better designed to handle dynamic environments. It is possible to populate the environment in the game engine with the data from the GIS software. Regarding 2D, 2.5D or 3D indoor applications, the game engine variation is more appropriate as the GIS software is designed mainly for outdoor environments.

[24] Source: https://docs.unity3d.com/Manual/OcclusionCulling.html

Regarding the 3D reconstruction methods discussed in the previous section for outdoor or indoor environments, in order to integrate the resulting data into a game engine or a GIS, there are some steps which should be taken. The most straightforward step is integrating reconstructed mesh into a game engine. Many game engines already support this type of data. The only step which should be taken is to manually mark the different objects or areas in the scene to add more context to the available 3D mesh. Regarding integration of the point cloud data in the game engine, it is possible to import them directly as a point cloud or to convert them first to 3D meshes and then use the method described previously. The problem with direct import is the performance of the game engine. As these systems are designed to work with polygons, their performance regarding visualization and analysis of the point clouds is not very high. There is also a third solution. Here it is possible to import the point cloud into a specific modeling software, and then model the environment manually based on the scanned data. Finally, one can export the manually modeled environment into an appropriate format and integrate it in the game engine. This technique should be used also for the GIS software. Depending on the target GIS software, the 3D mesh or the manually modeled data (based on the point clouds) can be transformed into appropriate formats and imported into the GIS software.

2.5 Position and Orientation Assessment

Assessment of the position and orientation of the user and objects in the 3D space play an important role in the human environment interaction. For example, in order to find out which object the user is pointing to or looking at, we have to know the position of the user, the orientation of his head/hand and also the position of the objects in the 3D environment. Regarding this specific example, there are some other video-based techniques which solve this problem without any positioning system. For this purpose, in the case of the user's visual attention, they examine the position of the user's gaze point with the bounding box of the objects in the video. This merging occurs in the world camera of a wearable eye-tracker. Although this technique provides a solution to many use-cases, it does not include the 3D information which can be used for further context-sensitive cases. Generally, assessment of the positions in the 3D space gives us a full overview on the current situation and the possibility to analyze future actions. For example, regarding human-robot collaboration, the assessment of the human and robot's position and movement orientation provides the possibility to analyze potential future collisions. This information can be used to re-navigate the robot or to alert the user appropriately.

Generally, it is possible to divide the problem of the position and orientation assessment into two categories: indoor and outdoor. The technologies in each case are

very different, however it is possibile to build hybrid systems which include both solutions. In the following, first the solutions for the assessment of the outdoor positioning/orientation are discussed and then the indoor variation is considered.

2.5.1 Outdoor Positioning

The most common technology for outdoor positioning is the Global Navigation Satellite System (GNSS). GNSS is the standard generic term for a constellation of satellites that is used for timing data and also to pinpoint the geographic location of a receiver device anywhere in the world. A geographic location is defined by its longitude, latitude, and altitude/elevation. There are several GNSSs such as Europe's Galileo, the USA's NAVSTAR Global Positioning System (GPS), Russia's GLONASS and China's BeiDou Navigation Satellite System.[25] The performance of a GNSS is assessed considering its accuracy, integrity, continuity, and availability. This performance can be improved by using additional regional satellite-based augmentation systems, such as the European Geostationary Navigation Overlay Service (EGNOS). EGNOS uses the GNSS information taken from several reference stations across Europe to measure the GNSS errors and transfer them to a to a central location where the differential correction of the data is calculated.[26] This corrected data can then improve the GNSS accuracy from several meters down to few centimeters. It is also possible to reach this level of accuracy with dual-frequency receivers.[27] This method removes ionospheric error from the position calculation. This error varies with frequency, so it impacts the GNSS signals with various signals differently. Another common technique for resolving positioning error is Multi-Constellation. In this technique the receiver gets the signals from several satellite positioning systems (GPS, GLONASS, BeiDou and Galileo) which results in accessing a larger number of satellites in the field of view. This, in turn, reduces the signal acquisition time and improves the position and time accuracy.[28]

There are also other solutions for resolving positioning error, for example dead reckoning or map matching. Dead reckoning is one of the oldest positioning concepts. It is used by sailors to extrapolate the current position from the information on how far and in what direction they have traveled from the last certain position. Now the same concept is used by leveraging the data from the internal device sensors. In the map matching technique, the raw GNSS positioning data are examined against a logical model of the real world, for example a road network. The map matching algorithm then combines these two information sources and predicts the actual position of the

[25] Source: https://www.gsa.europa.eu/european-gnss/what-gnss
[26] Source: https://www.gsa.europa.eu/egnos/what-egnos
[27] Source: http://www.gps.gov/systems/gps/performance/accuracy/
[28] Source: https://www.novatel.com/an-introduction-to-gnss

receiver. In order to determine the direction of a moving vehicle, the same model is used together with a series of position points. With these points, a direction vector is calculated on the described environment model.

One of the most useful use cases for outdoor positioning is its application in autonomous driving. There are six different levels of autonomous driving: Level 0 to Level 5. In levels 0 to 2 the application of GNSS in autonomous driving is optional. The Advance Driver Assistant Systems (ADAS) in these levels are, for example, a lane departure warning system (level 0), lane keeping assistant (level 1), and traffic jam pilot (level 2). For these systems a GNSS is not required. Level 3 autonomous driving is concerned with a highway driving assistant, level 4 is the highway pilot, and level 5 is the fully autonomous robot taxi with end-to-end navigation. For the last three levels (3-5) a very precise GNSS is crucial. While the required accuracy in the first three categories is at the street level, the accuracy in the last three categories should be at the lane level. As autonomous driving requires highly accurate position and velocity during the whole driving time and in any location with full availability, it combines the GNSS technology together with 3D dead reckoning and landmark positioning. The landmark positioning is performed using high quality maps fused with sensor data from different sources such as cameras, RADAR, and LiDAR.

2.5.2 Indoor Positioning

Similar to the satellite-based GNSS, various activities in the research and industry are aimed to design and implement a system for indoor positioning. Many kinds of applications can benefit from such a positioning system, for example locally based services in indoor environments, motion capturing or augmented reality applications. GNSS does not work properly indoors. Generally, design and implementation of an indoor positioning system is not very straightforward. [Mautz, 2012] provides a set of reasons why indoor positioning is particularly challenging:

- Walls and furniture cause signal reflection which in turn results in multipath problem.
- The line of sight is not always available.
- The density of the obstacles causes signal scattering and attenuation.
- The scene dynamic (presence of people and opening of doors) causes fast temporal changes.
- Indoor positioning demands high precision and accuracy.

Figure 2.26: Left: Visual markers can be coded into the texture of different products. The developed application can calculate the relative 3D position and orientation toward the product. Right: A Lenovo smartphone with built in depth sensor for positioning and augmented reality applications.[29]

The named reasons make it challenging for some positioning technologies to be used indoors. On the other hand, indoor environments have some other characteristics which facilitate positioning and navigation in other ways, for example good infrastructure, low movement speed of the user and fixed geometric constraints. [Mautz, 2012] divides the different approaches for the indoor positioning problem into 13 different technologies. [Schwartz, 2012] also provides a similar list of technologies. Each of these technologies is based on a measuring principle and has its own accuracy and coverage area. This makes each solution appropriate for a certain set of applications. Table 2.1 provides an overview of these technologies and their specific characteristics. In addition to the methods mentioned, in recent years depth cameras have been used more often for indoor positioning. The advantage of this technology is that it does not need any infrastructure in the environment (for example extra sensors etc.) and relative to other solutions, it is inexpensive. Moreover, its size is so small that it can be packed into a smartphone, and its energy consumption is so low that it can be operated by the smartphone's battery (see Figure 2.26). There are also systems which perform a similar positioning just with a built-in RGB camera (for example the ARKit from Apple[30]). In both cases the implemented algorithms and software which interprets the sensor data play a major role. The precision of this technique is however not very high. In order to reach a very high precision, visual markers are used (see Figure 2.26). These visual markers

[29] Source Left: https://library.vuforia.com
Source Right: http://www3.lenovo.com/us/en/
[30] Source: https://developer.apple.com/arkit/

have predefined patterns which make it possible for the software module to calculate the relative 3D position and orientation of the RGB camera to the marker with high accuracy and precision. It is very practical to combine these two techniques in order to reach a continuous and precise indoor positioning. For example, while using Microsoft HoloLens to interact with the environment, the depth camera can be used for general 3D positioning and additional markers are used for extra precision.

Technology	Accuracy	Coverage	Measuring Principle	Application
Cameras	0.1mm-dm	1-10	angle measurements from images	metrology, robot navigation
Infrared	cm-m	1-5	thermal imaging, active beacons	people detection, tracking
Tactile & Polar Systems	μm-mm	3-2000	mechanical, interferometry	automotive, metrology
Sound	cm	2-10	distances from time of arrival	hospitals, tracking
WLAN / WiFi	m	20-50	fingerprinting	pedestrian navigation, LBS
RFID	dm-m	1-50	proximity detection, fingerprinting	pedestrian navigation
Ultra-Wideband	cm-m	1-50	body reflection, time of arrival	robotics, automation
Highly Sensitive GNSS	10 m	'global'	parallel correlation, assistant GPS	location based services
Pseudolites	cm-dm	10-1000	carrier phase ranging	GNSS challenged pit mines
Other Radio Frequencies	m	10-10000	fingerprinting, proximity	person tracking
Inertial Navigation	1%	10-1000	dead reckoning	pedestrian navigation
Magnetic Systems	mm-cm	1-20	fingerprinting and ranging	hospitals, mines
Infrastructure Systems	cm-m	building	fingerprinting, capacitance	ambient assisted living

Table 2.1: An overview of indoor positioning technologies and their characteristics provided by [Mautz, 2012].

As it is mentioned before and also in Table 2.1, a camera is one of the dominating techniques for indoor positioning. [Mautz, 2012] and also [Schwartz, 2012] divide this kind of positioning into two categories: egocentric and exocentric. [Mautz, 2012]

also mentions different variations for the camera-based indoor positioning systems. One of these variations is the "Reference from 3D building models" which relies on matching the detected objects in images with position information of the building interior. Another class is "Reference from images" which relies on comparing the current view of the mobile camera with the previously captured view sequences in the same environment. "Reference from deployed coded targets" is the class of applications which (as described before) use coded markers to increase the robustness and accuracy of the reference points. A similar class is the "Reference from projected targets".

Another technology used for indoor positioning is based on infrared (IR) wavelengths. There are three methods which use infrared signals for this purpose: active beacons, infrared imaging using thermal radiation, and artificial light sources [Mautz, 2012]. The systems based on the tactile and combined polar technologies are not considered traditionally indoor positioning systems due to their high price which is not suited for mass market applications [Mautz, 2012]. Sound-based systems can be divided into two categories: ultrasound and audible sound. These systems are used for applications at cm-level accuracy, however they have only up to 10 m operating range. Another drawback is the temperature dependency. The indoor positioning systems based on WLAN/Wi-Fi technology is one of the most widespread approaches. This is due to the widespread nature of 802.11 networks and the possibility to use their RSSI (Received Signal Strength Indicators) for positioning. In the case of the RFID (Radio Frequency Identification) technology, there are two major categories: active and passive RFIS. The accuracy of this technology is highly dependent on the density of the tag deployed in the environment and the maximal reading range of the tags. Due to their hardware form, they can be installed in a way that be unobtrusive to the user. Another technology used for indoor positioning is Ultra-Wideband (UWB). A common setup here consists of a stimulus radio wave generator and receiver. As this technology needs an extra dedicated setup, it has not entered the mass market yet.

Pseudolites are land-based beacons which generate pseudo-noise codes similar to that of GNSS [Mautz, 2012]. The main purpose of these systems is to support GNSS with additional ranges where the satellite signals are not available or jammed. The "other radio frequencies" entry in Table 2.1 refer to communication technologies such as ZigBee, Bluetooth and others in the similar category. Like WLAN, these are designed for short-range wireless transfer. Generally, any radio signal, at any frequency or signal range with any protocol, can be used for indoor positioning [Mautz, 2012]. The "internal navigation systems" entry in Table 2.1 refer to dead reckoning approaches based on IMU and an Internal Navigation System (INS). Since this solution has a remarkable drift over time, it is better to combine it with other solutions. The indoor positioning systems which use magnetic localization technology

rely on an infrastructure of permanent magnets (or coils). This technique does not require a line-of-sight between the sensor and the source. This technology can have sufficient precision to be used in surgery. The "infrastructure systems" entry in Table 2.1 refer to positioning approaches which use available building infrastructure. For example, using floor tiles to detect a standing human in a 2D setup.

2.6 Role of Eye-Based Interfaces in Human-Environment Interaction

The first applications for eye-based human computer interaction were realized in the early 1980s. These systems were designed either for supporting disabled users (for instance [Levine, 1981]) or for interacting with computers in controlled environments. For example, Richard A. Bolt introduced in [Bolt, 1981] a dynamic gaze-based interface which allowed users to interact with 20 different windows. These windows were located in the same room as the user, and displayed different static or dynamic content. In addition to the eye-tracking, this system offered speech and manual input for controlling the windows. Thus, this prototype can be considered one of the first eye-based multi-modal human-environment interaction systems[31]. A decade later, Robert J.K. Jacob presented in [Jacob, 1990] a research prototype which used gaze as an input modality for desktop systems. This study compared human gaze with the computer mouse as an input device for desktops. The developed research prototype wanted to obtain information from the user's natural eye movements rather than requiring the user to make specific eye movements. One decade later, Wahlster presented in [Wahlster, 2003] a symmetric multi-modal system in which all input modes (including speech, gesture and facial expression) were also available for output, and vice versa. This way, the presented prototype not only understood the user's multi-modal input, but also responded to the user using its own multi-modal output via an embodied conversational agent. This prototype contained the full spectrum of dialogue phenomena that are associated with symmetric multi-modality, including back-channeling. In addition to backchannelling, Reithinger et al. presented in [Reithinger et al., 2006] the Virtual Human system which also included a turn-taking approach incorporating general gazing behavior as well as actions to take and yield turns. Regarding using human gaze in automotive user interfaces, Wahlster and Müller introduced in [Wahlster and Müller, 2013] a prototype which showcased using gaze instead of touch for selecting items on the main screen in vehicle. For selecting an item, the user could just look at it and push a button on steering wheel. Comparing with a touch screen, this system has many advantages. For example, the

[31]The famous Put-That-There system also by Richard A. Bolt did not include eye-tracking [Bolt, 1980].

driver's hand will remain on the wheel while selecting an item. Furthermore, his hand will not cover the information shown on the display.

Robert J.K. Jacob argued in [Jacob, 1990] that the eye-tracker as input device is far from perfect due to two main reasons: limitations of the eye-tracking device and the nature of human eye movements. Regarding the hardware, his complaints are directed at the size and accuracy of the device. With recent developments, the size of available eye-trackers are significantly reduced and their accuracy are improved as well. Bulling and Gellersen mention in [Bulling and Gellersen, 2010] that the stationary eye-trackers achieve a visual angle accuracy of about 0.5 degrees while mobile eye- trackers have a visual angle accuracy of about one degree. The reason why mobile eye-trackers are less accurate is that they are used to address varying distances between the user and the interface (or object). Thus, if a stationary eye-tracker is used for addressing interfaces with different distances, it will have the same accuracy reduction. Just like in the early days of the eye-based interactions, in recent years these systems have been used amongst others for medical purposes (for example for mental health monitoring in [Vidal et al., 2012]) and human-environment interaction. As both pervasive and wearable computing are getting more advanced, nowadays we carry several computers with us and also live in environments which consist of several computing devices. As eye-based interaction is very natural, it is very reasonable to use it for interacting with these devices and generally with the whole environment around us. As Robert J.K. Jacob tried, eyes can be used like a computer mouse as a pointer. Depending on the application, we can point with our eyes at a 2D surface or in the 3D environment. If the target is large enough, the discussed inaccuracies (from 0.5 to one degree) may be compensated. As Majaranta and Bulling state in [Majaranta and Bulling, 2014], gaze-based interfaces have a lot potential to be used either as an input method or as an information source for proactive systems. However, they also mention the problem with perception and control. If the user does not specifically control the system with other modalities (or with very specific eye movement, for example several blinks), the system needs to distinguish casual viewing from an intentional control. This problem is referred to as "Midas touch".

Majaranta and Bulling propose in [Majaranta and Bulling, 2014] a categorization of eye-tracking applications as a continuum which start from intentional and end with unintentional systems [32] (see Figure 2.27). Any gaze-based system can be positioned on a specific place of this continuum, however if the system is hybrid, it can contain features from different presented categories. At the far left of this continuum are the systems with explicit eye input. These are the applications which use eyes for command and control. They are very useful for people with physical disabilities

[32] Their categorization is based on the suggested four-group classification of Fairclough presented in [Fairclough, 2011] for describing the different kinds of physiological computing systems.

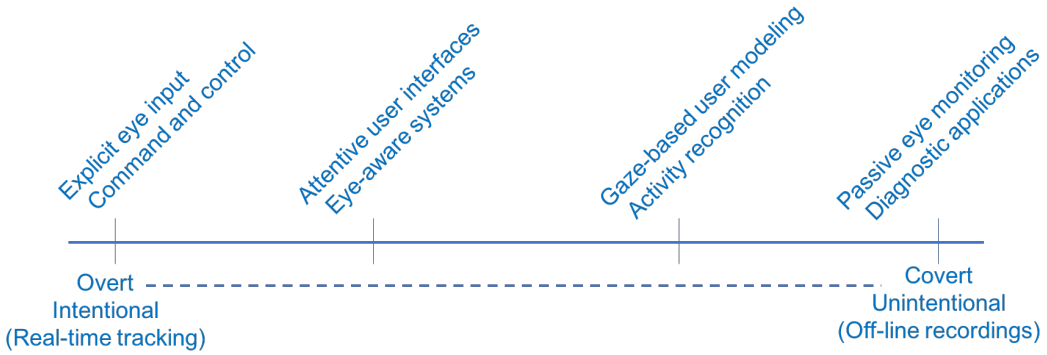

Figure 2.27: Continuum of different eye tracking applications presented by [Majaranta and Bulling, 2014], based on [Fairclough, 2011].

who mainly use their eyes to communicate with their environment. The project COGAIN (Communication by Gaze Interaction) is a European Network of Excellence that is working toward giving people with profound disabilities the opportunity to communicate and control their environment by eye gaze [Bates et al., 2007]. Currently, Tobii offers commercial eye-trackers which enable patients with disabilities to paint[33] using their eyes or as a small child talk with their family and friends[34].

Another category on the continuum of eye-tracking applications is the attentive user interfaces. In this kind of application, the user is not expected to change her or his gaze behavior in order to explicitly give commands. For example, Selker et al. present in [Selker et al., 2001] an application which analyzes the user's gaze fixations on different objects in the environment and based on these fixations, triggers a communication with the objects. The text 2.0 system presented in [Biedert et al., 2010] also belongs to this category. By analyzing the user's gaze in real time, this system aims to elevate the user's reading experience by augmenting text with automatic text translation, sound or visual effects. Another example for attentive user interfaces is the gaze-contingent displays. These displays show graphics with higher resolution on the focus area of the user and they also degrade the resolution in the periphery (to save resources).

Another category on the continuum is gaze-based user modeling. Based on the analysis of the user's gaze, these applications try to understand her intention, behavior or cognitive process. As the former two categories are concerned with where in space the user is looking, the applications in this category analyze how the user is looking [Majaranta and Bulling, 2014]. For example, Bulling et al. present in [Bulling et al., 2011] an application which uses eye data movement together with

[33] https://www.youtube.com/watch?v=rp4zHIhm0L0 (visited on 09.12.2017)
[34] https://www.youtube.com/watch?v=RJ35Zp8ZrJw (visited on 09.12.2017)

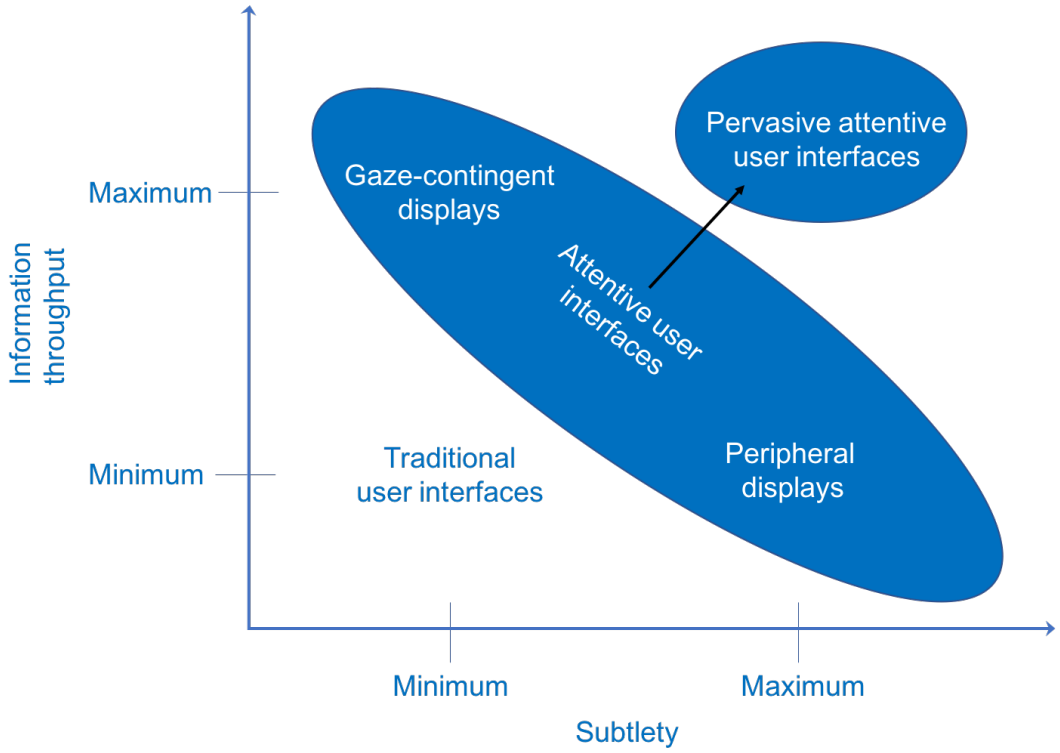

Figure 2.28: Classification of different display interfaces based on their possible information throughput and subtlety, presented by Bulling in [Bulling, 2016].

machine learning techniques to classify five common office activities (copying a text, reading a printed paper, taking handwritten notes, watching a video, and browsing the Web). In [Dietz et al., 2017], Dietz et al. present an application which uses eye and head tracking data together with machine learning techniques to make an automatic detection of visual searches for elderly people. They report a recognition rate of 97.55 percent. The last category on the right side of the continuum for eye-tracking applications is passive eye monitoring. These are mainly diagnostic applications which use recorded user's behavior for their analysis. As their processing happens offline, they do not have any effect on or immediate reaction to the user's interaction.

When we consider analyzing the user's eye for eye-based applications (eye movement, gaze, etc.), we are considering this data as an input for these systems. However, eye-based applications can have also a visual output. For this purpose, displays with different sizes and properties can be positioned in a variety of places relating to the user's line of sight. Due to technological advances in recent years, wearable displays have become more popular, affordable and useful. Seneviratne et al. provide an analysis of current available wearable displays and their pros and cons in [Seneviratne et al., 2017]. In [Bulling, 2016], Bulling categorizes the dis-

play interfaces by considering two factors: their information throughput and subtlety (see Figure 2.28). In his categorization, gaze-contingent displays have the maximum throughput, however as they require the attention of the user all the time, their subtlety is very low. Just like them, the traditional displays also have low subtlety. That is because they do not know about the current user's attentional capacity, nor do they consider the type or the amount of information which is present to the user [Bulling, 2016]. Peripheral displays on the one hand are very subtle by information delivery, on the other hand their information throughput is very low. Bulling argues that the Human Computer Interaction (HCI) community should aim for building pervasive attentive user inter- faces which are able to manage the user's attention so that the information delivery is optimized both for subtlety and throughput. As eye-based applications can manage the input and output of their interfaces based on the user's focus-of-attention, they aim for building applications with maximal information throughput and subtlety.

One of the aims of this thesis is to build a context-sensitive eye-based interface which reacts in real-time to the user's actions by analyzing the environment and also user's complete field of view, including gaze and peripheral vision. On the continuum of different eye tracking applications, this thesis can be positioned between the attentive user interfaces and explicit eye input. In the presented systems, the user should not change her behavior. However, as the user refers (with eyes) to objects in the environment, she has a sense of command and control. The presented prototypes can also be considered attentive user interfaces, as they are aware of the user's vision in real-time and react to it. Regarding the subtlety and information throughput, the presented systems try to utilize the whole bandwidth of the user's vision, both as input and output. For this purpose, they analyze the user's view field completely including the peripheral vision. They also use the whole spectrum to show information to the user. This information in turn is adapted in real-time to the user's focus-of-attention.

3 State of the Art in Eye-Based Human-Environment Interaction

The subjects discussed in this thesis cover a wide spectrum of research regarding eye-based human environment interaction. More specifically, the direct gaze interaction and also the peripheral interaction across the different realities from real environment up to mixed reality and virtual reality are considered. The literature corpus for of all of these research fields is very large. This chapter considers the studies which are highly relevant to the research presented in this thesis. In Section 3.1 the systems which use gaze for direct human-environment interaction are considered. Different aspects of each of these studies are analyzed, like for example the type of the environment, their position in the mixed-reality continuum, and also the number of users. Regarding the type of the environment, it will be analyzed whether the implemented research prototype used a 2D or a 3D model of the environment for the analysis. Furthermore, whether the scene is dynamic or not and also if the system is designed for indoor or outdoor use. Regarding the mixed-reality continuum, it will be analyzed where in the spectrum of mixed-reality the systems are implemented. Concerning the number of users, it will be analyzed how many users are involved in the implemented scenario. Section 3.2 answers the above questions for the systems in which the user's visual periphery is used for interaction. Figure 3.1 gives an overview for the studies which are mentioned in Sections 3.1 and 3.2. Section 3.3 provides a short summary and discussion.

3.1 Gaze Based Human-Environment Interaction

Interactive applications typically use eye trackers to employ gaze as a pointing modality [Duchowski, 2017]. There are also other applications which use gaze as an

indirect pointing modality, for example as an indirect pointing aid for different user interfaces or in the context of collaborative systems as a deictic reference. Each of the named direct or indirect uses of gaze for human-environment interactions should be aligned with the underlying models. If the applications are implemented in a dynamic environment, this alignment is even more important. The number of applications and their categories in the eye-tracking field are very vast. Section 2.2 described the types of eye-tracking applications which are considered in this thesis (see Figure 2.15). In his latest book on Eye Tracking Methodology Theory and Practice, Andrew T. Duchowski divides the eye tracking applications into the following categories [Duchowski, 2017]:

- Neuroscience and Psychology
- Industrial Engineering and Human Factors
- Marketing/Advertising
- Computer Science

Each of these categories is somehow related to the human visual attention. For example, in Neuroscience and Psychology, the scene perception and visual search are studied. In Marketing/Advertising, Ad placement and print advertising are studied. In the Industrial Engineering and Human Factors category, the different possibilities for using eye-tracking technology in simulators mainly for automotive and aviation fields are studied. The only category which is mostly related to the topics covered in this thesis is the computer science category. Here, interactive applications mainly developed by computer science researchers are discussed. In the following, different related work from this category is described. The examples will go beyond that which is presented in [Duchowski, 2017], as the recent studies and prototypes are not always mentioned there. The gaze-based interactive applications are divided into two categories by [Duchowski, 2017]:

- Human-Computer Interaction and Collaborative Systems
- Gaze-Contingent Displays

Section 3.2 introduces the different studies in peripheral vision in human-environment interaction. The current section covers the related work in the human-environment interaction and collaborative systems category across the mixed-reality continuum. The presented studies are selected from a wide range of research which contain to some extent an environment reconstruction module. Research such as the series of studies from Saraiji et al. and also Yanagi et al. who

mimic several aspects of the this thesis, however without environment reconstruction, are not covered ([SARAIJI et al., 2016], [Saraiji et al., 2014b], [Saraiji et al., 2014a], [Saraiji et al., 2015], [Saraiji et al., 2016], [Yanagi et al., 2015]). Another example here is the study of Stellmach and Dachselt or van Rheden et al. which do not include a 2D or 3D environment reconstruction ([Stellmach and Dachselt, 2013] and [van Rheden et al., 2017] respectively).

3.1.1 Gaze Based Human-Vehicle Interaction

There are several studies on analysis of gaze based interaction for the automotive context. However, many of these systems perform these analyses based on a display which shows a driving simulator and a driver who is sitting in front of the display and participates in the study. For example, Biswas introduces a series of such studies in [Biswas, 2016]. The focus in this section is, however, research prototypes which use a vehicle or build a similar environment to implement a new form of gaze-based human vehicle interaction. The user studies which perform pure analysis based on the display virtual simulator are not considered.

Dobbelstein et al. present in [Dobbelstein et al., 2016] a gaze-based interaction system that maps a single rotary control knob to multiple interfaces in a car. They use the driver's gaze for determining the interaction context. With this multimodal approach they want to reduce the complexity of the physical interface in a car to a simple, single physical control interface. As for the environment of their implemented prototype, they use a driving simulator setup with two Tobii Rex eye-trackers. These eye-trackers are mounted on the left and right side of the steering wheel to cover different regions of the instrument panel. They assess the user's gaze in different regions of the physical simulator setup for static objects and in a simulator setup which is located indoors, their system can be considered a 3D static indoor prototype. Although they use a virtual reality simulator, as they evaluate the focus-of-attention (gaze) of a single user setup in real environment, their prototype can be considered as a gaze-based single-user interactive system in a local real environment.

Tawari et al. introduce in their publications ([Tawari et al., 2014b] and [Tawari et al., 2014a]) different systems for the driver's attention estimation and its different applications. They use a distributed camera framework to estimate both the driver's head and gaze direction. With the proposed system it is possible to determine the driver's focus-of-attention in several different zones in the vehicle. They define six different zones, which include left and right mirror, back mirror, middle console, cockpit, and the straightforward windshield. They report much better results when using head pose together with eye gaze, rather than head pose alone. When using both modalities, the accuracy of their system varies between

79.2% and 98.2%. Tawari et al. also use a similar setup together with a Google Glass (worn by the user) to map the user's focus-of-attention to the pedestrians outside of the car. In none of the proposed systems is it possible to refer to the outside environment from within a moving vehicle, thus no outside-environment modeling is performed. As they test their research prototype in a real environment, their system can be considered a dynamic/static indoor prototype with high speed. The system is implemented for a single user.

Similar approaches like that of Tawari et al. are followed by two other research groups to determine the driver's focus regarding the defined 6 zones in the car. In [Vora et al., 2017] Vora et al. generalize the gaze estimation method using convolutional neural networks (CNN). Their research prototype is invariant to different subjects. The resulting accuracies of their system vary between 87.58% and 99.51%. In [Vasli et al., 2016] Vasli et al. also follow a similar approach for generalizing the gaze estimation on the named 6 zones. However, instead of CNN, they exploit the geometrical constraints in the car and propose a geometric based gaze estimation method and geometric plus learning based hybrid gaze estimation framework. The accuracy of their system varies between 89.29% and 100%.

Fletcher et al. introduced a system in a series of publications ([Fletcher et al., 2003], [Fletcher et al., 2005], [Fletcher and Zelinsky, 2008] and in [Fletcher and Zelinsky, 2009]) which can be considered one of the pioneer systems regarding driver-environment monitoring. Their research prototype fuses contextual (outdoor) environment information together with the driver's focus-of-attention to enrich safety features of the vehicle in real traffic. For the environment reconstruction, they perform a lane tracking and also obstacle detection. This tracking is conducted by using a stereo camera. Then from the camera images, a 3D surface together with a 3D depth flow will be calculated. This information is very helpful for e.g. object recognition. Finally, this information will be combined with the gaze information of the driver in a common 3D environment. The developed research prototype will then provide an immediate feedback based on this information, when for example a traffic sign is missed by the driver or a when the driver is distracted. This research prototype can be categorized as a 3D dynamic approach for (high speed) outdoor real environments. For the analysis, Fletcher et al. consider the gaze of the driver (single local user).

Kang et al. introduce in [Kang et al., 2015] a system which links the tracked gaze of the driver with annotated maps and the vehicle's position to provide the driver with information about the environment outside of the vehicle. Their approach is multimodal. The driver asks the car about the buildings and their system analyses the head pose and the eye gaze of the driver during his speech time. Based on this information and by using a custom algorithm, their system queries an annotated database to acquire the target building. For head pose and gaze estimation they use Microsoft

Kinect 1 together with their own gaze tracking algorithm. They report an accuracy rate of up to 69.1% when the user was allowed to move his head while referring to the outside buildings. They have modeled the environment in 2D and performed the tests in real traffic with speeds up to 30 KM/h. Hence, their system can be considered as a 2D dynamic system for (high speed) outdoors. The system is designed for one person and the utilized modalities are head pose, eye gaze, and user's speech.

3.1.2 Outdoor Human-Environment Interaction

Gaze-based human environment interaction outdoors is not restricted to the automotive applications. Several research groups have developed different prototypes for various kinds of interactions. For example, Baldauf el al. introduce in [Baldauf et al., 2010] a wearable system (KIBITZER) which enables the users to browse urban surroundings for annotated digital information. The main input modality of the system is the user's eye gaze. The user can trigger the system by closing his eyes for two seconds in order to acquire information about a specific object. When the user selects a building, the implemented prototype provides speech and non-speech auditory feedback. KIBITZER contains three main hardware components: a laptop computer, an Android Smartphone, and a mobile eye-tracker. The utilized eye tracker uses two cameras; one records the eye movement of the user and the other records the scene from the user's perspective. The Smartphone is mounted on the top of a helmet which is worn by the user. This Smartphone contains built-in GPS, an accelerometer, and compass sensors which will be utilized to acquire geographical coordinates of the user and the tilt and orientation of his head, respectively.

In [Giannopoulos, 2016], Giannopoulos introduces a gaze-based pedestrian navigation assistance approach called GazeNav. This system utilizes the user's gaze to inform him about taking turn onto the street he is looking at. This prototype also contains a vibro-tactile feedback as an interaction method with the user. GazeNav is compared with map-based turn-by-turn instructions in a user study with 32 participants. This user study is performed in virtual environments using Esri City Engine for the 3D environment reconstruction and Unity 3D for the implementation. The outdoor scene is projected on a wall: this scene also contains visual markers for the wearable which is used by the wearable eye-tracker for positioning purposes. The user navigates through the environment via a joystick.

Anagnostopoulos et al. describe in [Anagnostopoulos and Kiefer, 2016] a classification of gaze-based interaction based on different types of user and object movement. One of these categories is the location-constrained gaze-based interaction. In this case the user is standing, the object is stationary, and the interaction can occur only at certain pre-defined locations. For this category they have also implemented a re-

search prototype which allows users to refer to buildings in the environment by just looking at them (see [Anagnostopoulos et al., 2017]). Their prototype first uses position of the user for geo-fencing. When the user reaches a certain position, the location based system will be activated. Here, a reference image serves as a base for an environment model. In this image several points of interest are marked using bounding boxes. The implemented prototype uses these data together with the data coming from the eye-tracker (world camera and user's gaze) to estimate the building which is in the user's focus. Finally, this information is sent to the interaction module.

3.1.3 Indoor Human-Environment Interaction

Orlosky et al. introduce in [Orlosky et al., 2015b] a hardware and software framework which aims to improve the user experience in video see-through augmented reality setups. Their hardware setup includes several modular components: a binocular eye-tracker, several detachable camera modules with different lens types, and an Oculus Rift VR display. The software is composed of an interaction detection algorithm, a game engine for integrating virtual elements, an eye-tracking component and several computer vision algorithms. The interaction classifier detects squints, double blinks and long closes. Depending on these inputs, several zoom levels are implemented: binary zoom, gradual zoom, and sub-regional zoom. In addition, several other custom views are also implemented. Orlosky also introduces in [Orlosky, 2016] an indoor localization framework using OCR. For this purpose, a 2D area map is annotated. The hardware setup includes an eye-tracker with inward and world camera together with a head-mounted AR display. Orlosky et al. also introduce in [Orlosky et al., 2015a] Halo content. This research prototype uses different techniques for augmenting the user's peripheral vision with information regarding the people that user sees. The algorithms displace the information so that they do not occlude the target object or other important information in the scene. The peripheral vision of the user itself is however not considered in much detail. Preventing the virtual information and the real objects are more in the focus.

Kai Essig et al. present, in two different studies, applications for human environment interaction based on user's gaze. In [Essig et al., 2012] they present a system which combines mobile eye tracking data with motion capture data. Here they calculate and visualize the 3D gaze vector of the user together with the motion capture information in a common co-ordinate system. Furthermore, they tag different objects in the environment with reflective markers to track their 3D position in the common coordinate system at any given point in time. As a result, they can compute the fixations data regarding the different (moving) objects in the environment without manually evaluating the data. In [Essig et al., 2016], they combine different technologies such as Augmented Reality, eye tracking, object recognition and semantic analysis of a

user's mental representation. With this information fusion they aim to react in a context-sensitive way to human error and build a system which is able to provide individualized feedback on a transparent virtual plane which is superimposed on the user's field of view. Their focus group is elderly and cognitively impaired users and the use cases are the assembly of a birdhouse from wooden pieces and also usage of a modern coffee machine.

Khamis et al. present in several publications ([Khamis et al., 2015] and [Khamis et al., 2016]) thoughts on using eye gaze as a modality for interacting with public displays. They describe several benefits and challenges of such an interaction and also how it is possible to address these issues. In [Khamis et al., 2017] they present the EyeScout system for user interaction with public displays. In order to address the different limitations of eye tracking systems, they mount this module on a rail system. They also introduce a computational method to automatically detect and align the tracker with the user's movement. They introduce several concepts which enable the user to walk to an arbitrary position and then interact, or interact while on the move. For this purpose, they track user's body first, and when the body is in range they send the eye-tracker to the appropriate position on the rail and try to detect the user's gaze. While on the move, this procedure happens several times successively.

Gupta et al. present in [Gupta et al., 2016] a system for sharing gaze and also pointing cues between two users in different physical locations. The local user wears a head-mounted setup which consists of an eye-tracker, a world camera and an augmented reality display. The remote user sees the video stream of the local user's world camera and can mark different locations in this video. These positions are then transported and marked to the local user's augmented reality display. This way the two users can collaborate by exchanging their focus at any time. The local user sends his focus-of-attention via the eye-tracking setup, and the remote user sends his pointing cues by moving the mouse on the live video stream of the local user. The used this setup for building a sample LEGO house by the local user. The remote person then helped the local user with different cues. This research prototype is a good example of multimodal gaze-based remote collaboration. In their several experiments, they achieved best results on performance when using both modalities together.

Qodseya et al. present in [Qodseya et al., 2016] a system which builds in real time the 3D reconstruction of an indoor scene which also includes the user's focus-of-attention on the different points of regard. For this purpose, they have built a custom wearable hardware together with several software components. The hardware consists of a stereo camera rig with two cameras which are synchronized via an Arduino module. These cameras act as world cameras and their images are used for the environment reconstruction. Moreover, their wearable device consists of an IR LED, an eye camera, and an IR LED for monitoring the user's eye movements. Their various

software modules are the data acquisition component, the pupil detection module, the depth computation and also the environment reconstruction component. Their world camera runs at 25 fps, and thus it allows the user to move partially in the scene and the system works in a somehow real-time mode, however head movements should be smooth. Although the authors consider the module to function also outdoors, the environment reconstruction software is not appropriate for big objects such as buildings.

Mokatren et al. present in [Mokatren et al., 2017] a research prototype which integrates a mobile eye-tracker into an audio guide system for museum visitors. For this purpose, they use a wearable eye-tracker as a positioning tool and also for detection of the user's focus-of-attention. They use computer vision techniques for both these modules. They evaluate the image coming from the eye-tracker against an image database to find out about the position of the user, moreover they use the annotated regions in the reference image and the fixation points in the eye-tracker image to identify the object of interest for the user. When the target object is identified, a broadcaster fetches a specific audio for that object from the audio database. The user also has the possibility to interact with the system via hand gestures. The research prototype is implemented in two versions: reactive and proactive.

Renner et al. have published a series of studies regarding the gaze-based human environment interaction. In [Renner et al., 2011] they introduce an unobtrusive calibration procedure that blends in virtual environment in the user's field of view, and while the user follows the object, the calibration is performed. This way they ensure a continuity in the accuracy of their gaze-based interaction system. In [Renner et al., 2014] and [Renner and Pfeiffer, 2014] they combine the gaze and hand (pointing) gestures together with head position of the user in a route planning scenario for human-robot interaction. For this purpose, they perform an automatic marker tracking and 3D modeling of target objects in virtual environments. The world camera of a wearable eye-tracker is used to map the user's gaze to the target objects in a reconstructed 3D world. For the hand gesture tracking, they use a separate tracking device. In [Renner and Pfeiffer, 2017b] and [Renner and Pfeiffer, 2017a] they evaluated various visual attention guidance techniques for optical see-through devices. For this purpose, they use a virtual reality setup as a simulator for the augmented reality use cases. In the first study, they do not use any eye-tracking device, whereas in the latter study, they use a VR eye-tracking setup. There, they also integrate eye gaze information in the guidance techniques. One of their objectives is to study ways to display cues for attention guidance that can be followed by the user in his peripheral vision. They perform a peripheral flickering if the user looks somewhere close to the target. In this case, they also change the speed for the spherical waves which are artificially generated in the whole field of view of the user's AR display.

Mardanbegi and Hansen present in [Mardanbegi and Hansen, 2011] a mobile gaze-

based interaction system which enables users to control several devices in the environment. For this purpose, they use the different screens in the environment. As these screens are seen by the world camera of the eye-tracker, a signal is sent to the server. The server then shows a code (QR-like) on the screen which is in turn identified by the mobile system. The user is then localized relative to the pre-defined locations in the environment. Now a transformation is calculated from the gaze point of the user in the eye-tracker space to the image space of the screen. This image shows different control elements which can be chosen by the user (via blinking) to control several electronic devices in the environment.

In [Lee et al., 2016], Lee et al. present a system for remote collaboration using different technologies together. The whole system consists of a local and a remote part. The local worker wears a setup which includes eye-tracking and augmented reality components as well as two other components for monitoring the user's heart rate and facial expressions. The remote user has a desktop interface where the different information from the local user is displayed (camera view, visual focus point, heart rate, and facial expression). The remote user has the possibility to provide pointing feedback back to the local user's augmented reality display. The remote interface also captures the facial expression of the remote user and shares it with the local worker. The AR display of the local user shows a live feed of the remote user's desktop interface. This way the two workers can collaborate using different reference modalities (local user: gaze, remote user: mouse pointing). Although no environment modeling is involved in this prototype, the two workers can collaborate and solve problems using different objects in the environment.

Toyama et al. present various gaze-based interaction techniques in a series of publications. In [Toyama et al., 2014], they present a real-time translation system. This system uses users' eye gaze as an indicator to the regions of interest in text documents and activates optical-character recognition (OCR) and translation functions which will be then viewed in a see-through head-mounted display. In [Toyama et al., 2015], they present a gaze-based system which provides the user with proactive assist functions based on the user's attention engagement and cognitive state. Moreover, they control the HMD display by calculating the focus depth of the user. In [Toyama and Sonntag, 2015], they present a system which uses an eye-tracker to recognize everyday objects, faces and text that the user looks at. The system then supports users' memory by logging certain types of everyday information. This system was specially developed to help dementia patients. In [Prange et al., 2015], they also develop a system which uses human gaze and robot pointing gestures to extend a human-robot dialog in order to support dementia patients. In addition to the prototypes mentioned above, in [Toyama, 2015], Toyama presents different image analysis methods to recognize the user-attended visual content. This information is combined with eye gaze analysis in an application called Museum Guide 2.0 to rec-

ognize user-attended objects in a museum scenario (similar to the system of Mokatren et al. presented in [Mokatren et al., 2017]). When an object is recognized, the system presents corresponding AR meta-information on the user's wearable device.

Piumsomboon et al. present in [Piumsomboon et al., 2017b] the concept of empathic mixed reality. They investigate how mixed reality can be applied to creating Empathic Computing experiences. For this purpose, they have built a research prototype which enables sharing gaze for remote collaboration between Augmented Reality and Virtual Reality environments. In addition to the system presented by Lee et al. in [Lee et al., 2016], here they present also the CoVAR research prototype. This system includes remote collaboration via head pose, eye gaze, and field-of-view in both augmented reality and virtual reality. For the environment reconstruction, they use the Microsoft HoloLens to create a 3D model of the local environment. This model is then integrated in the VR. The local and remote collaborators see each other's avatar and gestures in the augmented or virtual reality. In the reconstructed environment, they do not mention whether they are tracking the moveable objects. Hence, their scene is static. Piumsomboon et al. also present in [Piumsomboon et al., 2017a] three different eye-gaze interaction techniques in virtual reality. However, none of these interaction techniques considers the user's peripheral vision.

Pfeiffer and Renner present in [Pfeiffer and Renner, 2014] EyeSee3D, a research prototype for analyzing mobile eye tracking data in real-time. For this purpose, they perform a geometric modeling of the scene with markers and calculate the 3D gaze of the user and his fixations in 3D. In addition, these markers can be attached to the different objects in the scene. This way they can be moved freely and they will be tracked at the same time with the help of these markers. In [Pfeiffer et al., 2016], Pfeiffer et al. present the second version of their system: EyeSee3D 2.0. In this prototype, in addition to gaze and movement of the objects, they also track the body parts of the user (for example head and hands). For this purpose, they have integrated various body tracking techniques from different vendors. It is also possible for multiple users to work with the system simultaneously. All of the data is integrated in a common 3D model. EyeSee3D and EyeSee3D 2.0 ease the process of analyzing eye tracking and interaction data. Here, the analysis can be performed automatically and no manual video evaluation is necessary.

3.2 Peripheral Vision in Human-Environment Interaction

In this section various studies which consider the peripheral vision of the user for their interaction design are considered. This thesis presents a peripheral view analy-

sis, in which the objects in the scene are tracked and depending on their position, the proposed system calculates how big they appear to the user. For this purpose, first the position of the user in the environment is tracked, then his head orientation and also eye gaze. Moreover, the 3D environment is also modeled, and the moving objects are tracked in real-time. This thesis proposes an interaction concept by considering the position of the target object in the user's peripheral view as well as the size of object that appears to the user. In the following, several studies are presented which consider using the user's peripheral vision for the human-environment interaction. Each study is analyzed to see whether they contain the different aspects mentioned above (environment modeling, object tracking, calculating the size of object which appears to the user, etc.). Moreover, where in the mixed-reality continuum the system is implemented.

Tönnis and Klinker present in two studies ([Tönnis and Klinker, 2014] and [Tönnis and Klinker, 2014]) placing the information in the augmented reality by some angular degree relative to the user's line of sight. In the presented system they do not perform any environment reconstruction or active object tracking. Hence, the size of any target object in the environment is not considered as an input to their module. Their main objective is to study the concept of putting information at an angular offset to the user's gaze in augmented reality. They also present different studies in this regard. In their setup, they monitor the user's eye gaze and also head pose.

In [Ishiguro and Rekimoto, 2011] Ishiguro and Rekimoto propose a gaze-operated information presentation method for mobile augmented reality systems utilizing the visual field model of Komatsubara. The gaze direction is used to control the level of detail of overlaid information located in the peripheral area of a user's field of view. In this context, the main focus lies in preventing user distractions by shifting annotation information to peripheral regions with lower visual capabilities. Although they do not perform any environment reconstruction, they do a virtual object monitoring. Based on the user interaction, for example when he looks at the icon in the periphery, the displayed information is changed from simple to detailed information.

In [Langlois, 2013] Langlois presents a system which creates light signals in the peripheral vision of the driver while driving. For this purpose, a box which is illuminated by LEDs reflects light into the windshield. The color, surface and the movement of the signals are changed depending on the urgency of the situation. The system has been evaluated in a driving simulator setup. The presented research prototype does not perform gaze or head tracking. No kind of environment reconstruction or (target) object recognition is implemented either.

In [Mauderer, 2017] Mauderer presents several experiments which investigate how gaze contingent displays can be used for supporting different aspects of depth and

color perception. For these studies, Mauderer performs gaze and head tracking, however an environment reconstruction or any form of human-environment interaction (in virtual or augmented reality) is not included.

In [Reddy, 1997] and [Reddy, 2001] Reddy introduces a model for predicting the perception of a virtual stimuli based on it's location in the user's field of view. With this model he wants to reduce the lags in virtual reality systems by degrading the LOD of objects which are located in the peripheral view field. He uses the results of a study by Rovamo and Virsu ([Rovamo and Virsu, 1979]) which shows that visual acuity can be predicted for any eccentricity by applying the cortical magnification factor.

3.3 Summary

Figure 3.1 provides an overview of the important features for the research questions addressed in this thesis and to what extent they are covered in the previous work by the mentioned research groups. These features are listed in the first three rows of the table. They are divided into four main categories: Environment, Mixed-Reality Continuum, Focus-of-Attention, and Users. Each of these categories and their corresponding sub categories are described in the following.

Environment

- Modeling: Is there any 2D or 3D modeling involved?
- Scene: Is the scene static or dynamic?
- Type: Is the prototype constructed for indoor or outdoor scenarios? In case of outdoor is it possible to move with high speeds in the scene?

Mixed-Reality Continuum

- Which of the three listed realities does the prototype cover?

Focus-of-Attention

- Which of the indicators of focus-of-attention is used in the prototype?
- Does the prototype consider the position of the objects in the peripheral field of view (peripheral)?

- Does the prototype consider the size of the object in the peripheral field of view?

User

- Number of users: How many users can interact with the system (or each other) at the same time?
- Location of users: Should the user be located locally or is it possible to perform interaction from a remote place?

In order to answer the questions mentioned above, Figure 3.1 uses three color codes depending on the implementation of each prototype. Red means that the feature is not implemented, orange means that the feature is partially implemented, and green means that the feature is fully implemented. The various research groups are listed in the first column. The row in front of each research group indicates to what extent they have covered each feature. In order to address the research questions in the automotive field, user interfaces and human robot collaborations, the listed features are essential. However, some of the features are more important for automotive use cases and others are more important for the use cases in human-robot collaboration. Regarding automotive use cases, the scene modeling, its outdoor and high speed capability are very important. In the case of human-robot collaboration, using the full spectrum of the mixed-reality continuum and providing features like remote intervention and multi user capability are important. In order to use the additional output channel in the peripheral view field, it is important to know the size of the object and also its position/mapping in the peripheral view field. In the following, a short analysis of the presented related work regarding each of these subjects is provided.

Just two research groups have addressed the topic of eye-based interfaces in vehicles in conjunction with the outside environment. Fletcher and Zelinsky use the direction of eye gaze together with the position of the road signs to infer whether the user has noticed the road sign or not. They do not perform any kind of interaction or 3D modeling of the outside environment. Kang et al. introduced such an interaction with buildings in an outside environment, however their environment modeling was just two dimensional. This way, the user does not have the opportunity to refer to a big building which is located behind a smaller building. Considering the human robot collaboration use case, the is no system which covers all the three realities in the mixed-reality continuum together with a dynamic scene and full features for the users (multiple and remote users). Piumsomboon et al. presents a very similar approach, however in the reconstructed environment, they do not mention whether they are tracking the movable objects. For the human robot collaboration scenarios, it is very important to keep track of the moving objects and robots. Regarding the additional output channel in the peripheral view field, none of the reviewed studies

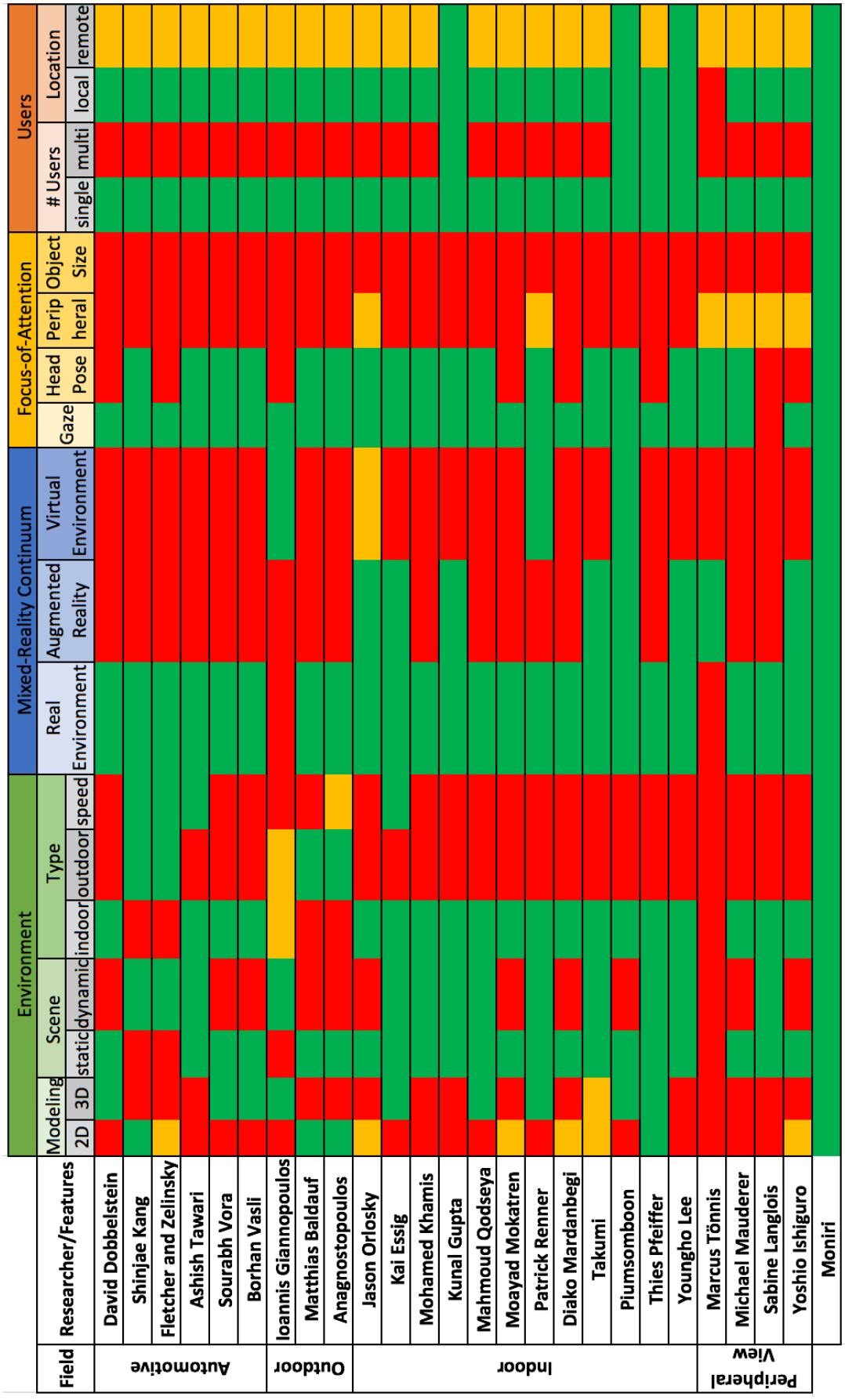

Figure 3.1: Comparison between the different related work and the presented thesis.

consider the size of objects in the periphery. With respect to the position/projection of the objects, there are some attempts in these directions (for example, Anagnostopoulos, Jason Orlosky, Marcus Tönnis, Michael Mauderer, Sabine Langlois, and Yoshio Ishiguro). However, none of these studies calculate where exactly the object is located in the peripheral view field in terms of vertical or horizontal eccentricity. As this information is not available, their interface will only react roughly to the indicators or objects in the periphery of the user.

4 3D GAZE AND PERIPHERAL VISION ANALYSIS

The interactions that are designed and implemented in this thesis mainly act on the basis of user's 3D gaze and peripheral vision. Thus, it is very essential to study the limits of these two aspects. Furthermore, as the user looks around or moves, the environment around her changes in her perspective. This changes affect the relative position and orientation of the objects as well as the solid angle of them. For the analysis purposes, in this chapter following two research questions are addressed[1]:

- **What are the limits for visual reference resolution when using gaze to refer to objects in an inside or outside environment?**
- **How can general peripheral vision be modeled for gaze-aware intelligent user interfaces?**

Section 4.1 studies the limits of reference resolution when using gaze to refer to objects. Section 4.2 provides detail on building a model for user's peripheral view analysis.

4.1 3D Gaze Analysis

The algorithms described in Chapter 5 perform a kind of ray-casting in the environment based on the user's gaze. The precision of the user's gaze is very important for the accurate result of the algorithms and consequently the accurate output of the whole system. The prototypes which were built based on these algorithms used off-the-shelf eye trackers. This section describes the different experiments which were

[1]Funded by the German Federal Ministry of Education and Research in the projects MADMACS (grant number 01IW14003) and SC_MeMo (grant number 01IS12050).

performed in order to determine the precision of the used eye-tracker in different scenarios. Generally it is possible to divide the scenarios into two categories: indoor and outdoor scenarios. The outdoor scenarios were concentrated on the automotive applications, the indoor scenarios more on human-robot interaction applications. In order to test the precision of the eye tracker it proceeds as follows. First, the precision of the same eye tracker is determined indoors for a 2D surface and also for a 3D environment. Then a similar experiment is conducted outdoors in a vehicle in actual traffic. Finally, the results of the eye tracking experiment are analyzed together with the characteristics of the different regions of the human peripheral vision. In the following, each of these analyses is presented in a separate section.

4.1.1 Experiment Setups

The experiment setup consists of two parts: an indoor and an outdoor part. In both cases the real environment is reconstructed in a 3D virtual setting. This 3D reconstruction is performed in each case with special scanners, so that a precise model of the environment is acquired. The eye tracking data is then evaluated in this environment. This way, the precision of the approach in the 3D space was ensured. In the following, each of these setups and subsequent analyses are described in detail.

4.1.1.1 Indoor Eye Tracking Data Analysis

The setup for indoor eye tracking data analysis consisted of three tables placed behind each other. Above each table a number of markers were placed at different horizontal and vertical distances (see Figure 4.1). This setup was used for the 3D experiment. For the 2D experiment, a display was positioned in front of the table. On this display the eye tracker was fixed on a printed pole. The participant was seated in front of the first table in both setups (see Figure 4.1). The whole environment was scanned (similar to the methods described before) for the precise 3D analysis. The eye tracker used for the experiment was an EyeX controller from Tobii. There were 22 participants involved with the experiment, 16 without optical aids, 3 using glasses, and 3 using contact lenses. Two measurement sets, one from a participant without optical aids and one from a user of glasses, were discarded due to the eye tracker being unable to detect their eyes at all times, meaning that the final set of measurements consists of recordings from 20 users. For each user there were about 5000 to 7000 individual data points collected. The experiment consisted of different test series combining several calibration levels (including no calibration) with 2D and 3D analyses. When the outliers where excluded, the accuracy range for the 3D eye tracking with free head movement and without calibration was between 0.5 and 5°. The median accuracy for this combination was 1.98°. The vertical and hor-

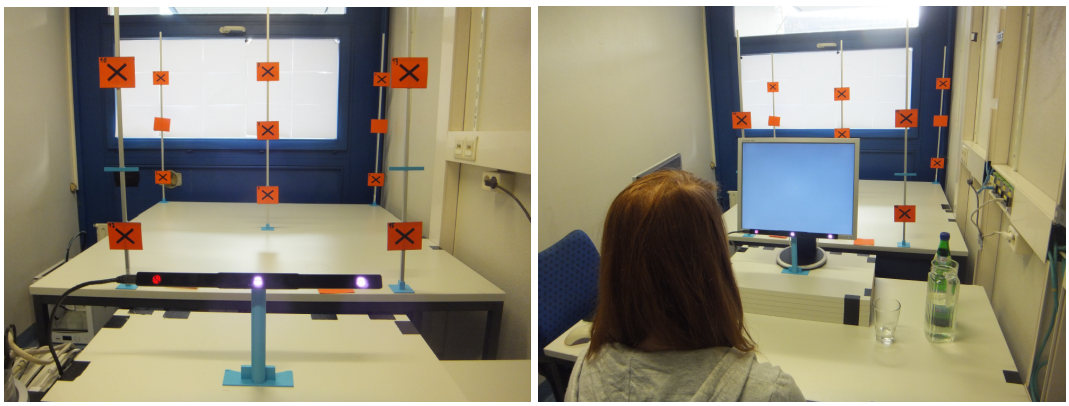

Figure 4.1: Setup for indoor eye tracking data analysis. Left: 3D analysis in room. Right: 2D analysis on display.

izontal median accuracies where 1.06 and 0.2°, respectively. For comparison, the median accuracy for the 2D case with free head movement and 9-point calibration was 0.48°. As in this research eye tracking for the outdoor applications (for example in vehicle) are interesting, a 3D analysis without a former calibration and with a free head movement was also considered. This was due to the fact that for now this is the practicable combination for easy eye tracker installation in vehicles. In the tests inside the vehicle, the user performed a 3D interaction without a former calibration including free head movement.

4.1.1.2 Vehicle Eye Tracking Data Analysis

This analysis is aimed to determine the accuracy of the system when referring to small objects in the outside environment. This analysis is performed in real traffic in a fully functional vehicle while maintaining safety measures. This accuracy analysis is aimed to determine the horizontal and vertical measurement errors of the system when referring to an object outside of the car. The analysis does not include the raw or the map-matched position information of the vehicle. Instead, the exact position of the car is entered manually to avoid propagating any positioning measurement error in the calculations. Regarding eye gaze, the presented information is the mean of the results from both the right eye and the left eye.

In this test, the driver looked at five different objects on the campus. The car was standing still in the middle of the street. The five objects were two traffic signs, two advertisement billboards, and one garbage can. Figure 4.3 depicts some of these objects. While the driver was looking at the center point of the objects, the data from the eye trackers and head trackers was logged. For each of these five objects, 60 measurements were logged. After the experiment, the logged data was checked against

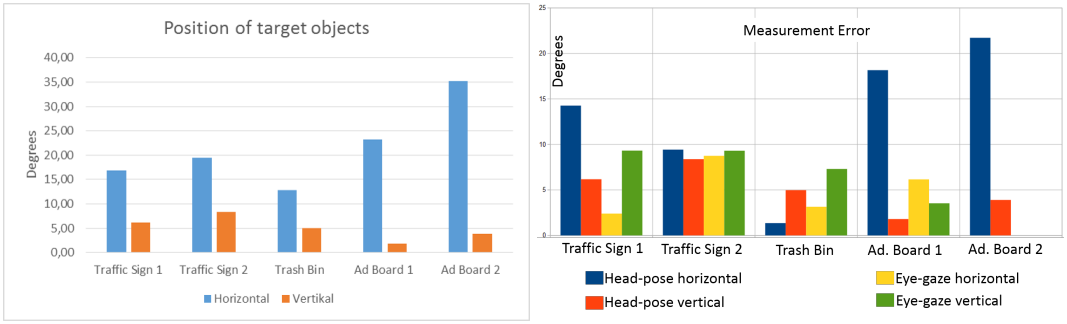

Figure 4.2: Left: The position of the target objects relative to the driver in horizontal and vertical degrees. Right: The measurement errors of eye gaze and head pose for each target object.

the reference data. The reference data was calculated by manually defining a reference ray from the driver's head in the model to the center of the target object. The horizontal and the vertical differences between these vectors were then calculated in degrees. Figure 4.2 shows the position of each target object towards the driver in vertical and horizontal degrees. It also depicts the measurement error of the system when referring to each of these objects by eye gaze or head pose.

As can be seen in Figure 4.2, there is no data for the eye tracker when the driver has referred to billboard 2. The reason becomes clear when we look at the position of this billboard. It is located horizontally at an angle more than 35° relative to the driver. This position is outside of the opening angle of the eye tracker (35° horizontally on each side). The measurement error of the head pose is also very high (more than 22° horizontally) for this object. For this modality, it seems that one can observe lower measurement errors as the horizontal distance of the target objects towards the driver becomes shorter. Besides billboard 2, the horizontal and vertical errors for all other objects for the eye gaze modality are always less than 10° and in some cases even as low as 2 to 3°. For this modality, there seems to be no relation between the position of the target object and the amount of the measurement error. Considering measurement error of the eye gaze and head pose in this analysis, it can be concluded that with the obtained accuracy it is possible to refer to big urban objects (like buildings, etc.). However, the resolution is not sufficient to refer to smaller object like traffic signs.

4.1.2 Effect of Eye Tracking Imprecision on Reference Resolution for Small Urban Objects

As it is described in the previous section, interaction with big urban objects is possible using the off-the-shelf eye trackers in the vehicle cockpit. In the tests on the

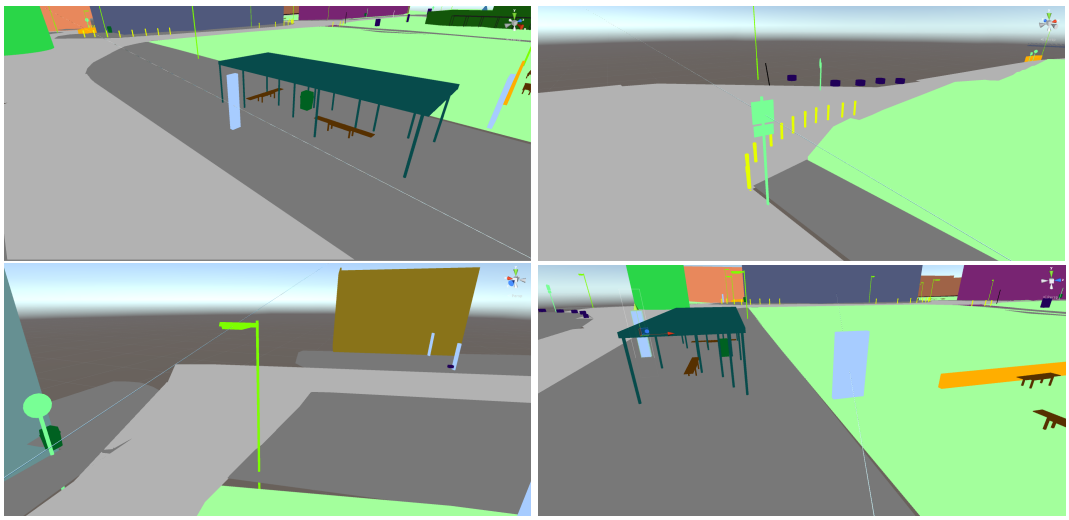

Figure 4.3: Examples of information board (top left), traffic sign (top right), street lamp (bottom left), and bus plan (bottom right). The line shows the baseline gaze data.

Saarland University campus, the precision of the developed reference resolution algorithm to determine the target building was more than 90%. However, the 3D eye tracking inside the vehicle (with the off-the-shelf components) has high imprecision and therefore cannot be used to refer to small objects in the outside environment. This section aims to measure this imprecision relative to the projection of the objects layout on the different regions of human peripheral vision. This information is valuable for developing infotainment or safety applications which need to be aware of the objects located in each peripheral region of the driver (pedestrians, tragic signs, etc.). If this shift is big, such applications cannot be implemented with the described setup. The amount of this shift also depends on the size of the object as well as the distance of the observer to the object. This explains why the presented eye tracking applications preformed well for the big objects in the environment despite the measurement errors.

For the tests, the normal distance[2] to each of the listed objects was always considered. The size of the objects is also clear by category. As a base for the analysis the Hatada model of peripheral vision ([Hatada et al., 1980]) is considered. In the following, the results of the analysis are presented.

[2]The object is visible and not too far or too close.

4.1.2.1 Shift of Object Position on Peripheral Vision

As described, when using an off-the-shelf eye tracker, performing a 3D eye tracking without calibration in a real traffic situation includes several degrees of measurement error. In the presented test scenario, the layouts of several objects in the environment are projected on the different regions of human peripheral vision. For this purpose, a baseline is used. The baseline here is the user's exact eye gaze toward the object. The measured sensor data is the measured gaze from the eye tracker. Because of the measurement error of the eye tracker, the place of the projected object on the peripheral vision differs from the baseline. Table 4.1 shows this shift for different objects.

Object	Type of Data	Object Occupation in Different Peripheral View Regions (in %)			
		Discriminatory	Effective	Induced	Supplementary
Information Board 1	Baseline	59.76	29.88	0	0
	Sensor Data	0	9.21	80.42	0
Information Board 2	Baseline	89.64	0.36	0	0
	Sensor Data	0	6.27	83.72	0
Bus Plan 1	Baseline	74.04	22.51	2.35	0
	Sensor Data	0	1.12	97.60	0
Traffic Sign 1	Baseline	99.6	0	0.4	0
	Sensor Data	0	0	100	0
Traffic Sign 2	Baseline	99.6	0	0	0
	Sensor Data	0	0	99.6	0
Traffic Sign 3	Baseline	100	0	0	0
	Sensor Data	0.4	0	99.59	0
Traffic Sign 4	Baseline	100	0	0	0
	Sensor Data	0.4	0	99.59	0
Traffic Sign 5	Baseline	85.79	13.8	0	0
	Sensor Data	0.09	29.66	69.84	0
Traffic Sign 6	Baseline	99.6	0	0.4	0
	Sensor Data	0	0.6	99.39	0
Road Lamp 1	Baseline	99.99	0	0	0
	Sensor Data	0.41	1.62	97.96	0
Road Lamp 2	Baseline	100	0	0	0
	Sensor Data	0.4	5.6	94	0
Road Lamp 3	Baseline	99.6	0	0	0
	Sensor Data	0	0.4	98.39	0
Traffic Sign 7	Baseline	100	0	0	0
	Sensor Data	0.42	0	99.57	0

Table 4.1: Shift of the projection of different objects on the regions of human peripheral vision. This shift is due to the measurement error of the eye tracker.

The test consists of 16 small urban objects (see Figure 4.3 for the categories). For each of these objects, 250 data points have been collected. The presented results are the average of these values. As can be seen in Table 4.1, the percentage for the baseline in the Discriminatory field of view is high. This is due the fact that the user has always looked at the middle of the object. This way, we can be sure that the baseline ray has been calculated correctly. Regarding the sensor data, the percentage in the induced field of view is high. This means that because of the measurement errors, the field of view has been shifted in a way that the focus object (object in the discriminatory field of view) has landed in the induced field of view. As this pattern occurs for all the listed objects, it is a strong indicator that with this measurement error no statement can be made regarding the object in focus or the position of the objects in the different locations of the peripheral view field.

4.2 3D Peripheral View Calculation Model

In order to build the 3D peripheral view calculation model a study with 68 participants is conducted. In the rest of this section, the conducted user study for the development and evaluation of the developed 3D peripheral view calculation model is described. At the end, this model is presented and discussed.

4.2.1 Participants

In total, 68 undergraduate and postgraduate students (aged between 19 and 40 years: mean = 26, STD = 4) from different disciplinary backgrounds were tested. Among the participants, 19 were female (male: 49) and 5 persons were left-handed. The focus was on people with normal and corrected-to-normal vision while excluding wearers of glasses due to technical restrictions of the involved eye-tracker (contact lenses were ok). The participants received 10 euros, for 40 minutes of participation. In addition, to keep people's commitment high, an additional fee to the four best performers in terms of accuracy and reaction times was paid. All candidates were unaware of the task of the experiment. For ensuring normal (20/20) vision of the participants, a short eye test as a pre-step of the actual experiment was conducted. This test will be described in the following section.

4.2.2 Eye Test

While there exist fully automated tests for human central vision [Bach, 2006] which can be conducted in front of a PC monitor, in terms of accessible web interfaces, our

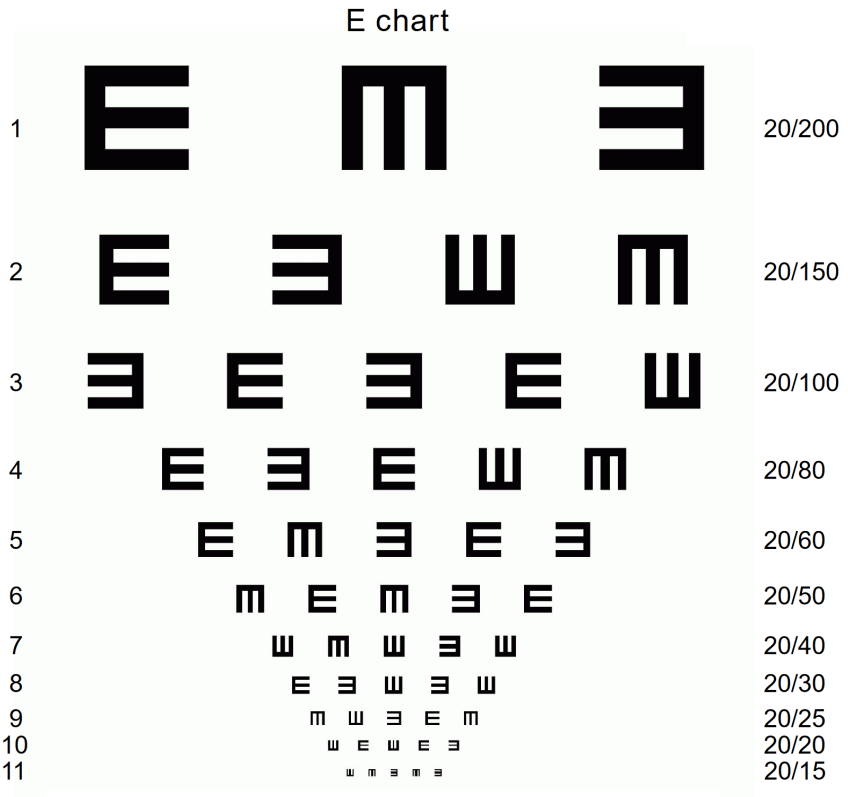

Figure 4.4: Eye test consisting of illiterate E symbols with four different orientations. For ensuring 20/20 vision, at least three items in line 10 had to be correctly identified.[3]

experiment opted for a freely available Tumbling E eye chart as shown by Figure 4.4 which we printed in the A4 standard format and attached to a wall. It shows eleven lines consisting of a number of illiterate E letters with four different orientations varying in size and number for each line. Here, a line indicates a fixed visual acuity value in terms of Snellen fractions, where the numerator is the testing distance, usually expressed in meters, and the denominator constitutes the distance of the smallest symbol read by a human while subtending a visual angle of 5 minutes of arc. As we were interested in normal, i.e. 20/20 vision, subjects were asked to stand at a marked position at a distance of 1.6 meters from the chart and to read the second-to-last line of the chart from left to right with both eyes open. If a person was able to correctly identify the orientation of at least 3 out of the 5 E symbols, he or she was allowed to participate in our experiment. It should be noted that the task of this procedure was not to determine a precise acuity value for human central vision by means of a

[3]Source: provisu.ch

clinical diagnosis, but only to ensure that people see well enough, thereby allowing for a meaningful visual performance in the experiment.

4.2.3 Experiment Setup

The apparatus used in our user study consisted of an HTC Vive (resolution: 1080×1200 pixels per eye, FOV: $110°$) with SMI's integrated eye-tracking component[4] offering gaze information at 250 Hz, MSI's GT73VR 6RF Titan Pro gaming notebook and a Logitech Extreme 3D Pro USB joystick for user input. Figure 4.5 depicts the components. People were asked to sit on a comfortable chair with adjustable height in front of a table onto which said input device was fixed in a physically stable position. While wearing the HMD, the participants rested in this seated position and were then asked to react to presented stimuli as accurately and quickly as possible by pressing the joystick in one of four directions. The visual content presented in the HMD was mirrored to the display of the laptop allowing the experimenter to monitor and assess the subject's performance and degree of task understanding during the test. The experiment was implemented in the Unity game engine[5] for which SMI offers a dedicated plugin providing access to the user's gaze data in real time and an easy-to-use, on-the-fly calibration routine for the eye tracker. Details on the visual stimuli which were used to assess the peripheral visual acuity of the participants will be given in the following.

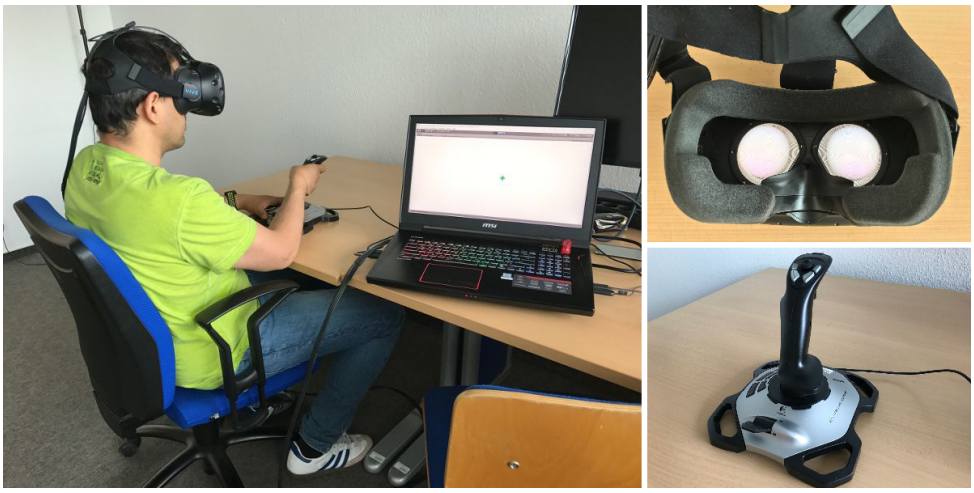

Figure 4.5: Experiment setup. Left: Participant in seated position with monitoring. Right: VR headset with integrated eye tracking and flightstick for recording user response.

[4] https://www.smivision.com/
[5] https://unity3d.com/

4.2.4 Visual Stimuli

The visual content which was presented to the subjects in a VR environment consisted of illiterate E symbols with varying, pre-defined scale, orientation and position on top of a perpendicular white plane which was drawn at a fixed distance from the subject's head position in the VR space. Four different object scales in terms of visual angle ($0.5°$, $1.33°$, $2.17°$, $3.0°$) were defined, while ensuring an appropriate rescaling of the virtual symbols for a given fixed distance. Having one of four different orientations (top, bottom, left, right) as indicated by the position of the small gaps, one E symbol was shown at one of eight possible, uniformly distributed positions in one of three circular regions of eccentricity ($6.5°$, $14.0°$, $21.5°$) around the center of gaze. This way, by means of the method of constant stimuli,$4 * 4 * 8 * 3 = 384$ different conditions were created under which a visual stimulus was randomly presented in peripheral regions of the human visual field. In this context, the presentation time of a visual stimulus was restricted to 150 milliseconds.

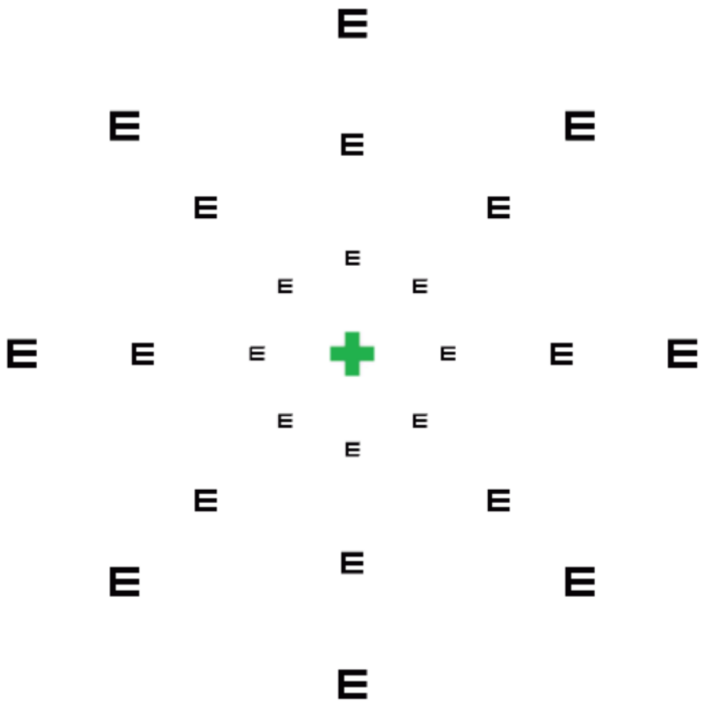

Figure 4.6: Visual stimuli (tumbling E's) with three varying scales presented at eight different positions over three peripheral, circular regions. The centered cross marks the desired gaze position and becomes green if focused on by a participant.

4.2.5 Procedure

4.2.5.1 Preliminaries

As has already been discussed, each participant had to pass a short eye test to ensure normal vision capabilities. After a short explanation of the general purpose of the peripheral vision study in a VR environment, several statistics were collected including age, gender, handedness and the participant's main subject. People received written, detailed information regarding the purpose of our research, including a declaration of consent and a privacy statement. The experimental environment had been designed in a way to prevent any disturbances during the testing.

4.2.5.2 Calibration

After an appropriate mounting of the VR headset, an initial calibration of the eye tracking system was performed. Figure 4.7 visualizes both the calibration and validation procedures. In the context of a triangular calibration process, probands were asked to follow the position of a red dot with their eyes over time while moving from a central starting position to three reference positions. The visual validation process then consisted of an overlapping of the subject's gaze (black dot) with several reference positions in a dot matrix.

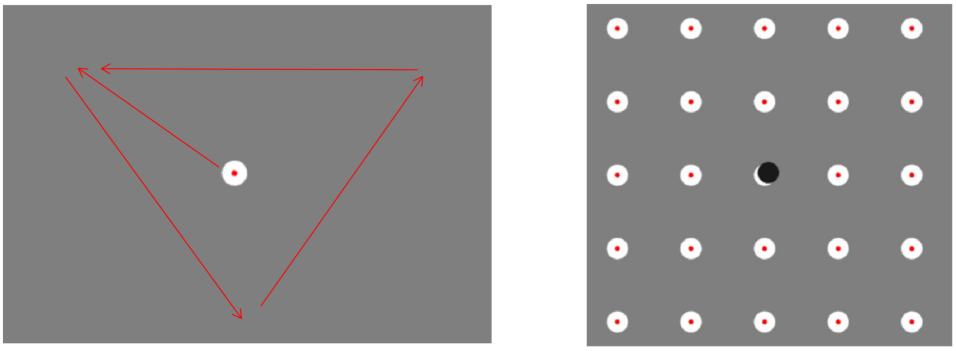

Figure 4.7: Instruction sheet for calibration/validation of the eye tracker. Left: Triangular calibration procedure. Right: Visual validation of tracking accuracy with dot pattern.

If the accuracy of the tracking was not good enough, the process could easily be repeated, especially as it could be conducted in a matter of seconds. This process was repeated after each break in the experiment where people were allowed to remove the HMD.

4.2.5.3 Trial

The experiment was designed in such a way that subjects had to observe and judge a peripherally presented visual stimulus under 384 different conditions, which we sequenced in terms of picking a trial from a randomly shuffled list. In order to avoid fatigue and dizziness, especially in a VR environment, the experiment was organized into three blocks with a five-minute break between the blocks, while allowing for further breaks if needed. Figure 4.8 depicts the typical steps of one trial of the experiment. First, the experimenter ensured that subjects were fixating on a centered black cross using the calibrated eye-tracker, such that the visual stimulus indeed appeared at the desired peripheral locations. As visual feedback, the color of the cross changed to green. People were then asked to keep their gaze focused on the cross during the whole trial. As this task turned out to be quite demanding over time, we provided our participants the possibility to relax their eyes between the trials and indicate when their gaze was stable enough by pressing the trigger button of the joystick. Soon after that, we presented a visual stimulus at pre-defined positions in the outer regions of the white plane and people were told to indicate the position of the gaps of an E symbol by pressing the joystick in the corresponding direction. While favoring acuracy over reaction time, we asked people to offer us their best guess if they were not sure about the correct answer. As maximum reaction time, we specified 2000 milliseconds and marked a trial as answered incorrectly if this time limit was exceeded. It should be further noted that if participants shifted their gaze away from the cross after having pressed the trigger button, the corresponding trial was immediately canceled and randomly repeated until it was handled in a valid manner.

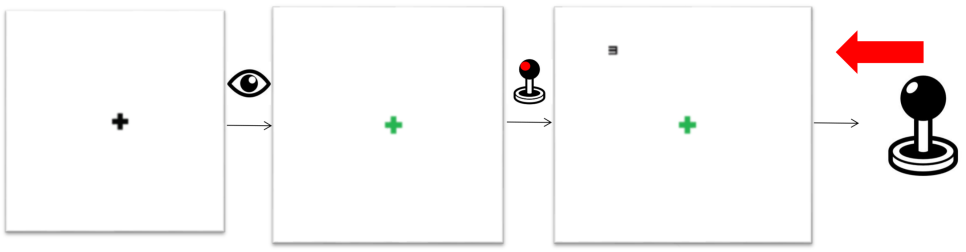

Figure 4.8: Instruction sheet for a trial (3-step procedure). After successful visual *fixation* of a centered cross (indicated by a change of color), a subject indicates *readiness* by pressing the joystick's fire button. Finally, subjects *respond* to the orientation of a shortly presented visual stimulus by pressing the joystick in the corresponding direction.

In order to ensure that the given task had been fully understood by the participants, a few practice trials were offered for getting used to the task and the system. In this context, the used scene monitoring on an external display was of great use for eliminating any misunderstandings of the subjects while giving feedback.

4.2.6 Measurements

The goal of the following sections is to find a parametric model involving the size of an object at a certain distance and its position in the human visual field in terms of eccentricity. The data observations from which the model will be learned from are based on the following measurements. One can interpret the visibility of a peripherally observed object as the probability that it will be correctly identified by a human with normal sight under pre-defined conditions. For this reason, a percent recognition performance (PRP) value is computed, i.e. the fraction of correctly judged trials across all trials following a specified condition. This is computed for all experimental conditions while averaging values over the four possible orientations of an object. The time taken by a participant is also recorded to produce a response in order to judge its meaningfulness as part of the process of cleaning our data.

4.2.7 Modeling Peripheral Visibility and Data Fitting

4.2.7.1 Data Preparation

Due to technical reasons, two of the 68 tested persons were excluded. Here the data was either incomplete or erroneous. The data was then split into two equal-sized parts: one for serving as training data for the model and one for evaluation purposes by means of test data. In order to find meaningful relationships between the collected data, one had to identify bad trials while filtering them out from our data set. Trials with very short reaction times can be assumed to constitute less reliable data. In this context, it can be safely assumed that a human cannot physically respond to a visual stimulus in less than 300 milliseconds. Hence, all trials where the reaction time was less than 300 milliseconds were removed. For these trials it can be assumed that the user input was either erroneous in terms of a non-neutral position of the joystick, or anticipatory, i.e. reactions had been initiated before the presentation of the visual stimulus. Besides cleaning the data, it was also extended by theoretical assumptions that were not directly measured. This additional information includes the following: It was assumed that all PRP values are of course zero for a zero object scale. At zero eccentricity, one assumed perfect (20/20) vision, i.e. a PRP value of one for all investigated scales. As visual observations become impossible at an eccentricity

level of 90 degrees in both directions, it was finally assumed that for these locations the PRP values becomes zero.

4.2.7.2 Model

It is well known that visual acuity of the human eye is best in the retinal center and rapidly falls off towards peripheral regions in a non-linear fashion. These ideas should also be reflected in our model, i.e. one assumes an exponentially decreasing visibility for an increasing eccentricity level and smaller object scales. In general, the model distinguishes between the visibility of a peripherally perceived object as it falls in the upper (positive vertical eccentricity) or lower (non-negative vertical eccentricity) part of the HVF. The reason for this was the hypothesis that humans perceive objects located in the lower half of the HVF "better" than those located in upper regions as numerically reflected in larger PRP values. To confirm this hypothesis, two sample left tail hypothesis tests at a 1% significance level were conducted. This means that parameters of the following functional models had to be estimated for both positive and non-negative vertical eccentricity values.

As a first step, it was sought for a functional relationship between an object's visibility and its position in the HVF. In this context, an exponential decay of an object's visibility for an increasing distance from the center of gaze was assumed, i.e. for a fixed object scale $s \geq 0$ and its position $\mathbf{e} = (e_V, e_H)^\top$ by means of vertical and horizontal eccentricities e_V, $e_H \in [-90, 90]$, its visibility V_s is assumed to follow the nature of a two-dimensional Gaussian

$$V_s(\mathbf{e}) = a \exp\left(-\left(\frac{(e_V - e_{V,0})^2}{2\,\sigma_V^2} + \frac{(e_H - e_{H,0})^2}{2\,\sigma_H^2}\right)\right) \quad , \tag{4.1}$$

where $(e_{V,0}, e_{H,0})^\top$ is the center of the bell-shaped function and σ_V, σ_H constitute the width.

Secondly, one was interested in modeling the relationship between an object's visibility and its angular size, which we also assumed to be exponential in nature. For a fixed radial eccentricity $\mathbf{e}$, the relationship between an object's visibility $V_\mathbf{e}$ and its scale s is assumed to follow the nature of a cumulative Weibull distribution function, i.e.

$$V_\mathbf{e}(s) = 1 - \exp\left(-\left(\lambda\, s\right)^k\right) \quad , \tag{4.2}$$

where $k > 0$ is known as the shape parameter and $1/\lambda > 0$ is the scale parameter of the function. In this respect, for larger sizes, an object's visibility increases with a varying steepness of the curve for different object scales.

Finally, for estimating the overall visibility V of a peripherally observed object, the

Equations 4.1 and 4.2 were combined into the joined model

$$V(\mathbf{e}, s) = \exp\left(-\left(\frac{(e_V - e_{V,0})^2}{2\,\sigma_V^2} + \frac{(e_H - e_{H,0})^2}{2\,\sigma_H^2}\right) \cdot (V_{\mathbf{e}}(s))^{-1}\right) \quad , \tag{4.3}$$

which is decreasing exponentially in an object's eccentricity and increasing in an object's scale. Basically, this model constitutes a 4D function mapping angular vertical and horizontal distances of an object from the center of gaze and its observed size to a scalar-valued visibility on the unit interval.

4.2.7.3 Data Fitting

In order to estimate the parameters of the model in terms of nonlinear regression techniques, MATLAB's curve fitting tool and nonlinear least square solvers were used. As a pre-step, it was confirmed that there is a statistically significant difference in the performance of human peripheral vision concerning the lower and upper half of the visual field. The corresponding two sample t-test was conducted on the whole cleaned data set and will be described in the following section. Then, using the curve fitting tool, the independent parameter sets for the models 4.1 and 4.2 were determined. Finally, in terms of a nonlinear least squares fit, the optimal parameters for the joined model 4.3 were determined. The results will be shown in the following.

4.2.8 Results and Discussion

Let us start with the numerical results of our investigations considering human peripheral vision in upper and lower parts of the visual field. Here, a two-sample independent t-test revealed a statistically significant difference between the corresponding PRP values ($\mu_{upper} = 0.7837$, $\mu_{lower} = 0.8167$, $t(134) = -3.0583$, $SD = 0.0629$). Just for comparison, the same test was conducted for data points in the left and right half of the visual field, where - as expected - no significant difference in the mean PRP values was found ($\mu_{left} = 0.8064$, $\mu_{right} = 0.7943$, $t(134) = 1.1372$, $SD = 0.0622$). For these reasons, the estimation for the parameters of the models 4.1 - 4.3 for the upper ($e_V \geq 0$) and lower ($e_V \leq 0$) half of the HVF were performed separately.

Figure 4.9 depicts the estimated relationship between an object's angular size and its visibility for different peripheral regions ($R0 : 6.5°$, $R1 : 14.0°$, $R2 : 21.5°$), each representing circular levels of object eccentricity $\mathbf{e}$.

The plots constitute different graphical representations of a cumulative Weibull distribution function, known as a standard predictor for a psychometric function, which

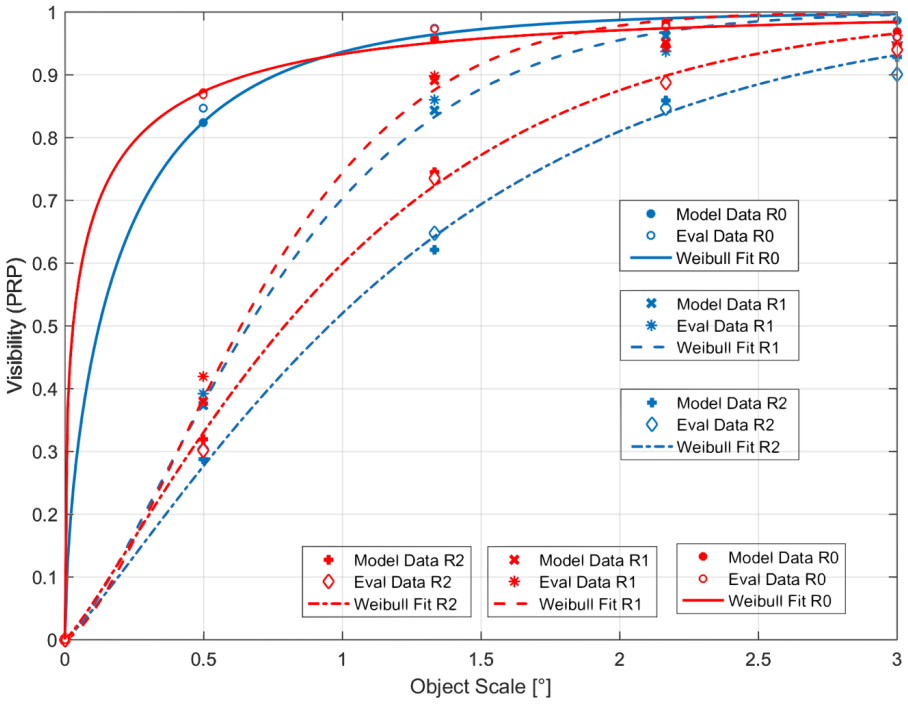

Figure 4.9: Estimated relationship between object scale and visibility for the upper (blue) and lower (red) half of the HVF. Function plots represent non-linear least square fits of the model data to a cumulative Weibull distribution function (Equation 4.2) for different peripheral regions of the HVF (R0 - R2).

Upper	λ	k	SSE	RMSE	$R-$square	**Validation** SSE	**Validation** RMSE
R0	4.63 (2.65, 6.61)	0.66 (0.4, 0.92)	3.42e-04	0.011	0.9995	0.002	0.022
R1	1.15 (1.0, 1.3)	1.36 (1.02, 1.69)	0.002	0.024	0.998	0.004	0.028
R2	0.77 (0.7, 0.84)	1.18 (0.99, 1.37)	9.795e-04	0.018	0.998	0.002	0.019
Lower	λ	k	SSE	RMSE	$R-$square	**Validation** SSE	**Validation** RMSE
R0	13.01 (-1.73, 27.76)	0.39 (0.2, 0.57)	3.82e-04	0.011	0.9995	0.001	0.015
R1	1.23 (0.97, 1.48)	1.49 (0.88, 2.1)	0.004	0.039	0.994	0.005	0.031
R2	0.93 (0.84, 1.01)	1.18 (0.99, 1.39)	9.92e-04	0.018	0.999	0.002	0.019

Table 4.2: Results for NLS fit of model data to Equation 4.2 for the upper and lower HVF and different eccentricity levels (R0-R2). Estimated parameters are shown with 95% confidence bounds together with goodness of fit statistics.

relates stimulus intensity, i.e. the angular size of an object in our case, to percent correct. Table 4.2 shows the corresponding estimation results for the two free parameters λ, $k > 0$ together with goodness of fit statistics. As one can see, the red curves, related to the lower half of the HVF generally exceed the level of the blue ones corresponding to human vision capabilities in upper peripheral regions.

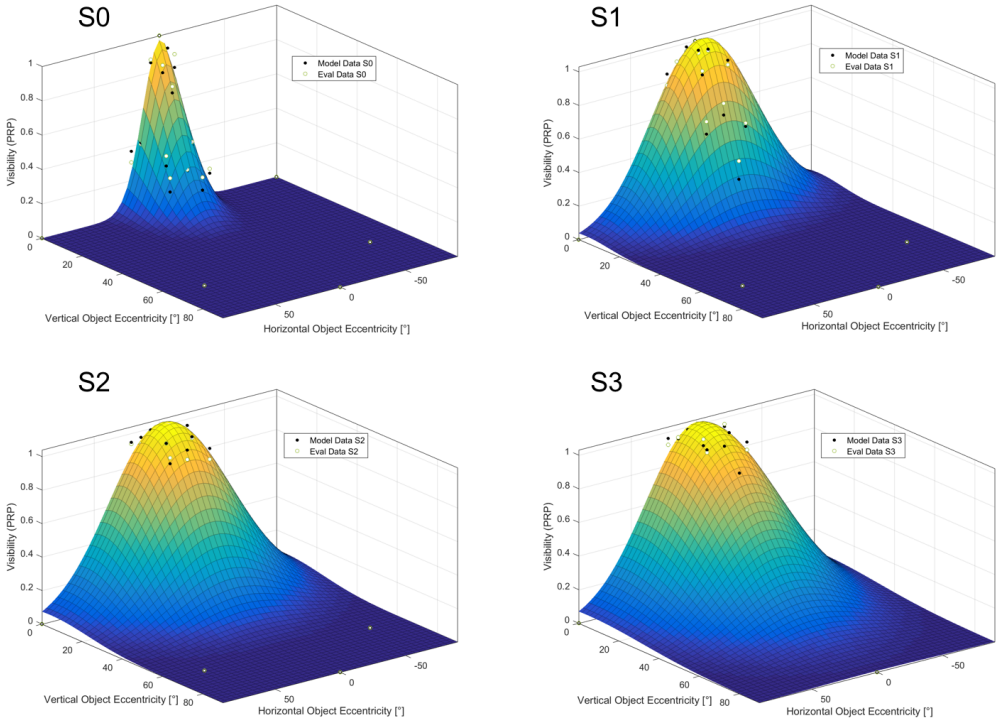

Figure 4.10: Estimated relationship between object eccentricity and visibility for the upper half of the HVF. Function plots represent nonlinear least square fits of our model data to a 2D Gaussian (Equation 4.1) for different object scales (S0 - S3).

In both cases, the proposed model fits our training data quite well, as reflected by small values for the sum of squared residuals (SSE) and the root mean squared error (RMSE). Furthermore, the fits are capable of describing the variation of our data as indicated by $\mathrm{R-square}$ values close to one. Also the corresponding error statistics retrieved from validating model 4.2 on the test data set are small. The way the size of an object influences the probability of correctly identifying it varies strongly among several peripheral regions as one can see from both the spread and steepness of the functions determined by the scale and shape parameters, respectively. In this context, for region R0 at a smaller eccentricity level of $6.5°$, smaller scale ($1/\lambda = 1/4.63 < 1$) and shape ($k = 0.66 < 1$) parameters generate a concave, compact cumulative distribution function, indicating that even small increases in object size lead to large improvements in object identification tasks. In comparison to this, for larger eccentricity levels (R2: $21.5°$), increased scale ($1/\lambda = 1/0.77 > 1$) and shape ($k = 1.18 > 1$) parameters result in a more spread distribution changing its curvature from convex to concave while indicating much smaller and slower increasing PRP values for increasing object sizes.

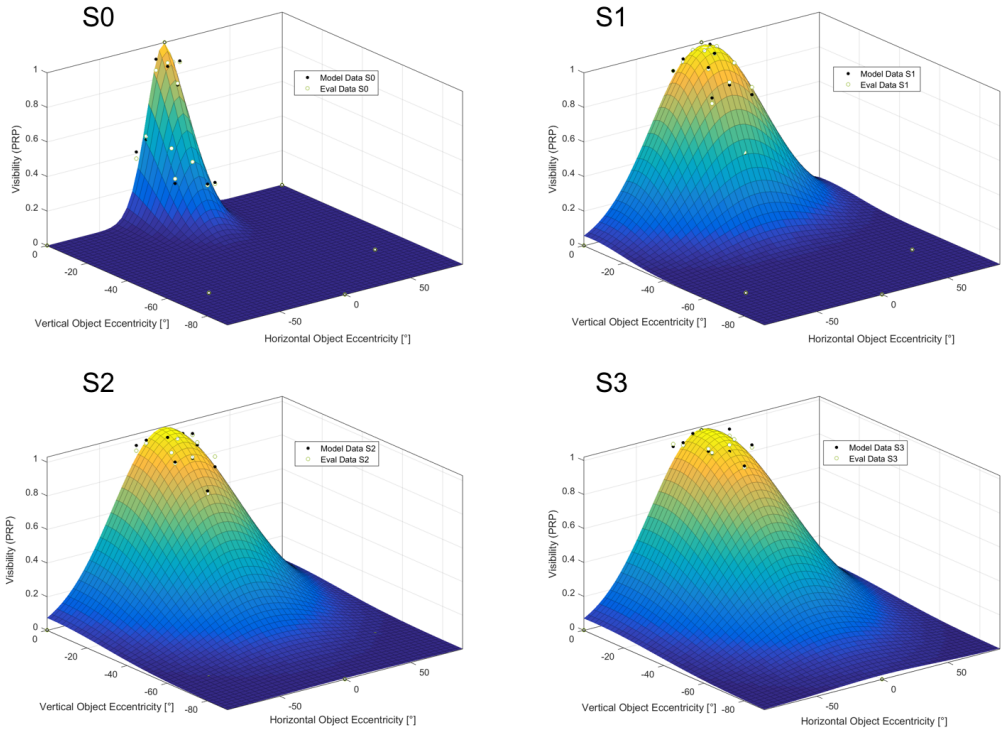

Figure 4.11: Estimated relationship between object eccentricity and visibility for the lower half of the HVF. Function plots represent nonlinear least square fits of the model data to a 2D Gaussian (Equation 4.1) for different object scales (S0 - S3).

Note at this point, that λ and k implicitly encode the dependency of an object's peripheral visibility on its eccentricity $\mathbf{e}$.

It is now possible to turn to the estimation results concerning the relationship between an object's visibility and its position relative to the center of gaze in terms of eccentricity as modeled by Equation 4.1. Figure 4.10 and Figure 4.11, show the results obtained from fitting our training data for the four different object scales (S0: $0.5°$, S1: $1.33°$, S2: $2.17°$, S3: $3.0°$) and the upper and lower half of the HVF, respectively. The corresponding parameter estimations together with the goodness of the fit can be found in Table 4.3. Again, as one can see from low values for both error measures and a coefficient of determination close to one, the model fits the data quite well in both cases. As expected, the height of the curve's peak by means of parameter a was determined to be close to one. For larger object scales, human peripheral vision declines slower in comparison to smaller ones which is reflected in increasing standard deviations of the function in both directions for an increasing object size. Furthermore, again as expected, the mean of the curve in a horizontal direction is generally close to zero, indicating the already discussed symmetry of human pe-

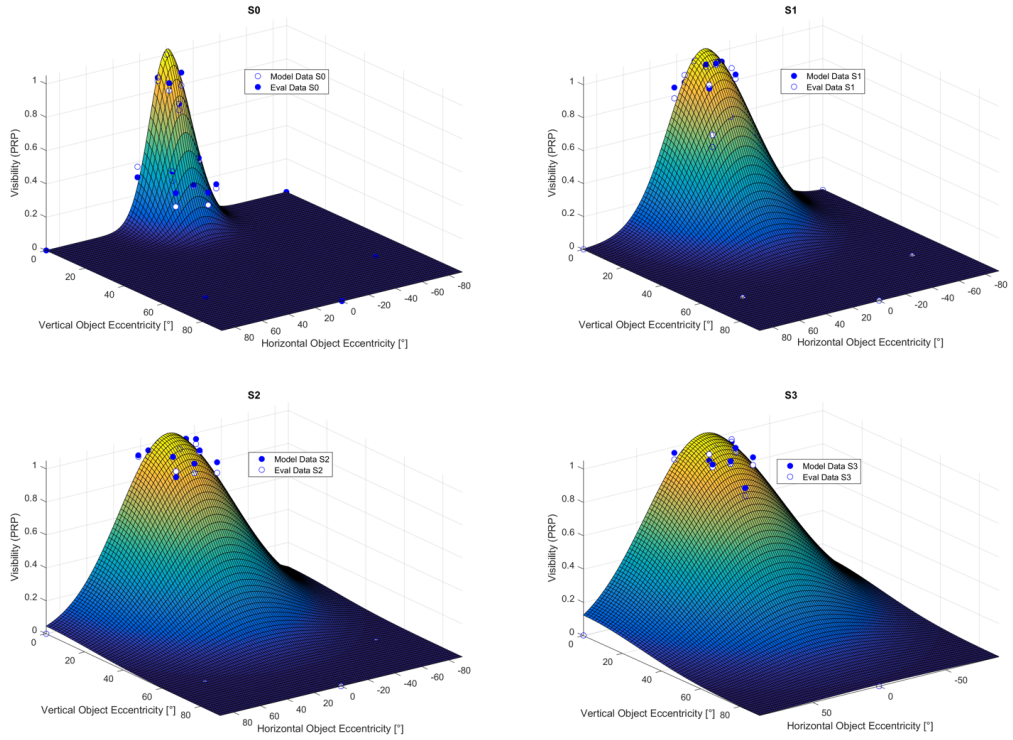

Figure 4.12: Estimated relationship between object eccentricity and visibility for the upper half of the HVF. Function plots represent nonlinear least square fits of the model data to Equation 4.3 for different object scales (S0 - S3).

ripheral vision in that direction. Estimated vertical mean values have been found a bit suboptimal however, as they result in an unwanted shift of the curve associating higher PRP values for non-zero eccentricities.

Nonetheless, both mean values were left as free parameters of the fitting process, in order to get better overall results. Additionally, as we will see in a moment, this problem will be eased in the context of fitting the data to the joined model. Again, at this point one should note that σ_H and σ_V, since controlling the width of the bell-shaped visibility curve, basically constitute an implicit encoding of the dependency of an object's peripheral visibility on its angular size s.

So far, we have evaluated two reasonable models for describing the relationship between the visibility of an object in the periphery of the HVF and its distance from the center of gaze in terms of horizontal and vertical eccentricity on the one hand and its angular scale on the other hand while parameterizing one of both independent variables implicitly. What is left to be done is combining both observations into a joined model by means of fitting Equation 4.3 to our data. Results can be observed in Fig-

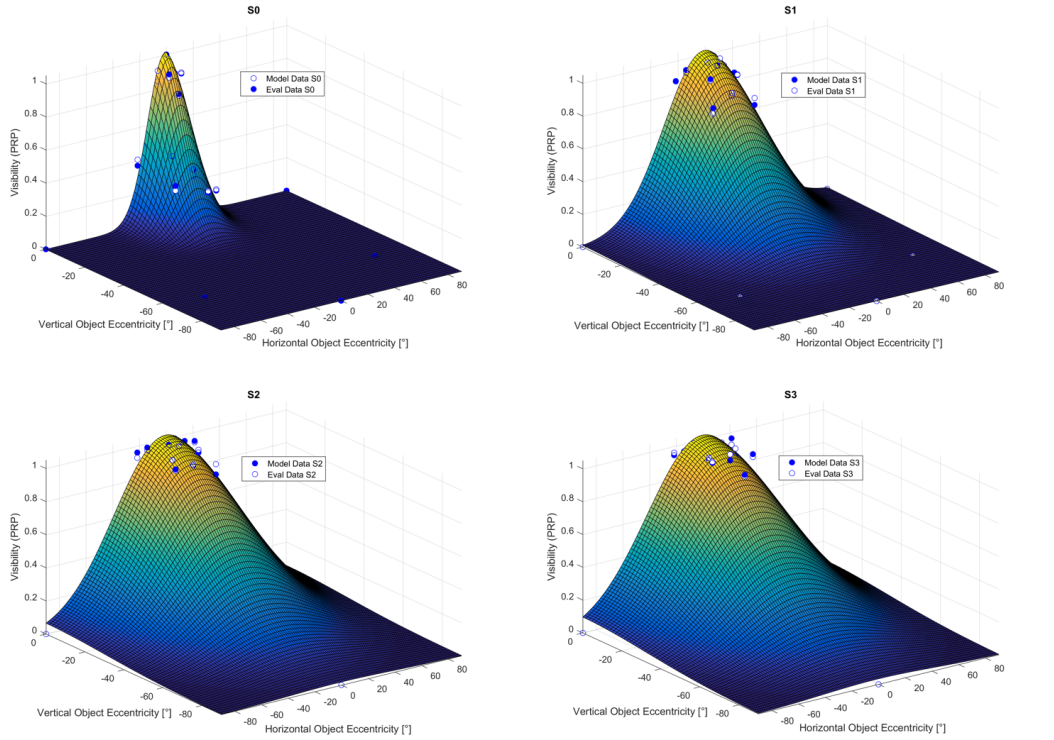

Figure 4.13: Estimated relationship between object eccentricity and visibility for the lower half of the HVF. Function plots represent nonlinear least square fits of the model data to Equation 4.3 for different object scales (S0 - S3).

ure 4.12, Figure 4.13 and Table 4.4. Note that the joined model is defined over two independent variables in terms of a scalar and a vector while yielding a real-valued visibility measure. Hence, its graphical representation can be thought of as a 3D surface evolving with an object's scale. Both figures actually illustrate four instances of the function for the angular sizes S0-S3. One observes that the functional plots look very similar to the ones of Model 4.1 with the difference that the mean of the curves is now closer to zero as desired.

Upper	a	σ_H	σ_V	$e_{H,0}$	$e_{V,0}$
S0	$1.06\,(0.65, 1.47)$	$13.53\,(11.54, 15.51)$	$13.26\,(5.38, 21.14)$	$-0.86\,(-2.55, 0.82)$	$-5.32\,(-18.96, 8.33)$
S1	$1.03\,(1.0, 1.06)$	$35.77\,(30.6, 40.93)$	$11.42\,(10.18, 12.66)$	$-1.44\,(-4.01, 1.12)$	$3.36\,(2.4, 4.31)$
S2	$1.04\,(0.99, 1.09)$	$39.61\,(31.46, 47.77)$	$18.81\,(12.34, 25.28)$	$-0.71\,(-5.61, 4.2)$	$4.15\,(0.75, 7.54)$
S3	$1.03\,(0.99, 1.07)$	$38.98\,(32.21, 45.75)$	$27.56\,(15.83, 39.3)$	$-0.095\,(-3.97, 3.78)$	$4.48\,(-0.48, 9.43)$

Upper	SSE	RMSE	$\mathrm{R-square}$	**Validation** SSE	**Validation** RMSE
S0	0.101	0.079	0.954	0.133	0.08
S1	0.02	0.035	0.994	0.041	0.044
S2	0.062	0.062	0.982	0.068	0.057
S3	0.042	0.051	0.99	0.045	0.047

Lower	a	σ_H	σ_V	$e_{H,0}$	$e_{V,0}$
S0	$1.07\,(0.62, 1.51)$	$13.16\,(10.95, 15.37)$	$14.41\,(4.23, 24.6)$	$-0.61\,(-2.51, 1.29)$	$5.24\,(-11.55, 22.02)$
S1	$1.01\,(0.97, 1.04)$	$37\,(30.89, 43.11)$	$15.8\,(12.78, 18.82)$	$-1.76\,(-4.97, 1.45)$	$-3.83\,(-5.71, -1.96)$
S2	$1.03(0.99, 1.07)$	$38.58(32.39, 44.78)$	$27.48(15.32, 39.63)$	$-1.32(-4.85, 2.2)$	$-1.32(-8.44, 5.8)$
S3	$1.02(0.98, 1.06)$	$38.89(31.08, 46.71)$	$31.19(17.5, 44.87)$	$-0.07(-4.55, 4.42)$	$-4.67(-11.06, 1.71)$

Lower	SSE	RMSE	$\mathrm{R-square}$	**Validation** SSE	**Validation** RMSE
S0	0.147	0.096	0.94	0.104	0.07
S1	0.029	0.043	0.991	0.023	0.033
S2	0.036	0.047	0.9897	0.04	0.044
S3	0.057	0.06	0.984	0.049	0.048

Table 4.3: Results for NLS fit of model data to Equation 4.1 for the upper and lower HVF and different object scales (S0-S3). Estimated parameters are shown with 95% confidence bounds together with goodness of fit statistics.

Upper											
a	σ_H	σ_V	$e_{H,0}$	$e_{V,0}$	λ	k	SSE	RMSE	$\mathrm{R-square}$	**Validation** SSE	**Validation** RMSE
1.04	67.12	45.55	-0.82	-0.78	0.23	1.47	0.394	0.069	0.972	0.379	0.067
Lower											
a	σ_H	σ_V	$e_{H,0}$	$e_{V,0}$	λ	k	SSE	RMSE	$\mathrm{R-square}$	**Validation** SSE	**Validation** RMSE
1.03	41.95	35.74	-0.71	1.72	0.62	1.91	0.341	0.064	0.975	0.298	0.06

Table 4.4: Results for NLS fit of model data to Equation 4.3 for the upper and lower HVF. Estimated parameters with goodness of fit.

5 Algorithms for Reference Resolution and Peripheral Vision Analysis

This chapter presents the various methods for environment modeling and also the algorithms which have been developed as a part of this research. Section 5.1 presents the developed algorithms for using gaze to refer to objects in highly mobile scenarios. Section 5.2 provides details on the algorithm for using head orientation as an indicator of user identification. Section 5.3 discusses the algorithms which have been developed as a part of the general peripheral view calculation model.

5.1 Using Gaze to Refer to Objects in Highly Mobile Scenarios

Highly mobile scenarios are challenging in several aspects. For instance, the relative position of the user to other objects in the environment is changing permanently; the orientation of the user's head or the direction of his focus-of-attention might change very fast. In addition, different objects may enter or exit the environment at any moment. An example of such a scenario is driving. As the car moves, new objects come into the driver's field of view and other objects disappear. Furthermore, the relative position of the driver (or co-driver) relative to the visible objects in the environment is changing permanently and as the driver monitors the traffic, her visual focus permanently switches from one object to another. Such highly mobile scenarios are not exclusive to road traffic. Intra-logistics in industrial environments or specific situations in human-robot-interaction also show the same characteristics.

As daily traffic exhibits all the characteristics of a highly mobile scenario, this use case was chosen as a test bed for the design and implementation of our algorithms.

The main research question here is how it is possible to use gaze to refer to objects in such scenarios. The main steps for such a procedure are as follows:

- Modeling the environment.
- Monitoring the position and orientation of the observer in real-time.
- Monitoring the position and orientation of the moving objects in real-time.
- Creating a set of potential target objects[1] depending on the user's current focus-of-attention.

The output of the module for determining the object in focus is a list of objects with their corresponding likelihood. It is possible to take the object on the top of the list as the output, however, the outcome of the whole system will be more accurate if this information is combined with other context-sensitive information such as the information from the user's speech (available in the dialog system). The presented research concentrates however on the geometrical analysis and does not perform the reference resolution on the semantic level considering the object characteristics and the user's speech. Thus, the output of the algorithm is a list of potential target objects. In the following, different stages of the procedure of using gaze to refer to objects in highly mobile scenarios are described.

Environment modeling is an important part of this procedure. It is possible to model an environment in 2D, 2.5D or 3D. Each of these variations can affect the performance and the accuracy of the system. The positions of the objects and the observer also should be updated permanently and in real-time, otherwise the results of the whole system will be affected in a negative way. The last part of the procedure is to develop an algorithm for creating a set of potential target objects. In this research, we refer to this algorithm as a "reference resolution algorithm". In the conducted studies, several iterations were performed in order to use the user's gaze to refer to objects in the environment. In each iteration, the modeling part or the setup was changed. When needed, the algorithm was adapted to the new setup. In the following, the different iterations are presented.

5.1.1 Reference Resolution Algorithm for Applications with 2.5D City Models

In this iteration, a 2.5D model of the city of Saarbrücken was developed to serve as the basis for further visualization and analysis. It consists of two synchronized

[1] A target object is the object in the user's focus .

Figure 5.1: The modeled 2.5D city center displayed in Google Earth.

parts: a 2.5D model in Google Earth and a corresponding model in the PostGIS spatial database. Both included 528 buildings in the city of Saarbrücken (see Figure 5.1). The buildings were modeled using their two-dimensional footprint and their maximum height. Google Earth was used to determine the footprint of each building by marking its contours. For the height information, an airborne LIDAR (Light Detection and Ranging) scan of the city center was acquired from the Saarland state office of cadaster, measurement, and mapping[2]. This laser scan consisted of distinguished height information on the Earth's surface, buildings, vegetation and bridges. The accuracy of these heights was about 15 centimeters and the distance between the measurement points was about one meter.

In this iteration, data collection was performed in a regular car in real traffic. All required hardware instruments and software modules had been installed and tested in the vehicle before starting the actual data collection. The tests performed included, amongst others, several tours through the city. In each data collection, two people were involved: driver and co-driver. The driver was not involved in the actual data collection and was only responsible for driving. A Tobii IS Z1 Eye Tracker (see Figure 5.2), a camera, an external GPS module and a laptop computer were used. In this setup, the Tobii Studio software was connected to the eye tracker and the camera. An external battery was used as an energy supply for the eye tracker. The camera was directly connected to the Tobii Studio to make use of the scene recording feature. The eye tracker was mounted on the dashboard of the car to record the eye gaze of the user in the space above it. The camera was mounted on the upper left side

[2] Landesamt für Vermessung, Geoinformation und Landentwicklung

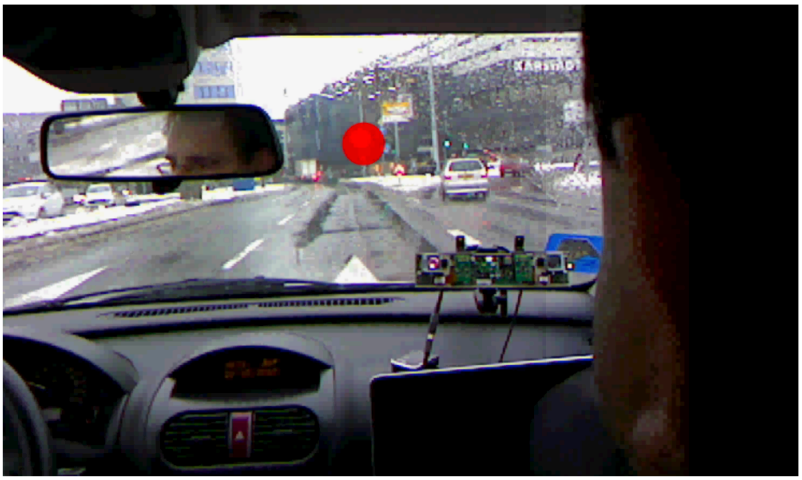

Figure 5.2: Tobii IS Z1 Eye Tracker used for data collection and its position in the vehicle. The red dot shows the eye gaze of the user while selecting the Karstadt department store.

of the co-driver's seat to be as close as possible to the user's eyes.

As different modalities were used in the data collection-phase, it was very useful to visualize the interaction of the user with the outside environment for each modality. Each interaction generates a vector with the user as the initial point and the target as the end point (assuming there are no measurement errors). The user's interaction can be depicted by visualizing these interaction vectors (see Figure 5.3). These vectors are three dimensional for the eye-gaze. The interaction of the user with the system while referring to a building may last for several seconds. In this period of time, in addition to the displacement of the user, several sensor snapshots were logged. Thus, in this context, visualizing the user's interaction vector means visualizing all the corresponding vectors when the user refers to one building.

In order to reproduce the context of the vehicle after the data collection, a scanning mechanism was developed by using the spatial data base. During the scanning phase, a ray was cast from a specific position in a specific direction (in the 2.5D spatial model), then the object which was hit by the ray was registered. A complete scanning procedure included several thousand cast rays. As a result of the scanning procedure, spatial features of the objects in the scanned directions were discovered. The results of the offline scanning phase were used by the reference resolution algorithm to detect the correct target building after the data collection was performed. In the following, the scanning procedure is described.

For the purpose of scanning, several hundred scan-points were spread along the data collection route. The distance between two adjacent scan-points was 12 meters. This

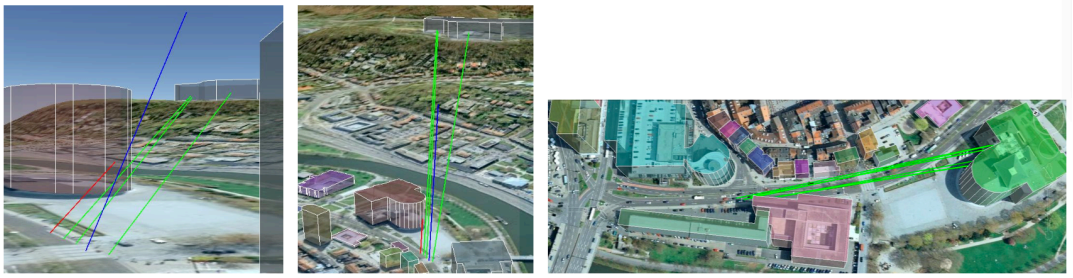

Figure 5.3: Visualization of two actions. Left and Middle: Five pointing vectors targeting a hospital on the hill. The red colored vector hit a non-target building, the blue vector hit nothing and the green vectors hit the target building. Right: Seven pointing vectors hit a target building.

distance was selected due to the fact that the calculation for each scan-point was very time-consuming. This study examined how the whole system performs with a 12 meter distance between each two scan-points. At each scan-point, an environment-scan in the driving direction was performed. The scan's coverage angles were 70 degrees vertically and 180 degrees horizontally (we consider a panoramic view of 180 degrees). These angles were selected due to the vehicle's structure: the passenger could observe the environment from 90 degrees left to 90 degrees right and from -10 degrees beneath his sight line up to 60 degrees above it. A unique ray was cast for all combinations of 71 vertical degrees (from -10 to 60) and 181 horizontal degrees (from 0 to 180). As a result, $181 \times 71 = 12851$ collision-points were detected. Some of these collision-points were buildings and others were the sky or the ground. The outcome of the scanning procedure was fitted in a table with 71 rows and 181 columns (scan-table). Each entry of this table corresponds to a coordinate of the object with which the ray cast from the position of the scan-point collided. The horizontal degree is depicted in the column and the vertical degree in the row. The whole table corresponds to the panoramic view of the environment in front of the vehicle, hence, it can be used as a lookup table for finding the visible object in each available direction (see Figure 5.5 and Figure 5.4). The reference resolution algorithm uses this table for its visibility analysis.

The developed spatial reference resolution algorithm will be described in the following. In order to better understand the algorithm, first some required definitions will be presented. Then, object-matrix Λ, which plays an important role in the algorithm, will be explained. The problem of spatial multimodal reference resolution can be defined as follows:

Definition 1 *A **physical object** $\omega = \langle f, h, s, d \rangle$, where f is the footprint of the object on the map and h is its height. Hence, a 2.5-dimensional geometrical description is used. Salience s and descriptions d are semantic characteristics. Ω is the set of*

Figure 5.4: The panoramic view matrix is used to calculate at each moment which part of each building in the environment is visible to the user.

Figure 5.5: The panoramic view of the user is transformed into a matrix with 181 columns and 71 rows.

physical objects in our domain (spatial knowledge base).

Definition 2 *A **spatial reference** $\sigma = \langle \rho, \epsilon, \mu, \omega \rangle$ is an action performed by a referer ρ in context ϵ using trigger mode or gesture mode $\mu^{t,g} \in M$ in order to refer to ω.*

Definition 3 *A **context** $\epsilon = \langle v, o \rangle$, where v is the velocity of the vehicle and o refers to the absolute orientation of the vehicle.*

Definition 4 *A **trigger** $\mu^t = \langle \psi, \tau \rangle$ is a mode $\in M^t \subset M$ specifying the geographical position $\psi(\rho)$, start time $\tau_1(\sigma)$ and end time $\tau_2(\sigma)$.*

Definition 5 *A **gesture** $\mu^g = \langle \mu^t, \delta \rangle$ is a mode $\in M^g \subset M$ consisting of a trigger and additionally specifying the three-dimensional direction $\delta(\sigma)$.*

Using the definitions above, a spatial-reference resolution algorithm can be defined as:

Definition 6 *A **spatial-reference resolution algorithm** θ is an algorithm that reads $\rho, \mu, \Omega, \epsilon$ and provides a list $\langle \langle \omega_1, \varsigma_1 \rangle ..., \langle \omega_n, \varsigma_n \rangle \rangle$, where $\omega_i \in \Omega, n = \mid \Omega \mid, 0 \leq \varsigma \leq 1, \forall i, j : \varsigma_i < \varsigma_j \rightarrow likelihood(\omega_i = \omega) < likelihood(\omega_j = \omega)$. $\theta \in \Theta$, the set of algorithms analyzed here.*

Hence, a spatial reference resolution algorithm outputs a list of objects, each annotated with a score, where higher scores imply a greater confidence of the algorithm in terms of the respective object actually being the "target". This paradigm is chosen as it facilitates passing the result list to a downstream semantic component which operates on ω's non-geometric characteristics (interestingness, number of facts in the knowledge base, history of information requests by the user or others, etc.) in order to create the final hypothesis. This study, however, is concerned with the geographical level only.

The presented algorithm is based on object-matrix Λ which is an approximation of the solid angle of the object. Solid angle is the two-dimensional angle in three-dimensional space that an object subtends at a point. It is a measure of how large that object appears to an observer looking from that point. A small object nearby may subtend the same solid angle (or even greater) as a larger object farther away. In the spatial reference resolution algorithm, object-matrix Λ of the buildings seen by the user are computed and considered. This value is an approximation of the solid angle of the corresponding object. Due to the different shape of the buildings, calculating their solid angle with the belonging formulas requires precise analysis and is rather complex. Hence, in this iteration of the study, the concept of object-matrix Λ was developed. This approach is based on the scan-tables (lookup tables) described before. Each entry of a building in the table counts for a part of its object-matrix. This way, the solid angle is discretized and approximated by storing the information of visible objects in a grid. That is to say, for computing the approximation of the total solid angle of a building, all the entries of that building in the scan-table are counted. The total number of entries in the scan-table correspond to the whole scene that can be seen by the user. Comparing the entries belonging to a specific building with the total number of entries determines the magnitude of the object-matrix of that building.

The algorithm weights the discretized object-matrix depending on the active modality. We call this weighted object-matrix the modality-dependent object-matrix. Figure 5.6 shows the weighting range for the eye gaze modality. The range depends on the angular dimensions of the modality. Angular dimension in this context means which of the two horizontal and vertical referring angles is determined by the modality. The reference vector of the eye gaze is three dimensional. In this case the weighted area in the lookup table is three nested circles. In the center of these circles is the exact collision-point of the reference vector with the environment. The entries in the lookup table are weighted differently for each circle. The inner regions are weighted more than the outer area.

The algorithm proposed for reference resolution within 2.5D City Models is as follows: First, the start and end times of the spatial reference (τ_1, τ_2), the positions of the referee along these two points in time $(\psi(\rho))$, and the absolute orientation of the vehicle (o) are determined. Two empty lists are created: S for the scan tables and

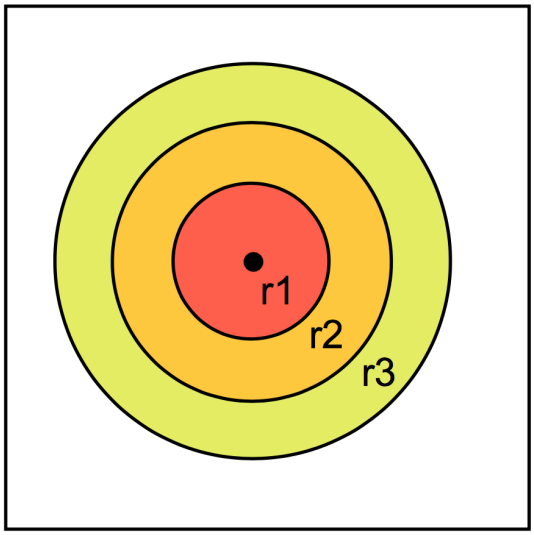

Figure 5.6: Different regions in the algorithm which are considered for different weighting.

$score$ for the final building scores. In addition, three empty mappings are created: $B_{r_1}, B_{r_2}, B_{r_3}$ for the potential target buildings in the regions r_1, r_2, r_3 respectively (see Figure 5.6). A precomputed environment scan (s_τ) is picked in regular spatial intervals d during the referring time period. This results in the set of scan tables for the current action (S). These environment scans are the described lookup tables. The environment scan stores all buildings visible at $\psi(\rho)$, taking into account o and further described constraints. The scan tables include a discretized approximation of the solid angle of the whole environment (an approximation), which is considered to be the view of the user at that point. For each Δt time interval in $[\tau_1, \tau_2]$ the corresponding scan table is selected. Then, by considering the reference vector, and the regions (r_1, r_2, r_3), the buildings which lay within each of these regions in the look up table (s_τ) are noted in the corresponding temporal mappings ($B_{r_1}, B_{r_2}, B_{r_3}$). The hitting position of the reference vector is used as the middle point for the regions (r_1, r_2, r_3). Then, for each table the number of occurrence for each small box (representing a small part of the solid angle of the object) which is shown in the Figure 5.5 is accumulated. Next, scores are accumulated over the number of mappings and weighted according to the different regions of the described circle in Figure 5.6. At the end the score list is sorted and returned. This score list contains the scores of all the visible buildings during the referring action.

Following is the analysis of the time complexity of this algorithm. The time complexity of the first four actions (acquiring $\tau_1, \tau_2, \psi(\rho), o$ and building empty sets and mappings $S, score, B_{I_1}, B_{I_2}, B_{I_3}$) is $\mathcal{O}(1)$. The time complexity of the loop **for** $(\tau = \tau_1;\ \tau \leq \tau_2;\ d)$ is $\mathcal{O}(n)$. The complexity of $s_\tau \leftarrow environmentscan(\psi(\rho), o)$

Reference Resolution Algorithm for Applications with 2.5D City Models

$\tau_1, \tau_2 \leftarrow \sigma$
$\psi(\rho) \leftarrow \sigma$
$o \leftarrow \epsilon$
$S, B_{I_1}, B_{I_2}, B_{I_3}, score \leftarrow nil$
for $(\tau = \tau_1;\ \tau \leq \tau_2;\ d)$ **do**
  $s_\tau \leftarrow environmentscan(\psi(\rho), o)$
  $S = S \cup s_\tau$
end for
for $(t_i = \tau_1;\ t_i \leq \tau_2;\ \Delta t)$ **do**
  $s_\tau \leftarrow (t_i, \epsilon, S)$
  $b_{r_1} \leftarrow targetbuildings(s_\tau, \mu^g, r_1)$
  $b_{r_2} \leftarrow targetbuildings(s_\tau, \mu^g, r_2)$
  $b_{r_3} \leftarrow targetbuildings(s_\tau, \mu^g, r_3)$
  $B_{r_1} = B_{r_1} + b_{r_1}$
  $B_{r_2} = B_{r_2} + b_{r_2}$
  $B_{r_3} = B_{r_3} + b_{r_3}$
end for
for all $buildings \in B_{r_i}$ **do**
  $score_b \leftarrow \alpha\big|building \in B_{r_1}\big| + \beta\big|building \in B_{r_2}\big| + \gamma\big|building \in B_{r_3}\big|$
end for
$score \leftarrow sort(all\ score_b)$
return $score$

is $\mathcal{O}(n)$ as it is in worst case a linear search. The complexity of the for loop **for** $(t_i = \tau_1;\ t_i \leq \tau_2;\ \Delta t)$ is also $\mathcal{O}(n)$. As $s_\tau \leftarrow (t_i, \epsilon, S)$ is in worst case a linear search, the time complexity here is $\mathcal{O}(n)$. For the lines $b_{r_i} \leftarrow targetbuildings(s_\tau, \mu^g, r_i)$, the time complexity is $\mathcal{O}(1)$, as they do lookups in a table with a maximum of 12851 entries ($181 \times 71 = 12851$). Time complexity of $B_{r_i} = B_{r_i} + b_{r_i}$ is in worst case linear, as it should find the building in the mapping and add b_{r_i} to it. The last loop in the algorithm consists of three loops which are executed successively (not nested). In each of these loops the $score_b$ for each building is computed and accumulated with the given formula. So in each loop, each building in the B_{r_i} is visited one time. The time complexity here is linear for each loop. The final action in this algorithm is sorting, for which a worst case complexity of $\mathcal{O}(n^2)$ can be considered.

Now we can add up the time complexity of each block of algorithm together. As

mentioned, first block has $\mathcal{O}(1)$. For the second block we have $\mathcal{O}(n) \cdot \mathcal{O}(n) = \mathcal{O}(n^2)$. For the third block we have $\mathcal{O}(n) \cdot (\mathcal{O}(n) + \mathcal{O}(1) + \mathcal{O}(n)) = \mathcal{O}(n^2)$. In the final loop we have three successive for loops (not nested) each with linear time complexity, so here time complexity will be $\mathcal{O}(n)$. As mentioned the time complexity for the final sort is in the worst case $\mathcal{O}(n^2)$. Thus, the overall time complexity of the algorithm is: $\mathcal{O}(1) + \mathcal{O}(n^2) + \mathcal{O}(n^2) + \mathcal{O}(n) + \mathcal{O}(n^2) = \mathcal{O}(n^2)$. It's obvious that algorithms presented in this chapter terminate since they operate on a finite model which is not manipulated by the algorithm.

5.1.2 Reference Resolution Algorithm for Applications with 2D Environment Models

Based on the results of the developed environment modeling and algorithm in Section 5.1.1, a new modeling mechanism together with a new variation of the algorithm was developed in order to be used in a vehicle operating live in real traffic. In the following this modeling and algorithm are described.

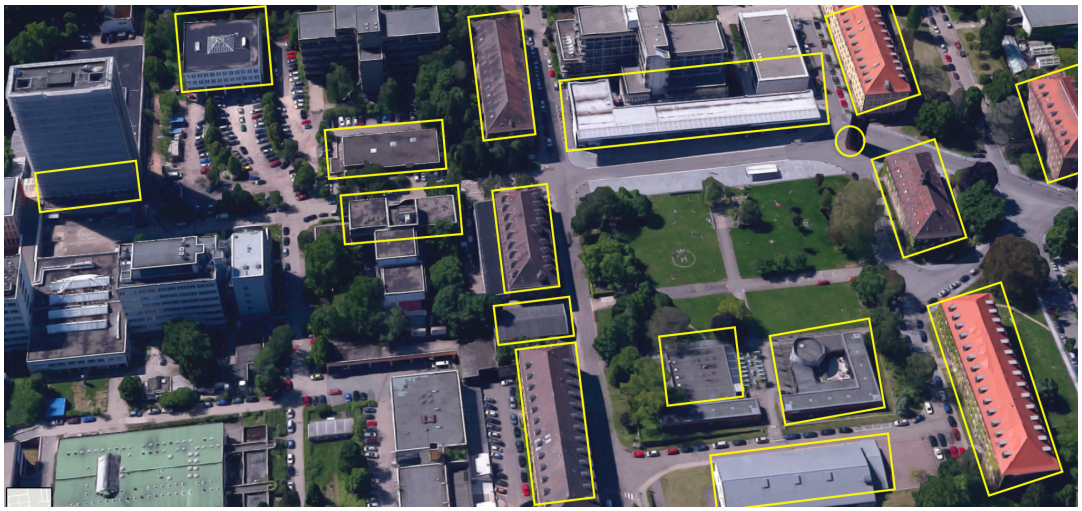

Figure 5.7: Part of the Saarland University campus and the footprint of the modeled buildings.

The modeling procedure here was reduced to analyzing the 2D footprint of the buildings (see Figure 5.7). The idea here was originated from the fact that the heights of the buildings are not totally visible to the users within the vehicle. Thus, a system which considers the width of the buildings might achieve good results as well. Furthermore, as this change results in a much smaller lookup-table. The number of comparisons (to the lookup-table) will be reduced drastically as this table will have just one row. In this iteration, however, the scanning distances were reduced to 2

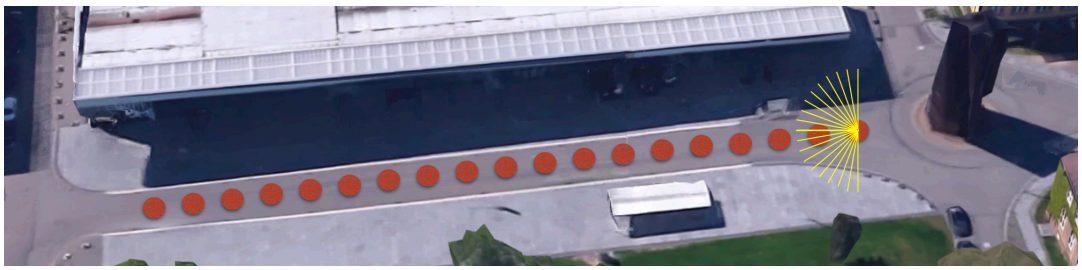

Figure 5.8: The different scan points (red points) and a sample of the scanning directions. The scanning was performed on each point in the driving direction for 180 degrees horizontally. The sampling rate was one degree.

Figure 5.9: The view from inside of the vehicle. As a result of the modeling, the system sees the buildings as one stripe, regardless of their height.

meters. For this purpose, the 2D footprints of the buildings were determined and the offline scanning mechanism was performed at 2 meter distances considering these footprints. The modeled environment in this case was the Saarland University campus which includes ca. 40 buildings (see Figure 5.7). As the scanning just considered the 2D footprint of the buildings, and the height was ignored, the result of the scans (lookup tables) had just one row and 180 columns (see Figure 5.8). Thus, in this case, instead of object-matrix, the algorithm will use object-vector.

As a result of these adaptions, the algorithm considered in each moment the visible width of the buildings (see Figure 5.9). This information was retrieved from the horizontal scanning mechanism using one row and 180 columns at each scan point.

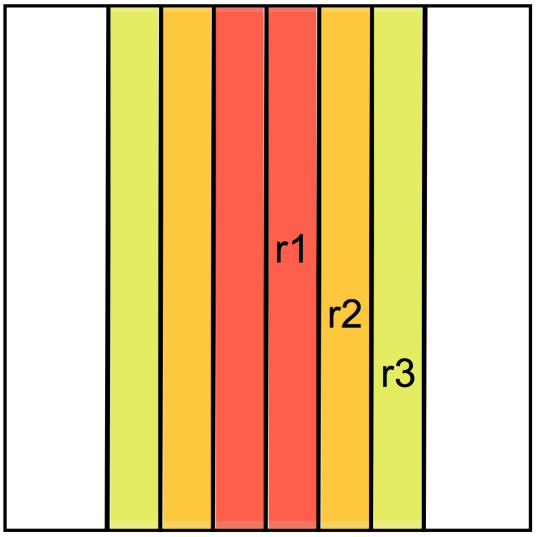

Figure 5.10: The 1D weighting mask used for the 2D modeling. Regardless of the height, the stripe which represents the building will be weighted ($r1 > r2 > r3$).

Furthermore, the weighting mask of the algorithm was adapted as well. Instead of a 2D circle (see Figure 5.6), a 1D horizontal mask was used (see Figure 5.10). This mask weighted the buildings left and right of the collision point of the gaze vector (see Figure 5.10). Further details of this algorithm remains similar to the algorithm described in 5.1.1, thus the detail description is omitted here. The time complexity here also remains the same: $\mathcal{O}(n^2)$. Based on this algorithm, two systems were implemented and tested. One in real traffic, and another one in a video simulation. Sections 7.1.1.1 and 7.1.1.2 describe these systems in detail.

5.1.3 Reference Resolution Algorithm and Modeling for Detailed 3D Environments

This modeling and algorithm technique has the following characteristics:

- Refine the scanning procedure so that all objects in the environment, including the small objects, can be considered by the developed system.
- Introduce a new spatial processing mechanism so that dynamic objects in the environment can enter or exit the scene at any time.

Regarding the environment reconstruction, it was important to model the details visible to the user. Some of these details were part of the outside environment (for

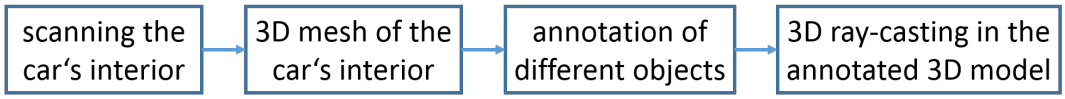

Figure 5.11: Different phases for the 3D environment reconstruction from scanning to ray-casting.

example trash can, street light, etc.) and some other details were part of the vehicle's dashboard (for example the navigation system, the air conditioner, etc.). Thus, it was important to model both these environments in detail and bring them into the Unity 3D game engine together. In the following this procedure is described in detail.

In this iteration, for the in-vehicle environment modeling, a depth camera in conjunction with surface reconstruction software was used. This way it was possible to create a color 3D model of the car's interior, which had the required precision for different interaction and analysis applications. This 3D mesh was then annotated in a 3D editor. In the annotation phase, every object in the car's interior model was marked. This annotated model was then used as a basis for further processing. Depending on the head pose and eye gaze of the user, it was possible to cast a ray in this model to reveal the object that is in the focus of attention. Figure 5.11 illustrates the described process. For modeling the outside environment, it was possible to use two alternative approaches. In the first approach the environment is scanned using a very precise 3D scanner. Then, based on this scan a mesh model of the environment is developed. In the second approach, an online map is imported in the 3D game engine. In both techniques the same ray casting method is used to reveal the object in the outside environment which is in the focus. In the following, we provide more details on the described scanning procedure for 3D reconstruction of the outdoor environment and also the interior of the vehicle.

In order to acquire an exact model of the vehicle's interior and the outside surroundings, these environments have been scanned using two different techniques. The outside environment is scanned with a professional 3D laser scanner. More specifically a Faro FOCUS 3D-S 120 (see Figure 2.23) was used to scan the university campus and all its details. The campus was scanned at 17 different places, and then these scans were merged using a special software. The output of these scans was a very detailed point-cloud of the environment (see Figure 5.12). Figure 5.13 shows an example of the outside environment model. For this purpose, more than three hectares of the university campus is scanned with centimeter accuracy (see Figure 5.13 (top)). The resulting point cloud was then used as a basis for a 2.5D polygon model (see Figure 5.13 (bottom)). This 2.5D model is then used together with a GPS map-matching algorithm to position the vehicle in the environment in real time. In addition, this model also contained different buildings in the environment as well as

Figure 5.12: Upside view of the 3D point cloud of the Saarland University campus. The small circles represent the scanning locations.

other smaller objects, such as bus stops and traffic signs and even small city garbage cans. As the point cloud model included all of these objects with high detail, it was possible to place them at the right position (relative to other objects) in the 2.5D model. Other available online maps, for example Google Maps or OpenStreetMap, do not contain these details. As this system aimed to be able to map the driver's focus to any object in the environment, the described approach was selected over the other available solutions.

In order to get an exact 3D model of the vehicle, its interior was scanned with a commercially available depth sensor. More specifically, the Structure Sensor from Occipital[3] and the Asus Xtion PRO[4] were used for this purpose (see Figure 5.14).

[3] http://structure.io/

[4] https://www.asus.com/de/3D-Sensor/Xtion_PRO/

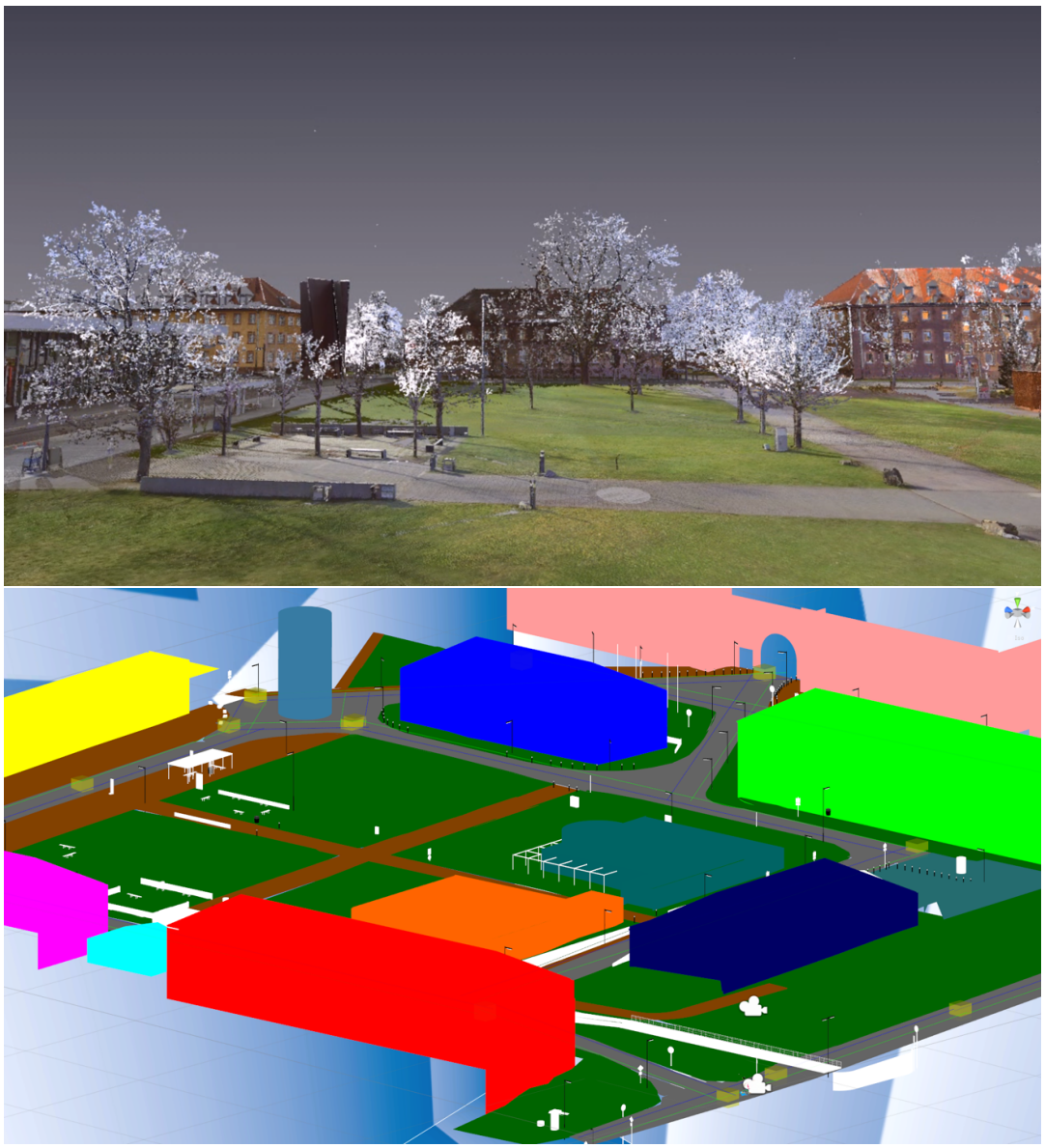

Figure 5.13: *Top:* A screenshot of the 3D point cloud of the Saarland University campus. *Bottom:* The 3D polygon model of the campus based on the point cloud data.

These pieces of hardware were used in conjunction with surface reconstruction software Skanect[5]. The resulting polygon-based model was then colored with respect to different regions in the vehicle (see Figure 5.14 (bottom)). These colors were then used to distinguish each specific area in the vehicle from another area while perform-

[5]http://skanect.occipital.com/

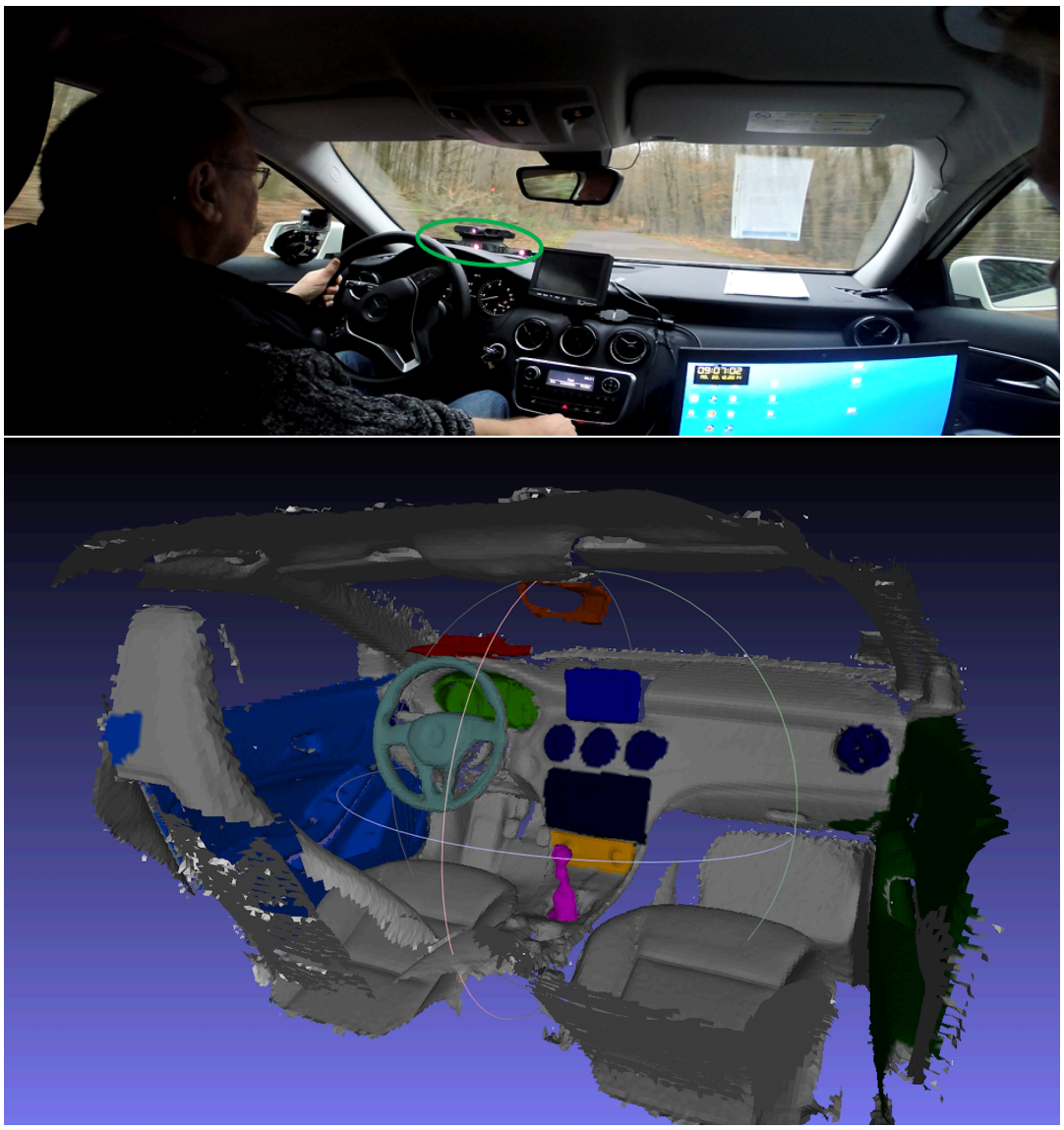

Figure 5.14: An example car setup and the corresponding annotated 3D model. Different colors represent different annotated regions.

ing ray casting depending on the direction of the user's focus. As the hardware of this system (eye-tracker and head-tracker) were part of the scanned environment in the vehicle, their positions were known. Relative to their coordinates, the information about the position and pose of the driver's head in the vehicle was reported by the software. The whole interior model of the vehicle was then placed in the 2.5D model of the outside environment. The position of the vehicle was updated with a 10 Hz GPS positioning system together with a map-matching algorithm in real time.

The information from the developed software about the driver's attention was then used together with the vehicle's interior model and the 2.5D model of the outside environment to identify which object in which environment (outside or inside the vehicle) was in the focus of the driver in real time. For this purpose, an algorithm was developed. The algorithm received two rays which were cast in the directions of the driver's left and right eye. Figure 6.2 depicts these rays. An algorithm was then developed to analyze these rays and identify the object in focus. This algorithm is described in the following. The intersecting objects (for the user's focus) can be divided generally into two categories: objects inside the vehicle and objects outside of the vehicle. This algorithm, however does not differentiate between these categories. The focus analysis is performed across all the objects inside and outside the vehicle.

The algorithm proposed for the spatial reference resolution is as follows: First, the start and end times of the spatial reference (τ_1, τ_2), the position of the referee during this time ($\psi(\rho)$), and the absolute orientation of the vehicle (o) are determined. Then an empty mapping for the objects in focus and their scores is defined ($score$). The 3D models of outside and inside environments are also imported (M_o and M_i respectively). Then all of the gaze vectors of the left eye and the right eye are analyzed in a time frame of Δt. As the whole algorithm is implemented in the Unity 3D game engine, this Δt is defined by the frame update of this engine (normally 20 to 25 milliseconds). In each time frame, a ray is cast from each eye in the direction of the user's focus, considering position and orientation of the car. Then the intersection of these two rays is defined in the 3D space ($p_{intersect}$). If these two rays are not intersecting in a point, the middle point of the line which defined the smallest distance between two rays is taken (beside the positions of left and right eye). Finally, a vector is defined from the middle point of the user's eyes, directly above the nose ($headposition$), through the intersection point ($p_{intersect}$). A ray is then cast in the direction of this vector in the M_o and M_i models. The colliding object is then logged (b). For this object a score Λ_{t_i} is noted (Λ_{t_i} can just be 1)[6]. Then over the interaction time (τ_1, τ_2) the scores of the different objects are accumulated. The output of the algorithm is the object with the highest score.

Following is the analysis of the time complexity of this algorithm. The time complexity of the first seven actions (acquiring $\tau_1, \tau_2, \psi(\rho), o$ and building empty mapping $score$ and also importing the two models M_o and M_i) is $\mathcal{O}(1)$[7]. Regarding the for loop **for** $(t_i = \tau_1;\ t_i \leq \tau_2;\ \Delta t)$ the time complexity is $\mathcal{O}(n)$, as it goes one time through all Δts. The time complexity for building v_{left} and v_{right} is $\mathcal{O}(1)$. In order to find the intersection between them, the time complexity is $\mathcal{O}(1)$, as it is simple

[6]The idea of Λ_{t_i} is that it makes possible to weight different part of the interaction differntly than other parts. For example, it is possible to have more weights for the objects which are in the focus of the user during middle part of the interaction. Λ_{t_i} is a time function.

[7]Importing M_o and M_i is done once when the whole program starts. The loading does not have any effect on run time of the algorithm.

mathematical calculation. The ray-cast operation however, is time consuming. As this operation is performed by Unity 3D internally, we do not know how it is done in details. Regardless, we can here consider a worst case of $\mathcal{O}(n^2)$. Accumulating the scores for each colliding object has the time complexity of $\mathcal{O}(n)$. The final sort has the worst case time complexity of $\mathcal{O}(n^2)$.

Reference Resolution Algorithm for Detailed 3D Environments

$\tau_1, \tau_2 \leftarrow \sigma$
$\psi(\rho) \leftarrow \sigma$
$o \leftarrow \epsilon$
$score \leftarrow nil$
$M_o \leftarrow 3D\ \ outside\ \ Model$
$M_i \leftarrow 3D\ \ inside\ \ Model$
for $(t_i = \tau_1;\ t_i \leq \tau_2;\ \Delta t)$ **do**
  $v_{left} \leftarrow vector(\psi(\rho), o, t_i, eye_{left})$
  $v_{right} \leftarrow vector(\psi(\rho), o, t_i, eye_{right})$
  $p_{intersect} \leftarrow intersect(v_{left}, v_{right})$
  $b \leftarrow raycast(p_{intersect}, headposition, M_o, M_i)$
  $score_b \leftarrow score_b + \Lambda_{t_i}$
end for
$score \leftarrow sort(all\ score_b)$
return $score$

Now we can add up the time complexity of each block of algorithm together. As mentioned, first block has $\mathcal{O}(1)$. For the second block we have $\mathcal{O}(n) \cdot (\mathcal{O}(1) + \mathcal{O}(n^2) + \mathcal{O}(n))$. As it is mentioned, for the sort we have $\mathcal{O}(n^2)$. So the whole time complexity is: $\mathcal{O}(1) + \mathcal{O}(n^3) + \mathcal{O}(n^2) = \mathcal{O}(n^3)$.

5.1.4 Reference Resolution Algorithm using Focus-of-Attention in Dynamic Mixed Reality Environments

This section presents how the procedure described in 5.1.3 was adapted for use in indoor applications. The scenario for realizing the algorithms described here is human-

robot interaction. In this use-case, many parts of the environment are moving. The moving objects include different parts of the robot and also the objects which are manipulated by the robot or humans. Each of these objects is moving with different speeds. For the modeling of the static objects in the environment, the same techniques are used as the indoor modeling methods described in Section 5.1.3. In order to integrate the dynamic elements, these objects are modeled in 3D. These 3D models are then integrated in the environment as individual polygon meshes. The 3D position and orientation of these elements are permanently synchronized in the 3D model by acquiring the data from the real world. As the same game engine as in 5.1.3 is used, the updates of the dynamic models are performed in each frame.

When referring to objects in mixed-reality environments, resolving the target object is more complicated. As in mixed-reality the real environment is augmented with virtual information, the vector which represents the user's gaze collides both with real objects and the virtual information which is overlaid on the real object. In order to solve this issue, in this research, the algorithm is designed under the assumption that the virtual augmentation of the real environment is located on the right 3D spatial position. That means that the virtual information is not just presented as a layer on the real environment, but rather it is integrated into this environment on the correct position and with the correct orientation. This way, if some virtual information is laid over a real object and the user looks at them, the virtual information (in the front) is selected by the algorithm as the target object. The developed algorithm acts on a virtual representation of a mixed-reality environment in which both the real and virtual elements are treated equally based on their spatial position in the 3D space.

The algorithm for reference resolution using focus-of-attention in dynamic mixed reality environments is as follows: First, the start and end times of the spatial reference (τ_1, τ_2), the current position of the referee $(\psi(\rho))$, and the absolute position and orientation of the objects in the virtual environment and real environment are determined ($OBV(p_{virtualobject}, o_{virtualobject})$ and $OBR(p_{realobject}, o_{realobject})$ respectively). One empty mapping for the objects in focus and their scores is defined as well($score$). Then, in each cycle of the game engine $OBV(p_{virtualobject}, o_{virtualobject})$ and $OBR(p_{realobject}, o_{realobject})$ are updated. In this case in the mixed-reality environment there is one gaze point available, thus a vector which represents the user's focus is used for identifying the object in focus. For this purpose, a ray is cast in the direction of this vector and the colliding object is logged (b). For this object a score Λ_b is noted. Then over the interaction time (τ_1, τ_2) the scores of the different objects are accumulated. The output of the algorithm is the list of objects with their scores.

Following is the analysis of the time complexity of this algorithm. The time complexity of the first actions until the loop is $\mathcal{O}(1)$. The time complexity of the loop is $\mathcal{O}(n)$. Updating the position and orientation of the real and virtual objects have

Reference Resolution Algorithm using Focus-of-Attention in Dynamic Mixed Reality Environments

$\tau_1, \tau_2 \leftarrow \sigma$
$\psi(\rho) \leftarrow \sigma$
$OBR(p_{realobject}, o_{realobject}) \leftarrow \epsilon$
$OBV(p_{virtualobject}, o_{virtualobject}) \leftarrow \epsilon$
$B, score \leftarrow nil$
for $(t_i = \tau_1;\ t_i \leq \tau_2;\ \Delta t)$ **do**
  $update(OBR(p_{realobject}, o_{realobject}))$
  $update(OBV(p_{virtualobject}, o_{virtualobject}))$
  $b \leftarrow raycast(\psi(\rho), o, t_i, gaze)$
  $score_b \leftarrow score_b + \Lambda_{t_i}$
end for
$score \leftarrow sort(all\ score_b)$
return $score$

$\mathcal{O}(1)$ as time complexity. As mentioned before, the time complexity for the ray-casting is considered to be $\mathcal{O}(n^2)$ (worst case). Accumulating the scores of each object has the time complexity of $\mathcal{O}(n)$. Finally, the sort function has a worst case time complexity of $\mathcal{O}(n^2)$. The overall time complexity of the whole algorithm is: $\mathcal{O}(1) + \mathcal{O}(n) \cdot (\mathcal{O}(1) + \mathcal{O}(n^2) + \mathcal{O}(n)) + \mathcal{O}(n^2) = \mathcal{O}(n^3)$.

5.2 Head Orientation as an Indicator of User Identification for Multi-Party Interaction

In this scenario users wear smart-glasses which in turn monitor their head-orientation. The algorithm presented here calculates the correlation between the yaw and pitch values of the user's smart-glass with yaw and pitch values delivered from the depth sensor concerning the head-orientation of the users in the scene. If the correlation value is higher than a specific threshold and also if there is significant difference between the one specific correlation value and others, the two IDs from the smart-glass and the depth sensor are matched. In other words, this algorithm identifies that these two users (one who wears the smart-glass, and one who is detected by the depth camera) are the same person. This way the information collected by the depth-sensor can be related to the person wearing the specific wearable. This

algorithm is designed for multi-party interaction scenarios.

Details of the algorithm is provided in the following. cr_T and cr_D are the threshold for the correlation and the mentioned delta difference between the correlations respectively. These two values are pre-defined with constants. $\Upsilon array_{depth-sensor}$ and $\Upsilon array_{smart-glass}$ are two mappings for arrays which contain depth sensor and smart glass information respectively (Υ always indicates a mapping). This mapping is between the ID of the user in the corresponding device and the array of the sensor data which is received from that device. $\Upsilon matchingFIN_{sensorID-glassIDs}$ is the final matching between these two IDs. The whole program runs in a *loop*. At the beginning of the loop $\Upsilon matchingTMP_{sensorID-glassIDs}$ and $\Upsilon matching_{sensorID-glassIDs}$ are initialized which can both be considered temporal mappings between the user IDs of the depth sensor and smart glass. Then, the sensor data is read from both devices and added to the already existing data from the previous runs of the loop. Now a for loop goes through all currently active IDs of the depth sensor and for each of these entries iterates over all of the active IDs of the available smart glasses. Then, for each pair it calculates the correlation coefficient and stores these values in $\Upsilon matchingTMP_{sensorID-glassIDs}$. The values which are stored in this mapping are ID of the user in the depth sensor device (as key), and the pair of ID of the glass device and the correlation coefficient (as value). The next for loop goes through each of these entries and sort them considering the value of the correlation coefficient (cr). Finally, the cr of the top element of the sorted mapping is compared with cr_T and in case it is bigger the difference between this correlation coefficient and the next one is compared to cr_D[8]. If this difference is also bigger than cr_D, the matching is added to $\Upsilon matching_{sensorID-glassIDs}$. After running though all IDs of the depth sensor, the result is returned by setting $\Upsilon matchingFIN_{sensorID-glassIDs} = \Upsilon matching_{sensorID-glassIDs}$.

For computing the correlation coefficient, the Pearson Correlation Coefficient (PCC) is used [Pearson, 1895]. PCC is widely used as a measure of linear correlation between two random variables X and Y. The outcome of PCC is between +1 and -1. If the outcome is 0, it means that no correlation between the variables exist. +1 indicates full positive linear correlation and -1 indicates total negative linear correlation. So when computing PCC between two variables, the closer the outcome is to $|1|$, the stronger is the relationship between X and Y. The PCC r is computed by the following formula:

$$r(X,Y) = \frac{C(X,Y)}{\sqrt{V(X)V(Y)}}$$

In the formula above, the functions C and V are the covariance and the variance functions respectively. The variance V indicates how far random values are spread

[8] If the mapping consists of just one entry, the second condition is ignored.

Head Orientation as an Indicator of User Identification for Multi-Party Interaction

$cr_T = K_1$
$cr_D = K_2$
$\Upsilon array_{depth-sensor} \leftarrow nil$
$\Upsilon array_{smart-glass} \leftarrow nil$
$\Upsilon matchingFIN_{sensorID-glassIDs} \leftarrow nil$

loop
$\Upsilon matchingTMP_{sensorID-glassIDs} \leftarrow nil$
$\Upsilon matching_{sensorID-glassIDs} \leftarrow nil$

$\Upsilon user_{depth-sensor} = \Upsilon(p_{user}, o_{body}, o_{head}, id_{depth-sensor}) \leftarrow depthSensor$
$\Upsilon user_{smart-glass} = \Upsilon(o_{head}, id_{smart-glass}) \leftarrow smartGlasses$
$\Upsilon array_{depth-sensor} = \Upsilon array_{depth-sensor} \cup \Upsilon user_{depth-sensor}$
$\Upsilon array_{smart-glass} = \Upsilon array_{smart-glass} \cup \Upsilon user_{smart-glass}$

for $(user_{depth-sensor\ in\ \Upsilon(user_{depth-sensor})})$ **do**
for $(user_{smart-glass}\ in\ \Upsilon(user_{smart-glass})$ **do**
$cr_{tmp} = corollate(array_{depth-sensor}(o_{head}), array_{smart-glass}(o_{head}))$
$\Upsilon matchingTMP_{IDs} \cup (id_{depth-sensor}, id_{smart-glass}, cr_{tmp})$
end for
end for

for $(id_{depth-sensor}\ in\ \Upsilon matching_{IDs})$ **do**
$sort(id_{depth-sensor}, id_{smart-glass}, cr)$
if $(cr[0] > cr_T) and ((cr[0] - cr[1]) > cr_D)$ **then**
$\Upsilon matching_{sensorID-glassIDs} \cup (id_{depth-sensor}, id_{smart-glass})$
end if
end for

$\Upsilon matchingFIN_{sensorID-glassIDs} = \Upsilon matching_{sensorID-glassIDs}$

end loop

from their mean. The covariance C shows how two random variables are related to each other. C and V are calculated as follow.

$$C(X,Y) = E[(X - E[X])(Y - E[Y])]$$

$$V(X) = E[(X - E[X])^2]$$

The expected value E for the random variable X can be interpreted as the long-run average for an statistical process. It is calculated as

$$E[X] = \sum_{\omega \in \Omega} X(\omega).P(\omega).$$

In the equation above, X is the random variable and $P(\omega_i)$ is the probability of ω_i. So if we reconsider the formula for PCC, we will have

$$r(X,Y) = \frac{E[(X - E[X])(Y - E[Y])]}{\sqrt{(X - E[X])^2(Y - E[Y])^2}}.$$

As in our use case we have finite sets of data incoming from two sources, we have to rewrite the formula given above. We consider the data samples for the depth sensor and smart glass to be $x = \{x_1, ..., x_n\}$ and $y = \{y_1, ..., y_n\}$ respectively. For these two data sets the formula will be

$$r(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}.$$

In the equation above $\overline{x}$ and $\overline{y}$ denote the mean of samples x and y respectively. This equation considers the two series to be temporally synchronized, however in practice this cannot be guaranteed over a long period. In order to overcome this problem one of the inputs can be shifted backwards. Fo a delay d, the PCC can be modified as follows.

$$r(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_{i-d} - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_{i-d} - \overline{y})^2}}.$$

Following is the analysis of the time complexity of the whole algorithm. The time complexity of the first actions until the main *loop* is $\mathcal{O}(1)$. As *loop* is the main loop of the program, it is not considered to have effect on the time complexity. For the

first four actions in the loop, the time complexity is $\mathcal{O}(1)$. The time complexity of the adding the sensor data to the mapping array is $\mathcal{O}(n)$. Regarding the two nested loops the time complexity is $\mathcal{O}(n^2)$ multiplied by the addition of the complexity of the PCC and matching function. So we will have $\mathcal{O}(n^2) \cdot (\mathcal{O}(n) + \mathcal{O}(n)) = \mathcal{O}(n^3)$. The time complexity of the PCC is considered to be $\mathcal{O}(n)$. For the last loop we have $\mathcal{O}(n) \cdot (\mathcal{O}(n^2) + \mathcal{O}(n)) = \mathcal{O}(n^3)$, when we consider the time complexity of the sort function to be in worst case $\mathcal{O}(n^2)$. So the overall time complexity of the algorithm is: $\mathcal{O}(1) + \mathcal{O}(n^3) + \mathcal{O}(n^3) = \mathcal{O}(n^3)$.

5.3 Peripheral View Algorithm

The model introduced in 4.2 gives the relation between the visibility of an object in periphery and three variables: vertical eccentricity, horizontal eccentricity, and angular size of the object. The angular size here is different than solid angle of an object described in 2.1.8. However, if considered together with the vertical eccentricity and horizontal eccentricity, it is possible to build a mapping from these three variables to a solid angle value. This way it is possible to build two categories of peripheral view algorithms:

- A peripheral view algorithm which is based on the solid angle of an object in the peripheral field and an approximation of visual acuity at that position.
- A peripheral view algorithm which uses the model described in 4.2 together with the required inputs: vertical eccentricity, horizontal eccentricity, and angular size of small fields in our periphery.

In the following one algorithm from the first category and two algorithms from the second category are described in detail.

5.3.1 Algorithm for Peripheral Vision Analysis based on Solid Angle and Hatada Model

In Section 2.1.7 the Hatada model is illustrated while giving a detailed review of defined regions of the visual field with their associated capabilities. The concept of solid angle is also examined in 2.1.8. In order to describe the current peripheral view calculation model, it proceeds as follows. Using the concepts of the Hatada model and solid angle, it is explained how to calculate intersections between each visual field defined by the Hatada model and an arbitrary object of the 3D environment. An

algorithmic realization of this model involves discrete aspects of solid angle computation where the calculations for complex objects is reduced to primitive types by dividing an object's outline into rectangular patches. Finally, the obtained measures are related to the concept of visual acuity to determine our final visibility measure.

5.3.1.1 Solid Angle of an Object's Bounding Box

In the following, the solid angle of the bounding box of an arbitrary object in a 3D environment is calculated. The position of the user's head is assumed to state the origin of a Cartesian coordinate system. At distance d from the user we define a "collision plane" which is perpendicular to the user's head orientation and whose origin is determined by the user's colliding gaze vector. The 2D projection of the 3D oriented bounding box of the target object onto this plane is assumed to have width a and height b.

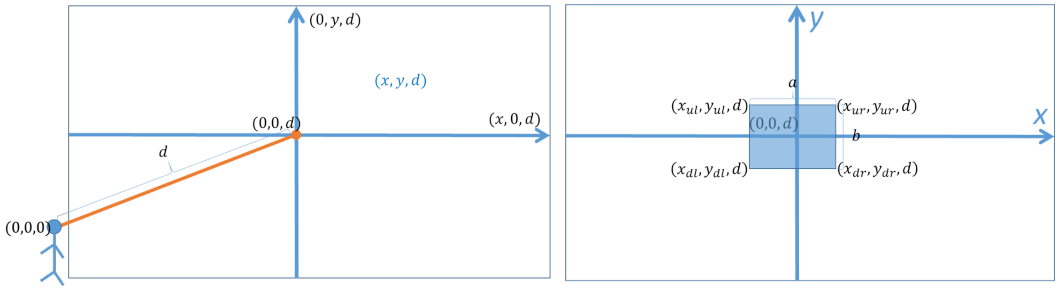

Figure 5.15: *Left:* User's head position as center of the 3D environment. The gaze direction yields the origin of the collision plane. *Right:* The bounding box of an object is located exactly at the center of gaze.

Figure 5.15 depicts the case in which the user is looking at the center of the target object. If the bounding box of the object is positioned exactly at the center of the collision plane, the solid angle of this rectangular shape can be calculated according to Mathar's formula [Mathar, 2014]:

$$\Omega(a,b,d) = 4 \arccos\left(\frac{\sqrt{1+(\frac{a}{2d})^2+(\frac{b}{2d})^2}}{\sqrt{1+(\frac{a}{2d})^2}\sqrt{1+(\frac{b}{2d})^2}}\right) . \tag{5.1}$$

Note that this is the case when the user is looking exactly at the middle of the object. However, this formula can not be applied in its current form if the bounding box of the object is not positioned exactly at this point. This is the case when the observer's gaze is focused at another point in the environment, which shifts the target object

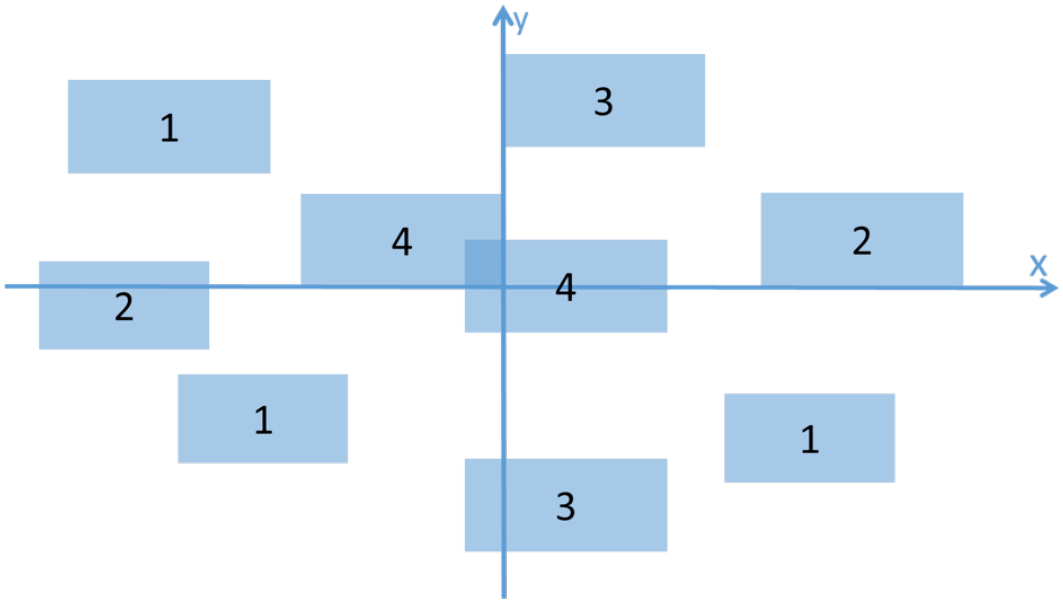

Figure 5.16: Four different categories for the position of an object's bounding box.

to peripheral regions of the user's visual field. The target object can have different positions relative to local coordinate axes of the collision plane.

As illustrated by Figure 5.16, four different positions of the object's bounding box with respect to the local coordinate axes of the collision plane can be categorized:

1. The rectangle does not intersect with any axes.
2. The rectangle intersects only with the x-axis.
3. The rectangle intersects only with the y-axis.
4. The rectangle intersects with both axes.

As it is described by Mathar in [Mathar, 2014], the solid angle of each of these rectangles can be calculated as follows. The first step is to build a big rectangle which is centered at the middle of the coordinate system and contains the smaller rectangle on its edge. The next step is to calculate the solid angle of the big rectangle using the provided formula. Then, the big rectangle is divided into different stripes horizontally and vertically (see Figure 5.17). Again, the solid angle of each stripe is calculated with the provided formula, as each of these stripes is centered at the origin of the coordinate system. The final step is to calculate the solid angle of the target

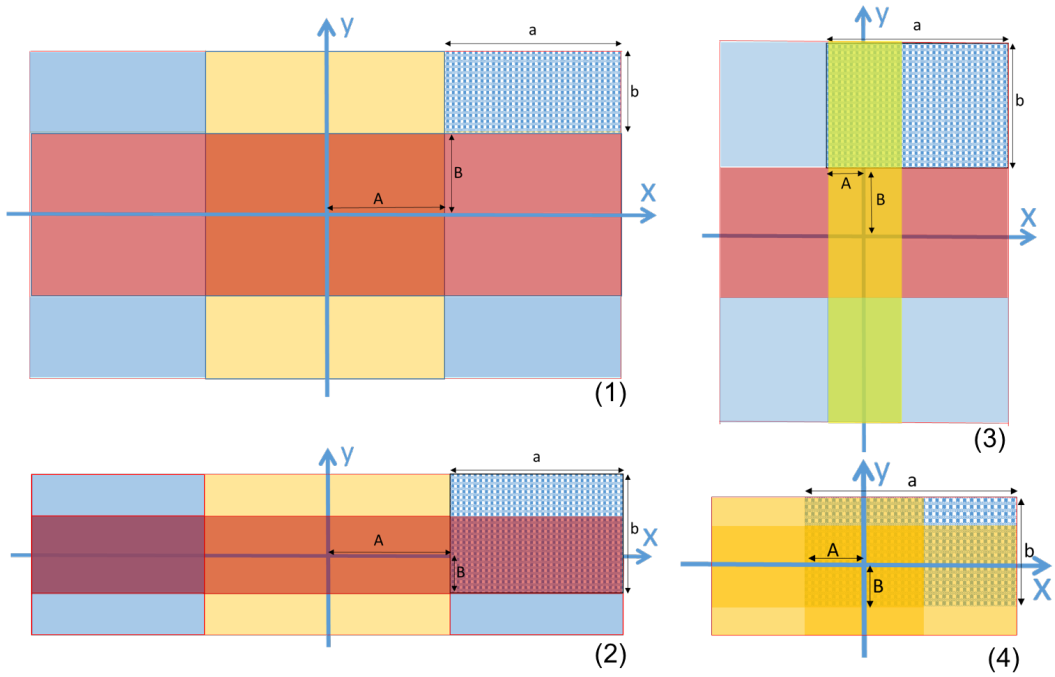

Figure 5.17: Calculating the solid angle of the object by combining the different values in each category. Numbers represent the corresponding categories.

bounding box by combining (adding or subtracting) the solid angles calculated before. This combining step differs depending on each of the four categories mentioned above. In the following, the calculation of the solid angle for each of these categories is described (see Figure 5.17).

Figure 5.17 depicts the case in which the bounding box of the object does not have any intersection point with the x-axis or the y-axis (Category 1). The patterned rectangle with width a and height b represents the 2D projected bounding box of the object on the collision plane. The closest distance of this rectangle to the y and x-axis is represented by A and B respectively. A rectangle with width $2(a + A)$ and height $2(b+B)$ is depicted. The middle of this rectangle is placed exactly at the point $(0, 0, d)$. Thus, Mathar's formula can be used to calculate the solid angle of the big rectangle. Finally, the solid angle of the patterned rectangle (target bounding box) is calculated as follows:

$$\Omega = \frac{1}{4}\left(\Omega(2(a+A), 2(b+B), d) - \Omega(2(a+A), 2B, d)\right.$$
$$\left. - \Omega(2A, 2(b+B), d) + \Omega(2A, 2B, d)\right) \ .$$

The solid angle of the bounding box for other categories is calculated in a similar

fashion, however with other combinations depending on each category. Regarding the second category, in which the rectangle intersects only with the x-axis, the calculating formula is the following:[9]

$$\begin{aligned}\Omega = \frac{1}{4} \left(\Omega(2(a+A), 2(b-B), d) + \Omega(2(a+A), 2B, d)\right. \\ \left. - \Omega(2A, 2(b-B), d) - \Omega(2A, 2B, d)\right) \ .\end{aligned}$$

In the case of the third category, in which the rectangle intersects only with the y-axis, the formula reads as[10]

$$\begin{aligned}\Omega = \frac{1}{4} \left(\Omega(2(a-A), 2(b+B), d) + \Omega(2A, 2(b+B), d)\right. \\ \left. - \Omega(2(a-A), 2B, d) - \Omega(2A, 2B, d)\right) \ .\end{aligned}$$

For the fourth category, in which the rectangle intersects with both axes, we get[11]:

$$\begin{aligned}\Omega = \frac{1}{4} \left(\Omega(2(a-A), 2(b-B), d) + \Omega(2(a-A), 2B, d)\right. \\ \left. + \Omega(2A, 2(b-B), d) + \Omega(2A, 2B, d)\right) \ .\end{aligned}$$

5.3.1.2 Intersection of Solid Angles

In order to determine the intersection of the solid angles of the different visual field regions and the solid angle of the bounding box of the object, the ellipses of the visual fields have to be projected onto the collision plane[12]. If we assume a visual field with parameters α and β at distance d to the user to be centered at the origin of the local coordinate system having its major and minor axes aligned with the local axes, a mathematical representation is given by

$$\left(\frac{x}{\tan(\alpha)\, d}\right)^2 + \left(\frac{y}{\tan(\beta)\, d}\right)^2 \leq 1 \ ,$$

where a 2D point $(x, y)^\top$ that fulfills this equation is located inside the associated elliptic region. It should be noted that this condition has to be adapted for each half ellipse with its corresponding angles given by the Hatada model. The projected 2D

[9] B is always selected to be $B \leq \frac{b}{2}$

[10] A is always selected to be $A \leq \frac{a}{2}$

[11] A and B are always selected to be $A \leq \frac{a}{2}$ and $B \leq \frac{b}{2}$

[12] Please refer to Appendix (A.1) for the calculation of the solid angles of the different visual field regions.

bounding box of the object is divided into very small rectangles (in the Cartesian coordinate system). For the center of each of these sample patches, we use the mentioned intersection condition in order to determine the corresponding region of the HVF the sample is located in. In the case of an intersection, we calculate the solid angle of the intersecting rectangles while accumulating them. The result of this accumulation is also a solid angle which determines the amount of intersection between the corresponding visual field and the bounding box of the object.

5.3.1.3 Integration of Visual Acuity

Visual acuity states an important factor influencing the visibility. As Gross mentions in [Gross, 1994], the visual acuity V is not equal on all parts of the retina; its maximum lies in the center of the retina and decreases towards the periphery. V can be measured by the inverse of the minimum visual angle α achievable when detecting a target. A numerical approximation is given by

$$V(\alpha) \approx c_1 + \frac{c_2}{\alpha + c_3} \ ,$$

with constants

$$c_1 = -0.0323 \ , \ c_2 = 0.0524 \ , \ c_3 = 0.0507 \ .$$

Note that so far this provides a one-dimensional measure defined on the unit interval for visual angles ranging from $0°$, yielding a maximized acuity for the very center of gaze, to $90°$, where visual observations become impossible. As already pointed out by Gross, it makes sense for each solid angle area covered on the retina to be weighted with the corresponding value of the visual acuity. We transfer these ideas to our model by weighting the fraction of the solid angle an object subtends in a visual field region with the visual acuity associated with this region, which is determined from half of the corresponding total vertical opening angle. This way, combining the concepts of solid angle and visual acuity, we end up with a two-dimensional visibility measure assigning weights of an appropriate scale to arbitrary objects in 3D space with respect to their position in the HVF.

5.3.1.4 Calculation

As mentioned before, the different steps for determining visibility are the following: First, we determine the solid angle of each peripheral field and the target object. Then, we calculate the intersection of these two solid angles. Finally, the different small fractions of this intersection will be weighted according to the visual acuity in different regions of the HVF. As a result, we get the total solid angle of the object,

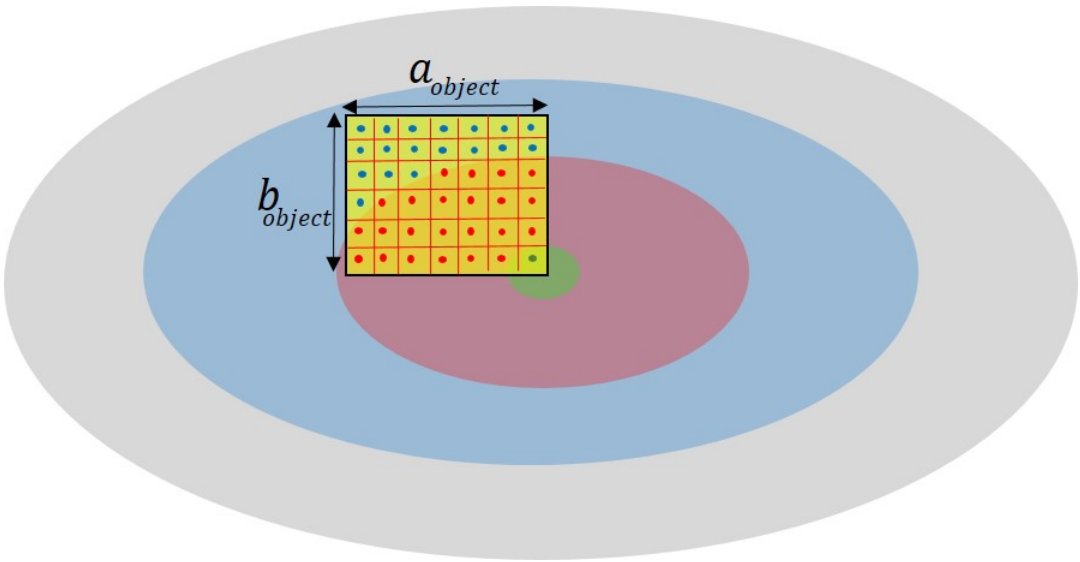

Figure 5.18: The solid angle of the intersection is calculated by accumulating the solid angles of the small rectangles of the object's bounding box which intersect with respective regions of the visual field.

the total visibility of the object based on visual acuity, and also the percentage of each of these values in each visual field (see Figure 5.18). This information can be used in different intelligent user interfaces to analyze the visibility of the objects in the environment and also to design interaction based on the visibility of the objects.

5.3.1.5 Algorithm

The algorithm proposed for calculating the visibility of an object in periphery based on its solid angle and visual acuity (considering the Hatada model) is as follows: First, one should define the target objects in the scene (Γ). Then a collision plane ($plane_{collision}$) should be placed between the observer and the closest target object in the scene. This plane should be perpendicular to the observer's head. Then, all of the target objects should be projected on this collision plane. By performing a min-max analysis, the 2D bounding box of all of the target objects on the collision plane should be determined. Then, one should fragment these 2D bounding boxes into equal patches ($\gamma_{patches}$). This can be performed by dividing the bounding box horizontally and vertically in equal parts. Then, from the position of the observer, the described "ellipses" of the Hatada model are projected onto the collision plane. Using the calculations described in Section 5.3.1.1, the solid angle of each patch of each target object is calculated (Ω_{patch}). Then, the Hatada region of the current

Algorithm for Peripheral Vision Analysis based on Solid Angle and Hatada Model

$\Gamma \leftarrow config$
$\psi(\rho) \leftarrow \sigma$
$p_{collision} \leftarrow nil$
$plane_{collision} \leftarrow nil$

$d = \infty$
for $(\gamma \; in \; \Gamma)$ **do**
  $d_{temp} = distance(\gamma, \psi(\rho))$
  if $(d_{temp} < d)$ **then**
    $p_{collision} = p(\gamma)$
    $d = d_{temp}$
  end if
end for

$plane_{collision} = plane(p_{collision}, \perp \psi(\rho))$
$n_{patches} \leftarrow config$
for $(\gamma \; in \; \Gamma)$ **do**
  $\phi_{temp} = project(\gamma, plane_{collision})$
  $\phi = minmax(\phi_{temp})$
  $\gamma_{patches} = divide(\phi, n_{patches})$
end for

$\Psi \leftarrow hatada$
for $(\psi \; in \; \Psi)$ **do**
  $project(\psi, plane_{collision})$
end for

$object\_visibility\_set \leftarrow nil$
for $(\gamma \; in \; \Gamma)$ **do**
  $\gamma_{visibility} \leftarrow nil$
  for $(p_{patch} \; in \; \gamma_{patches})$ **do**
    $\Omega_{patch} = solidangle(p_{patch})$
    $\psi = region(p_{patch})$
    $p_{patch\_visibility} = \Theta_{acuity}(\Omega_{patch}, acuity(\psi))$
    $\gamma_{visibility} = \gamma_{visibility} + p_{patch\_visibility}$
  end for
  $object\_visibility\_set = object\_visibility\_set \cup \gamma_{visibility}$
end for

$return \;\; object\_visibility\_set$

patch is determined (ψ). Based on ψ and by using the Formula of Gross mentioned in [Gross, 1994] (see 5.3.1.3), the visual acuity concerning this patch is computed. The visibility of each patch is calculated by function Θ_{acuity} which combines the computed solid angle and visual acuity. Finally, for each target object these visibilities are accumulated. The algorithm returns the visibility of all defined target objects in the scene.

Following is the analysis of the time complexity of this algorithm. The time complexity of the first actions until the first for loop is $\mathcal{O}(1)$. The time complexity of the for loop is $\mathcal{O}(n)$, as it goes through all the objects in the environment and calculates their distance to the observer. The time complexity for building the collision plane and reading the determined patch numbers from the config file is $\mathcal{O}(1)$. Next for loop, goes through all the objects in the environment, projects their bounding box into the collision plane, and divides these bounding boxes into smaller rectangles. In this division just the coordinates of the small boxes are noted, hence, the time complexity of this for loop is $\mathcal{O}(n)$. The time complexity of projecting Hatada ellipses is $\mathcal{O}(1)$, as these are maximally 5 ellipses. The final big operation in this algorithm is the final two nested loops. These nested loops have, however, the time complexity of $\mathcal{O}(n)$. That is due to the fact that the number of iteration of inner loop is predefined. This number is defined by the number of patches in the config file. The time complexity of the actions performed in the inner loop is $\mathcal{O}(1)$. By adding all the the time complexities, we reach the total of $\mathcal{O}(n)$.

5.3.2 Algorithms for Peripheral Vision Analysis based on Eccentricity and Size of the Object

Here two algorithms based on the peripheral model introduced in 4.2.7 are introduced. The main idea here is to divide the whole environment into smaller segments and calculate the visibility of these segments with the given formula. There are, however, two techniques for dividing the environment and calculating the visibility. Each of these techniques are described separately in the following.

5.3.2.1 Projection-Based Visibility Calculation

As in the algorithm described before, here the objects of the environment are projected into a collision plane. Then, using the model introduced by Barfield et al. in [Barfield et al., 1995], the visible part of this plane is divided into 12851 small rectangles (71 vertical degrees (from -10 to 60) and 181 horizontal degrees (from 0 to 180)). Then going though all these rectangles and adding up their visibility by considering the model introduced in 4.2.7, the total visibility of each object is

calculated. It is also possible to sample the plane in bigger rectangles and have less sampling rate.

Projection-Based Algorithm for Peripheral Vision Analysis based on Eccentricity

---

$\Gamma \leftarrow config$
$\psi(\rho) \leftarrow \sigma$
$p_{collision} \leftarrow nil$
$plane_{collision} \leftarrow nil$

$d = \infty$
for $(\gamma \; in \; \Gamma)$ **do**
  $d_{temp} = distance(\gamma, \psi(\rho))$
  if $(d_{temp} < d)$ **then**
    $p_{collision} = p(\gamma)$
    $d = d_{temp}$
  end if
end for

$plane_{collision} = plane(p_{collision}, \perp \psi(\rho))$
for $(\gamma \; in \; \Gamma)$ **do**
  $\phi_{temp} = project(\gamma, plane_{collision})$
  $\phi = minmax(\phi_{temp})$
end for

$n_{samples} \leftarrow config$
$object\_visibility\_set \leftarrow nil$
for $(sample \; in \; n_{samples})$ **do**
  $\omega_{sample\_visibility} = visibility(sample, \psi(\rho))$
  $\gamma = region(sample, \psi(\rho))$
  $\gamma_{visibility} = \omega_{sample\_visibility}$
  $object\_visibility\_set = object\_visibility\_set \cup \gamma_{visibility}$
end for

$return \;\; object\_visibility\_set$

Concerning the time complexity analysis, as until the last for loop the operations are similar to the last algorithm, the time complexity until this loop is $\mathcal{O}(n)$. The last for loop goes through all the samples which is defined by the config file. If the sampling rate is one degree, this number will be 12851. By changing the sampling rate, this number changes accordingly. As the visibility of each sample can be pre-computed, the line $\omega_{sample\_visibility} = visibility(sample, \psi(\rho))$, performs just lookups for these values. So here no calculations are involved and the time complexity is $\mathcal{O}(1)$. The line $\gamma = region(sample, \psi(\rho))$, checks in which bounding box of an object, each sample is located. In $object\_visibility\_set = object\_visibility\_set \cup \gamma_{visibility}$, the computed fraction of the visibility is added to the corresponding object. The time complexity of these two actions is $\mathcal{O}(n)$. As the time complexity of the for loop itself is $\mathcal{O}(1)$, the time complexity for the whole algorithm will be $\mathcal{O}(n)$.

5.3.2.2 Visibility Calculation with Ray Casting

Here, using the same pattern described before, 12851 rays are cast (71 vertical degrees (from -10 to 60) and 181 horizontal degrees (from 0 to 180)), and the colliding object are recorded. Using the model described in 4.2.7, the visibility of each small field for every object in the environment is calculated and accumulated. It is also possible to change the sampling rate of the ray casting.

Algorithm for Peripheral Vision Analysis based on Eccentricity and Ray Casting

---

$\psi(\rho) \leftarrow \sigma$
$n_{samples} \leftarrow config$
$object\_visibility\_set \leftarrow nil$

for $(sample\ in\ n_{samples})$ **do**
  $\omega_{sample\_visibility} = visibility(sample, \psi(\rho))$
  $\gamma = raycast(sample, \psi(\rho))$
  $\gamma_{visibility} = \omega_{sample\_visibility}$
  $object\_visibility\_set = object\_visibility\_set \cup \gamma_{visibility}$
end for

$return\ \ object\_visibility\_set$

The only for loop in this algorithm has the time complexity of $\mathcal{O}(1)$, as it goes through the number of samples which are pre-defined in the configuration. If sampling rate is one degree, the total number of samples will be 12851. This number changes by changing the sampling rate, however, the sampling rate is fix during the execution of the program. As mentioned before, $\omega_{sample\_visibility} = visibility(sample, \psi(\rho))$ is just a lookup command, so the time complexity of it will be $\mathcal{O}(1)$. The ray casting action, however, has a worst case time complexity of $\mathcal{O}(n^2)$. The time complexity of $object\_visibility\_set = object\_visibility\_set \cup \gamma_{visibility}$ is linear as it goes though all registered object. Hence, the overall time complexity of this algorithm is $\mathcal{O}(1) + \mathcal{O}(1) \cdot (\mathcal{O}(1) + \mathcal{O}(n^2) + \mathcal{O}(1) + \mathcal{O}(n)) = \mathcal{O}(n^2)$.

6 TIFoA: Toolkit for Eye-Based Interaction and Analysis

TIFoA is a toolkit for using the users's visual attention in order to build different interaction and analysis applications in mobile and mixed-reality scenarios. This toolkit consists of three different tools: XVIUS, SFT and PVA. These tools offer the possibility to design and implement different prototypes for human-environment interaction using the user's focus-of-attention. In the following, each of these tools is described briefly. Then, each section of the current chapter provides more details about each tool.

XVIUS is designed to implement gaze-based interaction applications. It can be used to build an application for interacting with the outside environment from within a moving vehicle. It can also be used to build indoor applications for gaze-based human robot collaboration. Regarding environment modeling, this tool provides three possibilities: 2D outdoor, 3D outdoor, and 3D indoor. It is possible with this tool to design and implement interactive multimodal applications based on 2D maps for advanced driver assistant systems. It is also possible to integrate a more precise 3D environment into XVIUS or to connect a driving simulator. This tool can also be used as an analysis platform. These feature expansions provide more precision in human-environment interaction and also the possibility for thorough analysis in the real world and simulator. Regarding human-robot interaction, it provides the possibility to track and integrate dynamic objects from the real world into the developed applications. Moreover, XVIUS offers the possibility to develop the same application for Augmented-Reality, Virtual-Reality, on mobile devices or a PC with minimal changes. All of these scenarios use the user's direct gaze for the interaction. The PVA tool uses this gaze together with the model of the environment to perform a peripheral view analysis in real time. Based on this platform, it is possible to develop applications which react to the objects in the peripheral vision of the user based on the visibility of the objects. The SFT tool utilizes the user's head direction and body pose for building different features of human-environment interaction. These fea-

tures include for example interacting with different objects in the environment (such as display, car, etc.) in a car workshop scenario.

The three mentioned tools use different algorithms described in Chapter 5 together with different software and hardware to enable developers to build different applications for human-environment interaction. A number of these applications are described in Chapter 7. The remainder of this chapter describes each of the mentioned tools in detail.

6.1 XVIUS: Human Environment Interaction Tool

6.1.1 2D Outdoor

This part of XVIUS provides the developers with the possibility of implementing applications for interaction with the outside environment from within a moving vehicle. This tool integrates the algorithm described in Section 5.1.2 in its core implementation. Furthermore, it also consists of several hardware and software components which enable system developers to implement applications based on this tool. Figure 6.1 depicts the overall architecture which consists of the XVIUS modules together with the Audio Manager, Dialog Manager and the Speaker Classification module (which are not part of this tool). The main software modules of the XVIUS are listed below.

- Vehicle Positioning
- Map Matching
- Environment Model
- Eye-Tracking
- Object-Recognition
- Interface to the Dialog Management and HMI

The Vehicle Positioning module monitors the raw Latitude and Longitude of the vehicle on the street and forwards this information to the Map Matching component. This module in turn uses the provided information in the Environment Model to match the raw Latitude and Longitude to the valid coordinate on the street of the model. All these procedures are performed in real time, several times a second. As the user sees an object in the environment, he has the opportunity to push the button

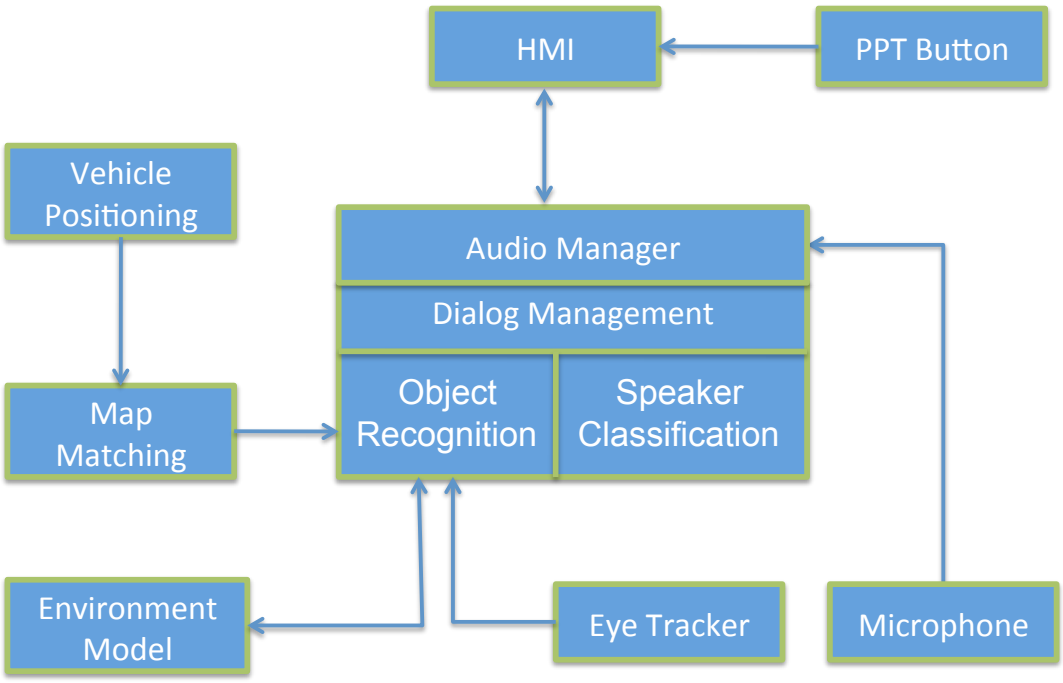

Figure 6.1: The system architecture and the information flow between different modules.

and ask the vehicle for more information about the object in the focus. The user's speech signal is forwarded to the Audio Manager. This component determines two timestamps in the speech of the user: the beginning time and the end time. These two timestamps are forwarded to the Dialog Management and from there to the Object Recognition module.

The Object Recognition module contains most of the core algorithm described in Section 5.1.2. This component receives the two mentioned timestamps regarding the beginning and end of the user's speech. Furthermore, it receives the positions of the vehicle together with the associated timestamps. The Eye-Tracking module also sends information regarding the gaze direction of the user with 20 Hz on a regular basis. Using the named algorithm, the Object Recognition module takes the gaze data coming from the eye-tracker and evaluates it on each vehicle position between the time window of the user's utterance. This way it detects the potential target buildings and delivers them as a list to the Dialog Management component. This list also contains information and pictures of these buildings. The Dialog Manager either picks up the building at the top of the list, or combines this information with the contextual information in the user's utterance. As a result, information on one building is sent to the HMI component. The module, in turn, shows the picture of the target building to the user (on the GUI) and describes the building via the

text-to-speech engine provided. Section 7.1 contains applications which have been implemented based on this tool.

6.1.2 3D Outdoor

This section describes a chain of tools and algorithms of XVIUS which have the following features:

- It is based on the Unity3D game engine. Thus it can contain environments with much more detail.
- It can perform detailed analysis on 3D eye-tracking applications for outdoors.
- It is possible to connect it to the driving simulator for joint analysis in real and virtual environments.

Different software and possibilities were explored to find a suitable method for real-time analysis in detailed dynamic environments. Finally, the Unity 3D game engine[1] was used due to its performance and versatility. This game engine offers the possibility for integration of big environments which are highly detailed (see Figure 6.2). Meanwhile, it provides a physics engine and other tools to perform real-time analysis in these large and detailed environments. It also allows for integrating or updating dynamic objects in real-time.

Architecture
The 3D Outdoor module is divided into three logical parts: Information Source, Processing Engine, and Visualization Module. Information Source contains all modules for capturing information in real-time or other components which already contain data about the environment, for example 3D model of the environment. In addition, Information Source includes sensors, which capture different features of the user's head. These features include eye-gaze, head-pose, and various facial expressions of the user. Processing Engine, another logical part of XVIUS, receives this information and computes the relationship between the different sources. For example, if the driver scowls at the navigation system, the Processing Engine uses the dashboard model and the gaze direction of the user to calculate the object, which is in sight of the driver (in this case the navigation system). It also uses the facial expression of the driver to find out about her current emotional state. In this scenario, Processing Engine infers that the driver is not happy while looking at the navigation system. This information is then forwarded to the third party application interface and the Visualization Module. The third party application interface prepares this information

[1] https://unity3d.com/

Figure 6.2: An example scene for interaction in outdoor environments containing three target objects. The environment model, the vehicle model and the direction of the focus are all integrated in one environment. The rays represent the direction of the eye gaze and head pose.

for other modules outside XVIUS, for example a Dialog System. It is up to these applications to act upon receiving this information. Processing Engine also includes an analysis part, which accumulates data over time and tries to infer useful information in a specific period of time. This information can be, for example, the number of times or the overall time, which the driver has looked at the speedometer.

The Visualization Module illustrates the output of the Processing Engine by using various modules. Each of these modules is responsible for showing the information

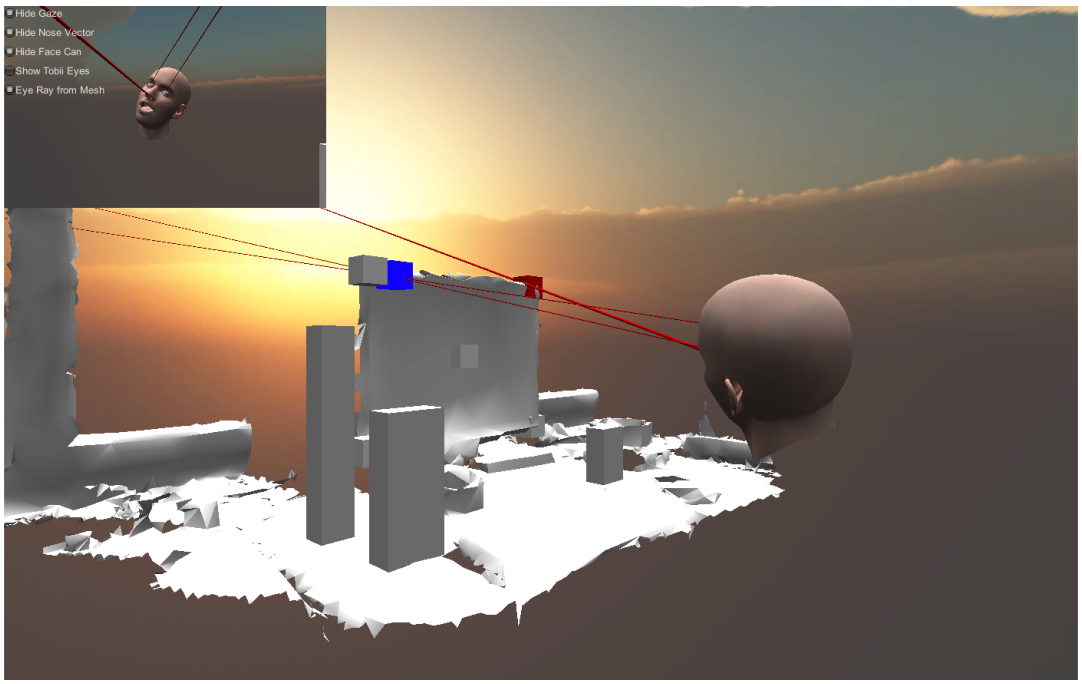

Figure 6.3: A virtual character representing the user. The extracted information is eye-gaze, head-pose, and facial expression. The colored cubes on the display show the intersection-point of the vectors with the environment.

in an appropriate format depending on the type of the data. For instance, if the analysis is based on a two-dimensional map of the city and the eye-gaze of the driver, a visualization module can produce a two-dimensional heat-map of the city illustrating the places which have been most in the focus of the driver on the map (specific buildings or monuments). In another use case, if the analysis is based on the three-dimensional model of the vehicle's interior and the driver's focus-of-attention, a three-dimensional model of the driver's floating focus can be produced using a game engine. In the following, technical details about the functionality of XVIUS for 3D outdoor, regarding eye-tracking and head-tracking, is revealed.

Figure 6.3 shows a virtual character representing the user, who is looking at a display with an open mouth in a scanned 3D environment (without texture). The projected line from the nose shows the direction of the head. The projected lines from the eyes show the direction of each eye respectively. The colored cubes on the screen show the intersection of the vectors with the environment. Figure 6.4 shows the head from the front view in more detail.

Eye-Gaze

In order to calculate the gaze direction of the user, the EyeX Controller from

Figure 6.4: Head-pose and gaze direction of the user.

Tobii was used. For this purpose, no video analysis is performed, instead a three-dimensional vector is calculated from the user's eye toward the environment. This gives us the flexibility to calculate the intersecting object independent of their position in the vehicle or outside the vehicle.

Head-Pose and Facial Expressions
In order to compute the head-pose of the user in conjunction with the facial expression, a depth cameras from Asus or Primesense was used together with the faceshift software. This combination provides the opportunity to position the head of the driver in the car and also to get the values yaw, pitch, and roll with regard to the head-pose. In addition, various raw data about the facial expression of the driver can be provided in real-time (see Figures 6.3 and 6.4).

Environment Modeling
For the in-vehicle environment modeling, the Kinect depth camera was used in conjunction with surface reconstruction software Skanect from Occipital. This way it was possible to create a color 3D model of the car's interior. This model had the required precision for different interaction and analysis applications. This 3D mesh was then annotated in a 3D editor. In the annotation phase, every object in the car interior was marked. This annotated model was then used as a basis for further processing. Depending on the head-pose and the eye-gaze of the user, a ray was cast in this model to reveal the object, which was in the focus-of-attention. Figure 5.11 illustrates the described process. For modeling the outside environment, two different approaches were used. In the first approach a very precise 3D scanner was used to scan the environment. A 3D mesh was then produced based on the point clouds of the scan. In the second approach, an online map was imported in the 3D game engine. In both techniques the same ray-casting method was used to reveal the

object which was in the focus of the observer.

6.1.3 3D Indoor

This part of XVIUS is mainly used to develop applications for human-robot collaboration. It offers the possibility to develop the same application for Augmented-Reality, Virtual-Reality, on a mobile device or PC with minimal changes. It implements the algorithm described in 5.1.4. Figure 6.5 depicts the different software and hardware modules of the tool for human-robot collaboration scenarios. The different components integrated in this tool are the following:

- Object Tracking
- Robot Head Controller
- Robot Left Arm Controller
- Robot Right Arm Controller
- TECS Server
- Tablet AR Module
- Wearable Eye-Tracker Module
- Wearable Gesture Control Module
- Mobile VR Module
- Stationary VR Module with integrated Gesture Control

The Object Tracking module tracks objects in the environment via a camera and a software module. The Robot Head Controller shows different feedbacks (to the user) on the display mounted on the head of the robot. The controllers for the left and right arms of the robot monitor the incoming messages regarding these two components and react to them, and they also send the pose of each robot hand to the other modules. The TECS[2] server is used as a communication framework between all the presented modules. As described before, TECS is a middleware that combines different ways of communication and it is publicly available. Tablet AR module is used for showing spatial information as a virtual overlay on the real environment. This tablet is calibrated in a way that this information is synchronized spatially between

[2]http://tecs.dfki.de/

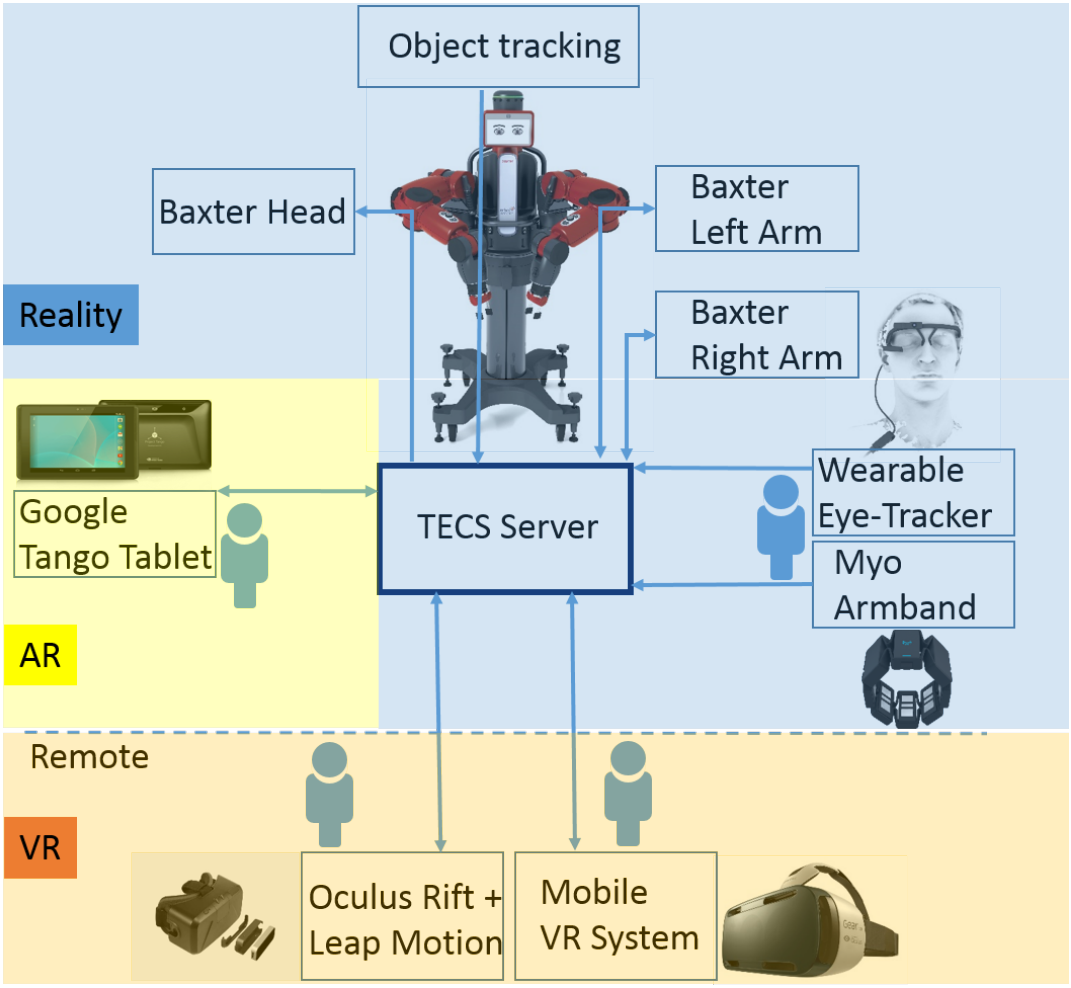

Figure 6.5: Different components including the local and the remote systems.

all VR and AR modules. The Wearable Eye-Tracker module sends gaze information from the user to the other components in real-time. The Wearable Gesture Control Module sends information regarding the user's gestures to the other modules. The Mobile VR Module sends (and receives) the direction of the user's focus to all other modules, furthermore it sends the 3D position of the user in the environment. The Stationary VR Module also sends (and receives) this information; in addition it sends information regarding the posture of the user's hand in the virtual environment. So it is possible for the user to point at an object in the virtual environment and the other participants, who are physically somewhere else, can see the direction and the posture of the user's hand.

The arrows in Figure 6.5 represent the information flow between different components. For synchronizing the communication between the different modules, the

publish-subscribe functionality of TECS was used. Each module in the system publishes its own data, and when required, subscribes for data from other modules. The Baxter[3] robot sends its body pose and receives pick-and-place commands. The remote users subscribe to this body pose to get the robot posture in real-time and send the pick-and-place commands. The objects on the table are tracked via a camera which is mounted on the head of the robot and the Vuforia[4] system. This module sends the object pose in real-time. The eye-tracker and the armband[5] of the local user also send their data in real-time. This information is also used by the remote users to update their environment in real-time. The AR tablet subscribes for the information from the VR users to get their head pose and position together with the information about the hand posture in real-time.

6.2 PVA: Peripheral View Analysis Tool

Peripheral View Analysis (PVA) is a comprehensive tool for analyzing the visibility of the objects which are not in the focus of an observer. This tool is implemented as a Unity3D module which can be used as an analysis instrument or as a core for various interactive applications. Figure 6.6 depicts the input data and output information of this tool. The input data is the same as mentioned before: position and the bounding box of the objects in the environment, position of the user's head/eye and also a 3D vector which represents the user's gaze direction. For each defined target object, the PVA tool has one output in each cycle of the game engine. The output of the PVA tool for each object can be divided into 6 categories (see Figure 6.6). In the following, these categories and each output are described in detail.

- Category 1
  - Timestamp: time stamp of the obtained results.
  - ObjectWidth: horizontal angular size of the target object from the user's view point (in Degrees).
  - ObjectHeight: vertical angular size of the target object from the user's view point (in Degrees).
- Category 2
  - AngleToGaze: The angle (eccentricity) between the center of the target object and the direction of the user's gaze (in Degrees).

[3]Baxter is a collaborative two-armed robot with an animated face built by Rethink Robotics (http://www.rethinkrobotics.com/).

[4]https://www.vuforia.com/

[5]https://www.myo.com/

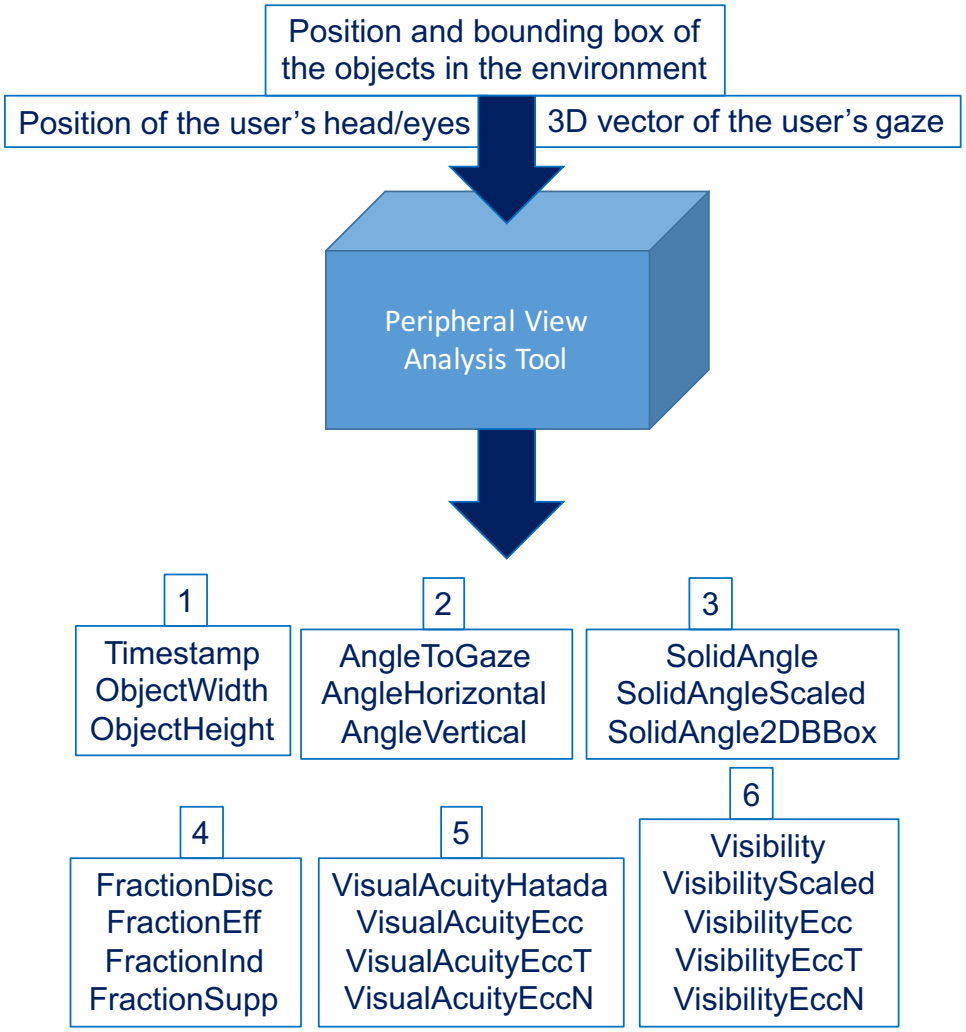

Figure 6.6: Input and output of the Peripheral View Analysis Tool. The numbers define the output categories.

  - AngleHorizontal: Horizontal fraction of the AngleToGaze (horizontal eccentricity in Degrees).
  - AngleVertical: Vertical fraction of the AngleToGaze (vertical eccentricity in Degrees).

- Category 3

  - SolidAngle: Accumulated result for the solid angle of visible parts of the target object.

  - SolidAngleScaled: Scaled representation of the solid angle on an alternative scale (e.g. log). This scale should be provided by the user.
  - SolidAngle2DBBox: Total solid angle of the target object's outline (without considering occlusions).

- Category 4
  - FractionDisc: Percentage of the target object located in the discriminatory region of the Hatada model.
  - FractionEff: Percentage of the target object located in the effective region of the Hatada model.
  - FractionInd: Percentage of the target object located in the induced region of the Hatada model.
  - FractionSupp: Percentage of the target object located in the supplementary region of the Hatada model.

- Category 5
  - VisualAcuityHatada: Visual acuity of the visual field region where the center of the target object is located.
  - VisualAcuityEcc: Visual acuity based on the eccentricity of the target object.
  - VisualAcuityEccT: Visual acuity based on the horizontal eccentricity of the target object.
  - VisualAcuityEccN: Visual acuity based on the vertical eccentricity of the target object.

- Category 6
  - Visibility: Overall visibility measure of the target object considering the Hatada model. Here the solid angle is weighted with the visual acuity of the involved regions of the Hatada HVF.
  - VisibilityScaled: Scaled representation of the Visibility on an alternative scale (e.g. log). This scale should be provided by the user.
  - VisibilityEcc: Overall visibility measure based on the eccentricity.
  - VisibilityEccT: Overall visibility measure based on the horizontal eccentricity.
  - VisibilityEccN: Overall visibility measure based on the vertical eccentricity.

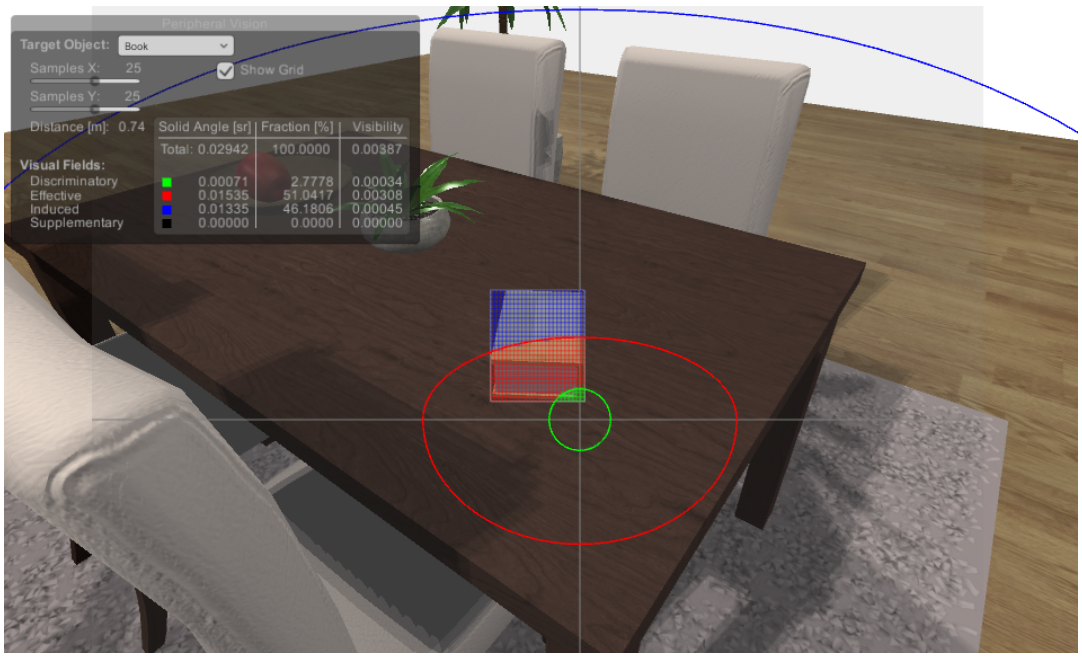

Figure 6.7: The web application provides an interactive visualization of the presented peripheral view calculation model. A control panel reveals underlying numerics and allows for choosing the target object and the number of samples. The gaze position is simulated by mouse input.

Based on this tool, several applications have been developed which will be described in Section 7.4. In addition, an interactive web tool was developed in order for researchers and system developers to test the tool before using it (see Figure 6.7). The interactive web tool shows a room with different objects inside. The user can freely move and look around in the room. The mouse cursor represents the center of the user's gaze. The borders of different peripheral fields are visualized with different colors around the gaze center. The dimensions of these fields are selected according to the peripheral field model of Hatada et al. [Hatada et al., 1980]. The green circle shows the discriminatory visual field, and the red and blue regions represent the effective and the induced visual field, respectively. The supplementary visual field is illustrated with a black line, however, due to the camera features of the game engine, it is not always visible. In the control panel on the upper left side of the scene, it is possible to select a target object from a selection of the objects available in the scene. In this case, the bounding box of the target object is calculated and visualized. The intersecting solid angles of the bounding box with different peripheral fields are also calculated. If the "Show Grid" checkmark is selected, the different intersecting regions are also visualized. In the control panel, it is possible to choose the number of the samples for dividing the object's bounding box in horizontal and vertical direc-

tions. In this respect, a higher number of samples leads to an increased accuracy of the calculations. The control panel also shows the following calculated information: the distance between the object and the observer, the total solid angle of the object, the percentage of the object visible to the user (interesting in the case of occlusion) and the visibility of the object considering the solid angle of the object and the visual acuity of the visual field. Each of the last three values is also calculated separately for each of the named fields of the Hatada model. Occlusion is also considered, and all of the calculations are performed for the visible part of the object.

As the user navigates through the scene or changes the gaze direction, the different measures are calculated and displayed in real-time. All of the calculations are based on the Algorithm for Peripheral Vision Analysis based on Solid Angle and the Hatada Model (see 5.3.1). The interactive website makes it possible to examine the changes in the different values as the position or the gaze direction of the user changes. Figure 6.7 shows a screenshot of this application. The web application can be reached under the address: http://madmacs.dfki.de/?page_id=544.

6.3 SFT: Spatial Fusion Tool

The Spatial Fusion Tool (SFT) utilizes the user's position, head direction and body pose for building different features of human-environment interaction. These features include for example extracting the user's focus-of-attention for interacting with different objects in the environment, detecting user groups and identification of the users in the environment. On the hardware side, the SFT includes one Kinect V2 depth camera and several Google Glasses. Figure 6.8 depicts the information flow between these hardware components and the four developed software modules. Each of the software modules allows for developing applications for multi-party human-environment interaction. The algorithm described in 5.2 (Head Orientation as an Indicator of User Identification for Multi-Party Interaction) is part of this tool.

The Gesture Detection module gets the input from the depth camera and extracts a set of hand gestures. When the user performs a gesture, this module recognizes it and records this information together with the ID of the user for the further use. This is a unique ID for all users who are seen by the depth camera. The Focus-of-Attention module has two components, one runs on the Google Glass, and the other one runs on the PC which is connected to the depth camera. The latter one determines the position of the users in the environment, their posture and also their head-pose. The module on the Google Glass also evaluates the video input of the glass to identify pre-defined objects. For this purpose, the glass uses the Augmented

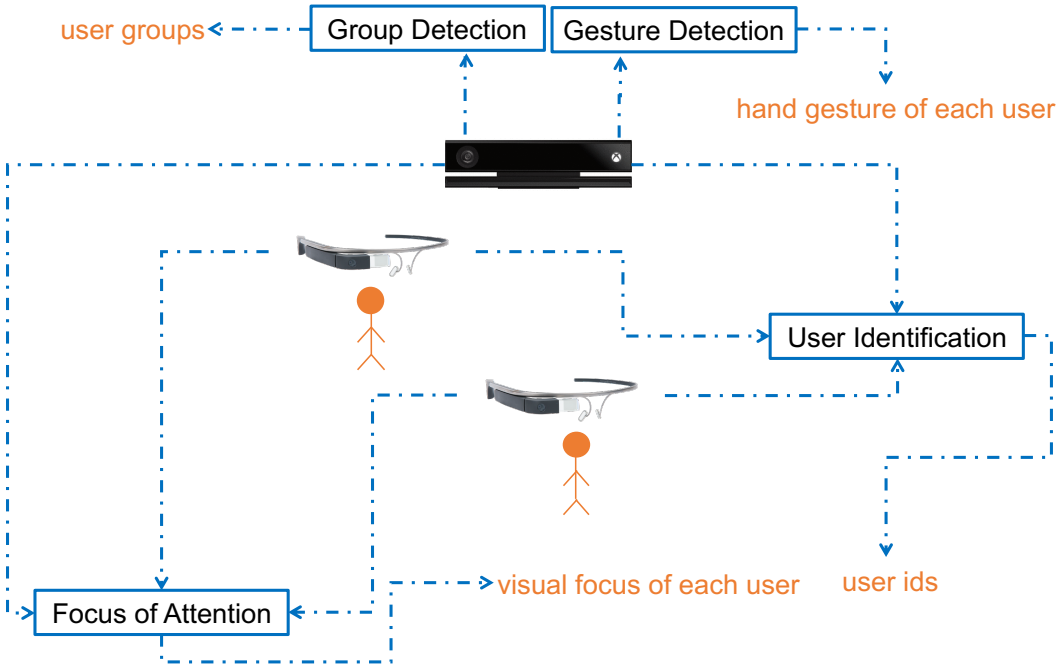

Figure 6.8: The information flow between the hardware and software modules of the SFT.

Reality module Vuforia[6]. For this purpose, several images of each specific object should be integrated in this module. As the glass camera registers the objects, an ID is sent to the PC module indicating the object in the focus.

The PC component of the Focus-of-Attention module uses the incoming information from the depth camera for further analysis of the focus of attention. From the user's nose a ray is cast in the direction of user's head. As this module includes the 3D model of the environment, it can determine the object which collides with the focus-ray in each frame of the game engine. The collided object is then registered. If there is no incoming input from the Google Glass, this object is evaluated as the object in focus. However, if the glass detects an object it will be registered as the object in focus and override the information from the depth camera. This is due to the fact that the glass module only registers objects and sends their ID when the user is very close to these objects and they are in the direct sight of the user. This circumstance is used to weight the information coming from the glass higher than the information which is calculated in the PC module based on the head pose of the user. The Focus-of-Attention component can perform this procedure for several people in the scene, each wearing one glass.

The User Identification module implements the "Head Orientation as an Indicator

[6]https://www.vuforia.com/

of User Identification for Multi-Party Interaction" algorithm described in Section 5.2, in order to identify users in the scene. Each user in the scene wears a Google Glass which includes the user's ID. The glass sends information regarding yaw and pitch values of the user's head orientation together with his ID to the main User Identification module. This module also gets information about the yaw and pitch values of the users' head orientation from the depth camera. By applying the function described in the algorithm, a correlation is found between these two data channels. This way it is possible to link the unknown ID of the Kinect user to the known ID of the Google Glass user. From this point, the Kinect user will not be unknown anymore. This way, any other incoming information from this user (speech etc.) can be linked to the actual person ID acquired from the personal Google Glass of the user. The Group Detection module uses the orientation of the users' body together with their position to detect the potential group of users. For this purpose, this module only uses the input coming from the Kinect sensor.

The different components of the SFT work together and other third party tools to enable developing human-environment interaction applications. With this tool it is possible, for example, to identify the different groups, their members, the visual focus-of-attention of the members, and their gesture if they perform one. In Section 7.3, applications which have been developed based on this tool are described.

7 Interaction Paradigms Realized with TIFoA

The three tools of the TIFoA toolkit were used to implement various use-cases. These use-cases cover scenarios like interacting with the outside environment from within a moving vehicle, to scenarios which implement novel interfaces for human-robot interaction. This chapter provides an overview over the implemented use-cases.

7.1 Spatial Interaction and in-Vehicle Infotainment

The systems[1] described here are applications which have been developed with the XVIUS tool. This tool integrates the algorithms described in Section 5.1 in its core implementation. Furthermore, it also consists of several hardware and software components which enable system developers to implement applications based on this tool.

7.1.1 2D-Based Applications

7.1.1.1 Interaction in Real-Life Traffic Scenario

The implemented scenario is the following. Two travelers rent a car and start their trip. When they enter the car, they are greeted by the system with a spoken welcome message from the rental service ”Welcome to AVIS. What is your destination?”. In the implemented demonstrator, speech input is always initiated by a push-to-talk

[1]Funded by the German Federal Ministry of Education and Research in the projects CARMINA (grant number 01IW08004), MADMACS (grant number 01IW14003), and SC_MeMo (grant number 01IS12050).

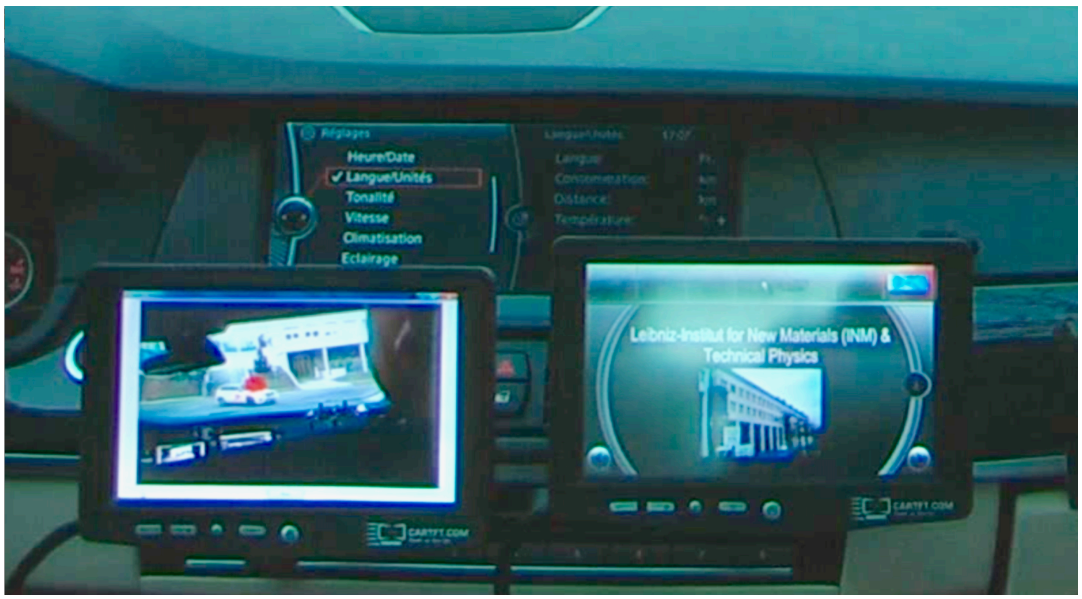

Figure 7.1: The user selects a building by looking at it and saying ”What’s this building?”. The system provides the user with information about it.

button. This button can be integrated into the vehicle and can be accessed via a vehicle bus such as CAN. In the implemented prototype, it is an external push-button mounted to the dashboard. After activating the button and saying the destination, the system performs the route planning and switches to the navigation screen, taking the role of a traditional navigation system, showing the current position, route and guidance. While the user (who is the front seat passenger in the demonstration) is navigating through the city, he can use voice commands to obtain information about the environment. With his eyes on the object of interest, he might, for example, ask ”What is this building?” or ”What is this object?” for particularly interesting buildings or landmarks. This works for every visible object surrounding the vehicle at each moment during the trip. Using an eye tracker and a physical object reference resolution technology that is described in Section 6.1.1, the system determines the target object. It can then retrieve information from an internal database or the Internet and present it to the user. In the implemented showcase, the information is accompanied by a close-up picture of the building (see Figure 7.1) shown on the center stack screen.

For purposes of visualization, the set-up in the demonstration vehicle features a video camera that is mounted to the co-driver’s seat. It covers approximately the same area as the eye tracker. It can be used to match the live road scene with the eye gazes and overlay them, one on top of the other in real time. The result is a video stream where an observer can see the regions where the passenger was looking (indicated by red

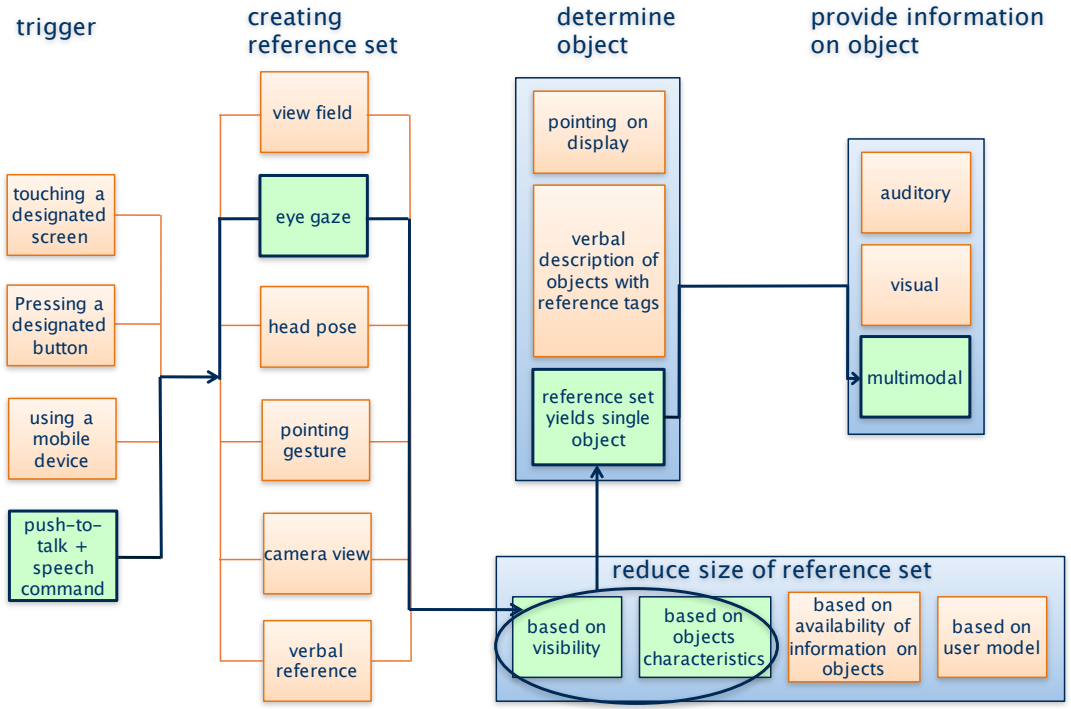

Figure 7.2: The four phases of eye-based reference resolution. The dark line shows the implemented control flow in the presented system.

circles). This stream is shown on the left screen in Figure 7.1. It can serve as a means of validation for building references.

Referring to the outside environment from within a moving vehicle can be performed using different modalities such as EYE GAZE, HEAD POSE, POINTING GESTURE, CAMERA VIEW and the user's VIEW FIELD [Moniri and Müller, 2012]. The interaction of the user with the outside environment can be divided into four phases: 1) Trigger the system. Here, we use a designated button. 2) Create a reference set for the potential target objects for which we use a combination of eye gaze and head pose tracking. Note that information contained in the user's utterance could also be useful here ("what is the blue building?"). However, it is not used in the demonstration system. 3) Reduce the size of the reference set by taking into account the visibility of the buildings. 4) Provide information about the first best hypothesis. Figure 7.2 illustrates this process highlighting the design choices of the system described here and showing alternatives at each stage.

Figure 7.3 shows the system setup in the vehicle. The user interacts with the system via speech, eye gaze or the touch screen. The video screen display shows a live view of the outside environment which is complemented with the eye gaze of the user. The main interaction of the user with the system is through speech (push to talk button)

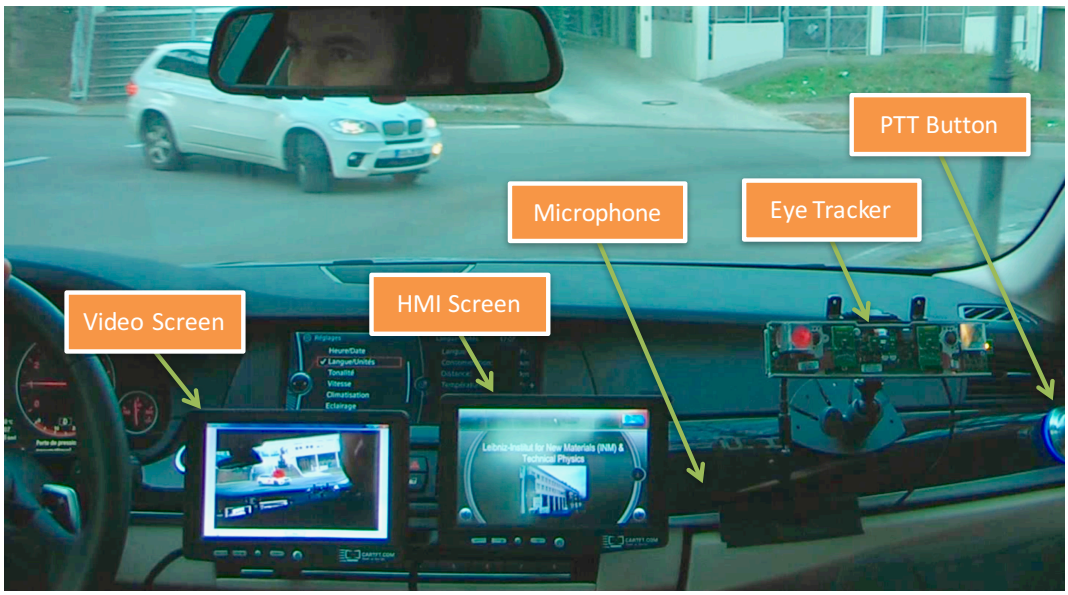

Figure 7.3: The system setup in the vehicle of the XVIUS application for urban environments.

and eye gaze. This module combines all this information with the speech time of the user to extract the target object. The reason for this interval extraction is the displacement of the vehicle from beginning till end of this speech interval as it is described in the algorithm ”Reference Resolution Algorithm for Applications with 2D Environment Models” (see section Section 5.1.2).

7.1.1.2 Interaction in Video-Based Setup

Similar to the application described in Section 7.1.1.1 for interaction in real-life traffic scenarios, the XVIUS tool was used to implement a system for video-based interaction. This system was used for testing new multimodal interaction technologies before they are introduced as a product. Figure 7.4 shows the setup inside and outside of the vehicle. In this setup the user is able to sit in the vehicle and see a video on the windshield. This video is taken from an ego-perspective view of a car driving through the campus. The user has the possibility to look at the buildings in the video and ask the car about more information. Using the XVIUS tool, the system performs a reference resolution and provides information to the user about the building in their focus.

Figure 7.5 depicts the information flow and various modules of the system. The video used for this application is annotated with GPS information. This way, it is clear at each frame in the video which latitude and longitude information should be delivered

Figure 7.4: The system setup for the video-based application.

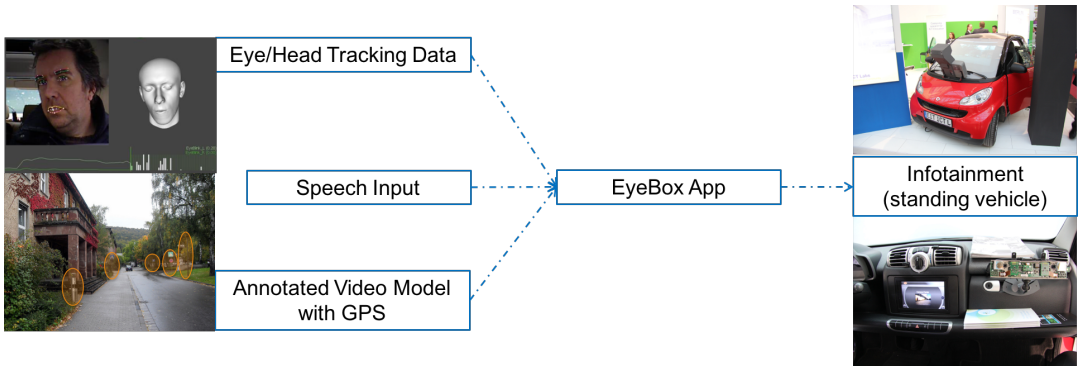

Figure 7.5: The information flow between modules of the video-based application.

to the application. The speech input is used as described in Section 7.1.1.1 for interaction in a real-life traffic scenario. This information together with the Eye-Tracking data is forwarded to the Eyebox application. This application in turn uses the algorithm "Reference Resolution Algorithm for Applications with 2D Environment Models" (see Section 5.1.2) to detect the building in focus. Information regarding this object is then forwarded to an infotainment GUI. This GUI then provides this information in an audio/visual form to the user (see Figures 7.4 and 7.5).

7.1.2 3D-based Applications

In this system the environment reconstruction and data processing are fundamentally different. Instead of using a spatial database, the Unity 3D Game Engine is used. This solution provides the flexibility and the granularity that is needed to refer to smaller objects in the environment (like for example traffic signs or city furniture). For this purpose, the precise 3D models of the environment and the vehicle (interior) play a major role in this system. In the following, first the use cases that can be realized

with the 3D-based approach are described, then some examples of the implemented applications are provided.

7.1.2.1 Use Cases

XVIUS can be applied to various automotive scenarios. It can be used in applications containing analysis or interaction use cases, both in simulated environments and in the real world. The use cases of this platform can be divided into four categories that are shown in Table 7.6.

Application / Domain	Analysis	Interaction
Simulator	Automated focus-of-attention analysis in simulator including off-screen set-up.	Extending the simulation set-up to include intuitive interaction.
Real World	Analyzing everyday traffic regarding the driver's focus-of-attention.	Adding intuitive interaction to current existing interaction types.

Figure 7.6: Various use case categories for XVIUS.

Analysis in Simulator Setup
In many evaluation studies, the analysis (for example for eye tracking) of a driving simulator run has to be performed manually by reviewing the recorded video and annotating different segments of the video. Using XVIUS, it is possible to perform this analysis automatically. The platform provides the ability to scan the whole simulator setup and perform an automatic analysis to discover, for example, how long the user has looked at relevant objects in the setup. Figure 7.7 shows an example simulator setup. In this example, XVIUS is integrated with OpenDS [Math et al., 2013]. OpenDS is a cross-platform, open-source driving simulation software. XVIUS can act as a plug-in for this driving simulator in order to widen its possibilities for user interaction and analysis both inside the virtual world and also in the physical simulator setup.

As the user drives through the virtual environment in OpenDS, the eye tracker component of the system sends the eye gaze data to this driving simulator. OpenDS then uses this information to perform a real-time ray casting in the virtual environment.

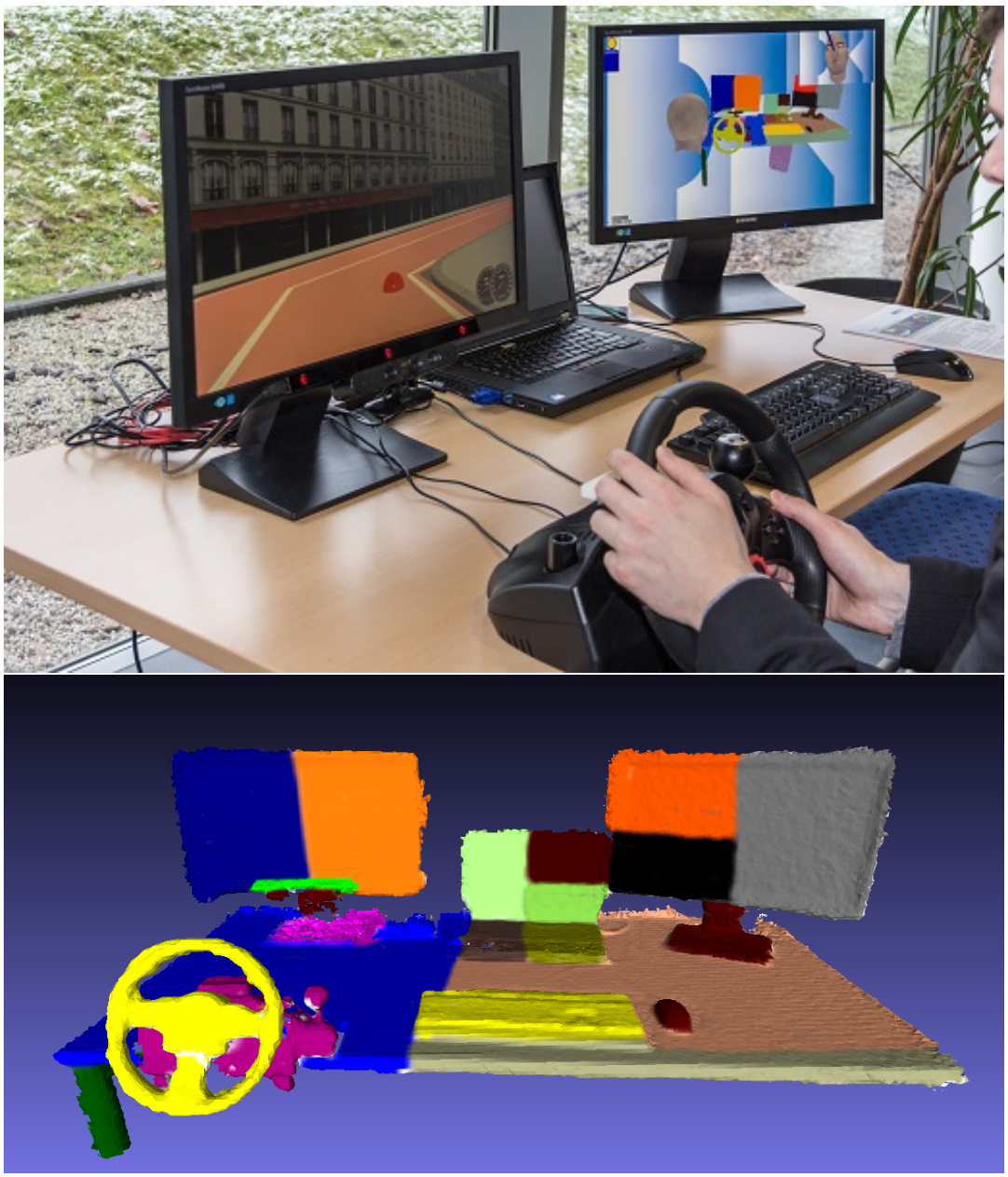

Figure 7.7: An experimental setup with the OpenDS integration and a corresponding 3D annotated model. Colors resemble different annotated regions.

The intersected object, together with the timestamp, is logged by OpenDS in the database. In addition, XVIUS uses the eye gaze and the head pose data to perform a ray casting in the 3D model of the experiment setup. Here too, the intersected object (annotated by color) together with the timestamp is logged in the database. If the user looks at another object besides the main display of OpenDS, this information is

logged by the XVIUS as well, provided that it is positioned within the opening angle of the eye tracker. In any case the direction of the head together with the intersecting region is logged. Both aforementioned logs are then accessed in the database to calculate the fixations of the user both in the virtual world (in OpenDS) and in the real world (experiment setup). As data from the same eye tracker is used in both the virtual and real environments, it is possible to determine the fixations in both worlds by analyzing the timestamps of the gaze data.

Analysis in Real-World Applications
XVIUS can be used together with a positioning system and a city model to perform an analysis of the driver's focus of attention. In this use case, it is possible to extract useful information for urban planning. For instance, we might want to find an answer for questions like "At intersection A, which buildings most attract the attention of the users? How long do drivers look at the billboard B on the highway? How distracting are the advertisements on the specific part of road C?".

Interaction in Simulator Setup
XVIUS provides third-party simulators the possibility to add interactive components to their off-screen setup. In other words, with XVIUS it is possible to map the focus of attention of the driver to each object in the physical environment and also get feedback in real time. For example, it is possible to add several small screens (e.g. small tablets or other objects) to the environment, and as soon as the user looks at each screen (or object), a message is sent by the system to the third-party application (see Figure 7.7).

Interaction in Real-World Applications
As described in the simulation part, with XVIUS it is possible to make different objects in a scene (for example the interior of a car) interactive. By scanning the car's interior via the described technique, it is possible for XVIUS to send messages to a third-party application as the driver looks at each predefined object in the scene. This object can be a part of the car console, e.g. the navigation screen or the speedometer. The third-party application can then use this information to deploy various services in real time. For example, if the third-party application is a dialog system, it can provide the user with useful information as the driver looks at different parts of the car and asks questions about their functionality, serving as an interactive manual.

7.1.2.2 Applications

The following sections present a number of actual automotive applications which have been realized with XVIUS.

Interaction with Buildings
Similar to the 2D module, it is possible to use XVIUS to interact with the buildings based on the 3D environment in real time. Instead of a spatial database, the physics engine in the Unity 3D game engine is used as a basis for reference resolution. Therefore, there is no need for predefined interaction points (and the related scanning and ray casting). Figure 7.8 shows a screenshot of such an implemented application in which the interior of the vehicle, together with the outside environment and information about driver's focus-of-attention (head pose and eye gaze), are rendered in the Unity 3D game engine. This makes it possible to perform ray-casting and reference resolution based on the "Reference Resolution Algorithm and 3D Modeling for Outside Environments" described in Section 5.1.3 in each frame.

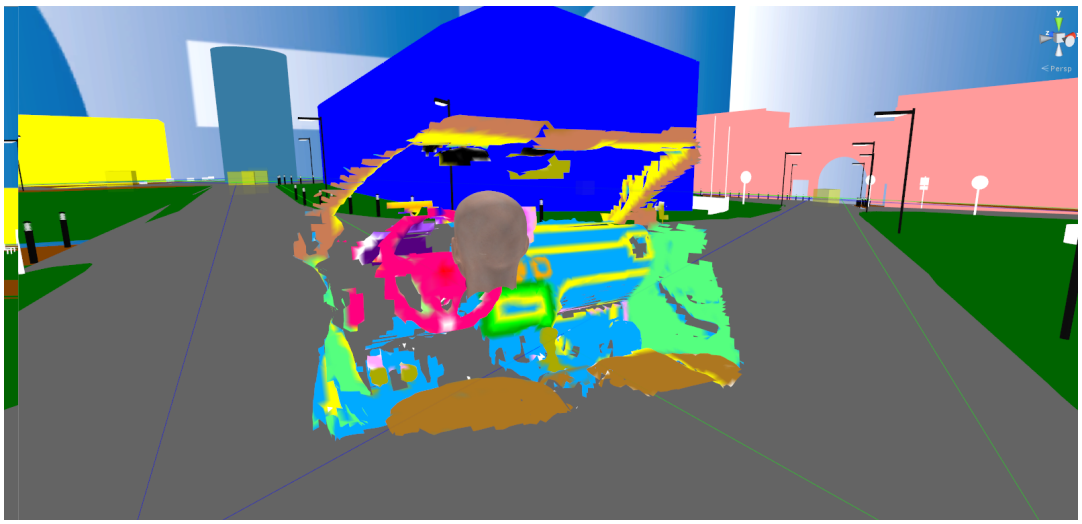

Figure 7.8: Screenshot of a scene containing the information regarding the focus-of-attention of the driver as well as the interior model of the vehicle and outside environment.

Interaction with Billboards
In addition to interaction with buildings in the 3D environment, it is possible to use XVIUS to interact with digital or analog billboards in a controlled environment. Figure 7.9 depicts an example setup. This setup resembles a situation in which a car is

Figure 7.9: Interaction with a digital billboard in a controlled environment. The driver can look at one of the two advertised movies and ask for more information.

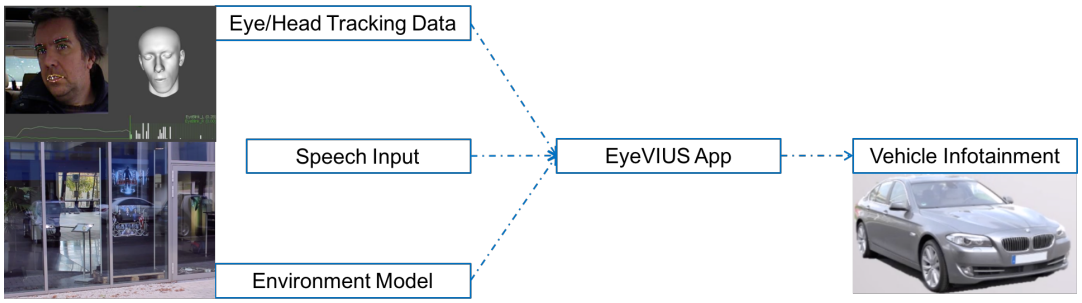

Figure 7.10: Information flow between different modules of the application for the billboard interaction.

waiting at a traffic light. The driver sees an advertisement of two different movies on a billboard. Using the XVIUS system, it is possible for the driver to look at different parts of the billboard and ask the car for more information about the respective movie. Figure 7.10 depicts the modules of this system and the information flow between them.

Interaction with In-Car Functions

With XVIUS, the driver has the opportunity to control in-car functions like the turn signal or the front windows. Figure 7.11 shows the in-car setup for this application. For this purpose, the driver has to look in the appropriate direction of the physical

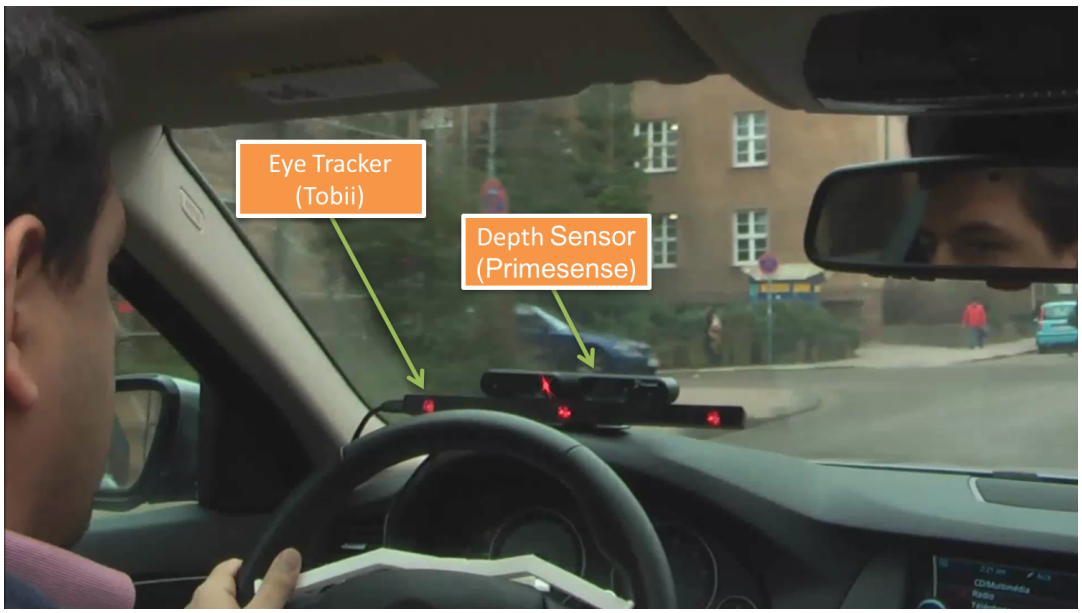

Figure 7.11: The setup in the interior of the vehicle for interaction with in-car functions.

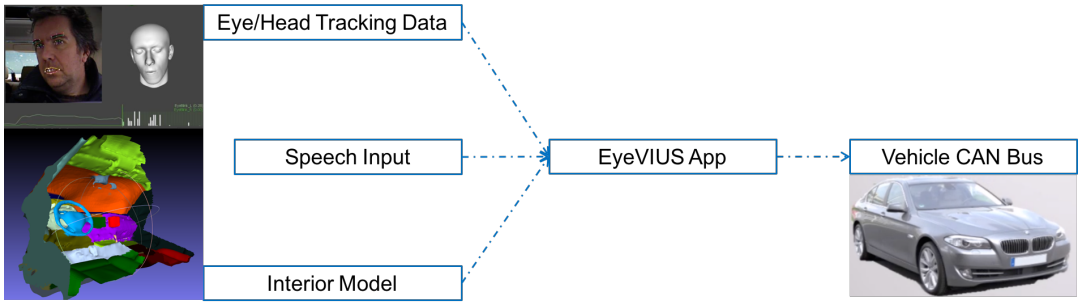

Figure 7.12: Information flow between different modules of the application for the interaction with in-car functions.

actuator to be manipulated and activate the operation through speech. A command is then sent to the CAN bus of the vehicle, and performed by the car's internal actuators. Figure 7.12 depicts the modules of this system and the information flow between them.

Analysis of Eye Tracking Accuracy

One of the main purposes of the development of the 3D version on the XVIUS system was to analyze the limits of accuracy for mobile eye tracking applications in vehicles. For this purpose, several tests have been performed in a lab and also in a vehicle in real traffic. Chapter 4 provided details of these experiments.

Analysis of Driver's Focus Direction

Besides the interaction applications, the system also performs a 3D analysis by evaluating the eye gaze and head pose data in user studies. Figure 7.13 (left) depicts an example setup for such a user study. Here, the role of XVIUS was to determine whether the driver is looking at the road or at some object inside the vehicle.

This system was composed of an eye-tracker, a head-tracker and a software which evaluates these data by using a detailed model of the target environment. In order to acquire a precise model of the vehicle's interior, this environment was scanned with a special depth camera. The resulting mesh was then divided into different regions and each region was colored differently. Figure 7.13 (right) shows the model of the vehicle's interior and the different colors used. Figure 7.14 (left) provides the description for each color. During the study, the position of the driver's head was determined relative to the hardware (green area in Figure 7.13) (right) in real time. From this position, the colored model was then used by XVIUS to evaluate the intersection of the resulting vector of the user's eye-gaze and head-pose with the different regions in the vehicle. This way it was possible to determine whether the users were looking straight forward on the street or if they were looking at objects in the vehicle. All the intersections were logged with a frequency of 25 Hz (see Figure 7.13 (left)).

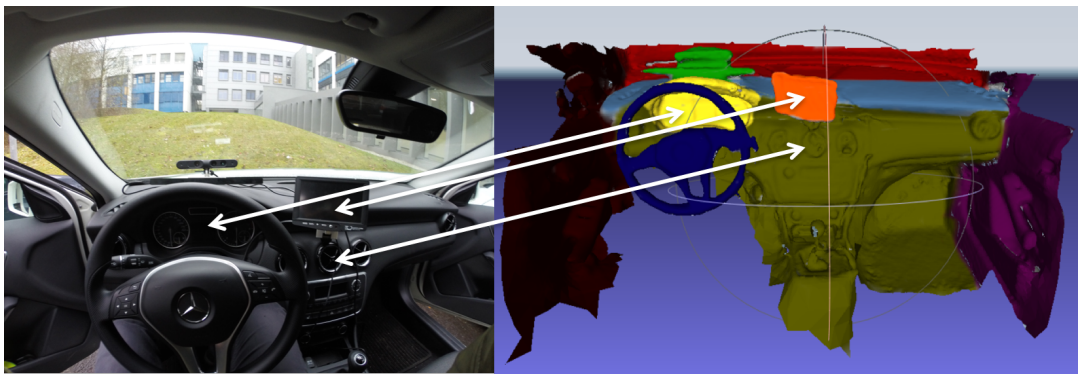

Figure 7.13: Left: Setup for the user study in the vehicle. Right: The corresponding 3D model of the vehicle's interior used for driver's focus-of-attention analysis.

The different regions of interest (ROIs) were summed up to derive two values: one value that describes the amount of time when the driver is looking at the street in front of him and one value, when the driver is not looking at the street. Figure 7.14 gives an overview over the allocation of the ROIs and also an example evaluation. In this evaluation, the gray column (outside) is one of the described summed up values.

Color	Region of interest	Encoding
grey	windscreen	street
green	eye tracker and head tracker	street
sky blue	part of the dashboard	street
red	lower part of the windscreen	street
maroon	left door	not street
blue	steering wheel	not street
yellow	speedometer	not street
orange	mounted device	not street
army green	inner part of the car	not street
purple	right car door	not street

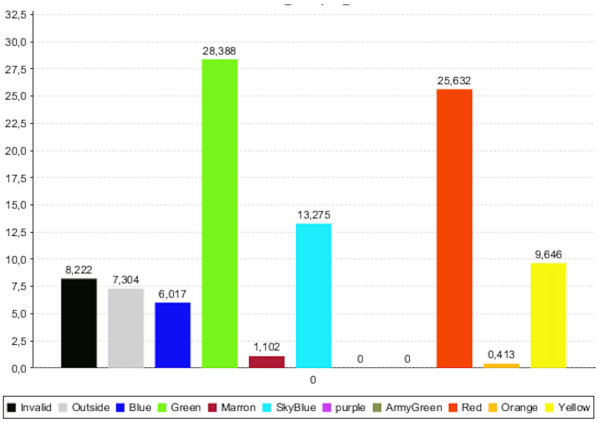

Figure 7.14: Left: Regions of interest (ROIs) in the 3D model. Right: Example evaluation of the regions of interest (ROIs).

7.2 Mixed Reality Human-Robot Collaboration

As described in Section 6.1.3, XVIUS provides the possibility to track and integrate dynamic objects of the real world into the developed applications. This part is mainly used to implement applications for human-robot collaboration[2]. Moreover, it offers the possibility to develop the same application for Augmented-Reality, Virtual-Reality, on a mobile device or PC with minimal changes. This tool implements the algorithms described in Section 5.1.4. Based on this algorithm three applications were developed and tested. In the rest of this section, these three applications are described.

7.2.1 Human Gaze and Focus-of-Attention in Dual Reality Human-Robot Collaboration

In this section a Human-Robot Interaction scenario which includes two humans and one robot is described. In this application one person has the same physical action space as the robot; the second user is positioned in a different physical location and monitors the setup through a VR system. The VR system provides the possibility for the user to manipulate objects in a 3D scene representation of the setup which is updated in real time. This representation includes the exact position and orientation of the objects on the working table as well as the other user's head position and orientation. Moreover, the user's gaze is tracked in both the real and virtual environments. Gaze information is then provided to the collaborating partner as additional

[2]Funded by the German Federal Ministry of Education and Research in the projects SiAM (grant number 01IW11004), Hybr-iT (grant number 01IS16026A), and MRK 4.0 (grant number 01IS16044).

information for tutoring purposes. As a result, with this research prototype, novel forms of collaborative factory work can be provided which enable the collaborators in different locations to follow the visual attention of the tutor in 3D and in real-time.

The entire system consists of one local server part and one or more telepresence clients. This configuration demonstrates a human-robot collaboration scenario as it is partially broadcasted to the remote user in 3D VR so that he or she has the opportunity to provide user input and interact from a remote location. The local server resides in the physical vicinity of the robot (Baxter[3]). The local prototypical interaction space of the pick-and-place scenario consists of a worktable with several objects placed on it. There is no limit for the number of objects on the table and no predefined position or orientation for them. Objects are square shaped and wooden, however, there are no limitations on the shape or material of the objects as far as they can be grabbed by the robot's grippers. The position and orientation of these objects are monitored by a smartphone which is located in the right hand of the robot. Thus it is possible to change the pose of the camera anytime and cover different objects on the table. The objects on the worktable can be freely manipulated and rearranged. As the system is running, the user can take objects from the table or add additional objects. The robot is located on one side of the table. The local user is located on the other side of the table. In front of the user is a tracking system that tracks the user's head position, pose, and eye-gaze. The left hand of the robot is free to perform actions on the table. These actions consist of picking or placing an object as well as pointing (to an object or a point on the table). There are no pre-specified or built-in movements or positions for these actions; the robot computes all of these actions within its arm range in real-time. The user can move the objects on the table by himself or perform this by clicking on the objects in 3D a representation of the working table which is also updated in real-time. Figure 7.15 depicts the local setup and the information flow between the different components. The information about the user's eye-gaze and head-pose are combined with the 3D reconstruction of the working table in a 3D environment. The whole setup is synchronized in real-time (milliseconds range) to the connected (VR) devices.

Figure 7.16 depicts the architecture of the system. A server routes the input and output messages between the different connected devices. The system on the local side sends eye-gaze and head-pose data to the server. The smartphone, which is used to monitor the setup via the built-in video camera, sends the information about pose and position of each object on the table to the server. For tracking purposes, the Augmented Reality software Qualcomm Vuforia[4] is used. This software also enables us to augment the objects on the working table with extra 3D information in real-time. The communication of the two modules (local focus-of-attention and

[3]http://www.rethinkrobotics.com/baxter/

[4]https://www.qualcomm.com/products/vuforia

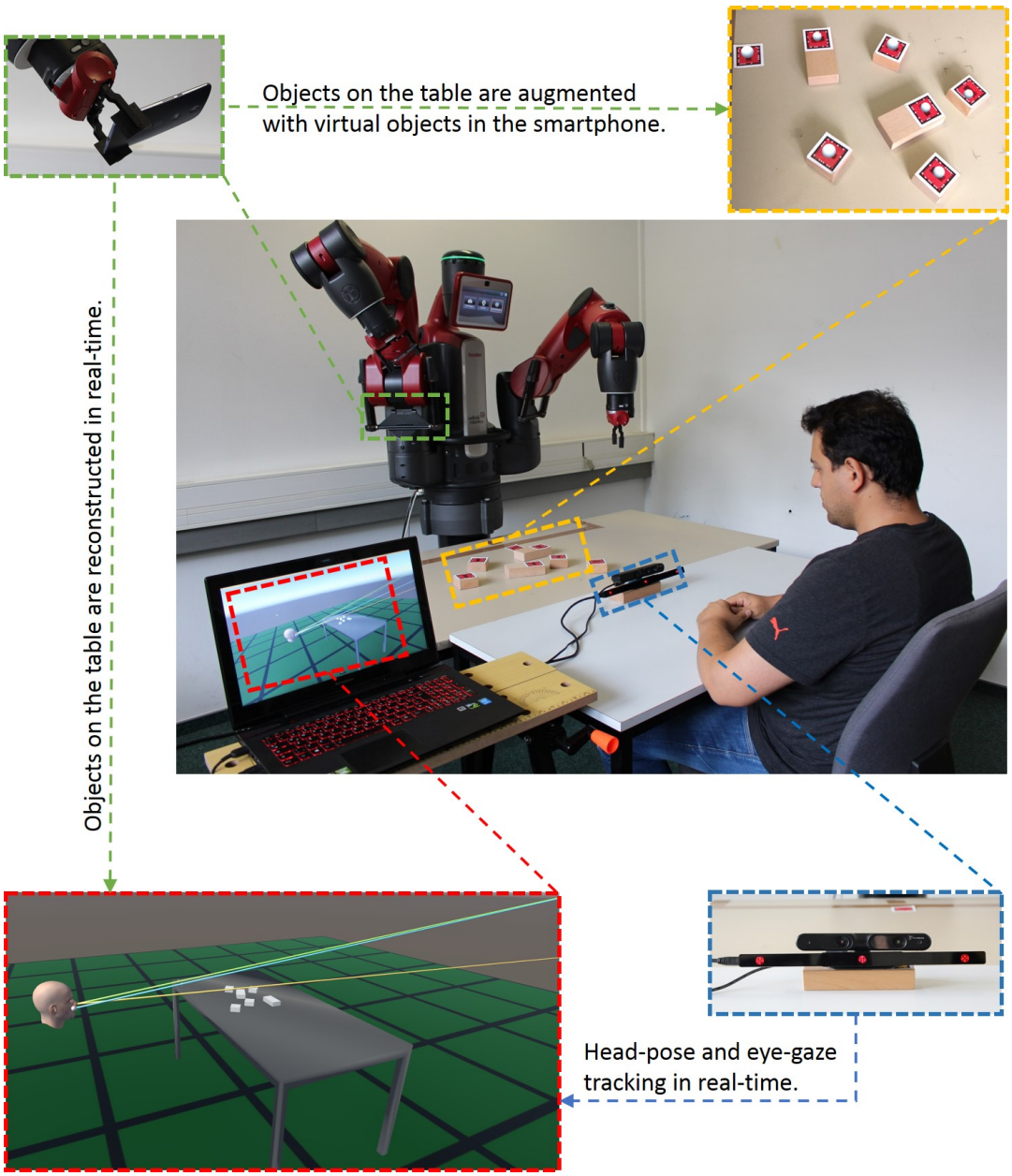

Figure 7.15: Information flow between different parts of the system for building a virtual 3D representation of the local setup. This representation is constructed by combining the eye-gaze and head-pose data of the user together with the objects' 3D information.

local object-pose-system) is one-directional and they do not receive any information from the server. The server can be connected to multiple clients. There are clients

implemented for PC, tablet PC, Oculus Rift DK2[5], and Google Cardboard[6]. It is possible to broadcast the local object setup to several instances of these devices simultaneously. Depending on the interaction types offered by the device, it is possible to send controlling commands to the server. These commands will then be forwarded to the Baxter robot. After physical actions (actuations) have been executed, the new object arrangements on the local table are updated on all monitoring clients. There are two sets of interactions available to choose from: picking/placing and pointing. On the PC client it is possible, for example, to click on any object or any location on the 3D representation of the local table to pick or place an object or to point to an object's position. The option of gaze-tracking on the clients is realized on the Oculus Rift DK2. For this purpose, a special integrated eye-tracking VR system which is commercially available[7] was used. As one of the focuses of this tool is the joint gaze-tracking of the local user and the user wearing the VR client, these clients are described in detail.

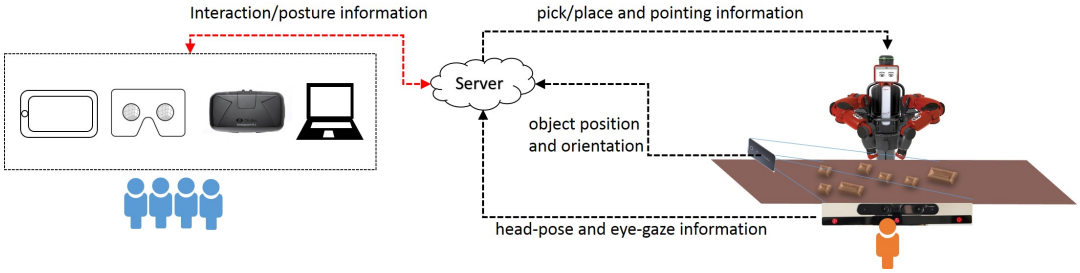

Figure 7.16: System architecture including the local and remote users. The remote client can run on different devices including PCs and VR systems.

The user of the VR client can interact with the local physical arrangement by looking at the reconstructed environment and operating a 3D mouse. In order for the eye-tracking to function properly, a 5-point calibration of the VR system is performed. After this procedure, as the user looks in any direction in the VR system a red dot representing his or her gaze will appear in the environment. This information is acquired in 3D for different interactions. To provide direct feedback, the color of the objects in the VR environment change as the user looks at them. In addition to moving eyes, the user has the opportunity to move the head and change the view direction of the camera in the VR world. In order to change position, the user can operate a 3D mouse. This gives the user the opportunity to move the viewing camera along the three axes of the local coordinate system of the VR camera. Combinations of these actions make it possible for the person using the VR client to move freely in

[5]https://www.oculus.com/en-us/
[6]https://www.google.com/get/cardboard/
[7]http://www.smivision.com/en.html

the reconstructed environment and look at any desired point from any desired angle and distance. In addition, the 3D mouse which is used has two buttons for interaction. These buttons can be used by the user of the VR client to pick or place objects in the scene or point at any position on the table. For this purpose, the following new multimodal interaction modes can be employed: the user looks at the target point and presses the dedicated button; manipulation commands will be sent to the server and from there forwarded to the actuation controller of the Baxter robot. Baxter executes the actuation. The VR user is able to see and monitor the 3D constructed and synchronized physical environment in real-time. Additional multimodal interaction modes are as follows: the VR system can also send focus-of-attention information to the local user by looking at the target location and using the pointing mechanism. Figure 7.17 depicts different views of the VR client and his gaze as well as some performed actions.

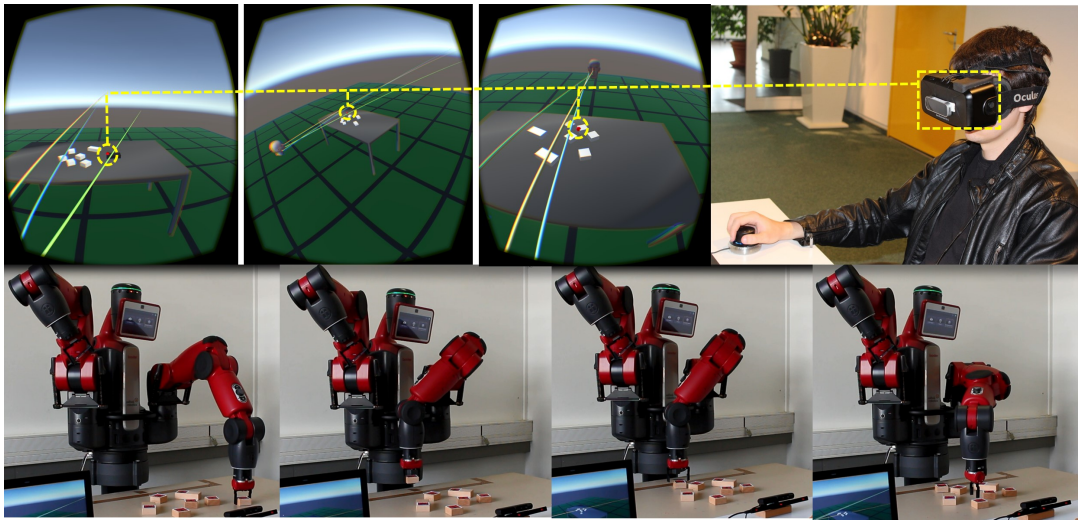

Figure 7.17: Top: the user wearing the VR glasses is able to inspect the 3D representation of the local scene from different angles and positions. The small circles show the gaze-information of the VR user in the reconstructed VR scenario. The lines in the VR scene represent the head-pose and eye-gaze of the local user. The VR user can look at objects and perform various actions. At the bottom (from left to right): the Baxter robot takes an object from one place to another and then points to another object.

7.2.2 Hybrid Team Interaction in the Mixed Reality Continuum

The scenario here is similar to that of Section 7.2.1, however this system has the following characteristics:

- Mobile eye-tracking
- Hand-tracking in virtual environment
- Multimodal interaction using different wearables
- Spatial augmented reality functionality

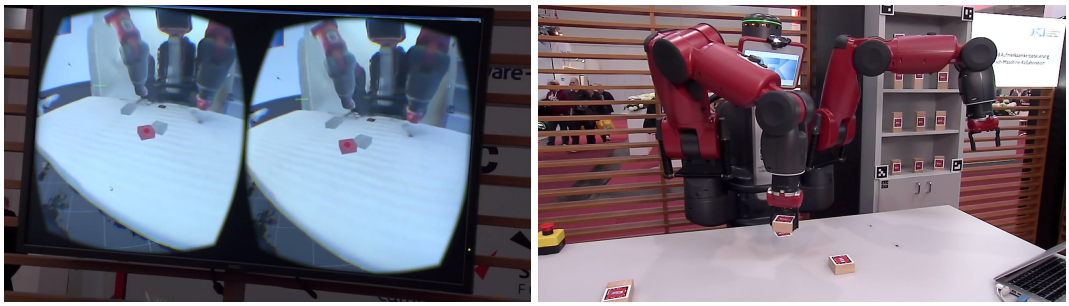

Figure 7.18: The environment setup (right) and its representation in the VR (left). The colors of the objects change when they are in the user's focus-of-attention.

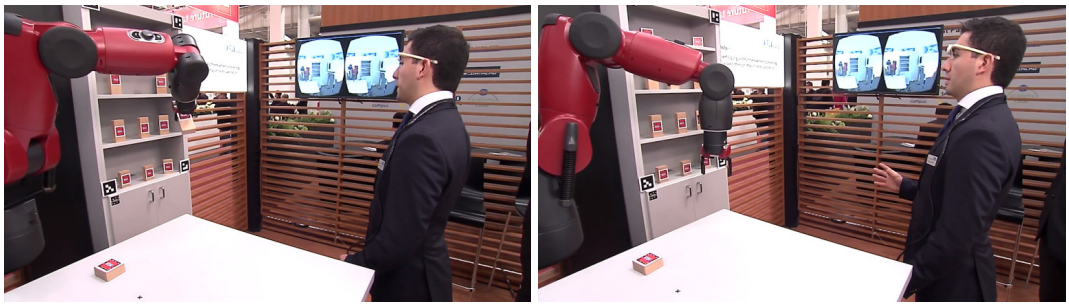

Figure 7.19: The user can look at a product on the shelf and make a hand gesture to instruct the robot to take the object from the shelf and put it on the work bench.

The system consists of two parts: a local and a remote part. The local environment includes a shelf, a working bench, and one Baxter robot[8] (see Figures 7.18 and 7.19). This environment is scanned and modeled in 3D for the remote users. The remote participants can posess a Samsung Gear VR[9] or an Oculus Rift DK2[10] VR system. The Oculus Rift DK2 is additionally equipped with a Leap Motion[11] sensor which is mounted on the VR system. The remote participants can control the Baxter robot from within these VR setups (see Figure 7.20). Their hand/head movement and their

[8] http://www.rethinkrobotics.com/baxter/
[9] http://www.samsung.com/us/explore/gear-vr/
[10] https://www3.oculus.com/en-us/dk2/
[11] https://www.leapmotion.com/

Figure 7.20: The setup for the user which monitors the environment through VR. The detailed hand movement of the user is also integrated into the environment.

position in the environment can be viewed by the local users via an AR solution in a Google Tango Tablet[12]. On this tablet, the users see, in addition to the environment, the avatar of the remote participants (see Figure 7.21). These avatars have exactly the position of the remote users in the VR setup. As the remote users move their head/hands or change their position, this information will be updated in real-time in the AR tablet. In addition, the user with this tablet can move around in the environment and view every 3D avatar from various distances and angles. The second local user is equipped with an eye-tracker from Pupil Labs[13] and a Myo gesture control armband[14]. This user can use these two devices to command the robot to take an object from the shelf and put it on the worktable. For this purpose, the user has to look at the object on the shelf and make a hand gesture (see Figure 7.19). After getting the feedback from the robot, the user can make another hand gesture to confirm the selection. From a different location, users can utilize a VR system to be virtually present in the factory. In the VR environment the remote users see a detailed 3D reconstruction of the factory (pre-scanned). They can freely move and look around in this environment. In addition, they can use a 3D representation of their hands, which will also be brought to the virtual environment. They can use their hand or visual focus point to interact with the remote robot. In the virtual environment, they can also see information about local users (for example, where the local users are or what are they looking at).

The scenario is as follows: Using a combination of a multimodal input (gaze and hand gesture), the local user selects an object on the shelf. The robot takes the object from the shelf and puts it on the table. Meanwhile, the remote users (experts) can monitor in the VR system which object the local user is selecting (by seeing the 3D gaze live information) and can assist him in the selection. When the proper object is selected and the robot has put it on the table, further processing can be done. This

[12]https://get.google.com/tango/
[13]https://pupil-labs.com/
[14]https://www.myo.com/

Figure 7.21: The user with the AR tablet can move freely in the environment. The AR application shows the remote colleagues who work virtually in the factory.

will be performed by the remote users in their VR setups. In this setup, they can see every character including other remote users. They can also investigate the objects on the table by looking/pointing at them. Finally, they can do the further processing of the object by controlling the robot remotely via their visual focus (for determining the action point) and a 3D mouse. The local user who is using the AR tablet can monitor the whole process from various angles/distances.

This system enables collaboration of hybrid teams consisting of a robot, physically present humans and remote humans. This represents a vision for the future communication of scattered teams in industrial environments. The goal of this communication can be training or collaboration for problem solving purposes. This scenario covers both of these goals. Here, two people have the same physical action space as the robot; the other users are positioned in different physical locations and monitor the setup through a VR system. The VR system provides the possibility for the user to manipulate objects in a 3D scene representation of the setup which is updated in real time. This research prototype can be scaled to include several virtual agents and local users. All of the participants can see each other either in the provided VR or the AR system.

7.2.3 Mixed-Reality Technologies for Multi-Site Human-Robot Team Production in Industrie 4.0

The scenario here is similar to that of Section 7.2.2, however this system has the following characteristics:

- Full interaction in Augmented Reality
- Scaling the whole scenario to simultaneously include 4 users and 8 robots over 3 locations

- Integration of the real-time actuator data

Figure 7.22 shows the system architecture and the information flow between the different components. The whole system is scattered over 3 locations (locations A, B, and C in Figure 7.22). Location A contains two users and three robots. These robots are UR10 from Universal Robots[15], ABB YuMi[16], and the logistic robot MiR100[17]. Each user wears a Microsoft HoloLens. The setup at location C is exactly the same as in location A. Figure 7.23 (top) shows such an example setup. The location B consists of one UR5 and one MiR100. There is no user at this location, which means that this site is controlled remotely. This location also includes two servers which distribute the messages between all of the components across the three locations. These messages can include snapshots of the robot's camera or another different form of information. In the following, first the various interaction possibilities are described and then the details about the implemented collaboration and interaction paradigms are presented.

Depending on the capabilities of each robot, a custom interaction technique was developed. As the MiR100 can move forward, backward, or rotate around its own center, in the Mixed Reality (MR) application, each of these directions was visualized with a specific arrow. These movement arrows were positioned on the robot in their action direction (see Figure 7.23, center). As the location and orientation of the MiR100 was tracked permanently, the position and orientation of these arrows were updated in each frame. The user had the opportunity to bring the focus point of the MR application to each of these arrows and make a tap gesture. As long as the user holds the tap gesture, the robot will move in the selected direction. Concerning the UR10 robot, a different kind of interaction was developed. Here, the user could open/close the gripper or put a 3D point in the space for the robot to move to. Combining these two actions makes it possible to bring the end-effector of the robot to any 3D position within the reach of the robot's arm and perform pick and place actions (see Figure 7.23 bottom). Regarding the ABB YuMi, as it has two hands, two different functionalities were implemented. For one hand, an application similar to the UR10 robot was designed. The other hand contained a camera. The user had the opportunity to activate the camera and make pictures of an object in front of the robot from different perspectives. This image was then displayed in the MR application (see Figure 7.23 center). A remote user can also perform the same interactions on these robots. In his MR application he sees a virtual representation

[15]UR10 and UR5 are one arm collaborative robots produced by Universal Robots (https://www.universal-robots.com/).

[16]YuMi is a two arm collaborative robot produced by ABB (http://new.abb.com/products/robotics/industrial-robots/yumi).

[17]MiR100 is a mobile robot for internal transportation and logistics (http://www.mobile-industrial-robots.com/en/products/mir100/).

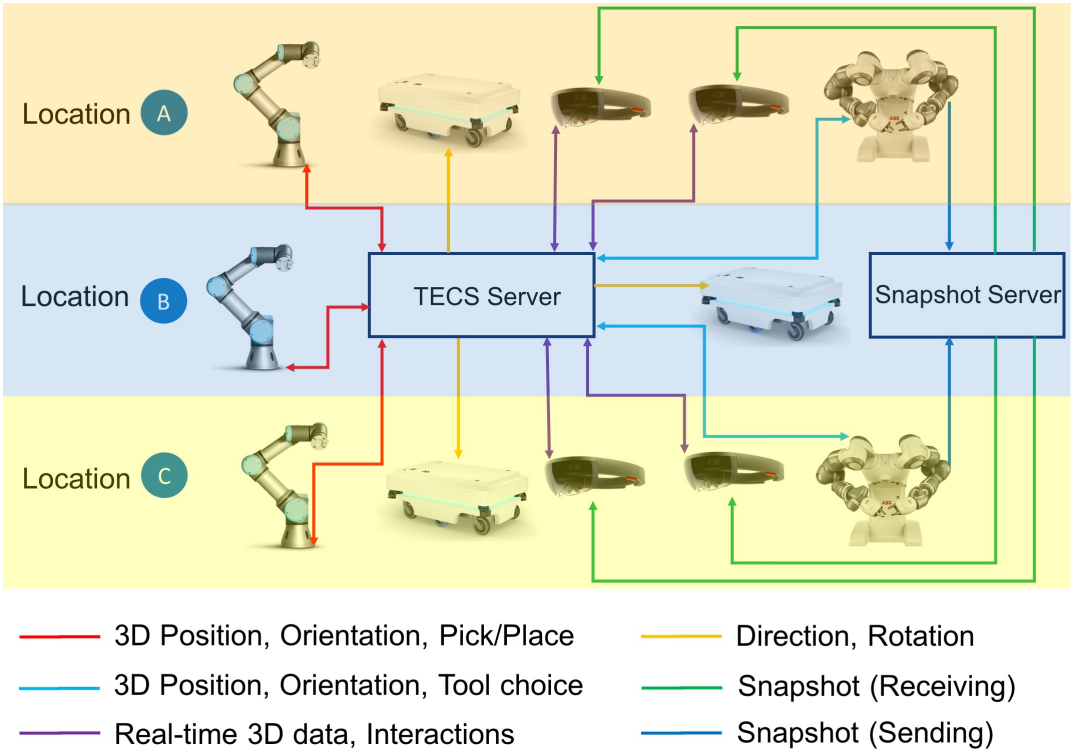

Figure 7.22: Various components of the scenario for Mixed Reality Human-Robot Team Collaboration. The lines visualize the information flow.

of these robots in addition to the avatar of the local person (see Figure 7.23 bottom). In turn, the avatar of the remote person is visible to the local user. The position and orientation of the avatar and all other objects in the scene are tracked and updated in each frame.

With the developed system it is possible to realize the following two human-robot interaction and collaboration paradigms which include Mixed Reality technology:

- Mixed Reality Human-Robot Interaction: Controlling different kinds of robots in a multistage use-case through a Mixed Reality application.
- Multi-Site Human-Robot Collaboration: Collaborating remotely with other colleagues in a different site via their virtual representation to control the robots in a Mixed Reality application.

Figure 7.24 depicts the details of the whole system which includes three scenarios. Each of these scenarios has a different focus. One of the implemented scenarios focuses on the spatial interaction, navigation and augmentation using Mixed Reality

Figure 7.23: Top: An Example setup of the developed scenario for Mixed Reality Human-Robot Collaboration. Center: Interaction with MiR100 and the ABB YuMi. Bottom: Interaction with UR10 as in the vicinity of the robot (left) and from a distance (right).

(Local Manipulation in Figure 7.24). In this scenario, the user is in the physical vicinity of the robots. The second scenario is similar to the first scenario, however here the user controls the robots from a distance (Remote Manipulation in Figure 7.24). In both these scenarios, one user is involved in the interaction task. The third

scenario combines different aspects of the first two scenarios in a team interaction setup. Here two people control different robots in cooperation. One person is in the vicinity of the robots and the other person joins them as a virtual character. Both users are equipped with mixed reality setup (Remote Collaboration in Figure 7.24). The user who is in the vicinity of the robots (similar to scenario 1) sees the robots and in addition to that sees the virtual character of his team colleague who is interacting with the robots from a distance. This person, who is physically somewhere else, sees the robots and the other user as virtual representations in his Mixed Reality setup.

In a demonstration setup, all of the described scenarios are realized over three different locations. This demonstration includes 8 robots and 4 users which work simultaneously. Figure 7.24 marks different locations of the system with the letters A, B and C. At the location A the first scenario is realized. At the locations B and C, the second scenario is realized. Here the user who is at location C controls the robots that are in location B. The third scenario is realized using the locations A and B. In this scenario, the robots are located at the location B together with one of the users, and the other user is located at the location A. Both users can control the robots and each sees a virtual representation of the other person in his Mixed Reality setup. Figure 7.24 depicts the different locations and the logical separations of the three scenarios.

Using the proposed dual reality human-robot collaboration systems it is possible to implement a variety of applications in different fields of Human-Robot Interaction (HRI). It provides the advantage that each participant can send and receive focus-of-attention information from the other person in real-time. In addition, as the system is designed and implemented in a way that several clients can monitor the local setup simultaneously, one expert can give demonstrations to multiple trainees at the same time. The trainees can also join the training using different devices including mobile devices. A possible scenario can be as follows: a specialist is traveling and not present at the office. Her/His colleagues call to ask about an abnormality which has happened at the main plant. By using this system and applying the introduced interaction technology, she/he takes control (of the robot) form far away and assists the colleagues to solve the problem together.

The advantage of this system over traditional telepresence systems (e.g. video conferencing tools) is twofold: the trainees can follow the gaze pattern of the expert and gain insight about how an expert analyzes a specific situation. This way they can get access to the expert's way of problem solving and find answers to questions like: which positions are important to look at and in which order? The trainees can also move freely in the 3D scene and see the expert performing the tasks from multiple angles and distances.

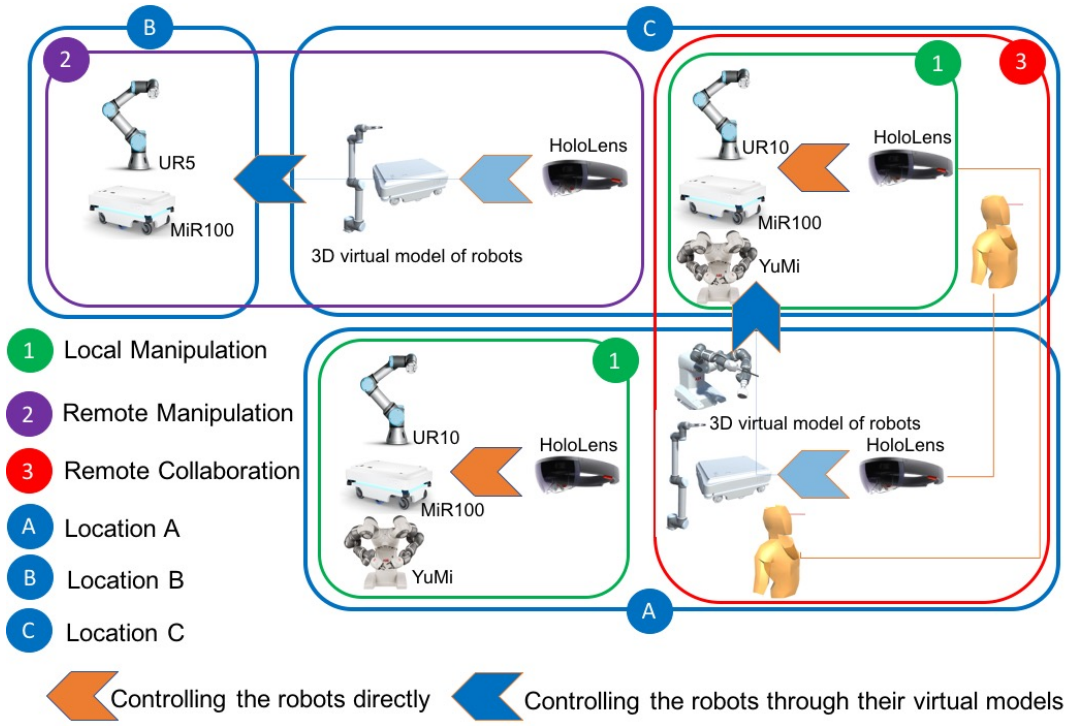

Figure 7.24: Three Scenarios for Mixed-Reality Human-Robot Collaboration across three different locations.

7.3 Focus of Attention in Spatial Human Environment Interaction

The Spatial Fusion Tool (SFT) is used to build applications for single and multi-party human-environment interaction. This tool implements the algorithm explained in Section 5.2. The following describes the applications realized with this tool[18].

EyePICA was the first system developed with SFT. This system utilizes the user's head and gaze direction for human-environment interaction. For this purpose, custom hardware was developed (see Figure 7.25). This hardware consists of different input and output components. The incoming data from the head-tracking and eye-tracking modules is directed to the SFT for further processing. The implemented system also included a 2.5D model of the environment. Based on this model and a ray-casting module, the object in the visual focus-of-attention of the user was detected. The output of the reference resolution algorithm was then sent to the installed augmented reality display. This application provided the possibility for a single user to get more

[18]Funded by the German Federal Ministry of Education and Research in the projects SiAM (grant number 01IW11004) and MADMACS (grant number 01IW14003).

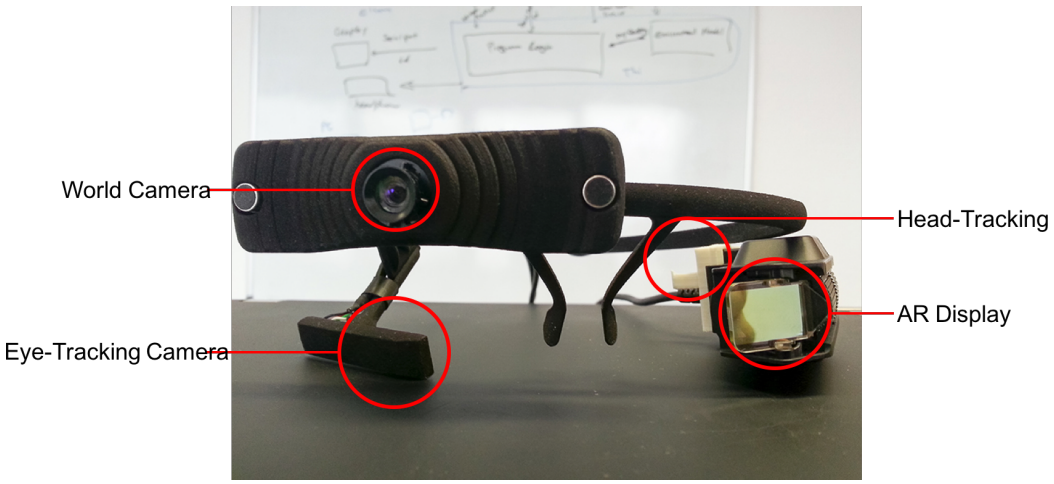

Figure 7.25: The EyePICA hardware consists of wearable head-tracking and eye-tracking components as well as an augmented reality display.

information about the objects in the environment.

SFT was also used to implement various use cases in a car repair garage scenario (see Figure 7.26). The implemented system utilized the user's position, head orientation and body pose for deriving different features in a multi-user scenario. These features include, for example, the user's identity, group membership and focus of attention. This system continuously received and merged sensor data from multiple devices and distributed the result as coherent user context to other applications. The user's visual focus on objects is derived from two sources: first, the nose vector from the Kinect was used to resolve objects of interest. Second, camera-based object recognition on a personal device was used to detect the focus on certain objects. In the implemented demonstrator, the application runs on a Google Glass or an Android tablet. Groups are resolved based on the users' distance to each other and the body orientation of the group members. In order to resolve the identity of users based on their position (e.g. when they perform a gesture or speak into a beam forming-capable microphone) one needs to combine the information from two data sources: the Kinect and the smart glass. The former captures the interaction and is aware of the user's position in the scene, but cannot generally identify the user. The smart glass on the other hand is a personal device, so it can have a user profile stored on it. Both devices, however, can measure movement of the head, hence we can correlate the information in order to match the patterns for all people in the scene from both data sources. More precisely, the algorithm measures the yaw and pitch correlation between each Google Glass and each body tracked by the Kinect. After reaching the threshold of the correlation coefficient, the user identity stored on the glass can be assigned to an unknown tracked user in the scene.

Figure 7.26: The car workshop scenario consisting of multiple users.

In the implemented scenario, a mechanic enters the scene to inspect the status of a car. He is tracked by the system (via Kinect) as soon as he approaches the area he is identified using his Google Glass. On the Glass screen, he sees which parts the car's diagnostic has identified as faulty through small Augmented Reality (AR) overlay icons. He asks the system for a full report via far-field speech interaction. The worker conveniently controls the car windows from outside the vehicle with a multimodal interaction involving speech, gesture and head direction. The car and bike have been registered by the Cyber Physical Environment (CPE) device platform, so the CPE can operate them e.g. via the CAN bus. A second mechanic, also wearing his Glass, arrives and they start a conversation. The system, physically tracking both via Kinect, assigns them to a single group due to their spatial proximity and body orientation. Based on their head movements, the system can quickly resolve their identities. When they look at the faulty part under the hood and then at a large display nearby, the CPE proactively presents some information available for the part previously in the focus on the now focused output device (the big display in Figure 7.26). Having a free choice of modality, both users can scale and rotate the displayed 3D schematic of the car part using speech, gestures, or a combination of both.

7.4 Application for Peripheral View Analysis

The PVA tool consists of a software module and implements the algorithms from Section 5.3.1. The input data for this algorithm is: position and the bounding box of the objects in the environment, position of the user's head/eye and also a 3D vector which represents the user's gaze direction. Based on these inputs, the PVA tool provides a comprehensive output regarding an object's visibility in the environment for an observer (see Section 6.2). The following are applications which have been implemented based on this output[19].

7.4.1 Peripheral View Analysis for Dementia Patients

This case is similar to the one explained in [Sonntag, 2015]. This study explores the mixed reality realm for helping dementia patients. As a point of the user's engagement with the environment, he uses gaze. In other words, for his different algorithms concerning object or face recognition or augmentation, he always uses the very center of gaze. The PVA application introduced here wants to build on this and bring the other objects in the user's periphery into the interaction space.

The application presented here approaches the [Sonntag, 2015] scenario by simulating daily life scenarios in a controlled virtual environment. The episodic memory logging and object augmentation of [Sonntag, 2015] is based on the gaze center. The presented application here applies the model described in Section 5.3 to perform these actions also in the periphery with complementing information like, for example, visibility of the object. For this application, a virtual reality setup with the option of gaze tracking is used. The utilized hardware is a special integrated eye tracking Oculus Rift DK2 system, which is commercially available[20]. In the virtual reality scenario, the user plays the role of the patient and can freely move within an apartment. Regarding the different objects in the apartment, there are some tasks that the user should accomplish. These tasks should be executed in a pre-defined order. The system supports the user by providing contextual information on the target object depending on the visibility of the object to the user. The information is stuck to the object and as the user's gaze moves in the scene, depending on the visibility, the application shows different messages on the object. These messages are related to the task that should be accomplished by the patient. Examples are eating fruit, reading a book or watering a plant. As the gaze of the user approaches the object and the visibility of the object increases, the message includes more details about the task. The

[19]Funded by the German Federal Ministry of Education and Research in the projects SiAM (grant number 01IW11004), MADMACS (grant number 01IW14003), and SC_MeMo (grant number 01IS12050).

[20]http://www.smivision.com/en.html

following three levels of detail depending on the visibility of the object are defined:

- Low Visibility: The visibility of the object is very low. Some examples for this case are large Cartesian distance between the object and the user or large angular distance between the user's gaze and the object.
- Medium Visibility: Either the object is in the center of gaze and the Cartesian distance of the user to the object is not very small or the user is near the object but he or she is not looking at it.
- High Visibility: The object is in the focus of the user and the user is in the vicinity of the object.

Note that for these classifications neither the Cartesian nor angular distances are used directly. These categories are set by the thresholds which are selected by weighting the solid angle of the visible part of the object with respect to its visual acuity in the according field. This novel approach provides a unique feature to categorize the objects in the environment depending on their visibility and not directly on their position and size. If the object is near to the user but partially occluded by any other surface, the application will provide the appropriate visibility measure. This value might be the same as when the object is in the periphery of the user. In any case the output of the algorithm in the application is a single visibility value which summarizes its visibility situation to the user.

Depending on the visibility of the object, the presented system augments it with more or less detailed information. If the object is in the 'Low Visibility' category, the system covers it with a blinking surface to attract the user's attention to it. As Hatada et al. mention, in the induced visual field (which has very low visual acuity), the observer has discriminatory capability to the extent of being able to recognize the existence of a visual stimulus [Hatada et al., 1980]. This indication is used to attract the user to the object, while augmenting a small part of the visual field. Figure 7.27 shows different screenshots of the VR environment. As it is shown, the three mentioned visibility classes can occur by varying the Cartesian or the angular distances between the object and the user. The two other visibility categories are the 'Medium Visibility' and 'High Visibility'. In the case of 'Medium Visibility', some hints about the patient's task will be augmented over the object. This augmentation is performed in short words with big fonts. It should help the patient to recall the corresponding activity without overloading the visual field. If the user can not recall the activity, he or she can approach the object and look at it directly. In this case, the state of the object in the application will enter the 'High Visibility' mode. Thus, the task of the patient will be recalled to him or her in the form of short sentences.

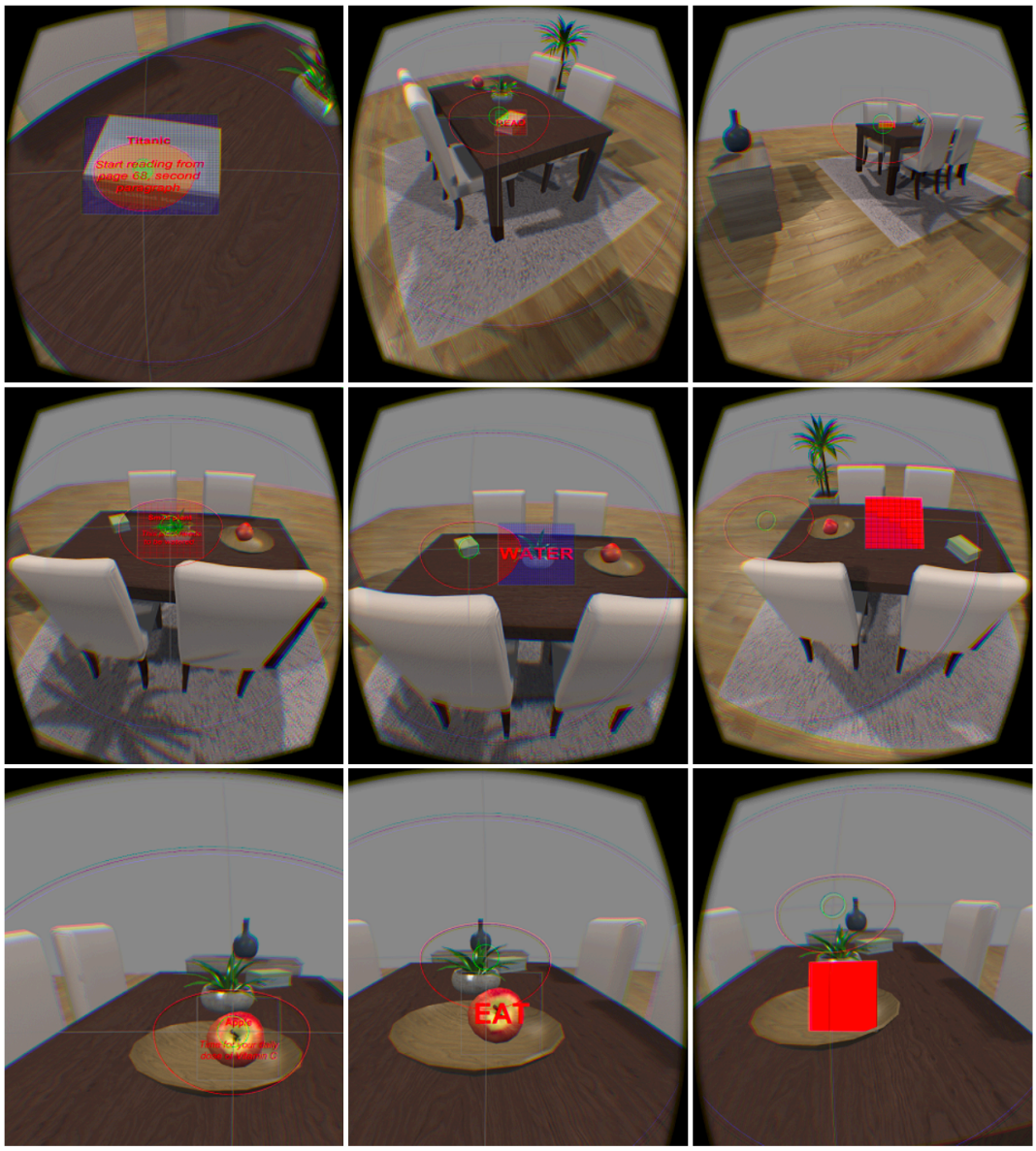

Figure 7.27: Three different levels of augmentation depending on the visibility of the target object. Each row shows a decrease in visibility from left to right. Top row: Decrease of visibility through increased Cartesian distance to the observer. Middle row: Decrease of visibility through increased horizontal angular distance. Bottom row: Decrease of visibility through increased vertical angular distance.

The augmentation is located on the object itself, disregarding its relative position to the user. Instead, depending on the visibility of the object, the appropriate form for communicating the message to the patient is chosen. First by attracting the user to-

wards the object, then by augmenting the task's description step by step. This way, we do not always bring the full information to the user, instead, we assist the patient to recall the task by providing the information gradually and helping her/him to recall. The visibility of the object to the user is permanently monitored so that the system can perform an episodic memory logging concerning the peripheral visual field of the patient. This is an added value to the episodic memory logging of Toyama et al. in [Toyama and Sonntag, 2015] as it can be effectively used to support an individual user's memory by logging certain types of everyday information that the user perceives in his or her peripheral vision and tries to organize in his or her memory.

7.4.2 Peripheral View Analysis for Automotive Applications

Based on the PVA tool, another VR application was developed which implements a simulated automotive use case. It includes an outdoor scene consisting of a 3D model of the Saarland University Campus with 492.3 thousand vertices, which has been reconstructed based on point clouds obtained from several laser scans. A scanned model of the test vehicle was also integrated into this model. Furthermore, the 3D models of other vehicles, pedestrians, and other small objects like trees, road signs, traffic lights, etc. were also integrated into the 3D environment. The VR setup includes a special version of the Oculus Rift DK2 system with an integrated eye tracker, which is commercially available[21]. All implementations are based on the Unity 3D game engine.

In the presented scenario, users play the role of a driver sitting in a car in front of a red traffic light. They are able to freely look around within the university campus. Regarding the different objects in the scene, some of them are tagged as "target object". Those objects are considered to be important in the current context. Users are expected to focus their attention and gaze on the frontal scenery or the red traffic light. In the moment it switches to green and users are expected to move forward, a pedestrian enters the scene, unexpectedly crossing the street. Consequently, besides the traffic light, the pedestrian constitutes a target object of high importance, as he or she induces a dangerous situation being situated in the user's peripheral visual field and is thus barely perceived. The situation is depicted by Figure 7.28. Both target objects are marked with their corresponding projected bounding box which serves as input for the algorithm. In addition, one can see the projected, elliptic visual field regions which are positioned around the center of gaze (red dot) and visualized with different colors. In each frame-based pass of the main application loop, the presented method delivers visibility measures for each of the target objects (OOEs) and enables

[21] http://www.smivision.com/en.html

Figure 7.28: The simulated automotive use case: the visibility of relevant objects (pedestrian, traffic light) in the peripheral view of the driver serve as input for assistive system functions (attention shift).

the system to trigger a shift of attention, in case a highly relevant object features a low visibility for the user[22].

7.4.3 Peripheral View Analysis for Augmented Car Workshop Scenario

The Peripheral View Analysis for Augmented Car Workshop Scenario application, presents how PVA can assist a user via a context-sensitive augmented reality application. The setup in this scenario consists of a video see-through augmented reality hardware with integrated eye-tracker. The implemented interaction here is similar to that of the system described in the Section 7.4.1. Here also three levels of detail depending on the visibility of the object are defined: Low Visibility, Medium Visibility, and High Visibility. Depending on the visibility of the object, the presented system augments it with more or less detailed information. If the object is in the ”Low Visibility” category, the system covers it with a blinking surface to attract the user’s attention to it. This indication is used to attract the user to the object, while augmenting a small part of the visual field. Figure 7.29 shows screenshots of the

[22]Please refer to Appendix (A.2) for the performance analysis.

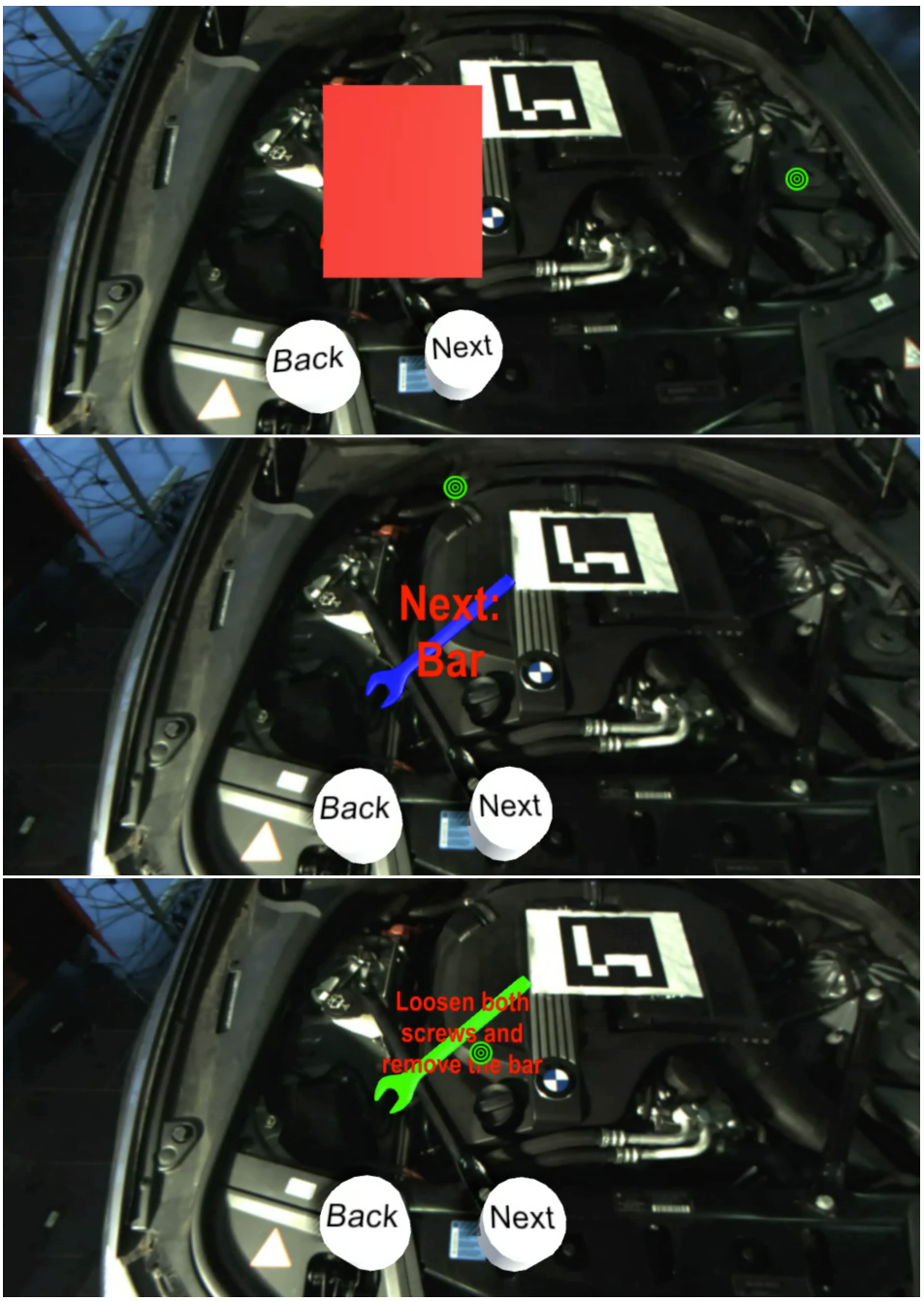

Figure 7.29: Three presentation levels for augmenting the object of interest when it is in the user's focus or peripheral vision.

augmented reality environment. The small green circle represents the user's gaze. As it is shown, the three mentioned visibility classes can occur by varying the Cartesian or the angular distances between the object and the user. The two other visibility

categories are ”Medium Visibility” and ”High Visibility”. In the case of ”Medium Visibility”, some hints about the user’s task will be augmented over the object. This augmentation is performed in short words with big fonts. In the ”High Visibility” mode, the object is augmented with short sentences.

8 CONCLUSION

This last chapter gives an overview of the work done in this thesis. First, in Section 8.1, the research questions are revisited. For each research question, a summary of results is presented. Then, in Section 8.2, the various contributions of this thesis are listed. These contributions vary from scientific publications to awards and industry projects. Finally, Section 8.3 discusses the next steps and the future work which can be performed on the basis of the research presented in this thesis.

8.1 Research Questions Revisited

In the current section, the research questions from Section 1 are revisited. For each question, a summary of the results which are achieved through this thesis are presented.

- **What are the characteristics of humans' visual perception in the fovea and peripheral vision and how are these characteristics used by previous eye-based interaction paradigms? What are the drawbacks for these systems and how can we overcome them?**
  Chapter 2 described the foundations of humans' visual perception. It described how we perceive the environment differently in fovea compared to other peripheral regions. It also described 3D model of the human visual field and also how the concept of solid angle can help us in the visibility measurement. The second part of this chapter gave an overview of the available software and hardware solutions for building eye-based interfaces. After covering the foundations, Chapter 3 gave an overview on the state of the art in the eye-based human environment interaction applications and also their drawbacks. The result of comparison and analysis of the current systems was, that we can overcome the drawbacks of these systems by adding several input and output channels.

Regarding input channel, gaze in highly mobile environments (such as driving situations) was not considered by many researchers as an input modality for selecting various urban objects. For output, none of the studies consider a model for integrating visibility of peripheral objects in their interaction paradigms.

- **What are the limits for visual reference resolution when using gaze to refer to objects in an inside or outside environment?**
  Chapter 4 provided details on two studies. One of these studies examined how the reference resolution differs in an outside environment from a controlled lab environment, when using an off-the shelf eye-tracking component. As described, interaction with big urban objects is possible using the off-the-shelf eye trackers in the vehicle cockpit. In the tests on the Saarland University campus, the precision of the developed reference resolution algorithm to determine the target object was more than 90%. However, due to the high imprecision it cannot be used to refer to small objects in the outside environment. Considering indoors, the accuracy range for the 3D eye tracking with free head movement and without calibration was between 0.5 and 5°. The median accuracy for this combination was 1.98°. The vertical and horizontal median accuracies where 1.06 and 0.2°, respectively. For comparison, the median accuracy for the 2D case with free head movement and 9-point calibration was 0.48°.

- **How can general peripheral vision be modeled for gaze-aware intelligent user interfaces?**
  In Section 4.2.7 a general peripheral visibility model was developed. First the data preparation was described. Then the model itself was presented in detail. This model showed the relation between visual perception and three variables: vertical eccentricity, horizontal eccentricity, and size of the object. For this purpose a two-dimensional Gaussian function was combined with a Weibull distribution function. Finally, for estimating the overall visibility of a peripherally observed object, the joined model 4.3 was introduced. Basically, this model constituted a 4D function, mapping angular vertical and horizontal distances of an object from the center of gaze and its observed size to a scalar-valued visibility on the unit interval.

- **How can we design new algorithms to enrich eye-based interactions in Automotive User Interfaces and Human-Robot Collaboration scenarios?**
  Chapter 5 described 8 algorithms for using eye-based information. Some of these algorithms can be used to refer to objects in highly mobile scenarios, others can be used for peripheral interaction or user identification. These scenarios addressed two different fields: automotive user interfaces and human robot collaboration. In the automotive scenarios, various city models were considered, from very simple 2D model to very detailed 3D models based on precise 3D scanning. In the human robot collaboration scenarios, different use

cases including mixed-reality and multi-party interaction were considered. A common feature between all these scenarios was that they evaluate the user's eye-based information in a dynamic 3D environment. Dynamic in the automotive context referred to the movement of the vehicle in normal traffic as well as head/eye movements of the user. In the human robot collaboration context, dynamic referred to the movements of the robots and objects as well as the user. The presented algorithms covered both these scenario fields.

- **How can we expand the spectrum of human-environment interaction by integrating additional eye-based input and output channels?**
  For this purpose Chapter 7 provided many example prototypes which were developed using the TIFoA toolkit described in Chapter 6. Section 7.1 presented two example applications for reference resolution in an automotive use case. These two systems were based on 2D maps. One of these systems was designed for a car driving in real traffic and the other was for a video simulation of a driving scenario. Section 7.1.2 presented 5 applications which use 3D environment reconstruction for interacting with buildings and billboards, as well as in-car functions and general analysis of the driver's gaze and head direction. Section 7.2 provided three example applications for using user's focus-of-attention in mixed-reality human robot collaboration. These examples covered many different features from dual reality to team and multi-site interaction. Section 7.3 provided two examples for single and multiparty human-environment interaction using user's position, head orientation and body pose for deriving different features in a multi-user scenario. Finally, Section 7.4 provided detail on three applications which were implemented based on the Peripheral View Analysis (PVA) tool. This tool is a part of the toolkit presented in Chapter 6. The first application of Section 7.4 explored the mixed reality realm for helping dementia patients. It gave an example how objects, depending on their position and size in user's periphery, can be brought into the interaction space. The second application, implemented a similar approach for the automotive field. In this example, the visibility of different objects were analyzed in a driving scene. Finally, the third application presented how peripheral analysis can assist a user via a context-sensitive augmented reality application in a car-workshop scenario.

8.2 Contributions

8.2.1 Scientific Publications

The following is the list of my publications during my PhD work.

Conferences

Mohammad Mehdi Moniri, Andreas Luxenburger, Winfried Schuffert, and Daniel Sonntag. 2016. **Real-time 3D peripheral view analysis**. In Proceedings of the 26th **International Conference on Artificial Reality and Telexistence and the 21st Eurographics Symposium on Virtual Environments (ICAT-EGVE '16)**. Eurographics Association, Goslar Germany, Germany, 37-44. DOI: https://doi.org/10.2312/egve.20161432

Mohammad Mehdi Moniri, Fabio Andres Espinosa Valcarcel, Dieter Merkel, Winfried Schuffert, and Tim Schwartz. 2016. **Hybrid team interaction in the mixed reality continuum**. In Proceedings of the 22nd **ACM Conference on Virtual Reality Software and Technology (VRST '16)**. ACM, New York, NY, USA, 335-336. DOI: https://doi.org/10.1145/2993369.2996318

Robert Neßelrath, Mohammad Mehdi Moniri and Michael Feld. 2016. **Combining Speech, Gaze, and Micro-gestures for the Multimodal Control of In-Car Functions**. In Proceedings of the 12th **International Conference on Intelligent Environments (IE)**, London, 2016, pp. 190-193. DOI: https://doi.org/10.1109/IE.2016.42

Mohammad Mehdi Moniri, Fabio Andres Espinosa Valcarcel, Dieter Merkel and Daniel Sonntag. 2016. **Human Gaze and Focus-of-Attention in Dual Reality Human-Robot Collaboration**. In Proceedings of the 12th **International Conference on Intelligent Environments (IE)**, London, 2016, pp. 238-241. DOI: https://doi.org/10.1109/IE.2016.54

Mohammad Mehdi Moniri, Dieter Merkel, Michael Feld and Christian Müller. 2016. **Incorporating the Driver's Focus of Attention into Automotive Applications in Real Traffic and in Simulator Setups**. In Proceedings of the 12th **International Conference on Intelligent Environments (IE)**, London, 2016, pp. 198-201. DOI: https://10.1109/IE.2016.44

Mohammad Mehdi Moniri, Daniel Sonntag, and Andreas Luxenburger. 2016. **Peripheral view calculation in virtual reality applications**. In Proceedings of the 2016 **ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp '16)**. ACM, New York, NY, USA, 333-336. DOI: https://doi.org/10.1145/2968219.2971391

Michael Barz, Mohammad Mehdi Moniri, Markus Weber, and Daniel Sonntag. 2016. **Multimodal multisensor activity annotation tool**. In Proceedings of the 2016 **ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp '16)**. ACM, New York, NY, USA, 17-20. DOI: https://doi.org/10.1145/2968219.2971459

Andreas Luxenburger, Alexander Prange, Mohammad Mehdi Moniri, and Daniel Sonntag. 2016. **MedicaLVR: towards medical remote collaboration using virtual reality**. In Proceedings of the 2016 **ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp '16)**. ACM, New York, NY, USA, 321-324. DOI: https://doi.org/10.1145/2968219.2971392

Tanja Schneeberger, Simon von Massow, Mohammad Mehdi Moniri, Angela Castronovo, Christian Mller, and Jan Macek. 2015. **Tailoring mobile apps for safe on-road usage: how an interaction concept enables safe interaction with hotel booking, news, Wolfram Alpha and Facebook**. In Proceedings of the 7th **International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '15)**. ACM, New York, NY, USA, 241-248. DOI: http://dx.doi.org/10.1145/2799250.2799264

Monika Mitrevska, Mohammad Mehdi Moniri, Robert Neßelrath, Tim Schwartz, Michael Feld, Yannick Körber, Matthieu Deru and Christian Müller. 2015. **SiAM - Situation-Adaptive Multimodal Interaction for Innovative Mobility Concepts of the Future**. In Proceedings of the 11th **International Conference on Intelligent Environments (IE)**, Prague, 2015, pp. 180-183. DOI: https://doi.org/10.1109/IE.2015.39

Nikolina Koleva, Sabrina Hoppe, Mohammad Mehdi Moniri, Maria Staudte and Andreas Bulling. 2015. **On the interplay between spontaneous spoken instructions and human visual behaviour in an indoor guidance task**. In Proceedings of the 37th **Annual Meeting of the Cognitive Science Society**. Austin, TX: Cognitive Science Society.

Mohammad Mehdi Moniri and Christian Müller. 2014. **EyeVIUS: Intelligent Vehicles in Intelligent Urban Spaces**. In Adjunct Proceedings of the 6th **International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '14)**. ACM, New York, NY, USA, 1-6. DOI:https://doi.org/10.1145/2667239.2667265

Mohammad Mehdi Moniri and Christian Müller. 2012. **Multimodal reference resolution for mobile spatial interaction in urban environments**. In Proceedings of the 4th **International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)**. ACM, New York, NY, USA, 241-248. DOI: http://dx.doi.org/10.1145/2390256.2390296

Angela Mahr, Michael Feld, Mohammad Mehdi Moniri and Rafael Math. 2012. **The contre (continuous tracking and reaction) task: A flexible approach for assessing driver cognitive workload with high sensitivity**. In Adjunct Proceedings of the 4th **International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12)**. ACM, Portsmouth, USA.

Mohammad Mehdi Moniri, Michael Feld and Christian Müller. 2012. **Personalized In-Vehicle Information Systems: Building an Application Infrastructure for Smart Cars in Smart Spaces**. In Proceedings of the 8th **International Conference on Intelligent Environments**. Guanajuato, Mexico, 2012, pp. 379-382.DOI:http://dx.doi.org/10.1109/IE.2012.40

Book Contribution

Mohammad Mehdi Moniri, Michael Feld. 2017. **Eye and Head Tracking for Focus of Attention Control in the Cockpit**. In: **Automotive User Interfaces**. HumanComputer Interaction Series. Springer, Cham. 2017. 249-272. DOI: https://doi.org/10.1007/978-3-319-49448-7_9

8.2.2 Supervised Theses

I have supervised the following theses at Saarland University:

Bachelor Theses:

- Framework for Analyzing Drivers Focus-of-Attention Based on Dynamic 3D Map Data, Bach-Thi Dinh (2016).
- System für die Aufmerksamkeitsanalyse des Fahrers Basierend auf Blickdaten, Dieter Merkel (2016).
- Accuracy Measurement of Low-Cost Eye Trackers in 2D and 3D Environments, Nicolas Erbach (2016).
- Detection of Saccades and Fixations in OpenDS User Studies: Combining the Data from Virtual and Real Environments, Oliver Jank (2016).
- Toolkit for Peripheral-View-Analysis in Interactive Mixed-Reality-Applications, Fabian Spaniol (2017).
- User Identification Based on Sensor Sata Analysis of Diverse Sources, Felix Scherzinger (2017).

Master Theses:

- Investigating the Limits of Bluetooth Low Energy for Indoor-Positioning - Based on Bancroft's Method and Particle Filters, Bernd Mechenbier (2014).
- Evaluating Visual Attention and Gestures Accuracy using Low Cost Motion Tracking Device, Vikrant Saxena (2016).

8.2.3 Engineering Contributions

Using the tools described in Chapter 6, 15 different research prototypes and demonstrations have been developed. These prototypes range from applications for reference resolution in automotive use cases (7 applications in Sections 7.1 and 7.1.2), mixed-reality prototypes (3 applications in Section 7.2), prototypes for single and multiparty human environment interaction (2 applications in Section 7.3) and prototypes for interactions based on peripheral view analysis (3 applications in Section 7.4).

8.2.4 Invited Talks and Demos for Industry

Based on the research and activities performed in this thesis, I was invited to give a talk and live demo at the following events.

- Talk and Demo: Application of Virtual Reality for Mid-Size Companies.
  Event: Industrie 4.0 Kongress (Germany, Hagen, 2016)
- Talk and Demo: Applications of Mixed Reality Technology for Human-Robot Collaboration.
  Event: 2. VDI-Fachkonferenz Augmented und Virtual Reality als Smart-Assistance (Germany, Munich, 2016)
- Demo: Remote Human-Robot Communication via Virtual Reality
  Event: DASA Arbeitswelt Ausstellung (Germany, Dortmund, 2016)
- Talk: Hand in Hand: Mensch-Roboter-Kollaboration in Industrie 4.0
  Event: Future Talk CeBIT 2017 (Germany, Hannover, 2017)
- Talk and Demo: Mixed Reality Production 4.0 - MR-Technologien für standortübergreifende Produktion in Industrie 4.0
  Event: Internationale Fachtagung für Qualität in der Kunststoffbranche (Germany, Würzburg, 2017)

8.2.5 Awards

During different stages of the thesis, I have received the following awards.

- Best Video/Demo Award at the Intelligent Environments Conference (IE12), 2012.
- Best Video/Demo Award at the Intelligent Environments Conference (IE16), 2016.
- Best Demo Award at the MATES 2016 conference, 2016.

8.2.6 Media Appearances

I gave the following press interviews about the results of my thesis.

- TV (2013): Sat1 (German TV channel)
  Topic: Auto der Zukunft (Car of the Future)
- TV (2016): SWR (German TV channel)
  Topic: Mixed-Reality Anwendungen für die Zukunft (Mixed-Reality Applications of the Future)
- TV (2016): SR (German TV channel)
  Topic: Mensch Roboter Kollaboration 4.0 (Human-Robot Collaboration 4.0)

- Press (2017): National Geographic
  Topic: Mixed Reality Human Robot Collaboration

- Press/TV (2017): ZDF, Euronews, Deutsche Welle, Reuters Thomson, Korea TV
  Topic: Mixed Reality Production 4.0

8.2.7 Research-Prototype Demonstrations

During the different stages of my thesis, the developed prototype was presented successfully at four CeBIT Expo events in Hannover (2012, 2014, 2016, 2017). Furthermore, the developed prototype was presented live to industry and political representatives at dozens of events between 2012 and 2017.

8.2.8 Industry Projects

Various companies have shown great interest in the results and the various prototypes of this thesis. I successfully acquired and managed a project in the exact field of this thesis (driver-environment interaction) for an industry customer. Currently there are several negotiations ongoing for future projects.

8.3 Future Work

The future work of the presented research can be divided into two general categories: science and engineering. Regarding the scientific future work, there are many open questions regarding human-environment interaction while considering the objects in the peripheral vision. The following is a list of interesting questions which can be addressed in the future work.

- What is the correlation between the model presented in Section 4.1.2.1 with a model which is based on real world data? As the presented model is based on an experiment conducted in virtual reality, the corresponding model for a real world scenario might be different.

- How do the colors of the object and its background (and the contrast) affect the model presented in Section 4.1.2.1?

- How does speed of the object or direction of its movement affect the presented model in Section 4.1.2.1?

- How do other moving objects in the scene affect the perception?
- How is the perception in the peripheral vision of a driver different than a person who is working closely with a robot?
- How does the perception of the virtual objects differ from real objects when they are both located in the peripheral vision of a user wearing a Mixed-Reality glass?
- How does crowding affect the presented model in Section 4.1.2.1?
- How is it possible to compensate for the eye-tracking measurement error in order to provide the possibility for a driver/passenger to interact with smaller objects outside of the vehicle? It remains a challenge to develop gaze-based interaction methods which compensate for the noise in the eye-tracking signals [Qvarfordt, 2017].

Considering the future engineering work, there are many promising applications which can be built on the basis of the research presented in this thesis. As mentioned before, many companies have introduced their plans for manufacturing autonomous cars. In such future cars, the passengers will have much more time to read, watch their favorite show, or interact with the environment outside the car. The latter is addressed in this thesis. Building such a system in a production car requires importing the environment modeling, and the algorithms described, into embedded systems inside the car. The following questions should be addressed in future work:

- How is it possible to import the described environment modeling and algorithms into embedded systems in a car?
- How is it possible to modify the algorithms in order to exploit the power of parallel computing of embedded GPUs?
- What is the performance of the system when it is used with an embedded dialog system? How long does it take for the passenger to get an answer from the system?
- If the model is imported into the cloud and the dialog system of the car should communicate with the cloud to receive the answer, how long does this communication take? Will it be still interesting for an automotive use case?
- How much resources in term of CPU power, SSD and RAM are needed for an embedded solution? How much resources are needed for an online solution?
- How should eye-tracking cameras be placed inside of the car so that the driver can refer to any place in the car via gaze?

Answers to these questions can encourage car manufacturing companies to integrate the presented research as part of a solution for their future driver-passenger environment interaction. The engineering task will be to develop a software which consumes fewer resources on such embedded devices. Regarding mixed-reality applications, there are also very interesting possibilities. As the described tools and techniques in this thesis are all standard and robust, in a very short time the presented mixed-reality prototypes can be engineered for use in a professional manufacturing line. As a matter of fact, there are now discussions ongoing with manufacturing plants to integrate the presented solution as a part of their existing platform. This way their manuals will not only be in traditional 2D material but also in interactive 3D mixed-reality application. In this case the user can move around and interact with the virtual content which is augmenting the physical device. However, it should be mentioned that it is not currently possible to include eye-tracking functionality in such systems. This is due to the fact that there are currently no commercial mixed-reality headsets which also include eye-tracking[1]. Nevertheless, current devices visualize the nose vector which can be used by the user to refer to objects in the environment. As an alternative solution it is possible to build custom hardware such as EyePICA as it is presented in Section 7.3.

[1]Magic Leap has introduced hardware which also includes eye-tracking. However, this hardware is not yet on the market. Source: https://www.magicleap.com/ (last time visited: 20.12.2017)

A APPENDIX

A.1 Solid Angle of a Visual Field

Each region of the HVF as defined by the Hatada model can be divided into two half ellipses with respect to their different vertical opening angles[1]. In order to calculate the solid angle of each visual field, the solid angle of each half ellipse is calculated separately in order to summarize the results afterwards. In this context, analytic solutions have been provided [Abbas et al., 2015]. Considering the case where an observer at distance h is looking at the center of an ellipse with axes a and b, we have

$$\Omega = \int_0^{\phi} \int_0^{\theta_1(\phi)} \sin(\theta)\ d\theta\ d\phi\ ,$$

with boundaries

$$\phi = 2\,\pi,\ \ \theta_1(\phi) = \arctan\left(\frac{r_1(\phi)}{h}\right)$$

and

$$r_1(\phi) = \frac{a\,b}{\sqrt{(a\sin(\phi))^2 + (b\cos(\phi))^2}}\ .$$

Since the visual fields of the Hatada model are defined by vertical and horizontal angles independent of a, b and h, the presented formula for calculating the solid angle of an ellipse by Abbas et al. is solved in order to convert these three parameters into the two given angles by the Hatada model. Integration over θ gives

$$\Omega = \int_0^{\phi} 1 - \cos\,\theta_1(\phi)\ d\phi\ .$$

[1] Circles are considered a type of an ellipse with equal axes.

Inserting the values for $\theta_1(\phi)$ and $r_1(\phi)$, we get

$$\Omega = \int_0^{\phi} 1 - \cos\left(\arctan\left(\frac{\frac{a\,b}{\sqrt{(a\sin(\phi))^2+(b\cos(\phi))^2}}}{h}\right)\right) d\phi \ .$$

Applying the trigonometric rule $\cos(\arctan(x)) = \frac{1}{\sqrt{1+x^2}}$ further yields:

$$\begin{aligned}\Omega &= \int_0^{\phi} 1 - \frac{1}{\sqrt{1+\left(\frac{\frac{a\,b}{\sqrt{(a\sin(\phi))^2+(b\cos(\phi))^2}}}{h}\right)^2}}\, d\phi \\ &= \int_0^{\phi} 1 - \frac{h}{\sqrt{h^2+\frac{a^2\,b^2}{a^2\sin(\phi)^2+b^2\cos(\phi)^2}}}\, d\phi \ .\end{aligned}$$

Figure A.1 illustrates a geometric interpretation of the parameters a and b in terms of the two visual field angles α and β for a given distance h. Here, we have

$$a = \tan(\alpha)\,h, \ \ b = \tan(\beta)\,h \ .$$

Using these values for a and b yields the final expression

$$\begin{aligned}\Omega &= \int_0^{\phi} 1 - \frac{h}{\sqrt{h^2+\frac{(\tan(\alpha)\,h)^2\,(\tan(\beta)\,h)^2}{(\tan(\alpha)\,h)^2\,\sin(\phi)^2+(\tan(\beta)\,h)^2\,\cos(\phi)^2}}} d\phi \\ &= \int_0^{\phi} 1 - \frac{1}{\sqrt{1+\frac{\tan(\alpha)^2\,\tan(\beta)^2}{\tan(\alpha)^2\,\sin(\phi)^2+\tan(\beta)^2\,\cos(\phi)^2}}} d\phi \ .\end{aligned}$$

This formula is used to calculate the solid angle of different visual fields of the Hatada model. In order to calculate the solid angle of a whole ellipse, it is necessary to set $\phi = 2\pi$ for the upper bound of the integral. However, as each of the visual fields is divided into two half ellipses, the upper bound of the integral of each half ellipse is set to be $\phi = \pi$. This is the final formula for calculating the solid angle of each visual field, only based on the horizontal and vertical angles of each field defined by the Hatada model. A numerical evaluation of the derived expression while inserting the corresponding angular values for each visual field yields to following total solid angles

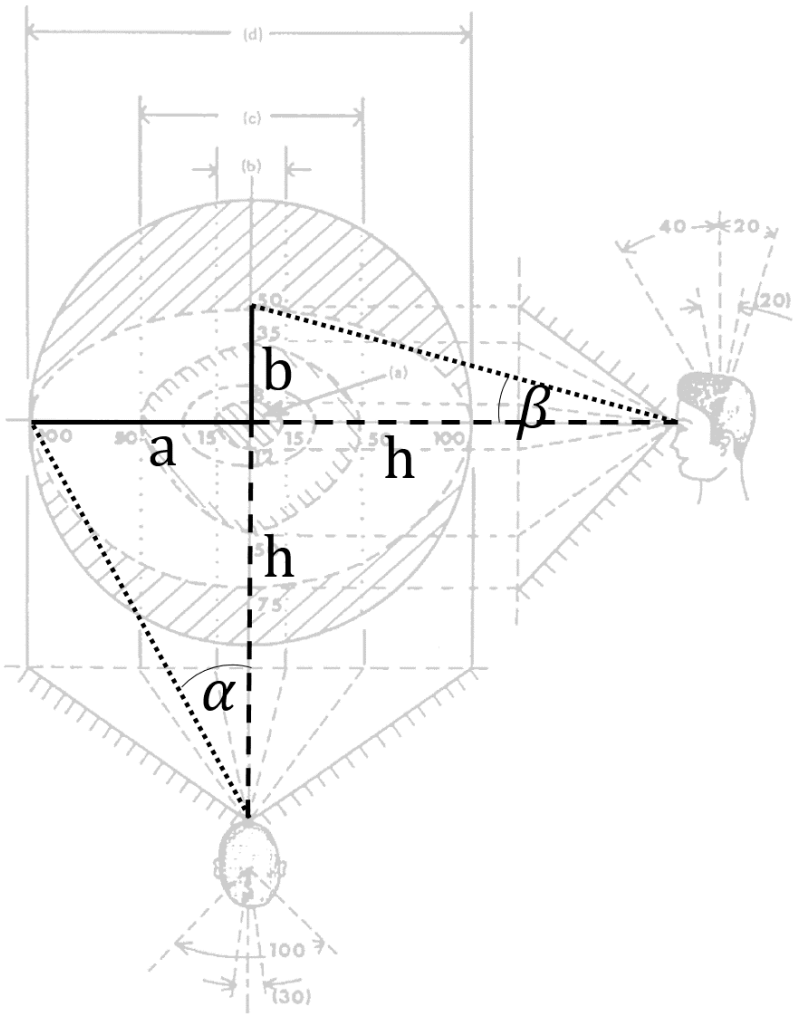

Figure A.1: Calculating the values a and b by considering the horizontal and vertical angle of the visual field together with the distance h. Figure based on [Hatada et al., 1980].

- $\Omega_{\text{discriminatory visual field}} = 0.0086\ sr$
- $\Omega_{\text{effective visual field}} = \Omega_{\text{effective-up}} + \Omega_{\text{effective-down}} = 0.1428\ sr$
- $\Omega_{\text{induced visual field}} = \Omega_{\text{induced-up}} + \Omega_{\text{induced-down}} = 2.0188\ sr$
- $\Omega_{\text{supplementary visual field}} = \Omega_{\text{supplementary-up}} + \Omega_{\text{supplementary-down}} = 4.1185\ sr$

Note that each of the listed values subsumes the total solid angle of the next smaller

visual field region. In order to obtain the exclusive solid angle of a visual field, the value of the next smaller field has to be subtracted.

A.2 Performance Analysis for Simulated Automotive Use Case

Any application based on eye tracking which has interest in the visibility of relevant objects in the peripheral visual field can benefit from the provided concepts and calculations. In this section, we describe our application which implements the described algorithm. In addition, we provide a quantitative evaluation in terms of a runtime analysis of the different steps of our approach. We also briefly introduce our online tool, which provides the possibility for researchers to become more familiar with the concepts and calculations presented in this paper.

In this section, we provide a quantitative evaluation of the different steps of our algorithm and give several benchmarks for a different number of objects of interest (OOEs) and sampling patches. The goal of our evaluation is to show the scalability and real-time performance of our approach for a varied number of OOEs. The corresponding runtimes have been derived from simple time stamps which were placed before and after the invocation of each procedure. Their differences encode absolute time spans in milliseconds. In order to reduce the influence of distortion factors like background processes, we averaged them over 10 frames and repeated each benchmark multiple times. Thus, the presented values constitute average values. Our hardware setup consists of an ASUS Desktop PC G20CB Series featuring an Intel Core i7-6700 CPU (3.40 GHz), 16 GB RAM, and a NVIDIA GeForce GTX 980 GPU. We measured the runtimes of each step of our algorithm including the projection of the 3D bounding boxes and the visual field regions onto a common plane, the sampling of the projected bounding boxes, the computation of the intersecting solid angle of the bounding box and each visual field, and finally the calculation of our acuity based visibility measure for each object. For the evaluation we used the described 3D model of the university, as it constitutes a typical outdoor scene with usual street layout and traffic. In the benchmarking, we varied the number of sampling patches for each projected bounding box, which determines the accuracy of the patch-based solid angle computation, and also the number of OOEs. In the following, we will describe each separately.

Table A.1 shows the measured runtimes for a varied number of sampling patches. The related scene contained one target object while featuring 508.3 K vertices in total. This means that for all variations the projection time is roughly the same (about 0.013 milliseconds). However, we find variations in the number of frames per sec-

# of patches	FPS	Sampling Time (ms)	Solid Angle Calculation Time (ms)
16 patches	75.04	0.0006	0.05380
100 patches	75.14	0.003	0.3017
400 patches	74.98	0.0035	1.0496
1600 patches	75.01	0.0184	3.9594
6400 patches	50.4	0.065	16.2518

Table A.1: Computation times for a varied number of sample patches. This parameter determines the accuracy of the solid angle calculation.

ond (FPS), the sampling time, and the time for calculating the solid angle. For all variations, the time for computing the eccentricity and the object's visibility were negligible as they were in the range of a few nanoseconds.

The number of rectangular sample patches of a projected bounding box is determined as the product of horizontal and vertical subdivisions. For example, in the case of 4 horizontal and 4 vertical subdivisions, we have 16 equal-sized patches. Table A.1 shows that an increased number of patches yields an increased sampling time, as expected, with a maximum computation time of 65 nanoseconds. The corresponding FPS value also does not undergo an explicit change until we reach a number of 6400 patches where it drops from 75 to 50.

The time for calculating the solid angle of the patches increases in the same manner as the number of the patches. Here, we can detect a linear relation between these two values. As we quadruple the number of the patches, the time for calculating the solid angle quadruples, too.

As we found a number of 100 patches for an OOE to constitute a reasonable compromise between the runtime (about 0.3 milliseconds) and the degree of approximation of the object's solid angle, we fix the number of patches for the following experiment and focus on a variation of the number of objects of interest. The following list shows our different variations in the number of OOEs (number of patches = 100):

- Baseline (0 target objects, 492.3 K vertices):
  The university model without any OOE
- Scene 1 (1 target object, 508.3 K vertices):
  The university model with one pedestrian as OOE
- Scene 2 (3 target objects, 510.1 K vertices):
  The university model with one pedestrian, one car, and one traffic sign as OOEs

- Scene 3 (9 target objects, 545.7 K vertices):
  The university model with three pedestrians, three cars, and three traffic signs as OOEs
- Scene 4 (27 target objects, 833.7 K vertices):
  The university model with twenty-one pedestrians, three cars, and three traffic signs as OOEs

For all of the listed variations, the number of FPS was between 75.14 for one target object and 74.5 for 27 target objects. The projection time and the sampling time were also less than 0.017 milliseconds and 0.0049 milliseconds, respectively. The solid angle calculation time for all variations was below 0.308 milliseconds and the time needed for computing the visibility measure and the eccentricity were negligible as again in the range of a few nanoseconds.

References

[Abbas et al., 2015] Abbas, M. I., Hammoud, S., Ibrahim, T., and Sakr, M. (2015). Analytical formulae to calculate the solid angle subtended at an arbitrarily positioned point source by an elliptical radiation detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 771(Supplement C):121 – 125.

[Abt, 1987] Abt, C. C. (1987). *Serious games*. University press of America.

[Admoni and Scassellati, 2017] Admoni, H. and Scassellati, B. (2017). Social eye gaze in human-robot interaction: A review. *J. Hum.-Robot Interact.*, 6(1):25–63.

[Akbarzadeh et al., 2006] Akbarzadeh, A., Frahm, J.-M., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Merrell, P., Phelps, M., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H., Nister, D., and Pollefeys, M. (2006). Towards urban 3d reconstruction from video. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 3DPVT '06, pages 1–8, Washington, DC, USA. IEEE Computer Society.

[Anagnostopoulos et al., 2017] Anagnostopoulos, V., Havlena, M., Kiefer, P., Giannopoulos, I., Schindler, K., and Raubal, M. (2017). Gaze-informed location-based services. *International Journal of Geographical Information Science*, 31(9):1770–1797.

[Anagnostopoulos and Kiefer, 2016] Anagnostopoulos, V. A. and Kiefer, P. (2016). Towards gaze-based interaction with urban outdoor spaces. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, pages 1706–1715, New York, NY, USA. ACM.

[Anton-Erxleben and Carrasco, 2013] Anton-Erxleben, K. and Carrasco, M. (2013). Attentional enhancement of spatial resolution: linking behavioural and neurophysiological evidence. *Nature Reviews Neuroscience*, 14(3):188–200.

[Bach, 2006] Bach, M. (2006). The freiburg visual acuity test-variability unchanged by post-hoc re-analysis. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 245(7):965–971.

[Bailey and Lovie, 1976] Bailey, I. L. and Lovie, J. E. (1976). New design principles for visual acuity letter charts. *Optometry and Vision Science*, 53(11):740–745.

[Balas et al., 2009] Balas, B., Nakano, L., and Rosenholtz, R. (2009). A summary-statistic representation in peripheral vision explains visual crowding. *Journal of Vision*, 9(12):13–13.

[Baldauf et al., 2010] Baldauf, M., Fröhlich, P., and Hutter, S. (2010). Kibitzer: A wearable system for eye-gaze-based mobile urban exploration. In *Proceedings of the 1st Augmented Human International Conference*, AH '10, pages 9:1–9:5, New York, NY, USA. ACM.

[Barfield et al., 1995] Barfield, W., Hendrix, C., Bjorneseth, O., Kaczmarek, K. A., and Lotens, W. (1995). Comparison of human sensory capabilities with technical specifications of virtual environment equipment. *Presence: Teleoper. Virtual Environ.*, 4(4):329–356.

[Bates et al., 2007] Bates, R., Donegan, M., Istance, H. O., Hansen, J. P., and Räihä, K.-J. (2007). Introducing cogain: communication by gaze interaction. *Universal Access in the Information Society*, 6(2):159–166.

[Biedert et al., 2010] Biedert, R., Buscher, G., Schwarz, S., Hees, J., and Dengel, A. (2010). Text 2.0. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 4003–4008, New York, NY, USA. ACM.

[Biljecki et al., 2017] Biljecki, F., Ledoux, H., and Stoter, J. (2017). Does a finer level of detail of a 3d city model bring an improvement for estimating shadows? In Abdul-Rahman, A., editor, *Advances in 3D Geoinformation*, pages 31–47. Springer, Cham.

[Biljecki et al., 2014] Biljecki, F., Ledoux, H., Stoter, J., and Zhao, J. (2014). Formalisation of the level of detail in 3d city modelling. *Computers, Environment and Urban Systems*, 48(Supplement C):1 – 15.

[Billinghurst et al., 2014] Billinghurst, M., Clark, A., Lee, G., et al. (2014). A survey of augmented reality. *Foundations and Trends® Human–Computer Interaction*, 8(2-3):73–272.

[Bimber and Raskar, 2006] Bimber, O. and Raskar, R. (2006). Modern approaches to augmented reality. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA. ACM.

[Biswas, 2016] Biswas, P. (2016). *Exploring the Use of Eye Gaze Controlled Interfaces in Automotive Environments*. Springer, Switzerland.

[Block, 2015] Block, N. (2015). The puzzle of perceptual precision. In Metzinger, T. K. and Windt, J. M., editors, *Open MIND*, chapter 5(T). MIND Group, Frankfurt am Main.

[Bolt, 1980] Bolt, R. A. (1980). Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '80, pages 262–270, New York, NY, USA. ACM.

[Bolt, 1981] Bolt, R. A. (1981). Gaze-orchestrated dynamic windows. In *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '81, pages 109–119, New York, NY, USA. ACM.

[Bulling, 2016] Bulling, A. (2016). Pervasive attentive user interfaces. *Computer*, 49(1):94–98.

[Bulling and Gellersen, 2010] Bulling, A. and Gellersen, H. (2010). Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing*, 9(4):8–12.

[Bulling et al., 2011] Bulling, A., Ward, J. A., Gellersen, H., and Troster, G. (2011). Eye movement analysis for activity recognition using electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):741–753.

[Campbell and Green, 1965] Campbell, F. and Green, D. (1965). Optical and retinal factors affecting visual resolution. *The Journal of Physiology*, 181(3):576.

[Cheung et al., 2005] Cheung, K.-m. G., Baker, S., and Kanade, T. (2005). Shape-from-silhouette across time part i: Theory and algorithms. *International Journal of Computer Vision*, 62(3):221–247.

[Clark, 1998] Clark, J. H. (1998). *Seminal Graphics*, chapter Hierarchical Geometric Models for Visible Surface Algorithms, pages 43–50. ACM, New York, NY, USA.

[Cohen, 2011] Cohen, R. A. (2011). *Cortical Magnification*, pages 718–719. Springer New York, New York, NY.

[Dewangan et al., 2016] Dewangan, K., Saha, A., Vaiapury, K., and Dasgupta, R. (2016). 3d environment reconstruction using mobile robot platform and monocular vision. In Choudhary, R. K., Mandal, J. K., Auluck, N., and Nagarajaram, H. A., editors, *Advanced Computing and Communication Technologies: Proceedings of the 9th ICACCT, 2015*, pages 213–221. Springer, Singapore.

[Dietz et al., 2017] Dietz, M., Schork, D., Damian, I., Steinert, A., Haesner, M., and André, E. (2017). Automatic detection of visual search for the elderly using eye and head tracking data. *KI - Künstliche Intelligenz*, 31(4):339–348.

[Dobbelstein et al., 2016] Dobbelstein, D., Walch, M., Köll, A., Şahin, O., Hartmann, T., and Rukzio, E. (2016). Reducing in-vehicle interaction complexity: Gaze-based mapping of a rotary knob to multiple interfaces. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, MUM '16, pages 311–313, New York, NY, USA. ACM.

[Duchowski, 2017] Duchowski, A. (2017). *Eye Tracking Methodology: Theory and Practice*. Springer, London.

[Duchowski, 2007] Duchowski, A. T. (2007). *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Duchowski et al., 2004] Duchowski, A. T., Cournia, N., and Murphy, H. (2004). Gaze-contingent displays: A review. *CyberPsychology and Behavior*, 7(6):621–634.

[Eldawy and Mokbel, 2015] Eldawy, A. and Mokbel, M. F. (2015). Spatialhadoop: A mapreduce framework for spatial data. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1352–1363. IEEE.

[Essig et al., 2012] Essig, K., Dornbusch, D., Prinzhorn, D., Ritter, H., Maycock, J., and Schack, T. (2012). Automatic analysis of 3d gaze coordinates on scene objects using data from eye-tracking and motion-capture systems. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 37–44, New York, NY, USA. ACM.

[Essig et al., 2016] Essig, K., Strenge, B., and Schack, T. (2016). Adamaas: Towards smart glasses for mobile and personalized action assistance. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '16, pages 46:1–46:4, New York, NY, USA. ACM.

[Fairclough, 2011] Fairclough, S. H. (2011). *Physiological Computing: Interfacing with the Human Nervous System*, pages 1–20. Springer Netherlands, Dordrecht.

[Fletcher et al., 2003] Fletcher, L., Apostoloff, N., Petersson, L., and Zelinsky, A. (2003). Vision in and out of vehicles. *IEEE Intelligent Systems*, 18(3):12–17.

[Fletcher et al., 2005] Fletcher, L., Loy, G., Barnes, N., and Zelinsky, A. (2005). Correlating driver gaze with the road scene for driver assistance systems. *Robotics and Autonomous Systems*, 52(1):71 – 84. Advances in Robot Vision.

[Fletcher and Zelinsky, 2008] Fletcher, L. and Zelinsky, A. (2008). Context sensitive driver assistance based on gaze – road scene correlation. In Khatib, O., Kumar, V., and Rus, D., editors, *Experimental Robotics: The 10th International Symposium on Experimental Robotics*, volume 39, pages 287–296. Springer, Berlin, Heidelberg.

[Fletcher and Zelinsky, 2009] Fletcher, L. and Zelinsky, A. (2009). Driver inattention detection based on eye gaze-road event correlation. *International Journal of Robotics Research*, 28(6):774–801.

[Fotheringham and Rogerson, 2013] Fotheringham, S. and Rogerson, P. (2013). *Spatial analysis and GIS*. CRC Press.

[Furukawa et al., 2011] Furukawa, R., Sagawa, R., Delaunoy, A., and Kawasaki, H. (2011). Multiview projectors/cameras system for 3d reconstruction of dynamic scenes. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1602–1609. IEEE.

[Gans, 2001] Gans, R. E. (2001). Video-oculography: A new diagnostic technology for vestibular patients. *The Hearing Journal*, 54(5):40–42.

[Gava and Stricker, 2015] Gava, C. C. and Stricker, D. (2015). Sphera - a unifying structure from motion framework for central projection cameras. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications - Volume 3: VISAPP, (VISIGRAPP 2015)*, pages 285–293. INSTICC, SciTePress.

[Geiger et al., 2011] Geiger, A., Ziegler, J., and Stiller, C. (2011). Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968. IEEE.

[Giannopoulos, 2016] Giannopoulos, I. (2016). *Supporting Wayfinding Through Mobile Gaze-Based Interaction*. PhD thesis, ETH Zürich.

[Gross, 1994] Gross, M. (1994). *Visual computing: the integration of computer graphics, visual perception and imaging*. Springer-Verlag Berlin Heidelberg.

[Guenther, 2015] Guenther, B. (2015). *Modern optics*. OUP Oxford.

[Guo et al., 2016] Guo, H., Li, X., Wang, W., Lv, Z., Wu, C., and Xu, W. (2016). An event-driven dynamic updating method for 3d geo-databases. *Geo-spatial Information Science*, 19(2):140–147.

[Gupta et al., 2016] Gupta, K., Lee, G. A., and Billinghurst, M. (2016). Do you see what i see? the effect of gaze tracking on task space remote collaboration. *IEEE Transactions on Visualization and Computer Graphics*, 22(11):2413–2422.

[Guptill et al., 1995] Guptill, S. C., , and Morrison, J. L., editors (1995). *Elements of Spatial Data Quality*. International Cartographic Association. Pergamon, Amsterdam.

[Hansen and Ji, 2010] Hansen, D. W. and Ji, Q. (2010). In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):478–500.

[Hatada et al., 1980] Hatada, T., Sakata, H., and Kusaka, H. (1980). Psychophysical analysis of the "sensation of reality" induced by a visual wide-field display. *SMPTE Journal*, 89(8):560–569.

[He et al., 1996] He, S., Cavanagh, P., and Intriligator, J. (1996). Attentional resolution and the locus of visual awareness. *Nature*, 383(6598):334–337.

[Hilliges et al., 2012] Hilliges, O., Kim, D., Izadi, S., Weiss, M., and Wilson, A. (2012). Holodesk: Direct 3d interactions with a situated see-through display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2421–2430, New York, NY, USA. ACM.

[Hitzel, 2015] Hitzel, E. (2015). *Effects of Peripheral Vision on Eye Movements: A Virtual Reality Study on Gaze Allocation in Naturalistic Tasks*.

[Irwin, 1992] Irwin, D. E. (1992). *Visual Memory Within and Across Fixations*, pages 146–165. Springer New York, New York, NY.

[Ishiguro and Rekimoto, 2011] Ishiguro, Y. and Rekimoto, J. (2011). Peripheral vision annotation: Noninterference information presentation method for mobile augmented reality. In *Proceedings of the 2nd Augmented Human International Conference*, AH '11, pages 8:1–8:5, New York, NY, USA. ACM.

[Izadi et al., 2011] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. (2011). Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 559–568, New York, NY, USA. ACM.

[Jacob, 1990] Jacob, R. J. K. (1990). What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 11–18, New York, NY, USA. ACM.

[Kán and Kaufmann, 2013] Kán, P. and Kaufmann, H. (2013). Differential irradiance caching for fast high-quality light transport between virtual and real worlds.

In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 133–141. IEEE.

[Kang et al., 2015] Kang, S., Kim, B., Han, S., and Kim, H. (2015). Do you see what i see: Towards a gaze-based surroundings query processing system. In *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '15, pages 93–100, New York, NY, USA. ACM.

[Kaňuk et al., 2013] Kaňuk, J., Hofierka, J., and Gallay, M. (2013). Using virtual 3-d city models in temporal analysis of urban transformations. In *Geoinformatics for City Transformations, Symposium GIS Ostrava 2013*, Ostrava.

[Khamis et al., 2016] Khamis, M., Alt, F., and Bulling, A. (2016). Challenges and design space of gaze-enabled public displays. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, pages 1736–1745, New York, NY, USA. ACM.

[Khamis et al., 2015] Khamis, M., Bulling, A., and Alt, F. (2015). Tackling challenges of interactive public displays using gaze. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, UbiComp/ISWC'15 Adjunct, pages 763–766, New York, NY, USA. ACM.

[Khamis et al., 2017] Khamis, M., Hoesl, A., Klimczak, A., Reiss, M., Alt, F., and Bulling, A. (2017). Eyescout: Active eye tracking for position and movement independent gaze interaction with large public displays. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 155–166, New York, NY, USA. ACM.

[Komatsubara, 2008] Komatsubara, A. (2008). *Human error*. Maruzen.

[Ladikos, 2011] Ladikos, S. (2011). *Real-Time Multi-View 3D Reconstruction for Interventional Environments*. PhD thesis, Technische Universität München, 2011.

[Langlois, 2013] Langlois, S. (2013). Adas hmi using peripheral vision. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '13, pages 74–81, New York, NY, USA. ACM.

[Lee et al., 2016] Lee, Y., Masai, K., Kunze, K., Sugimoto, M., and Billinghurst, M. (2016). A remote collaboration system with empathy glasses. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, Mexico.

[Lev et al., 2014] Lev, M., Yehezkel, O., and Polat, U. (2014). Uncovering foveal crowding? *Scientific reports*, 4(4067).

[Levi, 2011] Levi, D. M. (2011). Visual crowding. *Current Biology*, 21(18):R678 – R679.

[Levine, 1981] Levine, J. (1981). *An Eye-controlled Computer*. RC 8857. IBM Research Division, T.J. Watson Research Center.

[Lopez, 2010] Lopez, J. S. A. (2010). *Off-the-Shelf Gaze Interaction*. PhD thesis, IT University of Copenhagen, Denmark.

[Majaranta and Bulling, 2014] Majaranta, P. and Bulling, A. (2014). *Eye Tracking and Eye-Based Human–Computer Interaction*, pages 39–65. Springer London, London.

[Mardanbegi and Hansen, 2011] Mardanbegi, D. and Hansen, D. W. (2011). Mobile gaze-based screen interaction in 3d environments. In *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, NGCA '11, pages 2:1–2:4, New York, NY, USA. ACM.

[Masket, 1957] Masket, A. V. (1957). Solid angle contour integrals, series, and tables. *Review of Scientific Instruments*, 28(3):191–197.

[Mat et al., 2016] Mat, R. C., Nordin, N., Zulkifli, A. N., and Yusof, S. A. M. (2016). Suitability of online 3d visualization technique in oil palm plantation management. In *AIP Conference Proceedings*, volume 1761. AIP Publishing.

[Mat et al., 2014] Mat, R. C., Shariff, A. R. M., Zulkifli, A. N., Rahim, M. S. M., and Mahayudin, M. H. (2014). Using game engine for 3d terrain visualisation of gis data: A review. *IOP Conference Series: Earth and Environmental Science*, 20(1).

[Math et al., 2013] Math, R., Mahr, A., Moniri, M. M., and Müller, C. (2013). Opends: A new open-source driving simulator for research. *GMM-Fachbericht 75 -AmE 2013*.

[Mathar, 2014] Mathar, R. J. (2014). *Solid angle of a rectangular plate*. Max-Planck Institute of Astronomy, Königstuhl.

[Mather, 2006] Mather, G. (2006). *Foundations of perception*. Taylor and Francis.

[Mauderer, 2017] Mauderer, M. (2017). *Augmenting visual perception with gaze-contingent displays*. PhD thesis, University of St Andrews.

[Mautz, 2012] Mautz, R. (2012). *Indoor positioning technologies*. PhD thesis, Institute of Geodesy and Photogrammetry,Department of Civil, Environmental and Geomatic Engineering, ETH Zurich.

[Mckee and Nakayama, 1984] Mckee, S. P. and Nakayama, K. (1984). The detection of motion in the peripheral visual field. *Vision Research*, 24(1):25 – 32.

[Merlo et al., 2012] Merlo, A., Dalc, L., and Fantini, F. (2012). Game engine for cultural heritage: New opportunities in the relation between simplified models and database. In *18th International Conference on Virtual Systems and Multimedia*, pages 623–628, Milan, Italy. IEEE.

[Metzger, 1993] Metzger, P. J. (1993). Adding reality to the virtual. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, pages 7–13. IEEE.

[Michael and Chen, 2005] Michael, D. R. and Chen, S. L. (2005). *Serious Games: Games That Educate, Train, and Inform*. Muska and Lipman/Premier-Trade.

[Milgram and Kishino, 1994] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, vol. E77-D(12):1321–1329.

[Mohring et al., 2004] Mohring, M., Lessig, C., and Bimber, O. (2004). Video see-through ar on consumer cell-phones. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '04, pages 252–253, Washington, DC, USA. IEEE Computer Society.

[Mokatren et al., 2017] Mokatren, M., Kuflik, T., and Shimshoni, I. (2017). Exploring the potential of a mobile eye tracker as an intuitive indoor pointing device: A case study in cultural heritage. *Future Generation Computer Systems*.

[Moniri and Müller, 2012] Moniri, M. M. and Müller, C. (2012). Multimodal reference resolution for mobile spatial interaction in urban environments. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '12, pages 241–248, New York, NY, USA. ACM.

[Mordohai et al., 2007] Mordohai, P., Frahm, J.-M., Akbarzadeh, A., Clipp, B., Engels, C., Gallup, D., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewnius, H., Towles, H., Welch, G., Yang, R., Pollefeys, M., and Nistr, D. (2007). Real-time video-based reconstruction of urban environments. In Remondino, F. and Sabry, E.-H., editors, *3D-ARCH 2007: 3D virtual reconstruction and visualization of complex architectures : ISPRS WG V/4 workshop ; 12 - 13 July, 2007, ETH Zurich, Switzerland*, volume XXXVI-5/W47 of *International*

archives of the photogrammetry, remote sensing and spatial information sciences. 3D- ARCH.

[Obe and Hsu, 2011] Obe, R. and Hsu, L. (2011). *PostGIS in Action*. Manning Publications Co., Greenwich, CT, USA.

[Orlosky et al., 2015a] Orlosky, J., Kiyokawa, K., Toyama, T., and Sonntag, D. (2015a). Halo content: Context-aware viewspace management for non-invasive augmented reality. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pages 369–373, New York, NY, USA. ACM.

[Orlosky et al., 2015b] Orlosky, J., Toyama, T., Kiyokawa, K., and Sonntag, D. (2015b). Modular: Eye-controlled vision augmentations for head mounted displays. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1259–1268.

[Orlosky, 2016] Orlosky, J. E. (2016). *Adaptive Display of Virtual Content for Improving Usability and Safety in Mixed and Augmented Reality*. PhD thesis, Graduate School of Information Science and Technology, Osaka University.

[Palummo, 2017] Palummo, A. (2017). From the Road Sign to the Map: 3d Modeling in Support of the Urban and Rural Road Conditions. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 77–80.

[Pearson, 1895] Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242.

[Peddie, 2017] Peddie, J. (2017). *Augmented Reality: Where We Will All Live*. Springer.

[Pelli et al., 2004] Pelli, D. G., Palomares, M., and Majaj, N. J. (2004). Crowding is unlike ordinary masking: Distinguishing feature integration from detection. *Journal of vision*, 4(12):12.

[Pfeiffer and Renner, 2014] Pfeiffer, T. and Renner, P. (2014). Eyesee3d: A low-cost approach for analyzing mobile 3d eye tracking data using computer vision and augmented reality technology. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 195–202, New York, NY, USA. ACM.

[Pfeiffer et al., 2016] Pfeiffer, T., Renner, P., and Pfeiffer-Lemann, N. (2016). EyeSee3D 2.0: Model-based Real-time Analysis of Mobile Eye-Tracking in Static and Dynamic Three-Dimensional Scenes. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research and Applications*, pages 189–196. ACM Press.

[Piumsomboon et al., 2017a] Piumsomboon, T., Lee, G., Lindeman, R. W., and Billinghurst, M. (2017a). Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 36–39, Los Angeles, CA, USA.

[Piumsomboon et al., 2017b] Piumsomboon, T., Lee, Y., Lee, G. A., Dey, A., and Billinghurst, M. (2017b). Empathic mixed reality: Sharing what you feel and interacting with what you see. In *2017 International Symposium on Ubiquitous Virtual Reality (ISUVR)*, pages 38–41, Japan.

[Pollefeys et al., 2008] Pollefeys, M., Nistér, D., Frahm, J.-M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.-J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., and Towles, H. (2008). Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2):143–167.

[Prange et al., 2015] Prange, A., Toyama, T., and Sonntag, D. (2015). Towards gaze and gesture based human-robot interaction for dementia patients. *AAAI Fall Symposium Series*.

[Purves et al., 2001] Purves, D., Augustine, G. J., Fitzpatrick, D., Katz, L. C., LaMantia, A.-S., McNamara, J. O., and Williams, S. M. (2001). Neuroscience. sunderland. *MA: Sinauer Associates*.

[Qodseya et al., 2016] Qodseya, M., Sanzari, M., Ntouskos, V., and Pirri, F. (2016). A3d: A device for studying gaze in 3d. In Hua, G. and Jégou, H., editors, *Computer Vision – ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I*, pages 572–588, Cham. Springer.

[Quimby, 2006] Quimby, R. S. (2006). *Photonics and lasers: an introduction*. John Wiley and Sons.

[Qvarfordt, 2017] Qvarfordt, P. (2017). The handbook of multimodal-multisensor interfaces. chapter Gaze-informed Multimodal Interaction, pages 365–402. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA.

[Raskar et al., 2001] Raskar, R., Welch, G., Low, K.-L., and Bandyopadhyay, D. (2001). Shader lamps: Animating real objects with image-based illumination. In Gortler, S. J. and Myszkowski, K., editors, *Rendering Techniques 2001: Proceedings of the Eurographics Workshop in London, United Kingdom, June 25–27, 2001*, pages 89–102. Springer, Vienna.

[Reddy, 1997] Reddy, M. (1997). *Perceptually modulated level of detail for virtual environments*. PhD thesis, University of Edinburgh, UK.

[Reddy, 2001] Reddy, M. (2001). Perceptually optimized 3d graphics. *IEEE Computer Graphics and Applications*, 21(5):68–75.

[Reingold et al., 2003] Reingold, E. M., Loschky, L. C., McConkie, G. W., and Stampe, D. M. (2003). Gaze-contingent multiresolutional displays: An integrative review. *Human Factors*, 45(2):307–328. PMID: 14529201.

[Reithinger et al., 2006] Reithinger, N., Gebhard, P., Löckelt, M., Ndiaye, A., Pfleger, N., and Klesen, M. (2006). Virtualhuman: Dialogic and affective interaction with virtual characters. In *Proceedings of the 8th International Conference on Multimodal Interfaces*, ICMI '06, pages 51–58, New York, NY, USA. ACM.

[Renner et al., 2011] Renner, P., Ldike, N., Wittrowski, J., and Pfeiffer, T. (2011). Towards Continuous Gaze-Based Interaction in 3D Environments - Unobtrusive Calibration and Accuracy Monitoring. In Bohn, C.-A. and Mostafawy, S., editors, *Proceedings of the Workshop Virtuelle und Erweiterte Realität 2011*, pages 13–24. Shaker Verlag.

[Renner and Pfeiffer, 2014] Renner, P. and Pfeiffer, T. (2014). Model-based acquisition and analysis of multimodal interactions for improving human-robot interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 361–362, New York, NY, USA. ACM.

[Renner and Pfeiffer, 2017a] Renner, P. and Pfeiffer, T. (2017a). Attention guiding techniques using peripheral vision and eye tracking for feedback in augmented-reality-based assistance systems. In *3D User Interfaces (3DUI), 2017 IEEE Symposium on*, pages 186–194. IEEE.

[Renner and Pfeiffer, 2017b] Renner, P. and Pfeiffer, T. (2017b). Evaluation of attention guiding techniques for augmented reality-based assistance in picking and assembly tasks. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces Companion*, IUI '17 Companion, pages 89–92, New York, NY, USA. ACM.

[Renner et al., 2014] Renner, P., Pfeiffer, T., and Wachsmuth, I. (2014). Spatial references with gaze and pointing in shared space of humans and robots. In Freksa, C., Nebel, B., Hegarty, M., and Barkowsky, T., editors, *Spatial Cognition IX: International Conference, Spatial Cognition 2014, Bremen, Germany, September 15-19, 2014. Proceedings*, pages 121–136, Cham. Springer.

[Rovamo and Virsu, 1979] Rovamo, J. and Virsu, V. (1979). An estimation and application of the human cortical magnification factor. *Experimental Brain Research*, 37(3):495–510.

[Saraiji et al., 2015] Saraiji, M. Y., Fernando, C. L., Minamizawa, K., and Tachi, S. (2015). Development of mutual telexistence system using virtual projection of operator's egocentric body images. In *Proceedings of the 25th International Conference on Artificial Reality and Telexistence and 20th Eurographics Symposium on Virtual Environments*, ICAT - EGVE '15, pages 125–132, Kyoto, Japan. Eurographics Association.

[SARAIJI et al., 2016] SARAIJI, M. Y., FERNANDO, C. L., MINAMIZAWA, K., and TACHI, S. (2016). Study on telexistence lxxxv layered presence: Expanding visual presence using simultaneously operated telexistence avatars.

[Saraiji et al., 2014a] Saraiji, M. Y., Fernando, C. L., Mizushina, Y., Kamiyama, Y., Minamizawa, K., and Tachi, S. (2014a). Enforced telexistence: Teleoperating using photorealistic virtual body and haptic feedback. In *SIGGRAPH Asia 2014 Emerging Technologies*, SA '14, pages 7:1–7:2, Shenzhen, China. ACM.

[Saraiji et al., 2014b] Saraiji, M. Y., Mizushina, Y., Fernando, C. L., Furukawa, M., Kamiyama, Y., Minamizawa, K., and Tachi, S. (2014b). Enforced telexistence. In *ACM SIGGRAPH 2014 Posters*, SIGGRAPH '14, pages 49:1–49:1, Vancouver, Canada. ACM.

[Saraiji et al., 2016] Saraiji, M. Y., Sugimoto, S., Fernando, C. L., Minamizawa, K., and Tachi, S. (2016). Layered telepresence: Simultaneous multi presence experience using eye gaze based perceptual awareness blending. In *ACM SIGGRAPH 2016 Emerging Technologies*, SIGGRAPH '16, pages 14:1–14:2, Anaheim, California. ACM.

[Schmitz and Moniri, 2009] Schmitz, M. and Moniri, M. M. (2009). Burgomaster and pedro - a pervasive multi-player game for rural tourism. In *2009 Conference in Games and Virtual Worlds for Serious Applications*, pages 205–208.

[Schwartz, 2012] Schwartz, T. (2012). *The always best positioned paradigm for mobile indoor applications*. PhD thesis, Universität des Saarlandes, Postfach 151141, 66041 Saarbrücken.

[Seitz et al., 2006] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1 of *CVPR '06*, pages 519–528, Washington, DC, USA. IEEE Computer Society.

[Selker et al., 2001] Selker, T., Lockerd, A., and Martinez, J. (2001). Eye-r, a glasses-mounted eye motion detection interface. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, pages 179–180, New York, NY, USA. ACM.

[Seneviratne et al., 2017] Seneviratne, S., Hu, Y., Nguyen, T., Lan, G., Khalifa, S., Thilakarathna, K., Hassan, M., and Seneviratne, A. (2017). A survey of wearable devices and challenges. *IEEE Communications Surveys Tutorials*, 19(4):2573–2620.

[Sharkawi et al., 2008] Sharkawi, K., Ujang, M. U., and Abdul-Rahman, A. (2008). 3d navigation system for virtual reality based on 3d game engine. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(PART B4).

[Smith and Atchison, 1997] Smith, G. and Atchison, D. A. (1997). *The Eye and Visual Optical Instruments:*. Cambridge University Press.

[Snowden et al., 2006] Snowden, R., Thompson, P., and Troscianko, T. (2006). *Basic Vision: An Introduction to Visual Perception*. Oxford University Press.

[Snowden et al., 2012] Snowden, R., Thompson, P., and Troscianko, T. (2012). *Basic vision: an introduction to visual perception*. Oxford University Press.

[Sonntag, 2015] Sonntag, D. (2015). Kognit - cognitive assistants for dementia patients. In *Proceedings of FSS-15 Cognitive Assistance in Government and Public Sector Applications*. AAAI Fall Symposium.

[Spalton et al., 2013] Spalton, D. J., Hitchings, R. A., and Hunter, P. (2013). *Atlas of clinical ophthalmology*.

[Stellmach and Dachselt, 2013] Stellmach, S. and Dachselt, R. (2013). Still looking: Investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 285–294, New York, NY, USA. ACM.

[Strasburger et al., 2011] Strasburger, H., Rentschler, I., and Jttner, M. (2011). Peripheral vision and pattern recognition: A review. *Journal of Vision*, 11(5):13.

[Sun et al., 2002] Sun, Y., Paik, J. K., Koschan, A., and Abidi, M. A. (2002). 3d reconstruction of indoor and outdoor scenes using a mobile range scanner. In *Object recognition supported by user interaction for service robots*, volume 3, pages 653–656. IEEE.

[Sundstedt, 2010] Sundstedt, V. (2010). Gazing at games: Using eye tracking to control virtual characters. In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH '10, pages 5:1–5:160, New York, NY, USA. ACM.

[Tacca, 2011] Tacca, M. C. (2011). Commonalities between perception and cognition. *Frontiers in psychology*, 2(358):7–16.

[Tawari et al., 2014a] Tawari, A., Chen, K. H., and Trivedi, M. M. (2014a). Where is the driver looking: Analysis of head, eye and iris for robust gaze zone estimation. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 988–994. IEEE.

[Tawari et al., 2014b] Tawari, A., Mgelmose, A., Martin, S., Moeslund, T. B., and Trivedi, M. M. (2014b). Attention estimation by simultaneous analysis of viewer and view. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1381–1387. IEEE.

[Tönnis and Klinker, 2014] Tönnis, M. and Klinker, G. (2014). Boundary conditions for information visualization with respect to the user's gaze. In *Proceedings of the 5th Augmented Human International Conference*, AH '14, pages 44:1–44:8, New York, NY, USA. ACM.

[Tönnis and Klinker, 2014] Tönnis, M. and Klinker, G. (2014). Placing information near to the gaze of the user. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 377–378. IEEE.

[Toyama, 2015] Toyama, T. (2015). *Towards wearable attention-aware systems in everyday environments*. PhD thesis, Technische Universität Kaiserslautern.

[Toyama and Sonntag, 2015] Toyama, T. and Sonntag, D. (2015). *Towards Episodic Memory Support for Dementia Patients by Recognizing Objects, Faces and Text in Eye Gaze*, pages 316–323. Springer, Cham.

[Toyama et al., 2014] Toyama, T., Sonntag, D., Dengel, A., Matsuda, T., Iwamura, M., and Kise, K. (2014). A mixed reality head-mounted text translation system using eye gaze input. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, pages 329–334, New York, NY, USA. ACM.

[Toyama et al., 2015] Toyama, T., Sonntag, D., Orlosky, J., and Kiyokawa, K. (2015). Attention engagement and cognitive state analysis for augmented reality text display functions. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pages 322–332, New York, NY, USA. ACM.

[van Koningsbruggen and Buonocore, 2013] van Koningsbruggen, M. G. and Buonocore, A. (2013). Mechanisms behind perisaccadic increase of perception. *Journal of Neuroscience*, 33(28):11327–11328.

[van Rheden et al., 2017] van Rheden, V., Maurer, B., Smit, D., Murer, M., and Tscheligi, M. (2017). Laserviz: Shared gaze in the co-located physical world. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '17, pages 191–196, New York, NY, USA. ACM.

[Varadharajan, 2012] Varadharajan, L. S. (2012). *Spatial Vision and Pattern Perception*, pages 109–119. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Vasli et al., 2016] Vasli, B., Martin, S., and Trivedi, M. M. (2016). On driver gaze estimation: Explorations and fusion of geometric and data driven approaches. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 655–660. IEEE.

[Vidal et al., 2012] Vidal, M., Turner, J., Bulling, A., and Gellersen, H. (2012). Wearable eye tracking for mental health monitoring. *Computer Communications*, 35(11):1306 – 1311.

[Vora et al., 2017] Vora, S., Rangesh, A., and Trivedi, M. M. (2017). On generalizing driver gaze zone estimation using convolutional neural networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 849–854. IEEE.

[Wahlster, 2003] Wahlster, W. (2003). Smartkom: Symmetric multimodality in an adaptive and reusable dialogue shell. In *Proceedings of the human computer interaction status conference*, volume 3, pages 47–62. Berlin, Germany.

[Wahlster and Müller, 2013] Wahlster, W. and Müller, C. (2013). Multimodale dialogsysteme für interaktive anwendungen im fahrzeug. *at–Automatisierungstechnik*, 61(11):777–783.

[Ware, 2013] Ware, C. (2013). *Information Visualization (Third Edition)*. Interactive Technologies. Morgan Kaufmann, Boston.

[Whitney and Levi, 2011] Whitney, D. and Levi, D. M. (2011). Visual crowding: a fundamental limit on conscious perception and object recognition. *Trends in Cognitive Sciences*, 15(4):160 – 168.

[Wilkinson, 2016] Wilkinson, P. (2016). A brief history of serious games. In Dörner, R., Göbel, S., Kickmeier-Rust, M., Masuch, M., and Zweig, K., editors, *Entertainment Computing and Serious Games: International GI-Dagstuhl Seminar 15283, Dagstuhl Castle, Germany, July 5-10, 2015, Revised Selected Papers*, pages 17–41. Springer, Cham.

[Yanagi et al., 2015] Yanagi, T., Fernando, C. L., Saraiji, M. H. D. Y., Minamizawa, K., Tachi, S., and Kishi, N. (2015). Transparent cockpit using telexistence. In *2015 IEEE Virtual Reality (VR)*, pages 311–312, Arles, France. IEEE.

[Yu et al., 2012] Yu, C., Schermerhorn, P., and Scheutz, M. (2012). Adaptive eye gaze patterns in interactions with human and artificial agents. *ACM Trans. Interact. Intell. Syst.*, 1(2):13:1–13:25.

[Zhang et al., 2015] Zhang, Y., Gibson, G. M., Hay, R., Bowman, R. W., Padgett, M. J., and Edgar, M. P. (2015). A fast 3d reconstruction system with a low-cost camera accessory. *Scientific reports*, 5.

[Zlatanova, 2006] Zlatanova, S. (2006). *Innovations in 3D Geo Information Systems*, pages 1–14. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Zlatanova et al., 2012] Zlatanova, S., Stoter, J., and Isikdag, U. (2012). Standards for exchange and storage of 3d information: Challenges and opportunities for emergency response. In *Proceedings of the 4th International Conference on Cartography and GIS, Volume 2, Albena, June 2012, pages 17-28*. International Cartographic Association.