UNIVERSITÀ di VERONA

UNIVERSITY OF VERONA

DEPARTMENT OF COMPUTER SCEINCE

GRADUATE SCHOOL OF NATURAL SCIENCES AND ENGINEERING

PHD IN COMPUTER SCIENCE

With funding by National Institute for Insurance against Accidents at Work (INAIL)

CYCLE / YEAR of initial enrolment 34/2018

PERCEPTION-DRIVEN APPROACHES TO REAL-TIME REMOTE IMMERSIVE VISUALIZATION

S.S.D. (Disciplinary Sector) ING-INF/05

Coordinator: Prof. PAOLO FIORINI

Signature _____

Tutor: Prof. PAOLO FIORINI and DR. NIKHIL DESHPANDE

Signature _____

PhD candidate: YONAS TEODROS TEFERA

Signature _____

PERCEPTION-DRIVEN APPROACHES TO REAL-TIME REMOTE IMMERSIVE VISUALIZATION
Yonas Teodros Tefera
PhD thesis
Verona, 15 July 2022
ISBN ----------

University of Verona
Research Office – National and International PhD programmes
ph: 045.802.8608 – fax 045.802.8411 – Via Giardino Giusti 2 – 37129 Verona

To Workea, my Mum.

Acknowledgements

This thesis would have been impossible without numerous people's advice, feedback, inspiration, and love. I would like to thank my supervisors, Dr. Nikhil Deshpande and Prof. Paolo Fiorini, for their continuous support, mentorship, and guidance. I am grateful to have found two scholarly supervisors to share and discuss my ideas. They gave me inspiring ideas to test and challenge me to strive for higher goals. I would like to give special thanks to a collaborator and friend, Dr. Dario Mazzanti; Some ideas would never develop into successful projects without someone to share. His deep understanding of computer graphics helped me to implement some of the virtual reality interfaces found in this thesis. In addition, I am also grateful to UNIVR(University of Verona), IIT(istituto italiano di tecnologia), and INAL(Istituto nazionale per l'assicurazione contro gli infortuni sul lavoro) for their financial support.

I would also like to thank my thesis jury committee, Prof.Pedro U. Lima, Prof.Michael Weinmann and Prof. Cristian Secchi for their helpful comments and feedback on this thesis.

I was fortunate enough to become acquainted with several exceptionally talented individuals and close friends at IIT. I would like to express my special gratitude to Alexey Petrushin, Andre Geraldes, Andrea Bennati, German Lanzavecchia, Lilianne Beola Guibert and Nerea Iturrioz RodrÍGuez.

Before starting this research expedition, there were several people I would like to give my sincere thanks for their guidance, mentorship, teaching, and support over the years, including Dr. Girmaw Abebe, Prof. Nicola Conci, and Prof. Fabio Poiesi. It wouldn't have been possible for me to reach this far without them.

Finally, I would like to thank my mom Workea Yemane, my brother Mikiyas Teodros and my sister Miheret Teodros for their unfaltering love and support throughout this journey.

"Don't just say you have read books. Show that through them you have learned to think better, to be a more discriminating and reflective person. Books are the training weights of the mind. They are very helpful, but it would be a bad mistake to suppose that one has made progress simply by having internalized their contents." (Epictetus)

Abstract

In remote immersive visualization systems, real-time 3D perception through RGB-D cameras, combined with modern Virtual Reality (VR) interfaces, enhances the user's sense of presence in a remote scene through 3D reconstruction rendered in a remote immersive visualization system. Particularly, in situations when there is a need to visualize, explore and perform tasks in inaccessible environments, too hazardous or distant. However, a remote visualization system requires the entire pipeline from 3D data acquisition to VR rendering satisfies the speed, throughput, and high visual realism. Mainly when using point-cloud, there is a fundamental quality difference between the acquired data of the physical world and the displayed data because of network latency and throughput limitations that negatively impact the sense of presence and provoke cybersickness.

This thesis presents state-of-the-art research to address these problems by taking the human visual system as inspiration, from sensor data acquisition to VR rendering. The human visual system does not have a uniform vision across the field of view; It has the sharpest visual acuity at the center of the field of view. The acuity falls off towards the periphery. The peripheral vision provides lower resolution to guide the eye movements so that the central vision visits all the interesting crucial parts. As a first contribution, the thesis developed remote visualization strategies that utilize the acuity fall-off to facilitate the processing, transmission, buffering, and rendering in VR of 3D reconstructed scenes while simultaneously reducing throughput requirements and latency. As a second contribution, the thesis looked into attentional mechanisms to select and draw user engagement to specific information from the dynamic spatio-temporal environment. It proposed a strategy to analyze the remote scene concerning the 3D structure of the scene, its layout, and the spatial, functional, and semantic relationships between objects in the scene. The strategy primarily focuses on analyzing the scene with models the human visual perception uses. It sets a more significant proportion of computational resources on objects of interest and creates a more realistic visualization. As a supplementary contribution, A new volumetric point-cloud density-based Peak Signal-to-Noise Ratio (PSNR) metric is proposed to evaluate the introduced techniques. An in-depth evaluation of the presented systems, comparative examination of the proposed point cloud metric, user

studies, and experiments demonstrated that the methods introduced in this thesis are visually superior while significantly reducing latency and throughput.

Keywords: Telerobotics, Telepresence, Mixed Reality (MR), VR, Augmented Reality (AR), 3D Point cloud Compression, Simultaneous Localization and Mapping (SLAM), Rendering.

Contents

List of Figures

List of Tables

Acronyms

AR Augmented Reality vi, 10, 20

FOV Field of View 16, 27, 61
FR Foveated Rendering xi, 58, 61, 62, 63, 64, 65, 66, 68, 69, 71, 86, 87, 88, 102

HMD Head-Mounted Displays viii, 6, 28, 50, 51, 58, 59, 61, 62, 69, 72, 90, 102
HVS Human Visual System 2, 3, 4, 5, 6, 8, 12, 20, 42, 45, 46, 91, 97, 109, 111, 112, 113

LGN Lateral Geniculate Nucleus 23

MAR Minimum Angle of Resolution 25, 26, 47, 48, 49, 57, 97, 99, 112
MR Mixed Reality vi

PCL Point Cloud Library 43, 55, 65, 98
PSNR Peak Signal-to-Noise Ratio v, xi, xiii, 3, 66, 67, 72, 76, 77, 80, 81, 86, 87, 113

QoE Quality of experience viii, 3, 77, 82, 86, 112

RT Reaction Time 28, 70, 71, 79, 82, 85
RTP Real-time Transport Protocol 43
RTSP Real Time Streaming Protocol 43

SLAM Simultaneous Localization and Mapping vi, 6, 8, 32, 34, 35, 45, 52, 62

UE Unreal Engine 44, 59, 60, 61, 90

VR Virtual Reality v, vi, viii, x, 1, 2, 3, 5, 6, 8, 10, 20, 28, 42, 43, 44, 46, 47, 50, 58, 59, 61, 62, 69, 87, 90, 113

1

First things first

"The Gladdest moment in human life, methinks, is a Departure into Unknown Lands." (Sir Richard Burton)

1.1 Introduction

It is a human nature to explore, communicate, share experiences, and help each other in remote and close spaces. The rise of modern computing systems and the ability to transform a physical phenomenon into a digital representation that can be understand intuitively facilitate these natural needs. Specially, in situations when there is a need to explore and perform tasks in inaccessible environments, too hazardous or costly for humans. Recent advances in three-dimensional (3D) scanning sensors, remotely teleoperated robots, fast internet communication, and display techniques enable near-instant, realistic remote visualization without requiring physical presence. Remote visualization techniques have gained much attention with diverse applications in enabling the control of robots from a distance (Telerobotics), remote diagnosis, and monitoring of patients (Telemedicine), entertainment, teleconferencing, remote collaboration, and education. Most importantly, It has recently received increased interest due to the ongoing COVID-19 pandemic. Effective remote visualization systems would immeasurably improve the lives of frontline workers by giving visual feedback to respond to specific emergencies without requiring physical presence [178].

Different display technologies which rely on mono- or stereo-video displays for desktop computers have been proposed for remote visualization, which can display remotely acquired 3D data. However, 3D Visualization techniques such as VR often use more advanced displays methods to provide stereo viewing and allow remotely acquired 3D

data to be visualized with astonishing visual realism and immersion by the user, perceiving the color and the 3D profile of the remote scene simultaneously: This is a critical distinguishing factor from other displaying techniques, which suffers from limitations in terms of fixed or non-adaptable camera viewpoints, occluded views of the remote space, etc. [18, 75].

Nevertheless, achieving the goal of visualizing the 3D scene convincingly and compellingly that cannot be distinguished from the real world and potentially in real-time remains one of the most central challenges in VR. The increased data footprint (3D vs 2D) in real-time remote visualization imposes hard constraints regarding resolution, latency, throughput, compression methods, image acquisition, and the visual quality of the rendering of this information to the user [151, 130]. For instance, latency and low resolution have been shown to reduce the sense of presence and provoke cybersickness [102, 147]. For many applications, including Telerobotics for inspection and disaster response, these constraints are further exacerbated since the scene is *a priori* unknown and should capture the shape and appearance of the scene (3D reconstruction) in real-time from the RGB-D input data. Remote visualization techniques, therefore, presents the challenge of appropriately managing the typical data flow from remote data acquisition, processing, reconstruction, conversion, to compression, encoding, streaming, decoding, and visualization at the user, while allowing optimal visual quality [119, 130].

At the core of such visualization systems lies the question of how to turn a description of RGB-D data into a representation that can be presented to a user for visualization. Herbey, the target, is the essential perceptual channel: the Human Visual System (HVS). The HVS evolved uniquely: It developed mechanisms capable of detecting from a few photons to direct sunlight or switching focus from a close object to the distant horizon in a fraction of seconds. These capabilities are not random, but each detection and movement caused the central part of the vision to fall upon the environment's interesting region. Because vision is not uniform across the field of view, the central vision gives excellent detail of the interesting region, and the peripheral vision provides low-resolution cues to guide the eye movements so that the central vision visits all the interesting and crucial parts of the visual field. These characteristics show that the highest possible uniform visual quality across the field of view is not always necessary. These characteristics can be used to develop better and more efficient remote visualization systems.

The work presented in this thesis aims to research and develop novel methods that exploit the characteristics of the HVS to visualize complex, remotely acquired, and reconstructed point cloud data either in time or bandwidth constraint settings. In addition, it aims to enhance the quality of the rendering while still maintaining performance. The point cloud data is challenging because of its size, geometric complexity, and high visual fidelity requirements. Due to this, remote visualization systems suffer from network latency and throughput limitations. To this end, the high-level characteristics of the HVS that are involved when a physical scene is observed and changed into a percept are discussed. A deep understanding of these characteristics allows the HVS to be defined and

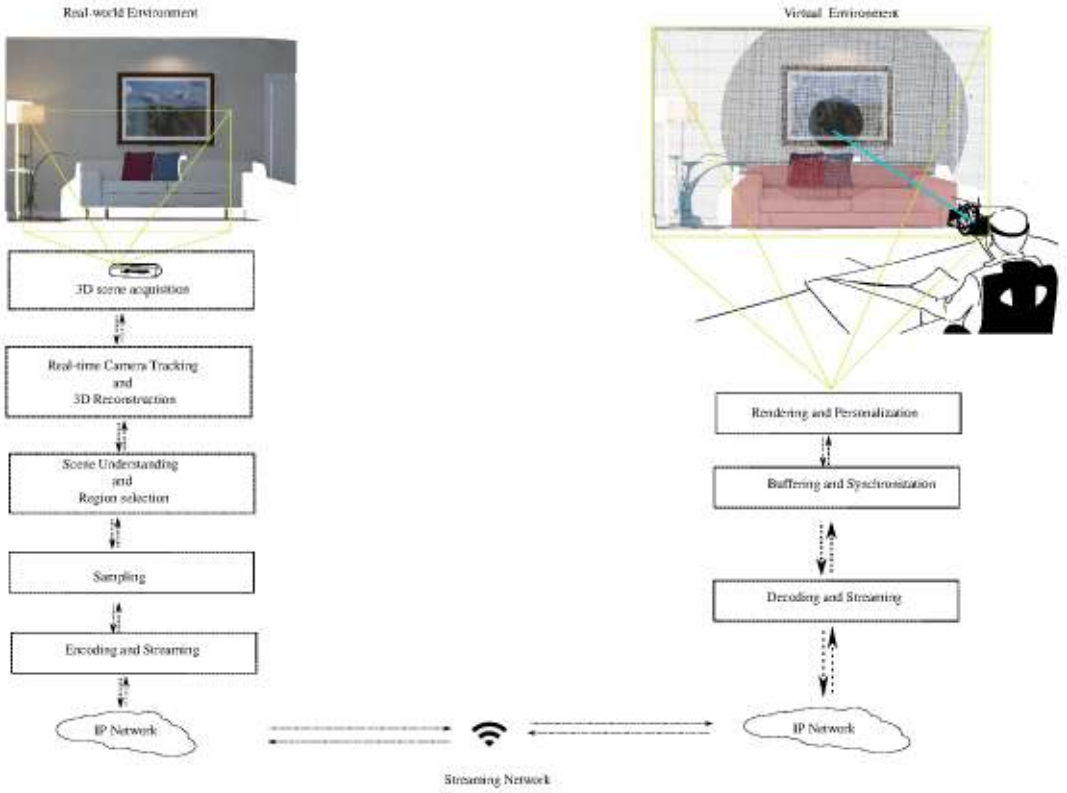


Figure 1.1: Schema showing *Immersive remote visualization* pipeline.

modeled.

A framework is developed from the knowledge of HVS, and a high-level schema of the proposed methods in this thesis is presented in Figure 1.1. The remote environment is captured using the 3D depth-sensing device such as Intel-real sense and Microsoft Kinect. A subsequent camera tracking and reconstruction stage can generate complete 3D representations (Map) of the remote scene in the form of 3D Point Cloud using the camera 3D position and orientation. Then it facilitates the processing, transmission, buffering, and rendering in VR of dense 3D reconstructed scenes while simultaneously reducing throughput requirements and latency. Finally, this thesis showcases how these theoretical insights can be applied to real-world applications such as remote inspection, which has a hard constraint on the networking infrastructure and bandwidth availability. An evaluation of these methods is carried out by performing and evaluating QoE subjective assessments and PSNR objective quality metrics, demonstrating the validity of the proposed methods.

1.2 Research Questions

The framework shown in schema 1 enables immersive remote 3D visualization. Specially, In situations when there are limited time and bandwidth constraints. The thesis defined the following research questions to evaluate the research gaps (unexplored ideas) and the

limitations carefully:

***Main Research Question**: How can we support a real-time immersive remote visualization system for remote Teleoperation and Telepresence applications with state-of-the-art streaming rates?*

This thesis addresses this question using an experimental integrated remote immersive visualization systems approach for remote robotic teleoperation and telepresence applications. It will build and test a prototype system that includes components from capturing to rendering, thus integrating networked transmission and 3D compression components in complete systems. The advantage of such an application-based approach is that it can shed light on integration issues and optimization strategies between components that may have been previously overlooked in the literature. In addition, the proposed approach will be tested with users, resulting in explorative studies on the benefits of different technical implementations.

This primary research question is further divided into three questions that are addressed throughout this thesis.

***Research Question 1:** What are the state-of-the-art immersive remote visualization systems for telepresence and teleoperation systems, and are there any technological, perceptual, and cognitive constraints in designing such systems?*

The research aims to explore related studies, determine what kind of visualization systems for remote telepresence and teleoperation systems have been proposed in the past, and identify which perceptual and cognitive constraints are the challenges in using such systems.

***Research Question 2:** What are the advantages and limitations of the HVS, and How can it be exploited in designing immersive remote viualization systems?*

The second research question objective is to study the essential perceptual foundation and evolution of the HVS that can be used to design efficient interfaces. Most importantly, the study will investigate different visual acuity, contrast sensitivity models, and eye physiology to describe the HVS as an optical system.

***Research Question 3:** How do we design an improved remote visualization system with reduced latency and throughput requirements using the HVS compared to the current state-of-the-art techniques?*

Immersive visualization system that can support many different bit rates and settings is beneficial to support real-time remote visualization, especially when they have hard

constraints on the networking infrastructure and bandwidth availability. Based on these reasons, this thesis investigates different setups, and it performs experimental analysis to understand the impact on visual quality for the user and the challenges in the computational and network performance of the system.

1.3 Summary of Contributions

This thesis builds on the research carried out in several prior works. It adds novel significant contributions for remote visualization and rendering, especially when there is limited computing power and bandwidth constraint. The major contributions and results are listed below:

1. An overview of the main challenges, building blocks, capabilities, and limitations of remote visualization techniques for remote teleoperation and telepresence applications (Chapter 2).

2. An in-depth discussion of the state-of-the-art HVS based systems, covering gaze- and non-gaze based methods to increase remote visualization efficiency (Chapter 2).

3. A novel approach for remote immersive visualization systems, i.e., differential sampling, streaming, and rendering real-time point-cloud / 3D reconstruction data in VR, exploiting the human visual acuity and the user's real-time gaze direction (Chapter 3).

4. 3D acquisition and reconstruction are the methods that capture the data to visualize, and they have to meet critical requirements for accuracy, completeness, and speed. Mainly, it requires robust camera pose tracking and mapping. This thesis presents a novel approach for 3D reconstruction and remote immersive visualization systems for dynamic environments. (Chapter 4).

5. A new volumetric density based peak signal-to-noise ratio (PSNR) metric for point-cloud data is presented to evaluate the proposed approaches (Section 3.5.3.4).

6. User studies and experiments evaluating the impact of the presented approaches on perceived visual quality, latency and throughput (Section 4.5).

1.4 Outline of The Thesis

The thesis is organized into five chapters, where this first chapter provides a brief introduction to the proposed system and research questions, **Chapter** 2 of the thesis presents

the most relevant theoretical foundations to remote visualization systems; It briefly describes technological and human factors which should be known when designing immersive interfaces for such systems. The subsequent section of this chapter gives an overview of the HVS. This section details the physiological parts involved in the vision process, most significantly the discussions of different visual acuity. In section 2.3 and section 2.4, it presents a brief theoretical foundation of real-world data acquisition sensors and recent literature on visual SLAM and 3D reconstruction problems. Section 2.5 presents efficient visualization (rendering) techniques used in graphics. Lastly, an overview of previous works on 3D tele-immersive systems is presented. The work in this **Chapter 2** is based on the litrature review and the following co-authored publication.

A. Naceri et al. "The *Vicarios* Virtual Reality Interface for Remote Robotic Teleoperation". In: *Journal of Intelligent & Robotic Systems* 101.80 (2021)

Chapter 3 introduces the (mathematical) models that are used to describe the HVS and presents a gaze-contingent remote visualization approach. Most importantly, the chapter discusses visual acuity models and the human eye's optical properties while using Virtual reality interfaces. Following these models, The chapter presents a server-client architecture that encapsulates the HVS models. This server-client architecture is divided into three major parts: the user site, the remote site, and a packetization and communication network between them. The remote site includes 3D data acquisition, reconstruction, sampling, compression, and transmission components, and the user site decode and renders the data in VR HMD. Finally, it compares the end-to-end performance (latency and bandwidth) with a user study conducted to evaluate the user quality experience of the proposed framework. This work has been presented in the following scientific publication.

Y. Tefera et al. "Towards Foveated Rendering For Immersive Remote Telerobotics". In: *The International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions at HRI*. 2022

Chapter 4 expands the models proposed in **Chapter 3** based on the HVS and presents a gaze-contingent object-level telepresence system. This chapter proposes a strategy that mainly focuses on understanding the visual information, extracting semantics from the data, and characterizing the present and future progress of the scene. This system follows a server-client architecture proposed in chapter 3 and adds system components presented in this section. The remote site adds an Information gathering understanding module. Finally, It compares the proposed framework's end-to-end performance (latency and bandwidth) with different quality settings to evaluate the usability.

The final part of the thesis in **Chapter 5** comprises a discussion of this research's main contributions. Some exciting developments and trends, as well as possible avenues for

future developments, are contemplated here.

2

Theoretical foundation

"Vicarious living is only slightly less impossible than vicarious eating." (Mason Cooley)

This thesis chapter presents the most relevant theoretical foundations to immersive remote visualization techniques and their application in telepresence and telerobotic systems. It briefly explains technological and human factors which should be known when designing immersive interfaces. The first section of this chapter gives an overview of the Human Visual System (HVS); It details the physiological parts involved in the vision process, most significantly the discussions of different visual acuity. In the subsequent section 2.3 and section 2.4, it presents a brief theoretical foundation of real-world data acquisition sensors and recent literature on visual Simultaneous Localization and Mapping (SLAM) and 3D reconstruction problems. Section 2.5 discusses rendering techniques and methods used for efficient rendering. Last, an overview of previous works on remote visualization systems for telerobotics applications is presented. This Chapter addresses the following research questions:

Research Question 1: *What are the state-of-the-art immersive remote visualization systems for telepresence and teleoperation systems, and are there any technological, perceptual, and cognitive constraints in designing such systems?*

Research Question 2: *What are the advantages and limitations of the HVS, and How can it be exploited in designing immersive remote viualization systems?*

The Vicarios Virtual Reality (VR) interface system for remote robotic teleoperation presented in Section 2.6, was done in collaboration with my colleagues Abdeldjallil Naceri. I contributed for point cloud streaming, video streaming as well as delay measurements

and details are provided in our paper:

A. Naceri et al. "The *Vicarios* Virtual Reality Interface for Remote Robotic Teleoperation". In: *Journal of Intelligent & Robotic Systems* 101.80 (2021)

2.1 Motivations and Background

Remote immersive visualization systems has received increased interest in recent times due in no small measure to the ongoing COVID-19 pandemic. Effective remote visualization systems would immeasurably improve the lives of frontline workers, being able to respond to certain emergencies without requiring physical presence [178]. This section defines why designing and studying immersive interfaces is valuable for improved interactions in remote visualization systems and to understand the expected effects of such interfaces on the user. Mainly the following motivations are defined in detail as follows :

1. **Improve situational awareness:** Situational awareness or situation awareness (SA) is defined as "the perception of environmental elements and events with respect to time or space, the comprehension of their meaning, and the projection of their future status" [39]. With adequately designed immersive interfaces, users can acquire information about the environment, i.e., the situation, then provide techniques to quickly interpret the information and help them with reasoning and decision-making.

2. **Risk prevention:** An unsafe (demanding) environment creates physical and psychological risks because the environment is too dangerous. For instance, nuclear, chemical, disaster response, construction/demolition, mining, submarine tasks, there are extreme risks to the health and safety of humans. Well-designed immersive interfaces can provide timely sensor feedback and allow humans to perceive the risk before it occurs or worsens: It could help to avoid risk and reduce the impact.

3. **Effective planning:** A user interface with elements that are easy to access, understand, and simple to use could help users to get information according to what the users needed. For example, interfaces in remote robotic-based tasks and motion planning help the robot operator to perceive environments where the robot is, decide how to navigate the environment, understand how to interact with objects. Before executing the actual task, the user could visualize information about the robot kinematics and its capabilities in the environment. Interactive interfaces can provide the operator information regarding the possible approaches for executing the tasks in the planning phase.

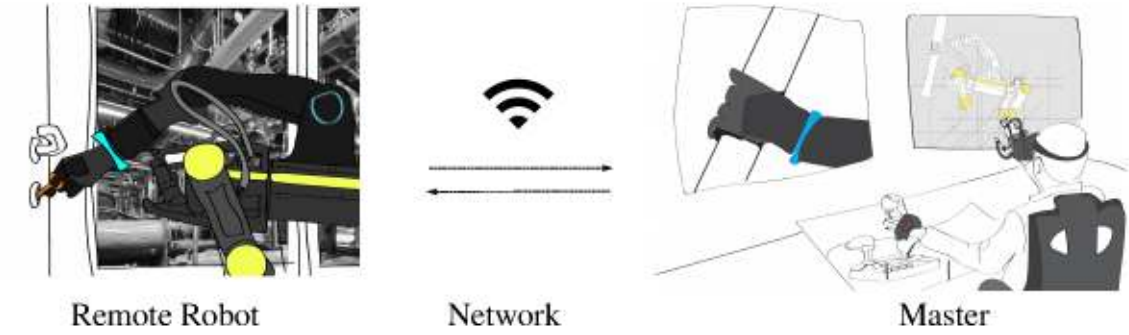


Figure 2.1: Schema shows an overview of remote teleoperation.

2.1.1 Telepresence and Teleoperation Systems

Researchers have defined Telepresence, or Telexistence, as the ability to be remotely present, i.e., exist, and function at a remote location [155] [120]. A telepresence system aims to enable a remote user to have the sensation of being present on the remote site and perform tasks as if the remote user is directly performing them there. Historically, telepresence systems are made for remote Teleoperation using a master-slave robotic system.

Teleoperation was defined as *"Teleoperation means "doing work at a distance", although by "work" we mean almost anything. What we mean by "distance" is also vague: it can refer to a physical distance, where the operator is separated from the robot by a large distance, ... Teleoperations comprise a robot technology where a human operator (master) controls a remote robot (slave). The system is formed by two parts, the control module, called cockpit and the telemanipulator, the slave robot at the remote location."* [90].

In remote robotic teleoperation Figure 2.1, with the operator using a robotic system from a remote place, the quality of perception plays a central role in allowing the operator to accurately estimate distances, sizes, movements, and spatial orientations in the remote environment to execute tasks successfully and efficiently. With the advancements in VR and Augmented Reality (AR) technologies, immersive user interfaces could improve human supervisory control of a teleoperation system by providing visual & sensory feedback from the remote environment in an intuitive and easy-to-understand manner. A brief discussion of different immersive interfaces for telepresence and telerobotic applications are discussed at the end of this chapter in section 2.6.

2.1.2 Human Factors

Bowman et al. [13] defined the term "human factors" as the capabilities, characteristics, and limitations of the human user; and includes considerations related to the body (acting), the sense (perceiving), and the brain (thinking). The learning capabilities and the performance of a human is limited by the amount of information they can remember,

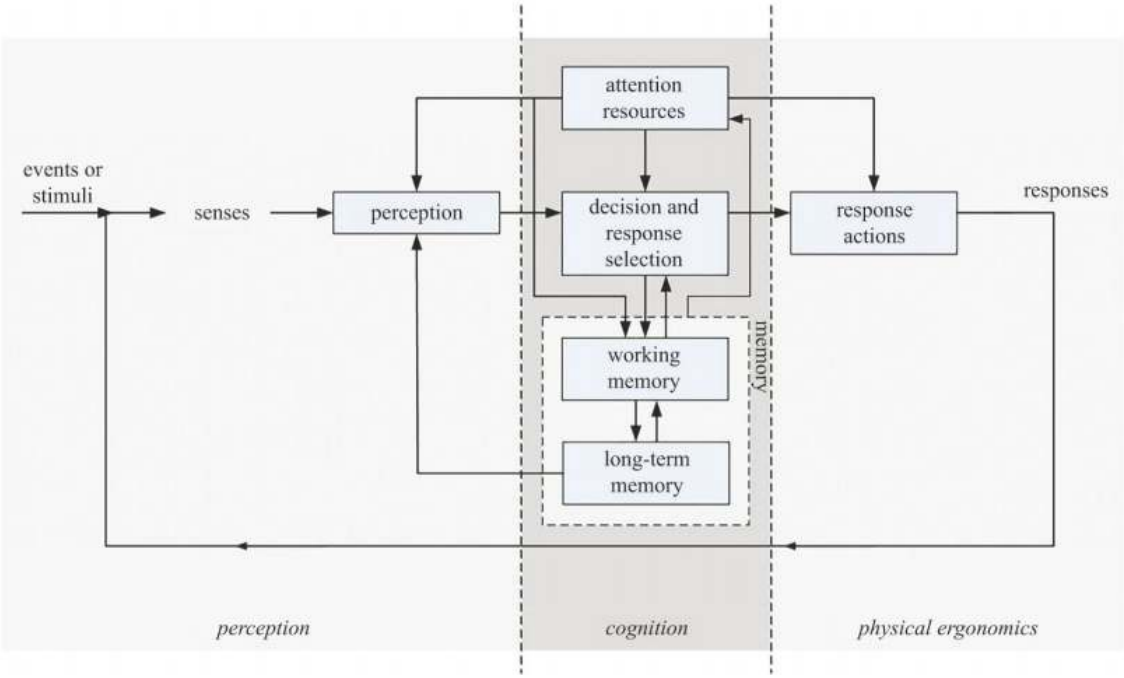


Figure 2.2: Information processing and related human factors adapted from [172].

and process [110]. Properly designed interfaces can potentially improve these limitations. To support and guide this study, the following questions are considered as primary questions, among others.

- How to find a balance between how much information to show and for how long to show it without confusion?
- How to represent multi-dimensional information (point-clouds, camera streams, etc.) effectively and efficiently?
- What are the perceptual and cognitive constraints in remote visualization systems?
- How to integrate different sensory information in an ergonomic and user-friendly manner, while utilizing human cognitive capabilities?

A basic knowledge of how human process information into useful (inter)action is very valuable in understanding the human factors [13]. Termed as "Information Processing", it has been studied for many years and different studies have designed different models to define it. Bowman et al. [13] adapted a high-level staged information processing model from [172] by mapping it into the three main factors: perception, cognition, and physical ergonomics Fig. 2.2.

2.1.2.1 Cognition

Cognition refers to the mental processes involved in gaining knowledge and comprehension. These cognitive processes include thinking, knowing, remembering, judging, and

problem-solving. Cognitive neuroscience studies consider cognition as the way a human perceives and conceptually structures the world. Hence, it leads to emotions and behaviors. According to Bowman et al. [13] model, stimuli or events which are perceived will be provided with a meaningful interpretation based on memories of past experiences. In response to what is perceived, actions may get selected, executed, or information may get stored in working memory (short-term memory).

Attention Resources: Attention can be seen as a selection process and it could be used to draw attention to specific information from the dynamic spatio-temporal environment. Attention process is prone to errors, which can be raised by limitations in our sensory system, which leads to an inability to notice visual (change blindness) and auditory changes. 'Change Blindness' refers to the surprising difficulty observers have in noticing large changes to visual scenes [143]). Similarly, errors occur on a temporal basis, especially when rapid sequences of stimuli occur.

Short and Long-term memories: Working memory has a limited capacity, and attention resources highly influence it. In contrast, the capacity of our long-term memory is vast, storing information about our world, concepts, and procedures while not being directly affected by attention [13].

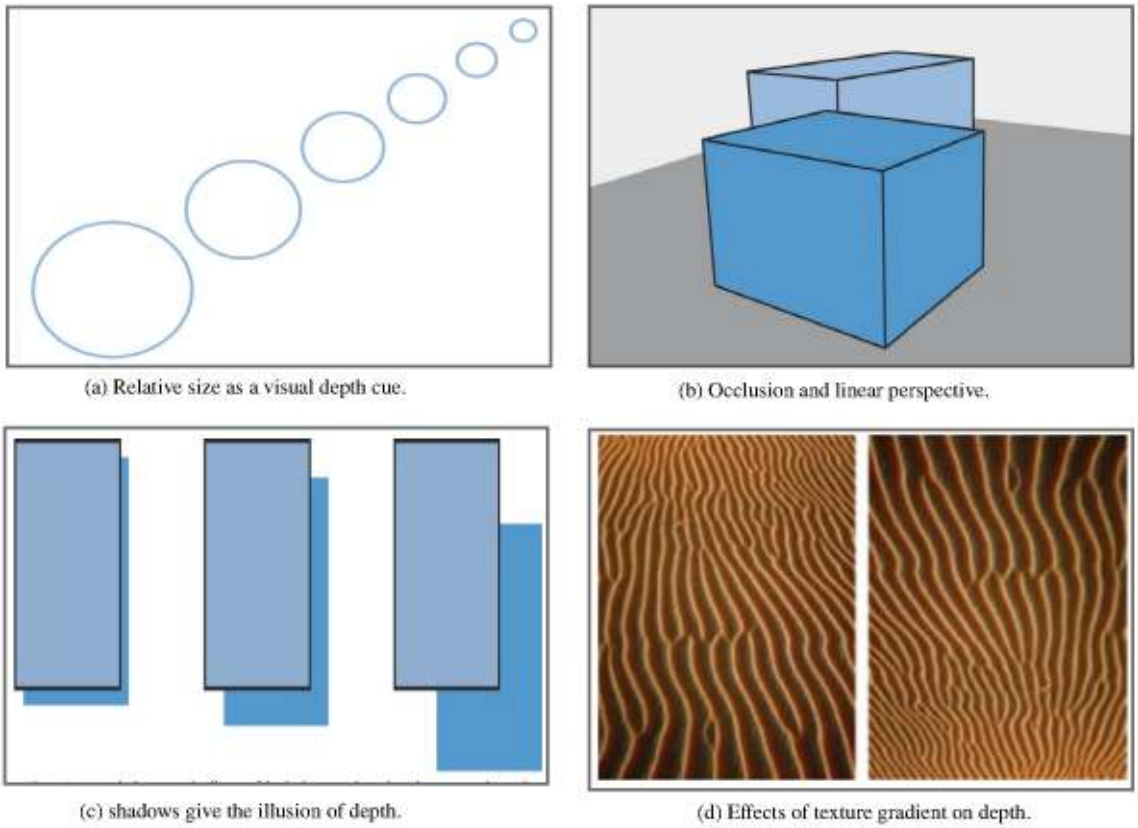
2.1.2.2 Perception

Visual information processing depends on a complex pattern of intertwined pathways in the human brain; The moment light meets the retina, the process of sight begins. The retinal has a layer of cells called photoreceptors. The distribution of the different kinds of photoreceptors (rods and cones) results in different abilities of the visual system in the center and the periphery. The center has the highest sensitivity to fine details, and these abilities deteriorate quickly at the periphery. However, peripheral vision is still reasonably good in processing motion; this is important for our fast reaction to moving objects. Details can be found in section 2.2.

Vision provides several cues about the spatial layout (such as depth cues) of objects in a scene that could be used for selection, manipulation, and navigation tasks. Therefore, understanding what depth cues the HVS uses and how visual displays provide such cues is another fundamental mechanism for designing immersive 3D interfaces.

Monocular depth cue is depth information in the retinal image gives us information about depth and distance. This depth information can be inferred with only a single eye. These cues consist of static information, including relative size, perspective, interposition, lighting, and focus cues (image blur and accommodation), as well as dynamic information such as motion parallax [67]. Figure 2.3 a shows depth cues from a relative size difference. The smaller circles appear farther away, and the larger object appears closer.

Oclusion happens when one object overlaps another, the partially obscured object is perceived as being farther away. Figure 2.3 b shows two boxes placed in the distance and one box overlapping and occluding the other. In that case, the user perceives the occluded



(a) Relative size as a visual depth cue.

(b) Occlusion and linear perspective.

(c) shadows give the illusion of depth.

(d) Effects of texture gradient on depth.

Figure 2.3: Important visual cues that are processed when interacting with a real or virtual environment, adapted from [13].

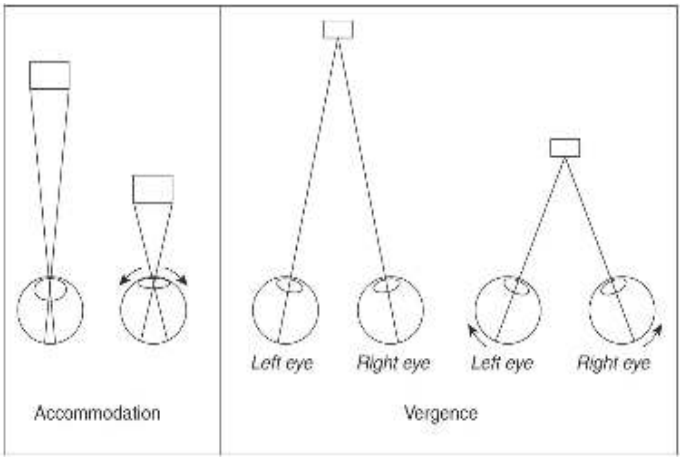
box as behind the non-occluded one. This effect is called Occlusion, also called contour interruption or interposition [13]; it allows the user to judge how objects are placed and contribute to the experience of depth perception.

Linear perspective is the phenomenon that makes parallel lines appear to converge as they travel away from the viewer. Figure 2.3 b shows an example of linear perspective; The closer together the two lines are, the greater the distance will seem from the viewer.

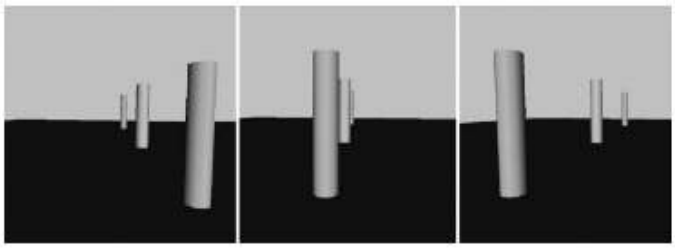
Aerial perspective is a cue that gauges relative distance by measuring the scattering and absorption of light through the atmosphere [13]. Objects farther away seem to be blurred or slightly hazy due to the atmosphere, and nearer objects seem to have more color saturation and brightness.

Shading and Lighting Light falls on objects, and the amount of shading present can also be an essential monocular cue. Things that are darkened and obscured may appear farther off in the distance than brightly lit ones. For example, Figure 2.3 c shows how shadows and lighting effects on depth estimation, objects that have more illumination are generally closer to the viewer.

Texture gradient is another monocular cue that uses texture to gauge depth and distance. This phenomenon is distortion in size, which is closer to texture patterns than patterns farther away . It also involves groups of texture patterns appearing denser as



(a) Accommodation and vergence.



(b) Motion parallax depth cues.

Figure 2.4: Accommodation (top left image), vergence (top right image) and motion parallax depth cues (bottom image), adapted from [13].

they move farther away. Figure 2.3 d, shows effects of texture gradient on depth cues: the image on the right is an inverted image of the left. In both images the depth is perceived [13].

Oculomotor cues are depth cues derived from muscular tension in the viewer's visual system, consisting of accommodation and vergence (Figure 2.5 a). Accommodation is the process by which the physical stretching and relaxing of the eye lens to focus an object on the retina. Far away objects require low lens convexity, whereas near objects require high lens convexity to become focused on the retina. Thus, The state of these eye muscles in stretching and relaxing provides a cue to depth. Vergence is the process by which the eyes rotate in equal and opposite directions to fixate an object. Near objects require both eyes oriented inwards to have the object foveated, whereas far objects require both eyes oriented along parallel lines of sight.

Motion parallax refers to the fact that objects moving at a constant speed across the frame will appear to move more if they are closer to an observer than they would if they were further away (Figure 2.5 b). This can happen when objects move relative to the viewer (stationary-viewer motion parallax), the viewer moves relative to stationary objects (moving-viewer motion parallax), or when there is a combination of the two.

Binocular disparity refers to the difference in image location of an object seen by the

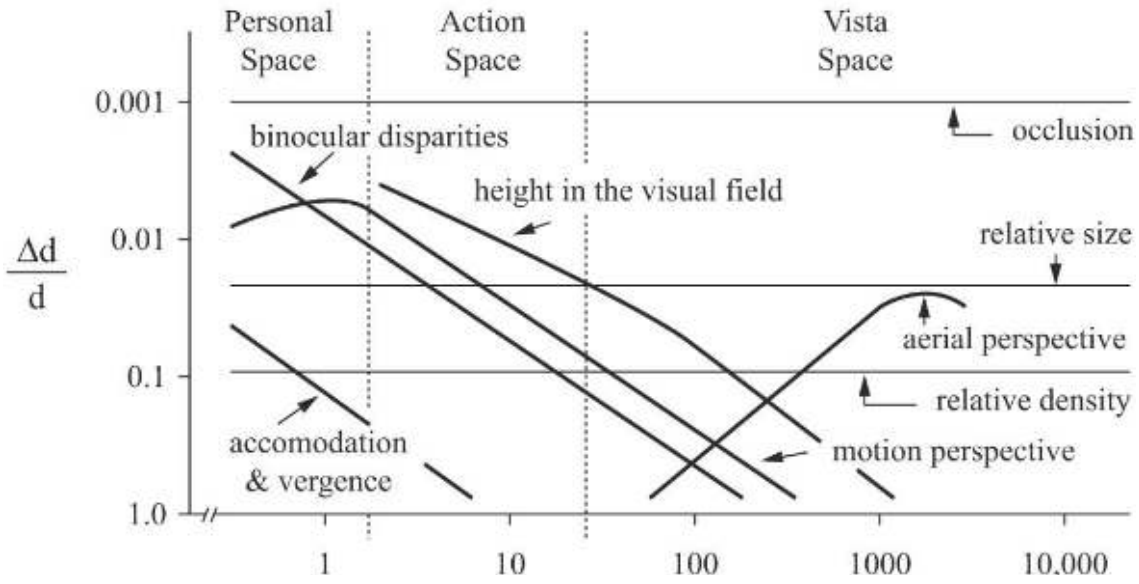


Figure 2.5: The relative importance of depth cues at different distances, adapted from [27].

left and right eyes. The simple way to understand binocular disparity is to focus on a near object and alternate opening and closing each eye. Combining these two images through accommodation and vergence provides a depth cue by presenting a single stereoscopic image: This effect is referred to as **stereopsis** [13].

The perception of an object of interest degrades as the distance from the observer increases. This quality reduction directly results from the diminishing quality and availability of precise visual cues and an evolutionary advantage given finite cognitive resources and the relative importance of close objects compared to distant ones. Cutting and James et al. [27] have discretized perceptual space into three distinct regions defined by the distance from the human observer: Vista space, Action space, and Personal space.

Figure 2.5 compares the relative strength of depth cues at different distances for comparing the depth of objects. Personal space surrounds the observer's head, generally within arm's reach and slightly beyond [26]. Thus, the observer does not typically generate motion perspective; instead, motion parallax and structure-from-motion information are generated by observer manipulation to reveal object shape [26]. In this region, occlusion, retinal disparity, relative size, and then convergence and accommodation are effective. The region beyond personal space is action space. In action space a person can move quickly within this space and talk within it; occlusion, height in the visual field, binocular disparity, motion perspective, and relative size are the dominant depth cues. Vista space occurs beyond 30 m, at least for a pedestrian. The only effective depth cues in this space are occlusion, height in the visual field, relative size, relative density, and aerial perspective. The effectiveness of binocular disparity and motion perspective are negligible.

In general, occlusion is a persistent depth cue for this sort of comparison. However, that occlusion and many other depth cues only provide relative depth information. Only accommodation, vergence, stereopsis, and motion parallax provide information about absolute depth.

2.1.2.3 Physical Ergonomics

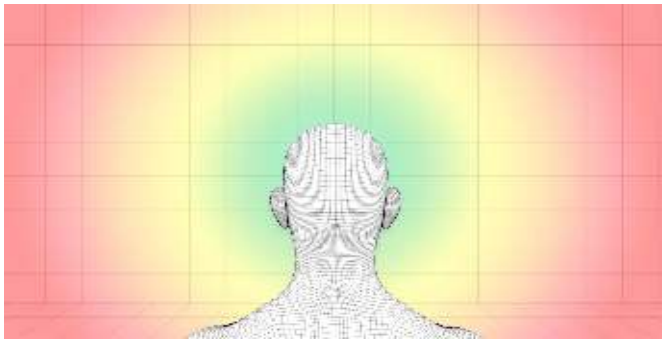
While this thesis mainly focuses on perception and cognition, physical ergonomics are equally important factors that researchers should study to design systems that can be used comfortably and effectively. Physical ergonomics concerns the human musculoskeletal system primarily. It depends on the anatomical capacities of the different human body parts, which defines how and how well we can perform a specific task. Different body parts have different comfortable and maximum range motions produced by joints and muscles. This section looks at the body parts, mainly the head, eye, and ankle, which are closely related to Head-mounted displays.

An average person can turn their head horizontally 30 ° comfortably and a maximum of 55 °. The degrees one can turn the head vertically differ between the head tilting up or down. Looking up 20 ° and looking down 12 ° is considered comfortable. The maximum for looking up is 60 °, and looking down is 40 ° [114]. To keep the interaction from getting uncomfortable, one should place the main user interface objects within a comfortable area. An angle greater than 25 ° down forces the neck to constantly keep the head up, and A bent neck also heightens the risk for pains and diseases [114]. Figure 2.6 a shows where the comfort zone is, areas with the green color are comfortable, whereas areas with the red should be avoided for user interaction design.

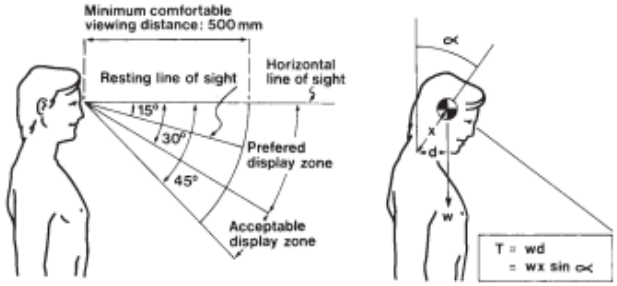
The neck angle only doesn't determine how comfortably the user can see in the virtual world; the Field of View (FOV) and the physiology of the eye determine it as well. In the relaxed line of sight, when the user is seated with the head up and looking ahead, the eyes will naturally assume a slight downward gaze of some 10 or 15 ° from the vertical. The eyes could be raised by 48 ° and lowered by 66° without head movements, but in practice, the downward eye movement is limited to 24-27°, beyond that point, the head and neck are inclined forwards, and the neck muscles come under tension to support the weight of the head. For this reason, the preferred display zone is between 0 to 30 ° down [125]. Figure 2.6 b Left shows the preferred viewing conditions as described in text and the Right shows the postural stress to neck muscles resulting from a downward line of sight. T is the torque about the neck; w is the weight of the head and neck; x is the distance from C7 to the centre of gravity of the head and neck.

2.1.3 Technological Factors

In addition to human factors, This section study various technological factors, which can affect remote visualization and the user specialy in remote telerobotics application,



(a) Safe head movements zones



(b) Preferred viewing conditions

Figure 2.6: Safe head movements zones and preferred viewing conditions, Image adopted from [72] and [125].

especially related to glsFOV, limited depth perception, network latency, etc. To support and guide this study, the following questions are considered as primary, among others.

- How to present information without confusion?
- How to find a balance between how much information to show and for how long to show it?
- How to represent the multi-dimensional information (environmental sensors, point-clouds, camera streams, etc.) effectively and efficiently?
- How operators perceive different information?
- What are the perceptual and cognitive constraints in remote robotic teleoperation?
- How to integrate different sensory information in an ergonomic and user-friendly manner, while utilizing human cognitive capabilities?

Field of View : Cameras with limited angular view create the so-called "Keyhole" effect (a sense of trying to understand the environment through a narrow "Soda Straw" glsFOV). Major consequences of this effect include missing new events, increased difficulty in navigating novel environments, gaps or incoherent models of the explored space, etc. [175].

Orientation and Attitude of the remote robot: When the remote environments are complex and cues to a robot's pose are sparse, it becomes easy for a teleoperator using an egocentric (camera) display to lose situational awareness. To successfully navigate locally, while globally knowing where the object / region of interest is, the operator needs to know the robot's attitude. Attitude (i.e., pitch and roll) of a robotic vehicle may be easy to reference when there are other familiar objects (e.g., horizon, buildings, trees, etc.) in the remote environment. However, if those reference points are absent and the on-board cameras are fixed ones, operators sometimes find it surprisingly hard to accurately assess the attitude of their robotic vehicles [164]. .

Orientation in the remote environment: Navigation with a traditional (north-up) map can be challenging at times because of the demand of mental rotation. Track-up (ego-referenced; rotating viewpoints) maps consistently perform better for local guidance (i.e., navigation) and north-up maps are better for global awareness [164].

Attitude of the robot: Attitude (i.e., pitch and roll) of a robotic vehicle may be easy to reference when there are other familiar objects (e.g., horizon, buildings, trees, etc.) in the remote environment. However, if those reference points are absent and the on-board cameras are fixed ones, operators sometimes find it surprisingly hard to accurately assess the attitude of their robotic vehicles [164].

Multiple cameras and viewpoints: The capabilities to see the robot and its local environment gives the operator a better sense of the robot's location with respect to obstacles, victims, or other potential situations [84]. However, the difference in eye-point and camera viewpoint could create motion sickness [84].

In addition, when handling multiple robots, it can be challenging for the operator to acquire different contexts rapidly when switching among robots: information in one scene may not be encoded sufficiently to be compared/integrated when accessed subsequently (change blindness).

Degraded Depth Perception : Degraded depth perception affects the teleoperator's estimation of distance and size and can have profound effects on task effectiveness. In the case of monocular cameras, the operators have to rely on other cues: such as shadows, linear perspective, and size consistency.

The effect of degraded depth perception has a higher impact when working in unfamiliar and difficult terrain due to lack of apparent size. In addition, remote manipulation that involves fast movements or analysis of three-dimensionally complex scenes would be highly affected [34].

Degraded Video Image and Time delays: Degraded video feeds could leave out essential visual cues for building teleoperators' mental models of the remote environment. Different factors such as low bandwidth, low frame rate, low resolution, high latency, and the number of bits per pixel can create degraded video feeds.

Motion: Teleoperation can be difficult and distracting because of the vibrations and oscillations of the moving robot, which makes viewing the visual displays and the manual control/action more challenging [137].

Factor	Effects	Suggested solution	Ref.#
Field of View	• Restricted FOV affects target detection and identification. • Distance cues may be lost and depth perception may be degraded. • Degraded remote driving. • Increased difficulty in navigating novel environments.	• Wider FOV (changeable FOV) can be used. • Stereoscopic 3D displays. • Multiple cameras and single cameras with special optics.	[19] [145] [161] [20] [154]
Orientation and Attitude of the Robot	• Difficulty knowing the robot orientation in the environment. • North-up and Track-up maps. • Mismatch between actual and perceived attitude of robot. • Unawareness of robot's inclination and shape.	• Track-up map for navigation. • North-up map for tasks involving integration of spatial relations in the environment. • Gravity referenced view.	[164] [20] [43] [35]
Multiple cameras and Viewpoints	• Attention switching and change blindness. • Motion sickness. • Egocentric, cognitive tunneling and Exocentric, loss of immediacy and true ground view.	• Auditory alerts. • Multi-modal solutions. • Peripheral cues for egocentric.	[84] [20] [116] [84] [20] [116]

Depth perception	• Underestimation of distance and size. • Degraded navigation, driving, and telemanipulation.	• Sterioscopic displayes (SDs). • Inter-camera distance should be less than inter-ocular distance.	[13] [43] [20]
Video quality and time delays	• Degraded motion perception and spatial orientation. • Degraded target identification and latency. • Motion sickness. • Over actuation when delay is variable.	• Utilize the human cognitive processing speed which is around 170 ms (range: 75 – 370 ms). • Augmented reality / overlaying information. • Predictive Display (simulation in VR/AR). • Robust adaptive algorithm for video streaming.	[98] [20] [79]
Motion	• Degradation on accuracy and latency. • Motion sickness.	• Multi modal user interface. • Tailor interface to vibratory and motion effects.	[76] [137]

Table 2.1: Summary of different technological factors.

2.2 The human Visual System

Humans perceive visual information through sensory receptors in the eyes. The process begins when light passes through the cornea, enters the pupil, and gets focused on the lens onto the retina. This is then processed in the brain, where an image is formed. The visual system can be divided into three major processing components: The Eye, Visual Pathway and Visual Cortex(see Figure 2.7). Each component performs a particular analytical process on the visual information, and the following section describes them in detail in the subsequent sections. This section gives a brief description of the HVS and the biology around visual acuity in the central and peripheral regions of the retina.

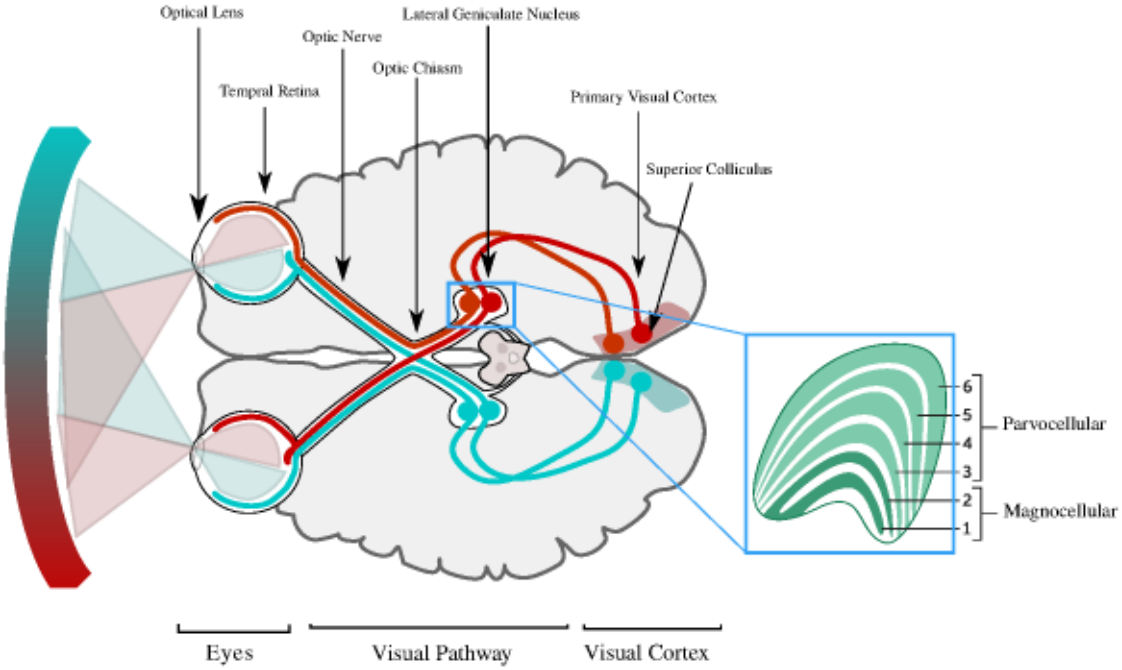


Figure 2.7: The human visual pathway showing the three major physiological components.

Some questions of interest this chapter will try to understand:

1. What are the Limitations and potentials of our vision?
2. How does our vision in the periphery differ from that near the center?
3. Are there separate brain areas that determine our perception of different qualities?
4. How has the operation of our visual system been shaped by evolution and by our day-to-day experiences?

2.2.1 The Eye

The eye is where vision begins; It initiates when light passes through the cornea, enters the pupil, and gets focused on the lens onto the retina. The eye is approximately spherical, with a diameter of around 24 mm, An overview of the eye is illustrated in Figure 2.8-(a).

The cornea, the transparent covering of the front of the eye, reflects light receives by an object into the eye, which accounts for about 80 percent of the eye's power. It's focussed by the lens and passes through the vitreous chamber before reaching the retina at the back of the eye. The light must pass through several layers of neurons in the retina before finally reaching the photoreceptor cells (see Figure 2.8-(b)). There are two kinds of photoreceptors on the retina: rods and cones, which contain light-sensitive chemicals called visual pigments that react to light and trigger electrical signals. Cones are capable of color vision and are responsible for color vision high spatial acuity, and

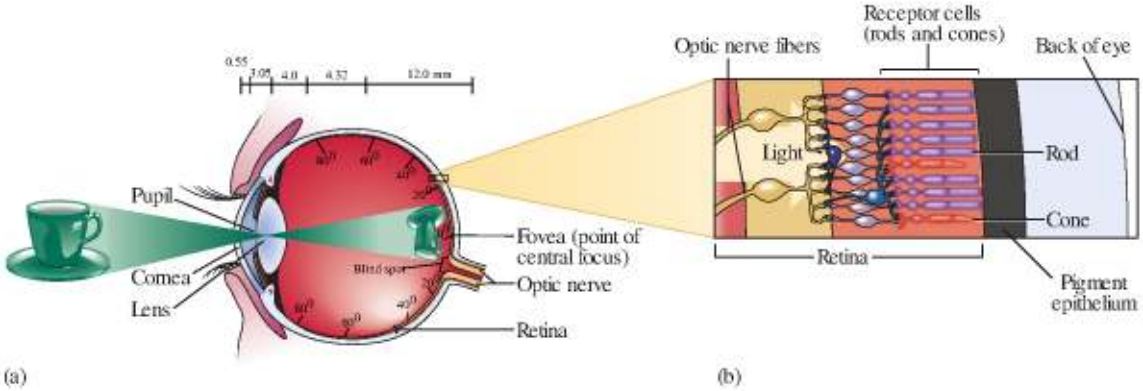


Figure 2.8: The human eye schematics, Image adapted from [47].

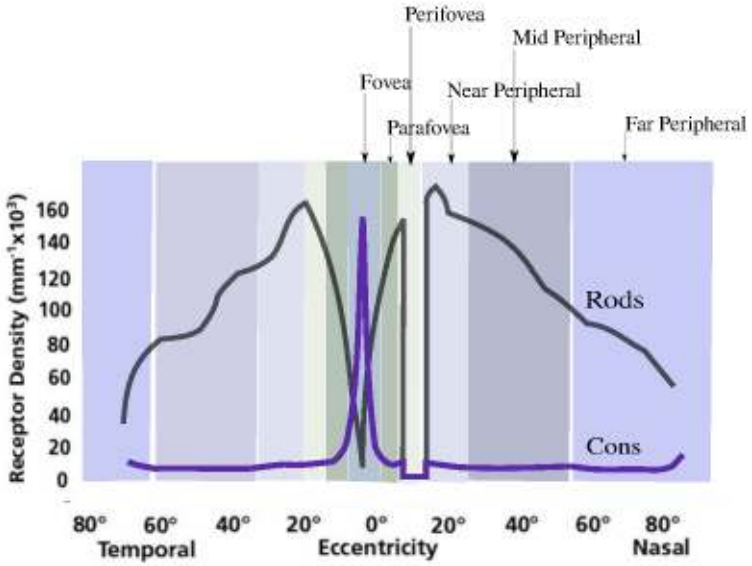


Figure 2.9: Graph that shows the rod and cone densities in the retina (°).

rods are responsible for vision at low light levels. As shown in Figure 2.9, The distribution of the rods and cones depends on location in the retina; the cone density is highest in the central region of the retina and reduces monotonically to a reasonably even density into the peripheral retina region.

- The region from the center of the retina up to 5° of eccentricity is the *Fovea* region. The Fovea is only about 1% of the retina, but has the highest density of cone photoreceptor cells and the brain's visual cortex dedicates about 50 % of its area to information coming from the Fovea [60]. Therefore, the Fovea has the highest sensitivity to fine details.

- The region that surrounds the Fovea is commonly known as the *Parafovea*, which goes up to 8° of the visual field [57]. The parafoveal region provides visual information as to where the eyes should move next (saccade), and supports the Fovea to

process the region of interest in detail. Previous research has investigated if meaningful linguistic information can be obtained from parafoveal visual input while reading [66, 139].

- The next region that surrounds the Parafovea is called *Perifovea*, which extends approximately up to 18° of eccentricity. In this region, the density of rods is higher than that of cones, about 2:1. Consequently, unlike the Fovea and Parafovea, only rough changes in shapes are perceived in this region [69]. The region beyond 18°, and up to about 30° of the visual field, is known as the *Near-Peripheral Region*. It has the distribution of 2–3 rods between cones [128]. This region is responsible for the segmentation of visual scenes into texture-defined boundaries ("texture segregation") and the extraction of contours for pre-processing in pattern and object recognition [152].

- The region between 30° up to about 60° of eccentricity is called the *Mid Peripheral Region* [144]. Although acuity and colour perception degrade rapidly in this region, researchers have shown that color perception is still possible even at large eccentricities, up to ~ 60° [48, 52].

- The region at the edge of the visual field (from 60° up to nearly 180° horizontal diameter) is called the *Far Peripheral Region*. This region has widely separated ganglion cells, and visual functions such as stimulus detection, flicker sensitivity, and motion detection are still possible here [152].

There is one area approximately 1.5mm across in the retina with no receptors, where the optic nerve leaves the eye. Because of the absence of receptors, this place is called the blind spot. Preprocessed electric signals are transmitted over the optical nerve to higher-level visual pathways. It carries the impulses from the retina's ganglion cells to the visual centers in the brain.

2.2.2 The Visual Pathways

Once the signal leaves the eye via the axons of the retinal ganglion cells, they are transported by the visual pathways to the higher visual center of the cranium, i.e., the skull (see Figure 2.7). However, these pathways do not passively transport the signal: some reorganization and processing are performed. Most of the signals from the retina travel out of the eye in the optic nerve to the Lateral Geniculate Nucleus (LGN) in the thalamus through the optic chiasm: an X-shaped structure formed by crossing the optic nerves. Because of this crossing arrangement, the right LGN receives information about the left visual field, and the left LGN receives information about the right visual field.

Neurons in the LGN have receptive fields that are concentric receptive fields much like those of the retinal ganglion cells. It is arranged in multiple layers (6 layers) that are segregated according to the origin of the retinal signal emerging from the retinal

ganglion cells. As shown in Figure 2.7, The inner two layers are magnocellular, while the outer four layers are parvocellular. The names come from the retinal ganglion cells with large (Magno) or small (Parvo) cell bodies. An additional set of neurons, known as the koniocellular layers, are found ventral to each of the magnocellular and parvocellular layers.

Magnocellular cells (also called M-cells) are relatively large cells in the ventral region. Two magnocellular layers(layer 1 and 2) lie inward. They have a large receptive field on the same side as the retina, but they cannot provide detailed or colored information but still provide useful static, depth, and motion information. These cells have high light/dark contrast detection and are more sensitive at low spatial frequencies than high spatial frequencies [126]. Due to this contrast information, M cells are essential for detecting luminance changes, performing visual search tasks, and detecting edges [21].

Parvocellular cells, also called P-cells, are relatively small and sensitive to color and can discriminate fine details than their magnocellular counterparts. By comparison, Parvocellular cells have a greater spatial resolution. Still, lower temporal resolution [177], demonstrating that the visual system consists of several separate and independent subdivisions that analyze different aspects of the same retinal image [126].

2.2.3 The Visual Cortex

Ultimately, the signal passes by the visual pathways reach the primary cortical region of the brain. Based on function and structure, the visual cortex divides into five different areas (V1 to V5). The approximate positions of different areas of the brain and retinotopic maps of the human visual cortex are illustrated in Figure 2.10.

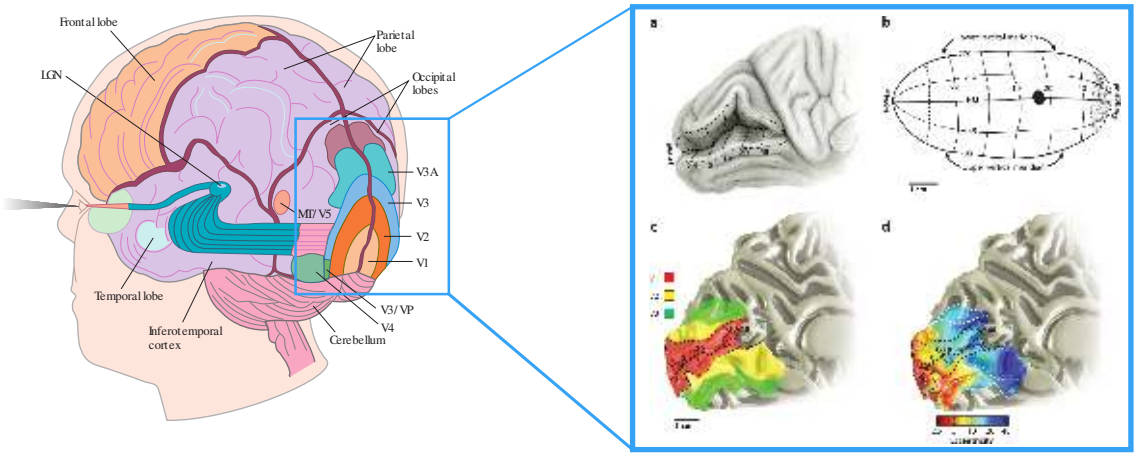


Figure 2.10: Approximate positions of different areas of the brain that are responsible for vision and retinotopic maps in early human visual cortex. Image adapted from [146, 12]

The signal from visual pathways first passes through the thalamus, where it synapses in a nucleus called the lateral geniculate. This information then leaves the lateral geniculate and travels to an area called V1 or striate cortex. Area V1 is located in the occipital

lobe at the back of the head 2.10). This area is a large cortex area involved in vision: More than 80 percent of the cortex responds to visual stimuli. Objects close together in the retina will activate neighboring neurons of the visual cortex. Moreover, the left V1 maps the right visual field, and the right V1 maps the left visual field with minimal overlap (figure 2.7). However, within the map, the central area of the visual field is represented by a greater amount of neural cells to receive a disproportionately large representation [47]. Cells in the visual cortex respond to particular visual features or objects' attributes in the visual world. For example, cells in V1 respond most strongly when an edge or contour placed at a specific orientation. The study by Hubel and Wiesel et al. [63] identified two functionally different classes of cortical cells in cat and monkey primary visual cortex: The simple and complex cells. Simple cells respond to stationary or slow-moving stimuli, and complex cells respond maximally to moving stimuli of a particular orientation.

The Visual cortex 2 (V2) is the second major area in the visual cortex, It receives strong feedforward connections from V1 (direct and via the pulvinar) and sends strong connections to V3, V4, and V5. It also sends strong feedback connections to V1. Researchers have seen that the V2 cells collectively encode information about many complex shape characteristics: differences in color, spatial frequency, moderately complex patterns, and object orientation [56]. V2 sends feedback connections to V1 and has feedforward connections with V3-V5. Information leaving the second visual area splits into the dorsal and ventral streams, which specialize in processing different aspects of visual information. The dorsal area is often concerned with object recognition, while the ventral streams focus on spatial tasks and visual-motor skills.

2.2.4 Visual Acuity

As highlighted briefly in section 2.2.1, the center of our vision, known as the fovea, is only about 1% of the retina. But, it has the highest density of cone photoreceptor cells, and the brain's visual cortex dedicates about 50 % of its area to information coming from the Fovea [142], which is a very high magnification, and This means that the other bits have far less cortex. The large representation of the fovea in the cortex is also illustrated in Figure 2.10. The amount of cortex given to each 1° gets smaller further from the center of vision the cells are encoding.

Daniel and Whitteridge [31] invented the term *linear cortical magnification factor* (M_c) to refer to the millimeters of cortex representing 1° of visual field at any given eccentricity. The most popular way of determining visual acuity is to quantitatively represent it in terms of *minimum angle of resolution* (Minimum Angle of Resolution (MAR), measured in arcminutes). In fact, visual acuity is represented as the reciprocal of MAR, but MAR itself is now quite common in literature [168, 49, 152]. MAR can be understood as the smallest angle at which two objects in the visual scene are perceived as separate [168].

Since the MAR is the reciprocal of the visual acuity, It can also be understood in relation to the cortical magnification factor, M_c [24]. M_c accounts for the number of neurons

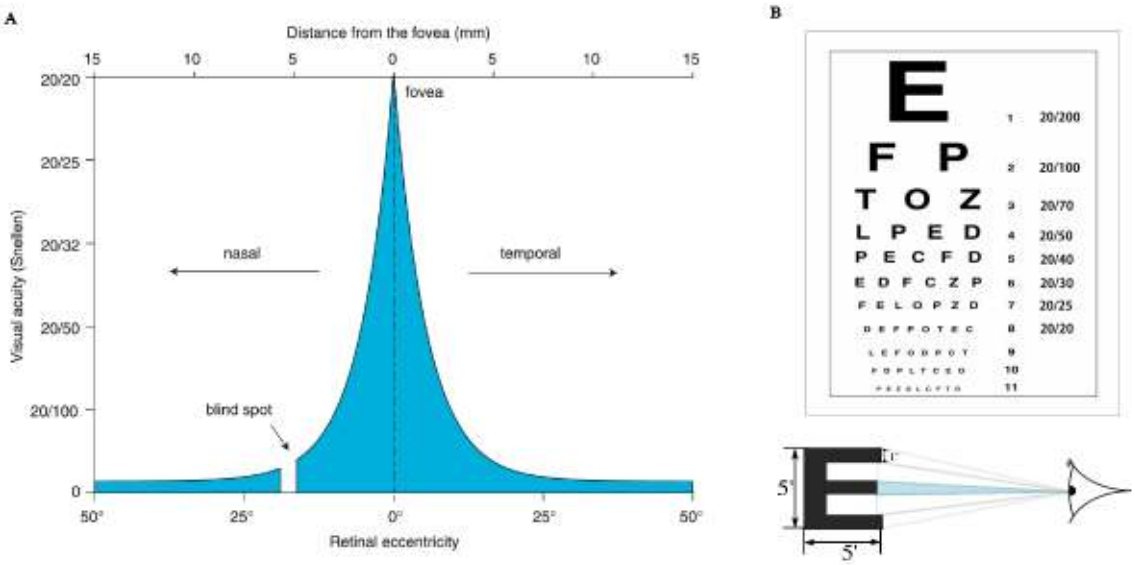


Figure 2.11: Snellen visual acuity as a function of degrees of retinal eccentricity. Image adapted from [**https://doi.org/10.1371/journal.pone.0174020.g001**].

allocated to process the information from the visual field, as a function of the eccentricity [24]. Authors in [24] showed that the reciprocal of M_c is directly proportional to MAR and, therefore, to visual acuity itself. This relation between M_c, MAR and eccentricity can be approximated as a linear model, which has been shown to closely match the anatomical features of the eye [168, 49, 152, 15].

$$M_c^{-1} = M_{c_0}^{-1}\left(\frac{E_2 + E}{E_2}\right) = M_{c_0}^{-1}\left(1 + \frac{E}{E_2}\right) = \frac{M_{c_0}^{-1}}{E_2}E + M_{c_0}^{-1} \tag{2.1}$$

where $M_{c_0}^{-1}$ denotes the smallest resolvable angle and represents the cortical magnification at the center of the fovea, i.e., 0°, E is the eccentricity in degrees of visual angle, and E_2 is a constant of approximately 2°.

Acuity can also be represented as a Snellen Eye Chart Figure 2.11 B, helps determine visual acuity using letters on a grid, and The chart has eleven block letters. The first line has one very large letter, "E". Subsequent rows have increasing numbers of letters that decrease in size. The smallest row that can be read accurately from a certain distance indicates the visual acuity in that eye, making it possible to determine the Snellen fraction S.

$$S = \frac{Viewer's\ distance\ from\ the\ chart}{The\ distance\ at\ which\ anormal\ viewer\ could\ read\ the\ smallest\ letter}$$

For example in (20/40) fraction, The numerator indicates the testing distance (20 feet) and the denominator represents the distance at which a person with normal vision can correctly identify at 40 feet (i.e., the smallest Snellen letter has an angular size of 5 min of arc). The fraction 20/20 (6/6 Meter) indicates the normal visual acuity in the clinician's office and a person with this value can discern a detail of one arc minute. for example,

the gap between the ends of letter "E" subtends 1 min of arc, each letter subtends a total of 5 arcmin [11].

2.2.5 Visual Attention

Our lives take place in an overwhelmingly rich visual world — it contains far too much information to perceive at once, and the visual system's processing capacity is limited by the high metabolic cost of cortical computations. Given these limits, we need mechanisms to optimally allocate processing resources according to task demands. Typically, we scan the scene to aim the fovea at a place (area) we want to process more deeply and shift the fovea to another item [47].

Even though eye movements are an important mechanism of selectively focusing the fovea, it is also essential to recognize that there is more to attention than just moving the eyes to look at objects. There are three main types of visual attention:

1. Spatial attention: can be either overt or covert attention; overt attention is defined as selecting one location over others by moving the eyes to point at that location. Covert is defined as paying attention without moving the eyes: detecting objects and locations in the peripheral FOV, followed by an eye movement to that area to bring the spotlight of our overt attention to the task of seeing [42].

2. Feature-based attention : can be deployed covertly to a specific stimulus feature (e.g., color, orientation, or motion direction) of objects in the environment, regardless of their location.

3. Object-based attention: in which attention is influenced or guided by object structure [117].

The critical question in scanning the scene is which items or what determines where we give attention in a scene? The answer to this question is complicated because our looking behavior depends on several factors, including the scene's characteristics and the observer's knowledge and goals [47]. The scene characteristics include physical properties of the stimuli such as color, brightness, contrast, or orientation that is stimulus's *saliency* and they stand out to catch our attention. Saliency refers to the visual "attractiveness" or importance of components and features in the environment.

Capturing attention by stimulus salience is a *Bottom-up* process mechanism. These are thought to operate on raw sensory input, rapidly and involuntarily shifting attention to salient visual features of potential importance. But attention is not just based on what is bright or stands out. Cognitive factors are also essential; Attention mechanisms that implement our longer-term cognitive strategies are referred to as *Top-down* mechanisms [23].

In this section, we use the term priority to describe the degree to which a location captures attention due to combining top-down and bottom-up mechanisms. A saliency

map is a map of saliency values of the visual field. In contrast, a priority map is a map of priority values. Naturally, when a scene is viewed, a selection process follows the priority map, such that attention or gaze is more likely to be directed first to locations of higher priorities, and lower priority areas will be directed following the order of their priorities. A priority map can be assessed behaviorally by measuring how well observers discriminate or identify a visual target at the location, given a fixed viewing duration (the accuracy) and alternatively by their Reaction Time (RT) associated with finding or identifying a target at its location [182]. Accuracy should increase with the amount of time the target spends. Therefore, given a fixed viewing duration, greater accuracies should be coupled to shorter RT for selecting the target location. For example, studies of visual search often assume that a shorter RT indicates a larger saliency at the location of the search target. Later in Sec. 3.5.3.6, the thesis evaluates the proposed approach by conducting visual search experiments to assess the effect of the proposed system.

2.2.6 Eye Tracking

Eye- or Gaze-tracking can provide useful information with regards to the user's intent, attention, and their point of regard [85]. In this section, We will review, how the eye tracking techniques used and progressed, what kind of eye tracking techniques has been used and what kind of eye-tracking measurements are used to understand the user's intent and attention.

Dating as far back as the 18^{th} century, eye tracking has fascinated researchers to help understand human emotions, needs, as well as mental state [53]. One of the first eye tracking devices was built by Edmund Huey [64]. Made to understand the reading process, the trackers were a kind of contact lens with a hole for the pupil, and the eye movements were tracked using an aluminum pointer connected to the lens. A similar approach was used by Fitts et al. [44] for their study of the eye movements of pilots during aircraft landing. Another significant contribution in eye tracking was made by the Alfred Yarbus [180], showing that the gaze trajectories depended on the task to be executed. He developed a novel set of devices for recording and compensating for rapid eye movements [157]. The last three decades has seen a major revolution in eye tracking research and commercial applications due to the apparent ubiquity artificial intelligence algorithms and portable and consumer-grade eye tracking devices. Commercially available Head-Mounted Displays (HMD)s that include eye trackers are the Fove-0, Varjo VR-1, PupilLabs Core, and the HTC Vive Pro Eye [148].

Commonly used eye-tracking techniques in research are the following [100]: (1) Videooculography (**VOG**): video-based eye tracking using head-mounted or remotely-mounted visible light video cameras; (2) Video-based infrared (IR) pupil-corneal reflection (**PCR**): infrared lights to illuminate and measure the intensity of reflected infrared light; and (3) Electrooculography (**EOG**): measuring the corneo-retinal standing potential between the front and the back of the human eye.

The most commonly used eye tracking metrics are as follows [8]:

- **Fixation:** Fixation is the maintaining of the eye on a target for some time: long enough for the brain's visual system to perceive it. The time span is at least 100 ms, typically between 200 and 600 ms. Standard metrics used to evaluate fixations are the number of fixations, the fixation duration, and the fixation position [8].
- **Saccade:** A saccade represents the rapid eye movement between two consecutive fixations. This quick jumps of 2° or longer that take about 30–120 ms each. Standard metrics used to evaluate are the saccadic amplitude (i.e. the distance the saccade traveled), the saccadic duration, and the saccadic velocity in degrees per second [8].
- **Smooth Pursuit:** A Smooth Pursuit is a much slower tracking movement of the eyes intended to stabilize the moving stimulus on the focus. These movements are under voluntary control meaning the observer can choose whether or not to track a moving stimulus [8].
- **Scanpath:** A scanpath is a sequence of fixations and saccades in chronological order that represents the pattern of eye movements. Standard metrics used to evaluate scanpath are the distances between sequence fixations and scanpath duration [8].

2.3 3D Scene Capture

The advances in the field of telepresence and 3D reconstruction are especially attributed to the ready availability of good quality, low cost, consumer grade 3D scene capturing sensors (RGB-D cameras) [183]. RGB-D cameras, such as Microsoft Kinect, Intel Realsense, ZED cameras, etc. [82, 181, 118], have contributed profoundly to the advancement of novel algorithms in real-time point-cloud acquisition. These cameras combine their low-cost advantage with being lightweight, capturing pixel-level color and depth images at different resolutions, and at real-time rates (25 $\sim$ 30 Hz) [183].This section briefly summarizes the technologies for 3d scene capture and their use in this thesis.

State of the art in 3D capture indicates that several techniques have been developed and used for 3D scene acquisition. One could perhaps classify 3D scanning techniques into two types: contact and non-contact. Non-contact solutions can be further divided into two main categories, active and passive [160].

2.3.1 Contact

Contact 3D capturing techniques use Coordinate Measuring Machines (CMM) composed of mechanical arms that touch the surface of objects along user-defined profiles. The arm could be autonomous and touches the surface using a predefined, regular, grid, The precisions of such 3D captures can be in the order of microns. The scanner mechanism may have three different forms [160] :

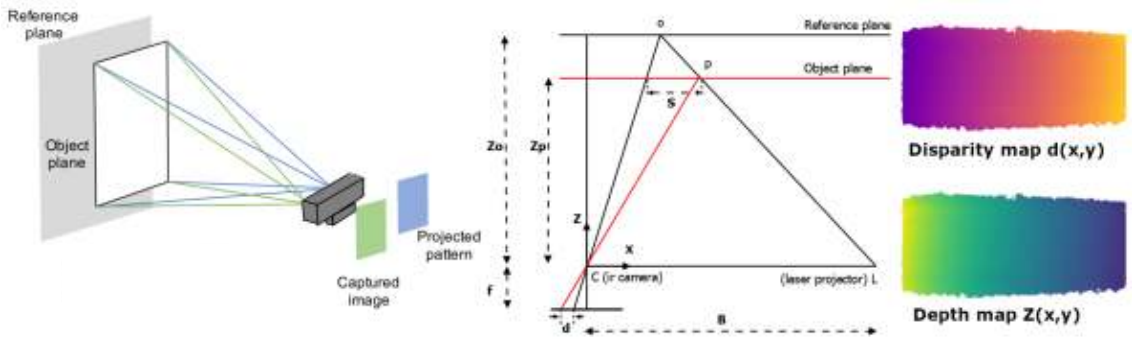


Figure 2.12: Relation between relative depth and measured disparity, Image adapted from [162].

- A carriage system with rigid arms held tightly in perpendicular relationship and each axis gliding along a track.
- An articulated arm with rigid bones and high precision angular sensors. The location of the end of the arm involves complex math calculating the wrist rotation angle and hinge angle of each joint.
- A combination of both methods may be used, such as an articulated arm suspended from a traveling carriage, for mapping large objects with interior cavities or overlapping surface .

2.3.2 Non-Contact Active

Active scanners work by emitting an infrared dot pattern with an infrared projector and receiving it. Possible types of emissions used include light, ultrasound, or x-ray. Microsoft introduced the popular Kinect 1 sensor which is based on *structured light*, and Intel Realsense later introduced active infrared (IR) stereo depth sensors.

The first generation Kinect V1 camera is a light-coded range camera that contains rgb camera, infrared projectors and detectors that map the depth through structured light. The depth information is computed by triangulation of the received and original infrared patterns. The principle is shown in Figure 2.12 where Z_p is the distance to the object, Z_o the distance to the reference plane, the distance between the IR camera (C) and laser projector (L) and d is the disparity between the two triangulated patterns, B is baseline between the infrared camera center and the laser projector (L) and f is the focal length. Then the depth of a scene point Z_p can be computed using the Equation 2.2.

$$Z_p = \frac{Z_o}{1 + \frac{Z_o}{f.B}d} \tag{2.2}$$

Alternatively, the Microsoft Kinect V1 has been replaced by a new device (Kinect V2) which is based on Time of Flight (ToF) principle. The basic operating principle is the one of continuous wave ToF sensors . The ToF depth sensing principle is based on measuring

the distance the light has travelled. This is done by modulating the light and measuring the phase shift introduced at the receiver compared to the original signal [30]. Some more details of this technology is listed on Table 2.2.

Meanwhile, Intel built an active IR stereo depth sensing family, Intel RealSense [16]: the D415 and the D435 models. Such devices differ from each other mainly in the glsFOV angles and in the exposition time of the camera-integrated shutter. The infrared laser projector projects non-visible structured IR pattern to improve depth accuracy in scenes as shown in Figure 2.13. The left and right infrared cameras capture the scene and the depth values for each pixel can be calculated by correlating the points on the left image to the right image. The focal length f and the baseline b between the two cameras are assumed to be known and the depth estimation problem becomes a disparity search along the scan line. Given the output disparity d, the the maximum Z value (depth) is obtained by the following formula:

$$Depth_{zmax} = \frac{f * b}{d} \tag{2.3}$$

where

$$f(pixels) = \frac{1}{2}\frac{Xres(pixels)}{\tan\frac{HFOV}{2}} \tag{2.4}$$

Where Xres is the horizontal resolution of the imager. HFOV is the horizontal field of view which is 90° for Model D435 and 65° for Model D415. The minimum Z value (MinZ) is defined by the following equation, taking into account that the camera searches in a disparity range of 126 bits [16] .

$$Depth_{zmin} = \frac{f * b}{d + 126} \tag{2.5}$$

Additional technical specification for Model D415 is detailed on Table 2.2.

2.3.3 Non-Contact Passive

Non-contact passive 3D scanning solutions do not emit any kind of radiation themselves, but instead rely on detecting reflected ambient radiation. Most solutions of this type detect visible light because it is a readily available ambient radiation. One of the most recent developed stereo sensors is the ZED camera, it has two high-resolution cameras that capture images (left and right) at the same time and transmit them to an external computer device for processing [118]. The depth is estimated using triangulation(re-projection) from the geometric model of non-distorted rectified cameras Figure 2.14. Assuming that the two cameras are co-planar with parallel optical axes and same focal length $f_l = f_r$, the depth Z of each point L is calculated by the Equation 2.6, here B is the baseline distance and $xi^l - xi^r$ is the disparity value.

$$Z = \frac{f * B}{xi^l + xi^r} \tag{2.6}$$

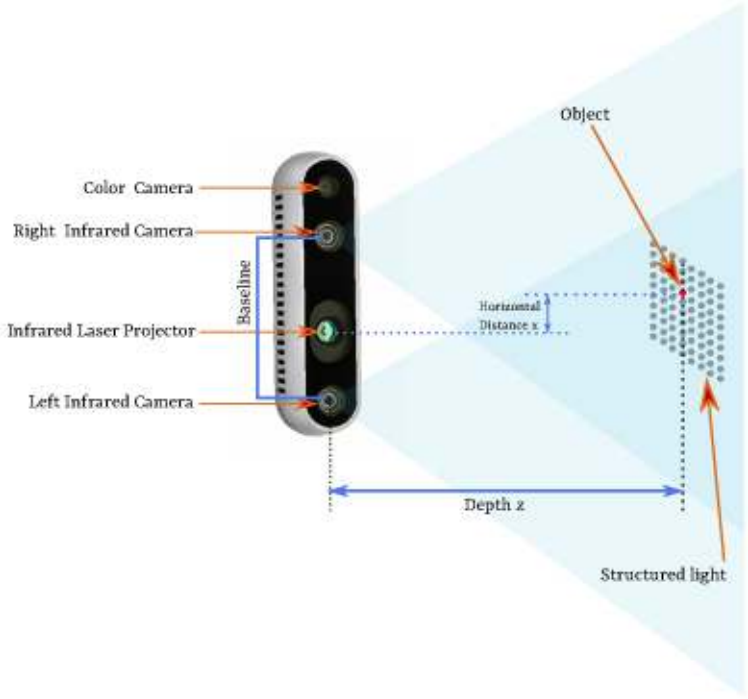


Figure 2.13: An overview of the Intelrealsense camera and the depth perception based on active infrared stereo vision.

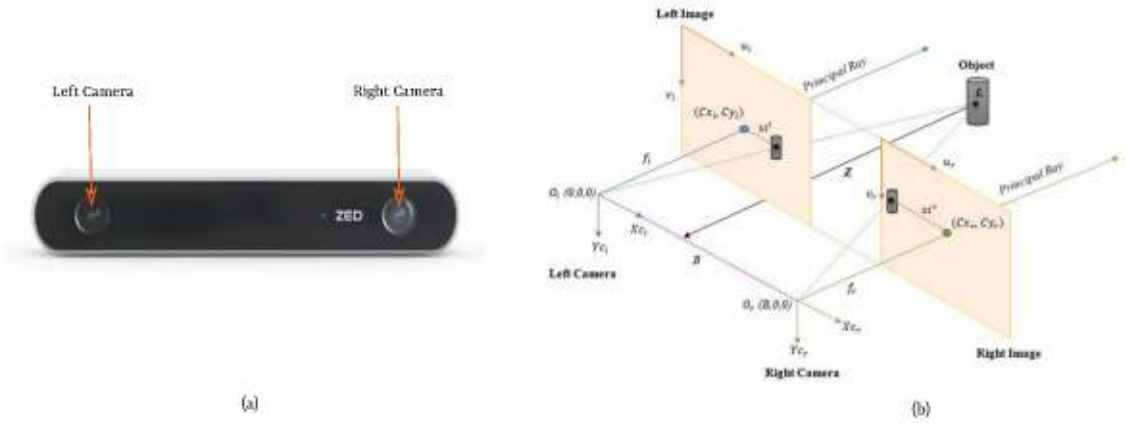


Figure 2.14: (a) A stereo Zed camera (b) The operation principle of the ZED camera [118].

2.4 Visual SLAM and 3D Reconstruction

Recent years have seen increasing research on visual SLAM problems for dense 3D reconstruction with various forms and functionality since the emergence of low-cost consumer-grade 3D scene capture sensors briefed in section 2.3. Most of the early works on V-SLAM using both monocular and steroe camera were based on tracking and mapping feature points: feature-based approach [32],[37], [86], [101]. Although these techniques permit the construction of an accurate map of an environment, the fact that they are feature-based means that they result in sparse point cloud maps Figure (a) 2.15 and suffer from textureless or featureless environments make them unsuitable for many applications.

Item	Value		
	IntelRealsense D415	**Kinect v2**	**ZED**
Depth technology	Active IR stereo	Time of Flight (ToF)	Stereo Camera
Depth Resolution	1280x720@30fps	512x424@30fps	2208x1242@15fps
Color Resolution	1920x1080@30fps	1920x1080@30fps	2208x1242 @15fps
Min Depth Distance	0.3 m	0.5 m	0.3 m
Max Range	~ 10 m	4.5 m	40 m
Depth Field of View	69.4°x42.5°(±3°)	70°x60°	90° x 60°
Baseline	55 mm	75 mm	120 mm

Table 2.2: The technical specification of 3D scene capture sensors

(a) (b)

Figure 2.15: (a) Sparse reconstruction of a map using feature-based approaches [101] and (b) a dense reconstruction of an office [170].

These challenges have motivated the development of dense mapping techniques that aim to use information from every pixel from the input frames to create 3D maps Figure (b) 2.15. The most widely used dense 3D reconstruction algorithms have mostly followed two strains: volumetric (voxel-based (see subsection 2.5.2.3)) and point-wise (surfel-based (see subsection 2.5.2.2)). After the popularity of KinectFusion [70], volumetric reconstruction has become the predominant approach. The depth map from the RGB-D camera is used to store the truncated signed distance to the closest surface in each voxel. This is parallelizable, whereas the 3D mesh surface representation is extracted from the voxel volume using the Marching Cubes algorithm [13]. The camera tracking is done with the Iterative Closest Point algorithm (ICP). KinectFusion matched the increasing popularity in GPGPU with the high-quality real-time depth maps provided by the Kinect sensor 2.3 to produce a system capable of camera frame rate dense 3D reconstruction. However, in spite of their popularity, volumetric methods have been shown to lack flexibility, e.g., expensive loop closures, fixed voxel resolution and voxel size limiting the quality of thee reconstruction, etc. [138].

The surfel-based approach instead represents the scene with a set of points [81, 170]. The point, i.e., surfel coordinates can be updated very efficiently, and the method is inherently adaptive for higher resolution requirements. It applies local model-to-model surface loop closure optimizations frequently as possible. This allows the system to stay

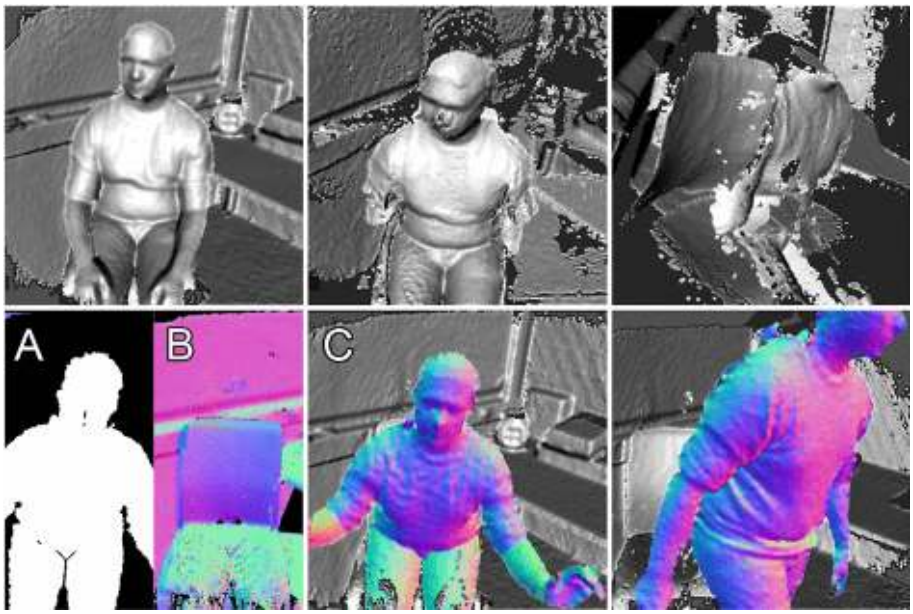


Figure 2.16: The Moving Person scene. Person sits on a chair, is reconstructed, and then moves. Dynamic parts occupy much of the field-of-view and cause ICP errors with previous approaches (top row). Segmenting the dynamics (A) and ignoring them during pose estimation (B) allows increased robustness (bottom row) [80]

close to the mode of the map distribution while utilizing global loop closure to recover from arbitrary drift and maintain global consistency. The main limitation is the computationally expensive meshing method required to generate continuous surfaces [138]. Both these approaches are currently being investigated in the research community [138, 28, 74].

Most sparse and dense 3D Reconstruction for static environments assumes the environment is static. Moving objects might be detected as outliers and ignored, which creates a failure of camera tracking and cannot robustly handle scene motion. For these challenges, a real-time dynamic environment reconstruction system is necessary. More recently, many researchers have begun exploring this area. Keller et al. [80] presented a simple and flat point-based representation, which directly works with the input acquired from range/depth sensors and dynamic objects are initially indicated by outliers in point correspondences during ICP and used a point-based region growing procedure to identify dynamic regions. These regions are excluded from the camera pose estimate, and their corresponding points in the global model are reset to unstable status, leading to a natural propagation of scene changes into our depth map fusion as shown in Figure 2.16.

More recently, researchers have begun to leverage Deep Neural Networks and their ability to learn from large amounts of training data to improve 3D reconstruction. This technique can add semantic information to the SLAM, enhancing the dense, dynamic 3D reconstruction. This approach is usually referred to as "semantic SLAM" that includes the semantic information into the SLAM process to improve the performance and representation by providing high-level understanding, robust performance, resource awareness,

and task-driven perception.

Xujie et al. [77] categorized the approaches of dealing with dynamic objects in visual SLAM into three main directions:

- Deforming the whole world in a non-rigid manner in order to include a moving object [112].
- Specifically building a single static background model while ignoring all possibly moving objects and thus improving the accuracy of camera tracking [71] [141] [4][7].
- Modeling the dynamic components by creating sub-maps for every possibly rigidly moving object in the scene while fusing corresponding information into these sub-maps [132] [5][131] [176] [109] [105] [153] [50].

The reconstruction of dynamic scenes is **computationally and algorithmically more challenging** than its static reconstruction counterpart. Modeling the non-rigid motion of general deforming scenes requires orders of magnitude more parameters than the static reconstruction problem [112]. In general, finding the optimal deformation is a high-dimensional and highly non-convex optimization problem that is challenging to solve, especially if **real-time performance** is the target.

A recent research approach in dynamic 3D reconstruction working at a maximum of 5Hz [131], demonstrates the challenges in this domain. The proposed approach is a multi-instance dynamic RGBD SLAM system takes full advantage of using instance-level semantic segmentation: Mask R-CNN [55]. Although, the semantic segmentation provides good object masks, it suffers from latency (around 5 Hz) and object bounderis leak into boundaries. For this reason the proposed approch applied a geometric segmentation algorithm, based on an analysis of depth discontinuities and surface normals. The same semantic segmentation technique Mask R-CNN is used by Xu et al. [176] and Bescos et al. [7] to find semantic instances. The research work by Xu et al. [176] (see Figure 2.17) used an octree-based volumetric representation, that follows geometric edge refinement to solve leaked mask boundaries and this work can run at 2-3 Hz on a CPU, excluding the instance segmentation. The work by Bescos et al. [7] named : DynSLAM tracking, mapping and inpainting in dynamic scenes. Using the semantic segmentation most of the dynamic objects can be segmented. However, there are objects that cannot be detected by this approach because they are not a priori dynamic, but movable. For this reason they proposed to use multi-view geometry models to assign keypoints in to static and dynamic objects.

Dense semantic SLAM beyond indoor scene reconstruction was proposed by Bescos et al. [5]. The techniques uses a volumetric representation based on voxel block hashing for large scale environment. A stereo camera is used to infer the egomotion and reconstruct the surrounding world. The system estimate the 3D motion of each new detection using

Figure 2.17: Qualitative demonstration of the work MID-Fusion [176] input RGB (top row), semantic class prediction (middle row) and geometry reconstruction result (bottom row).

the scene flow and semantic segmentation information, comparing it to the camera ego-motion to classify each object as static, dynamic, or uncertain. The semantic segmentation process uses Multi-task Network Cascades (MNC) [29]. The system is capable of running on a PC at approximately 2.5Hz.

2.5 Efficient 3D Rendering

A rendering pipeline generates 2D images of a 3D scene, given a specific camera pose and light sources. Once a real-time point cloud is acquired or reconstructed, the graphics pipeline turns the 3D scene into 2D displays for visualization. The pose and shapes of the 3D scene are determined by their geometry, the environment's characteristics, and the camera's placement in that environment. The appearance of the objects is affected by material properties, light sources, textures (images applied to surfaces), and shading equations [3]. The steps required for the rendering process rely on the software, hardware, and desired display characteristics. The most used universal graphics application program interfaces are Direct3D [150] and OpenGL [83]. The most common rendering pipeline is presented in the following section.

2.5.1 Rendering Pipeline

A real-time graphics rendering pipeline consists of several stages, each performing part of a larger task. These stages execute in parallel on hardware on graphics cards and general-purpose graphics processing units (GPGPU's). A coarse division of the real-time rendering pipeline into four main stages — application, geometry processing, rasterization, and pixel processing as shown in Figure 2.18.

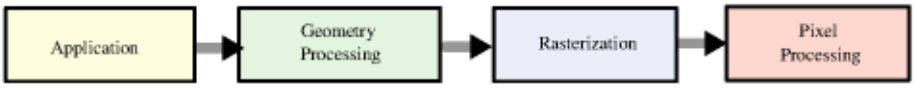


Figure 2.18: Basic representation of the graphics rendering pipeline.

Application

The application stage is driven by the application and is typically implemented in software running on general-purpose CPUs. These CPUs include multiple cores and can utilize multiple threads of execution in parallel to execute many various tasks that the application may need, like preparing the geometry data before submission to the GPU for rendering, physics simulation, animation, AI, and many others more. The application stage is important to prepare and submit the geometry to the geometry stage for rendering. This stage consists of rendering primitives like points, lines, and triangles that might end up being displayed on the screen.

The most common process commonly implemented on the application stage is to take care of input sources, such as eye trackers, a keyboard, a force feedback device, a mouse, or a head-mounted display.

Geometry Proccessing

This processing stage is responsible for most of the per-triangle and per-vertex operations. It applies transformations of 3D primitives and turns them into 2D coordinates that can be fed into the next stage (Rasterization) to do the actual drawing. This stage is further divided into the following functional stages: Model and View Transformation, vertex shading, projection, clipping, and screen mapping as shown in Figure 2.19.

Figure 2.19: Geometry Processing functional stages.

(1) Model and View Transformation: Before geometry reaches the stage of being displayed on the screen, it gets transformed into several different coordinate systems, also known as coordinate spaces. The main idea is to place the objects in a scene and view them from a viewpoint.

A object's geometry is first defined by its coordinate space called local or model space. To define the model space into a local coordinate system, it will be defined relative to the origin of the local coordinate system, which is usually the center of the 3D model. For geometry to be positioned and oriented within the three-dimensional world, it must be transformed from the local/model coordinate system to the world coordinate system or world space.

As noted earlier, The main idea is to place the objects in a scene and view them from a standpoint, and only the models that the camera sees are rendered. The camera has

a location in world space and a direction used to place and aim the camera. After the camera has been placed and oriented in world space, the view transform is applied. The view transform's purpose is to place the camera in the origin of the world coordinate system and aim its direction down the positive or negative z-axis (depending on the implementation), aligning the geometry in the same way as shown in the Figure 2.20. After the view transformation is applied, the new coordinate system is known as the view coordinate system or view space.

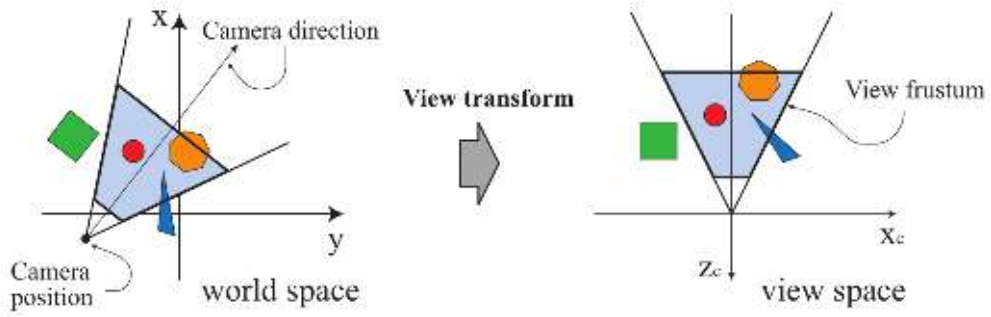


Figure 2.20: The view transform, adopted from [3].

(2) Vertex Shading: To produce a realistic scene, it is not sufficient to render the shape and position of objects, but their appearance must be modeled. Shading is the operation performed to determine the effect of a light source on the geometry. To produce a realistic scene, it is not sufficient to render the shape and position of objects, but their appearance must be modeled. Shading is the operation performed to determine the effect of a light source on the geometry. The shading process involves computing a shading equation at various points on the object. Typically, some of these computations are performed during geometry processing on a model's vertices, and others may be performed during per-pixel processing [3].

(3) Projection: The rendering pipeline perform projection and clipping, which transforms the view volume into a unit cube with its extreme points at $(-1,-1,-1)$ and $(1,1,1)$. That makes it easier for the rasterizer to render things. This unit cube is called canonical view volume. There are two main types of view volumes of interest: orthographic and perspective projections.

Orthographic projection is the simplest type: it consists of simply projecting points and vectors parallel onto a plane . The view volume is a rectangular box that is then transformed into a unit cube by the projection transform. The main characteristic of an orthographic projection is that lines that are parallel before the transformation remain parallel after it (see Figure 2.21-right).

The perspective projection is a more complex case. In this projection, the farther an object is from view, the smaller it appears after the transformation. In contrast to the orthographic projection, parallel lines may converge at the horizon (see Figure 2.21-left).

This change of object size with distance from the point of view mimics our perception of the real world regarding the relationship of size and distance. Geometrically, the view volume, called a frustum, is a truncated pyramid with a rectangular base. The frustum is transformed into the unit cube as well. Geometry that has undergone the perspective or orthographic projection resides in a coordinate space known as normalized device coordinates.

(4) Clipping: After applying the view and projection transformations, primitives that reside inside the view frustum pass to the next stage of the pipeline, and primitives that reside outside the view frustum do not. These primitives must undergo the process of clipping. The sides of the view frustum parallel to the camera's lens are called near and far clipping planes. Only objects between the near and far clipping planes are rendered. The near clipping plane also serves as the plane on which objects are projected and is generally positioned close to the eye or camera (see Figure 2.21).

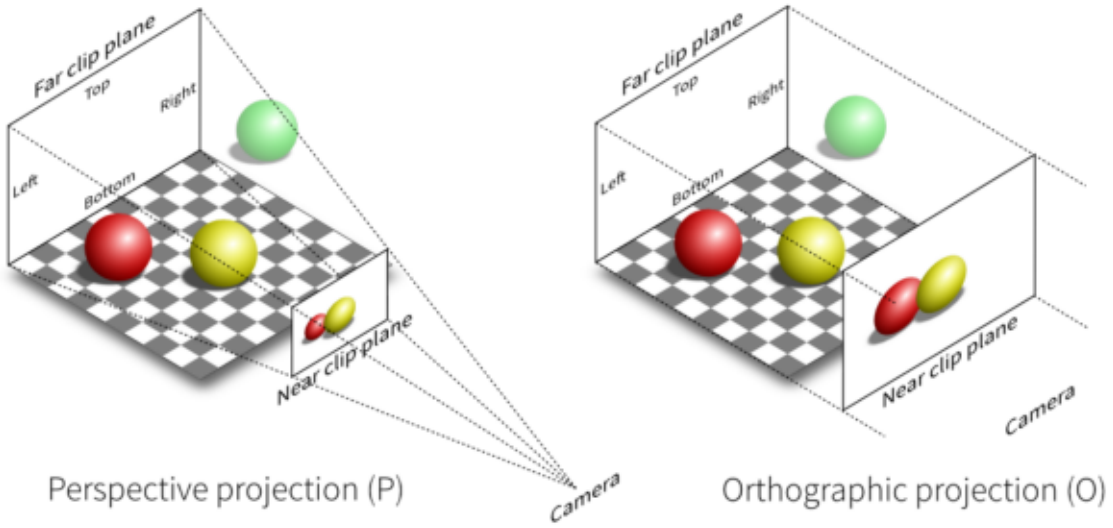


Figure 2.21: The perspective and orthographic projection.

(5) Screen Mapping: Only the (clipped) primitives inside the view volume are passed on to the screen mapping stage, and the coordinates are still three-dimensional when entering this stage. The x- and y-coordinates of each primitive are transformed to form screen coordinates. Screen coordinates together with the z-coordinates are also called window coordinates. After screen mapping is complete the data are passed to the rasterizer stage for further processing [3].

Rasterization

All the primitives that pass through the clipping process are rasterized: which means that all pixels inside a primitive are found and sent further down the pipeline to pixel processing. The rasterization split into triangle setup and triangle traversal functional

stages and the pixel Processing into pixel shading and Merging functional sages as shown in Figure 2.22.

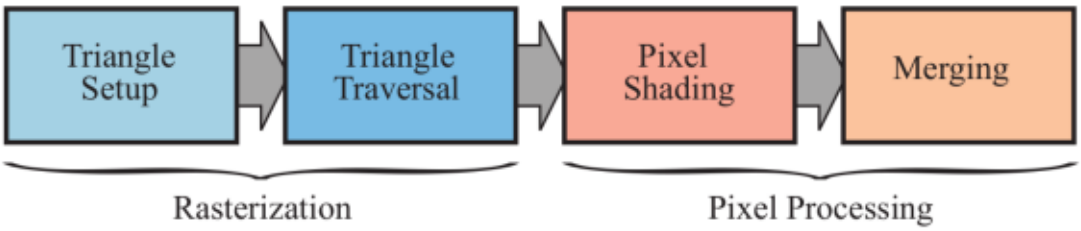


Figure 2.22: The rasterization and pixel processing functional stages, from [3].

(1) Triangle Setup: Triangle setup is the stage responsible for calculating various data about the tringle's surface. This data is used to interpolate the shading data originating from the geometry stage and scan conversion.

(2) Triangle traversal: Each pixel with its center covered by the triangle is checked, and a fragment is generated for the part of the pixel that overlaps the triangle. Fragments are a collection of values produced by the rasterizer and represent a sample-sized part of the primitive that is rasterized. The fragment size is relative to the pixel area, but the rasterizer can produce multiple fragments from the same triangle per pixel.

(3) Pixel shading: Any per-pixel shading computations are performed here, and These computations use the interpolated shading data produced by the triangle traversal stage. The purpose of the fragment shading stage is to compute a final color value, considering all the shading data provided from the previous stage for each fragment that is then submitted to the next stage.

(4) Merging: The information for each pixel is stored in the color buffer, which is a rectangular array of colors (a red, a green, and a blue component for each color). The merging stage's responsibility is to combine the fragment color produced by the pixel shading stage with the color currently stored in the buffer [3].

This pipeline resulted from decades of API and graphics hardware evolution targeted to real-time rendering applications and a very high-level overview: It is important to note that this is not the only possible rendering pipeline.

2.5.2 Efficient Rendering

For realistic rendering, a high pixel density and high refresh rates are necessary for realism, interaction, and immersion. Efficiency and realism are the two main goals in rendering realistic scenes. If the efficiency of an algorithm is improved immensely, a more significant proportion of computational resources can be spent on creating more realistic imagery. The graphics community has proposed several methods to improve the efficiency of rendering algorithms—this thesis focuses on explaining more general concepts and focuses closely on relevant perception-driven approaches. This section presents

an overview of general strategies that improve the efficiency of 3D geometry rendering.

2.5.2.1 Polygon Simplification

One of the early works used to improve rendering performance is by adapting a sampling function that reduces the complexity of the input 3D scene. A technique used to simplify a 3D scene is a static *polygon simplification* method; this technique can be categorized into two parts: geometric simplification and topology simplification. The most basic geometric simplification techniques are vertex clustering or vertex removal techniques [58] [165]. However, as discrete changes between the models from the set can become visible, dynamic simplification methods have been developed to simplify the model continuously at runtime, which makes the algorithm for a dynamic continuous level-of-detail and easier for network streaming [3].

2.5.2.2 Point-Based Approaches

Levoy et al. [133] presented a 3D geometry representation and rendering technique based on points. A more lightweight rendering of large-scale hundreds of millions of polygon can be derived by removing or averaging nearby points in the set, directing to *Point-Based approaches*. Pfister et al. [124] presented point primitives without explicit connectivity representation technique called surface elements (surfels). These surface elements store positions, normals, weight, radius and color. Surfels are rendered using an octree-based approach and splatting. A more detailed survey on point-based rendering techniques can be found here [135].

2.5.2.3 Voxel-Based Approaches

Voxel-Based approaches are another way of simplifying the rendering process. Voxel-grids are a regular grid structure of different attributes; the regularity in the structure makes them well suited for Level of Detail approaches as coarser representations of a scene can be represented by downsampling the 3D grid to a grid with a lower resolution. Laine and Karras [88] presented a compact data structure for storing voxels and an efficient algorithm for performing ray casts using voxel grid structures.

2.5.2.4 Model-Driven Approaches

Further researchers have proposed a Model-driven approach that augments the techniques described above: This techniques exploit human perception. One of the early works is an image-driven simplification, a framework that uses images to decide which portions of a model to simplify [92]. Similar to this work, Scoggins et al. [140] presented a method to enable matching of level-of-detail (LOD) models to image-plane resolution. A relationship is developed between image sampling rate, viewing distance, object projection, and expected image error due to LOD approximations. These techniques measure

the perceived quality of the output based on the view or image contrast and the spatial frequency. However, they do not focus specifically on textures and effects caused by dynamic lighting, as this needs a deeper knowledge of low-level perceptual processes.

Research work by Williams et al. [173] presented a model of low-level human vision to estimate the perceptibility of local simplification operations in a view-dependent Multi-Triangulation structure. Their algorithm improves on prior perceptual simplification approaches by accounting for textured models and dynamic lighting effects. A high-level perception and attention mechanism have been investigated to preserve the salient features of the mesh using attention mechanisms [61].

2.5.2.5 Gaze-Contingent Approaches

Head and eye trackers as well as inertial measurement units can be utilized to simplify a 3D geometry Level of Detail. This techniques is called *Gaze-Contingent approaches* [165][96, 122]. This approach can be used to simplify simplify geometry progressively based on the gaze. The degree of mesh simplification is controlled by a perceptual model that exploits the visual acuity and the contrast sensitivity of the HVS. A large and growing body of research has investigated how to utilize the HVS. Guenter et al. [49] presented one of the first foveated rendering techniques to accelerate graphics computation. They proposed the rendering of three eccentricity layers around the user's fixation point and each layer's parameters were set by calculating the visual acuity. Stengel et al. [149] proposed gaze-contingent rendering that only shades visible features of the image while cost-effectively interpolating the remaining features, leading to a reduction of fragments needed to be shaded by 50% to 80%. The work by Bruder et al. [15] used a sampling mask computed based on visual acuity fall-off using the Linde-Buzo-Gray algorithm. The sampling mask is used to reconstruct the image based on Voronoi cells using natural neighbor interpolation, and apply temporal smoothing to attenuate sampling artifacts. In the commercial domain, more and more VR headsets are exploiting foveated rendering for increased realism and reduced graphical demands [17].

2.6 Immersive Visualization Systems For Teleoperation and Telepresence Applications

The majority of research works for Telepresence or Teleoperation applications, provided video interface through monocular and stereoscopic 360° videos [97]. Unfortunately, these interfaces only allow users to see from a fixed direction [33]. Furthermore, they do not respond to any head motion such as moving left/right, forward/backward, or up/down. Immersive Teleoperation interfaces require a freedom of motion in six degrees of freedom so that users can see the correct views, regardless of where the user is and where the user is looking. Researchers have long seen the advantages of using 3D VR environments in telepresence [6, 106]. The more recent investigations have focused on

using affordable VR devices in live human telepresence as well as remote human-robot interaction [119, 93], integrating VR graphics engine softwares (e.g., Unity3D, Unreal) with compatible hardware. However, as noted previously, most traditional approaches have been limited in their scope for telepresence [97, 18, 75].

On the other hand, VR-based immersive interfaces for robotic teleoperation have gained a lot of traction in recent literature, which include models of the remote robots along with the real-time point-cloud rendering inside VR as well as gesture tracking approaches and real-time stereo video [123, 87, 93, 159, 171, 130]. Most of these approaches rely on standard encoding and communication protocols, e.g., those included in ROS [127], Point Cloud Library (PCL) [134], UDP-based streaming protocols (Real-time Transport Protocol (RTP), Real Time Streaming Protocol (RTSP)) for video / image formats.

Maimone et al. [99] were one of the first researchers to investigate a telepresence system offering fully dynamic, real-time 3D scene capture and viewpoint flexibility through head-tracked stereo 3D display. Orts-Escolano et al. [119] presented "holoportation" doing high quality 3D reconstruction for small fixed-sized regions of interest. Authors in [41, 107] present remote exploration telepresence systems for large- and small-scale regions of interest with reconstruction and real-time streaming of 3D data. Researchers at the University of Bonn have taken this idea further with simultaneous immersive live telepresence for multiple users for remote robotic telepresence and collaboration [167, 151]. These systems use one entity (robot or another user) to capture the data, a cloud-based real-time reconstruction framework that does camera localization and volumetric fusion in real-time. It consists of a cloud server to manage the global scene model, control the data transmission according to the requests by remotely connected users, and visualization components that update the locally generated meshes for the individual remote users. This system uses voxel block hashing for low latency streaming and reconstruction of the environment for remote users. The authors evaluated the proposed system for live-telepresence as well they evaluate its use for robot teleoperation and showing it's benefits over purely 2D video-based teleoperation. User experience evaluation showed that the proposed system was well-suited for teleoperation and allowed moving the robot to target positions more efficiently; users had a high degree of situation awareness and self-localization in the simultaneously captured scene and could easily assess the terrain for navigation purposes.

A good example of an immersive visualization system for remote telerobotic applications can be found in research work by research work by Naceri et.al. [108]. In this research work, the authors showed a system named "Vicarios", a VR based interface to facilitate intuitive real-time remote teleoperation while utilizing the inherent benefits of VR, including immersive visualization, freedom of user viewpoint selection, and fluidity of interaction through natural action interfaces. As shown in Figure 2.23, The setup have different components for the user, the remote environment, and a visualization interface. A gesture/motion controllers at the user site convey the commands to the remote robots.

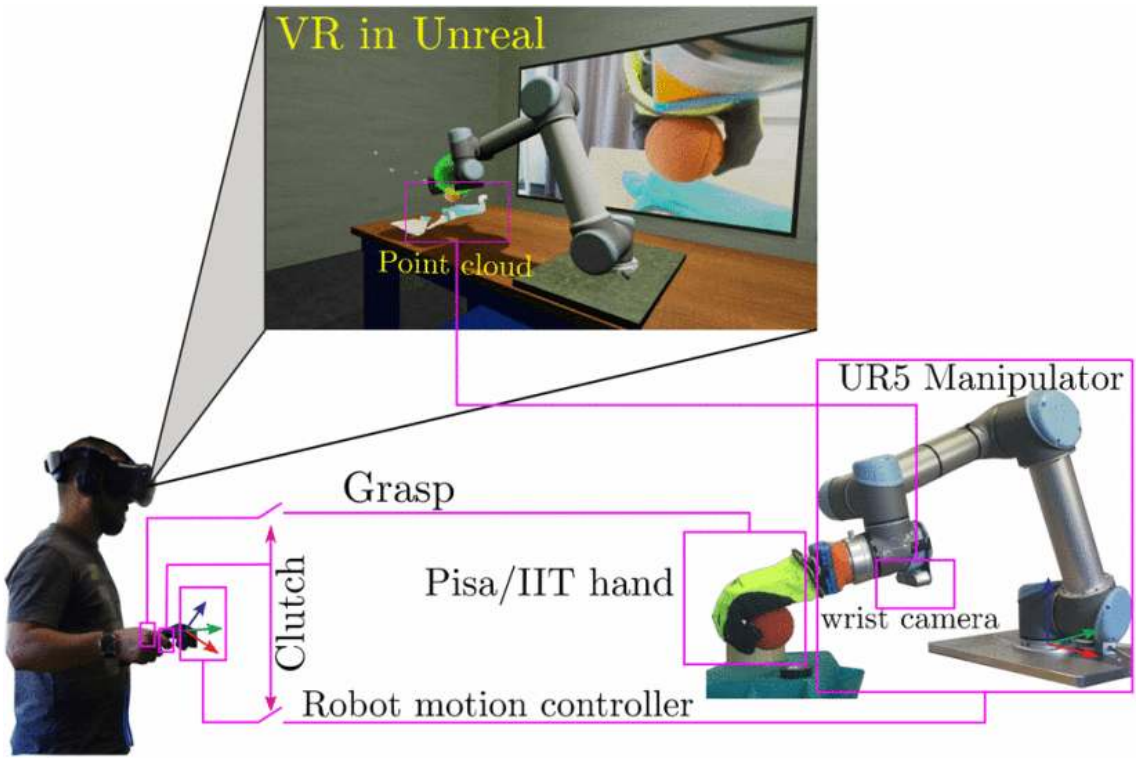


Figure 2.23: Vicarios interface: the user teleoperating a remote robotic arm using the HTC vive pro VR system, Unreal Engine (UE) VR graphics engine, Remote camera provides video and depth feedback in real-time, image adapted from [108].

The remote environment and the robot model are rendered virtually in the VR interface. Between the user and the remote environment a communication network is established to allow a real-time data exchange, i.e., sending commands, receiving remote robot status, and receiving real-time video and point-cloud information. The authors did user studies to understand the performance and utility of the interface and test the effectiveness of the different features, including teleporting and viewpoint-independent motion mapping. In particular, they performed hypothesis testing to assess the impact on the user, positive or negative, of the Vicarios VR-based interface, against the traditional video-only interface. The findings of this study suggest that users' performance with the VR-based interface was either similar to or better than the baseline traditional stereo video feedback, supporting the realistic nature of their VR-based immersive interfaces. This study concludes that latency is an inevitable problem in teleoperation interfaces that profoundly alters the operators' performance.

2.7 Conclusion

This chapter presented the most relevant theoretical foundations to remote telepresence and telerobotics systems; It briefly described the technology and gives an overview of recent research works on 3D tele-immersive systems (section 2.1.1). Furthermore, It explained why designing and studying immersive interfaces is valuable for improved interactions in remote visualization systems, especially how to present complex remote scenes into a representation that can be presented to a user 2.1. The section presented the essential psychological and physiological principles of the human user, and following it presented technological factors in Immersive visualization systems. Section 2.2 presented a general overview of the HVS and its limitations and the associated models to describe key visual processes or mechanisms; it briefly presented the limitations and potentials of the human vision and presented how the vision in the periphery differs from that near the center. In addition, It looked briefly at the separate brain areas that determine our perception of different qualities.

Later, Section 2.3 provided more insight into remote sites (environment) by studying state of the art in environmental scene acquisition techniques, mainly 3D capture techniques, and then provided an insight on how to create a realistic, high-quality virtual environment from the acquired information using visual SLAM techniques (section 2.4) and finally, it briefly described how to visualize (render) the information to the user, once a real-time point cloud is acquired or reconstructed, the graphics pipeline renders the 3D scene for the head-mounted display, This implies rendering the scene twice, once for each eye (section 2.5).

In Section 2.5.2, the thesis discussed the problem of efficient rendering when there is a requirement for efficient and realistic visualization. It discuss on how to have high pixel density and refresh rates for realism, interaction, and immersion. It discussed different approaches for efficient rendering. The final section in this chapter has discussed different state of the art literature works in immersive visualization systems for teleoperation and telepresence applications.

3

Gaze contingent Remote-Immersive Visualization Framework

"Every day the urge grows stronger to get hold of an object at very close range by way of its likeness, its reproduction."
(Walter Benjamin, 1936)

The previous chapters presented a brief investigation of the state-of-the-art immersive interfaces for telepresence and teleoperation systems and technological, perceptual, and cognitive constraints in designing such systems. The next step taken in this chapter is to develop immersive remote visualization framework that utilizes acuity fall-off in the HVS to facilitate the processing, transmission, buffering, and rendering in VR of dense 3D reconstructed scenes while simultaneously reducing throughput requirements and latency. This chapter addresses research question 3:

Research Question 3: *How do we design an improved remote visualization system using the HVS with reduced latency and throughput requirements compared to the current state of the art techniques?*

The work in this chapter is based on the following publication:

Journal: Towards Foveated Rendering For Immersive Remote Telepresence, Yonas Tefera, Dario Mazzanti, Sara Anastasi, Darwin G. Caldwell, Paolo Fiorini, and N. Deshpande, The Eurographics Association, 2022.

Conference: Towards Foveated Rendering For Immersive Remote Telerobotics, Yonas

Figure 3.1: *Foveated* rendering in VR of a real-time 3D reconstructed remote scene. The user's gaze itself is used to foveate the remote point-cloud. The scene is rendered with dense points in the fovea region and progressively sparse rendering in the peripheral regions.

Tefera, Dario Mazzanti, Sara Anastasi, Darwin G. Caldwell, Paolo Fiorini, and N. Deshpande, Workshop on Virtual, Augmented, and Mixed Reality for Human-Robot Interaction (VAM-HRI), Workshop at IEEE HRI, 2022.

3.1 Exploiting Human Visual System Acuity

As briefly discussed in section 2.2.1, humans perceive visual information through sensory receptors in the eyes. The process begins when light passes through the cornea, entering the pupil, and then gets focused by the lens onto the retina. This is then processed in the brain where an image is formed. The retina has two kinds of photoreceptors: cones and rods. Cones are capable of color vision and are responsible for high spatial acuity. Rods are responsible for vision at low light levels. As shown in Figure. 2.9 in section 2.2.1 the cone density is highest in the central region of the retina, and reduces monotonically to a fairly even density into the peripheral retina region. Retinal eccentricity (or simply, eccentricity) implies the angle at which the light from the image gets focused on the retina. This distribution of the photoreceptors gives rise to the concept of *Foveation*, and helps define the idea of visual acuity.

3.1.1 Foveation

As noted, the density of photoreceptors (cones and rods) declines monotonically and in a continuous manner from the center of the retina to its periphery. Nevertheless, approximating the retina as being formed of discrete concentric regions, where the density of the photoreceptors corresponds to eccentricity angles, helps simplify the concept. This is the concept of *Foveation*, seen in Fig. 3.2-A.

3.1.2 Visual Acuity

As described in detail in section 3.1.2, The most popular way of determining visual acuity is to quantitatively represent it in terms of *minimum angle of resolution* (MAR, measured in arcminutes). In fact, visual acuity is represented as the reciprocal of MAR, but MAR

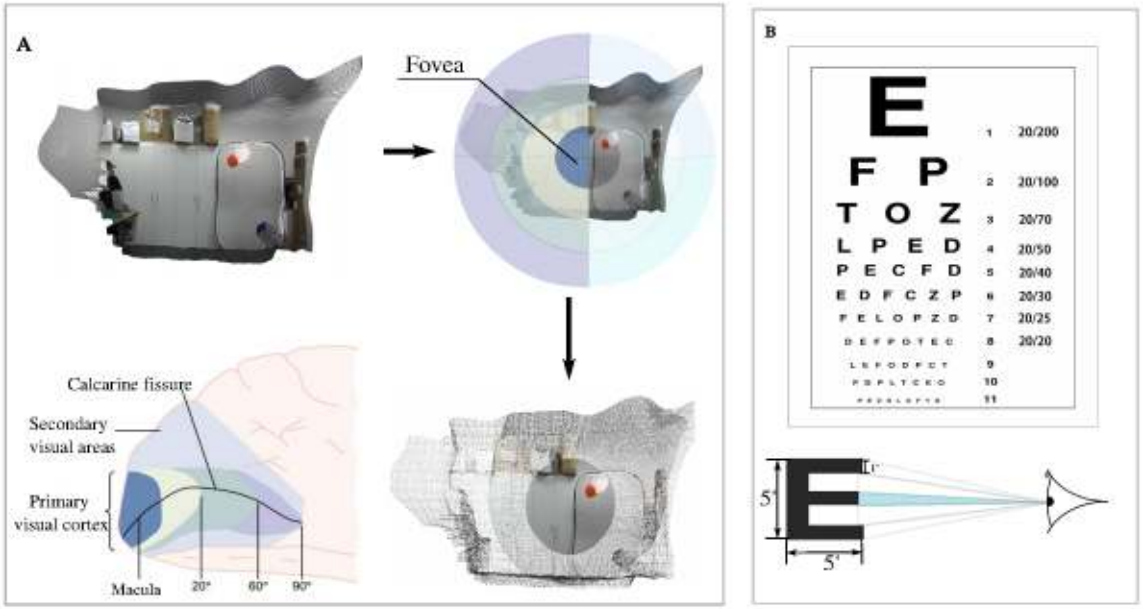


Figure 3.2: A) Retinotopic organization of the primary visual cortex. Foveation applied to a sample point-cloud is seen at the bottom-right and B) Snellen Eye Chart.

Table 3.1: Human retinal regions and their sizes in diameter and angles.

	Region	Size in Diameter	Size in Angle
R_0	Fovea	1.5 mm	5°
R_1	ParaFovea	2.5 mm	8°
R_2	PeriFovea	5.5mm	18°
R_3	Near Peripheral	8.5 mm	30°
R_4	Mid Peripheral	14.5mm	60°
R_5	Far Peripheral	26 mm	> 60°

itself is now quite common in literature [168, 49, 152]. MAR can be understood as the smallest angle at which two objects in the visual scene are perceived as separate [168] or as a Snellen Eye Chart Figure 3.2-(B). Snellen Eye Chart is expressed as a fraction, such as (20/40) as shown in Figure 3.2-(B). The numerator is the testing distance (20 feet) and the denominator represents the distance at which a person with normal vision can correctly identify at 40 feet (i.e., the smallest Snellen letter has an angular size of 5 min of arc). The fraction 20/20 (6/6 Meter) indicates the normal visual acuity in the clinician's office and a person with this value can discern a detail of one arc minute. For example, the gap between the ends of letter E subtends 1 min of arc, each letter subtends a total of 5 arcmin [11]. Since the MAR is the reciprocal of the visual acuity, the minimum angle of resolution MAR can also be understood in relation to the cortical magnification factor, M_c [24]. M_c accounts for the number of neurons allocated to process the information from the visual field, as a function of the eccentricity [24]. This relation between MAR and eccentricity can be approximated as a linear model, which has been shown to closely

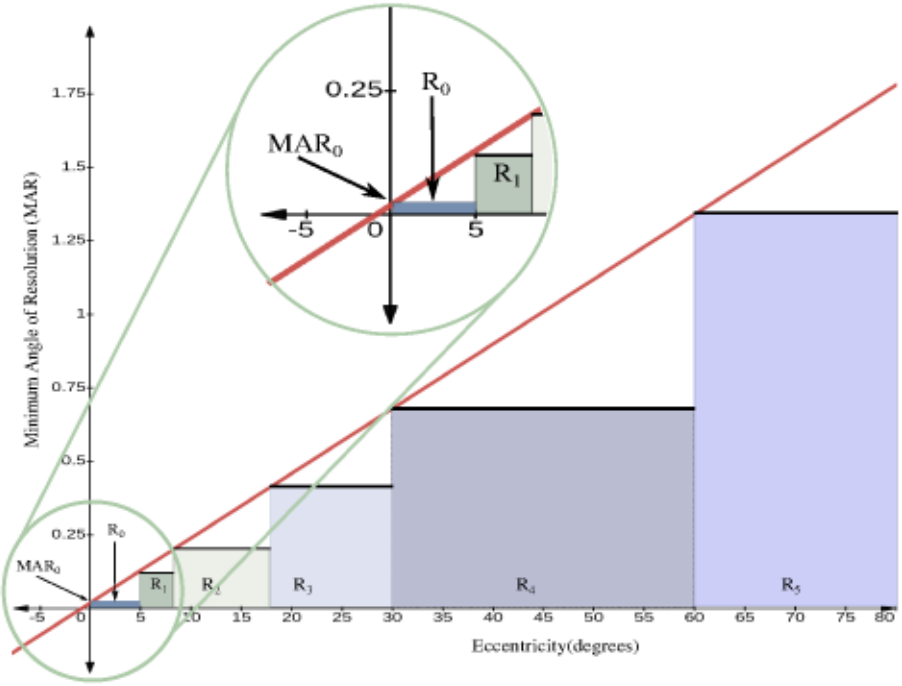


Figure 3.3: Minimum Angle of Resolution against the eccentricity values for the retinal regions ($R_0 - R_5$) defined in Table 3.1, based on Eq. (3.2).

match the anatomical features of the eye [168, 49, 152, 15].

$$M_c^{-1} = M_{c_0}^{-1}\left(\frac{E_2 + E}{E_2}\right) = M_{c_0}^{-1}\left(1 + \frac{E}{E_2}\right) = \frac{M_{c_0}^{-1}}{E_2}E + M_{c_0}^{-1} \tag{3.1}$$

Where $M_{c_0}^{-1}$ denotes the smallest resolvable angle and represents the cortical magnification at the center of the fovea i.e., 0° in eccentricity, E is the eccentricity in degrees of visual angle, and E_2 is a constant of approximately 2°. Since M_c^{-1} is directly proportional to MAR, this can be rewritten as,

$$MAR = mE + MAR_0 \tag{3.2}$$

Here MAR_0 is the intercept, which signifies the smallest resolvable eccentricity angle for humans, and m is the slope of the linear model. The smallest resolvable angle that a healthy human, with a normal visual acuity between 20/20 and 20/10, can discern varies between 1 and 2 arcminutes, i.e., 1/60° -to- 1/30° (1° = 60 arcminutes). Authors in [49] experimentally determined the values of m based on observed image quality, ranging between 0.022 to 0.034. Figure 3.3 captures this linear relationship. Eq. (3.2) establishes how the visual acuity degrades as a function of eccentricity, i.e., since increasing MAR implies degrading acuity. Similar to the discrete retinal regions in Table 3.1, the linear relationship can also be utilized with a piecewise constant approximation, i.e., each retinal region having a distinct constant MAR value [49]. Figure 3.3 shows the bands of the constant MAR values for each retinal region, based on the eccentricity values in Table 3.1.

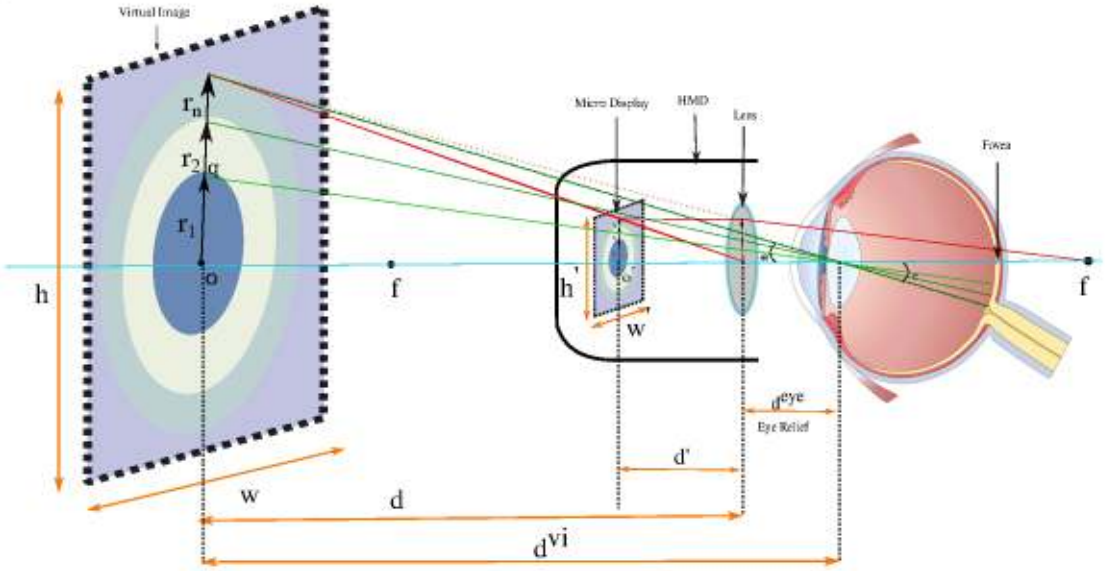


Figure 3.4: A 2D rendering of the perceptual model designed to approximate the properties of human visual system, adapted from [65].

3.1.3 Image Formation In VR HMDs

The optical system in HMDs used for VR environments allows light rays to converge and focus on a point on the retina, thus producing virtual images. To do this, there are lenses placed between the HMD displays and the viewer's eyes. Such an arrangement gives the illusion that the images are not on the HMD display, but out to a distance where they can be viewed comfortably. Figure 3.4 shows the perceptual optical model of an HMD, which is designed to approximate the properties of the human visual system. The HMD display, a *microdisplay*, is positioned at a distance of $d^{'}$ from the lens and at a distance of d^{eye} from the eye relief (distance from the last lens surface to the eye pupil), with a focal length of f, a width $w^{'}$ and a height $h^{'}$. The lens creates a magnified virtual image with a width w and height h on the retina and located at distance of d^{vi} from the eye and d from the lens. This relationship between microdisplay and the virtual image can be defined using the Gaussian thin lens Eq. (3.3).

$$\frac{1}{d} + \frac{1}{d^{'}} = \frac{1}{f} \tag{3.3}$$

$$M_l = \frac{h}{h^{'}} = \frac{w}{w^{'}} = \frac{-d}{d^{'}} = \frac{f}{f - d^{'}} \tag{3.4}$$

Eq. (3.4) gives the formula for the linear magnification of a thin lens. This formula helps calculate the distance d, from the lens to the virtual image, in terms of focal length and distances, given by Eq. (3.5). Similarly, the distance between the virtual image and the eye can be calculated by adding up the distances from the lens to the virtual image d

with the eye relief distance, d^{eye} equation, given by Eq. (3.6).

$$d = d^{'} * \frac{f}{f - d^{'}} \tag{3.5}$$

$$d^{vi} = d + d^{eye} \tag{3.6}$$

$$\begin{aligned} h = h^{'} * M_l = h^{'} * \frac{f}{f - d^{'}} \\ w = w^{'} * M_l = w^{'} * \frac{f}{f - d^{'}} \end{aligned} \tag{3.7}$$

Eq. (3.7) gives the width and height (in units of length) of the virtual image, which is essentially that of the virtual image scaled by M_l. Eq. (3.8) instead confirms that the resolution of the virtual image in pixels is the same as that of the microdisplay.

$$\begin{aligned} h_p = h_p^{'} \\ w_p = w_p^{'} \end{aligned} \tag{3.8}$$

3.1.3.1 Foveation In The Virtual Image

The virtual image inside the HMD can be understood to be the 3D data being shown to a remote user in immersive remote visualization system. *Projecting* the foveated regions of the human eye on to the virtual image implies introducing concentric regions in it that correspond to the retinal fovea regions. These concentric regions would be centered on the center of the human eye gaze on the virtual image. That is, the regions would move around the virtual image as the eye gaze moves. Each region would have a specific radius and its associated visual acuity, i.e., degradation in quality. Fig. 3.4 shows an example of the foveal regions on the virtual image.

Since the intention is to project the retinal regions, to calculate the radius of each concentric region, the eccentricity values noted in Table 3.1 can be used. The radii r_n $\forall n \in \{0...N\}$, for each of the N retinal regions are calculated using the tangent of the corresponding eccentricity, scaled by the distance d^{Vi}, as noted in Equation (3.9).

$$r_n = \tan(e_n) * d^{Vi} \tag{3.9}$$

Conversely, to know which part on the virtual image lies in which concentric region, the angle subtended by the objects in the virtual image at the eye can be used. As seen in Fig. 3.4, the visual angle subtended by an object on the virtual image in the region r_1 at the eye is given by e_1 in degrees of arc. The projected length of the object is measured

from the center O which is on pixels (x_o, y_o) and point q at a position (x_q, y_q) (in pixels), and is calculated as,

$$\mathbf{d}(O,q) = \sqrt{\left(\frac{(x_q - x_o) * w}{w_p}\right)^2 + \left(\frac{(y_q - y_o) * h}{h_p}\right)^2} \tag{3.10}$$

The subtended angle can then be calculated using,

$$e_1 = \arctan\left(\frac{\mathbf{d}(O,q)}{d^{Vi}}\right) \tag{3.11}$$

3.2 Real-time 3D Data Acquisition and Mapping

The acquisition and reconstruction module acquires RGB-D images from the RGB-D cameras, e.g., Intel RealSense, ZED stereo camera, section 2.3. The mapping pipeline leverages the state-of-the-art real-time dense visual SLAM system, ElasticFusion [170], and adds functionalities, as explained in the following.

The map $\mathcal{M}$ is represented using an unordered list of surfels [124], where each surfel $\mathcal{M}^s$ has a position $\mathbf{p} \in \mathbb{R}^3$, a normal $\mathbf{n} \in \mathbb{R}^3$, a color $\mathbf{c} \in \mathbb{R}^3$, a weight $w \in \mathbb{R}$, a radius $r \in \mathbb{R}$, an initialization timestamp t_0, and a current timestamp t. The camera intrinsic matrix $\mathbf{K}$ is defined by: (i) the focal lengths f_x and f_y in the direction of the camera's $x-$ and $y-$ axes, (ii) a principal point in the image (c_x, c_y), and (iii) the radial and tangential distortion coefficients k_1, k_2 and p_1, p_2 respectively. The domain of the image space in the incoming RGB-D frame is defined as $\Omega \subset \mathbb{N}^2$, with the color image $\mathbf{C}$ having pixel color $\mathbf{c} : \Omega \rightarrow \mathbb{N}^3$, and the depth map $\mathbf{D}$ having pixel depth $d : \Omega \rightarrow \mathbb{R}$.

Given $\mathbf{K}$, the 3D back-projection of a pixel $\mathbf{u}_i = [x_i, y_i]^T \in \Omega$ for a given depth value $d(\mathbf{u}_i) \in \mathbf{D}$ is defined as $\mathbf{p}_i(\mathbf{u}_i, d(\mathbf{u}_i)) = \mathbf{K}^{-1}[\mathbf{u}_i, 1]^T d(\mathbf{u}_i)$. Over all pixels $\mathbf{u}_i$, this converts the RGB-D frame into a 3D map model. Further, the perspective projection of the 3D point $\mathbf{p}(x, y, z)$ is defined as $\mathbf{u} = \pi(\mathbf{K}\mathbf{p})$, where $\pi(\mathbf{p}) = [x/z, y/z]^T$ denotes the dehomogenization operation. The intensity value of the pixel $\mathbf{u} \in \Omega$ in the color image $\mathbf{C}$ with color $\mathbf{c}(\mathbf{u}) = [c_1, c_2, c_3]^T$ is defined as $I(\mathbf{u}, \mathbf{C}) = (c_1 + c_2 + c_3)/3$.

At each time step t, $\mathbf{C}_t$ and $\mathbf{D}_t$ are registered into the map model $\mathcal{M}$ by estimating the global pose of the camera $\mathbf{P}_t$, with rotation $\mathbf{R}_t \in \mathbb{SO}(3)$ and translation $\mathbf{t}_t \in \mathbb{SE}(3)$ with respect to the previous pose estimate $\mathbf{P}_{t-1}$. This registration provides the relative change from t to $t-1$.

$$\mathbf{P}_t = \begin{bmatrix} & \mathbf{R}_t & & \mathbf{t}_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{SE}(3) \tag{3.12}$$

The alignment between the current color $\mathbf{C}_t$ and depth map $\mathbf{D}_t$ with those of the active map model from the previous pose estimates, is achieved by minimizing a joint tracking

error E_{track}, composed of the geometric and photometric error functions E_{icp} and E_{rgb} respectively.

$$E_{track} = E_{icp} + w_{rgb} E_{rgb} \tag{3.13}$$

The weight w_{rgb} is empirically set to 0.1 reflecting the difference in units between the two error terms [169]. The geometric error function E_{icp} estimates the back-projection error from the current depth map $\mathbf{D}_t$ to the model depth map from $t-1$.

$$E_{icp} = \sum_i \left(\left(\mathbf{v^i} - \left(\exp(\hat{\xi}) \cdot \mathbf{T} \cdot \mathbf{v_t^i} \right) \right) \cdot \mathbf{n^i} \right)^2 \tag{3.14}$$

Here $\mathbf{v_t^i}$ is the back-projection of the i^{th} vertex in $\mathbf{D}_t$. $\mathbf{v}^i$ and $\mathbf{n}^i$ represent the corresponding vertex and normal in the model depth map from $t-1$. $\mathbf{T}$ is the current estimation of the transformation of the camera pose from $t-1$ to t, and $\exp(\xi)$ is the matrix exponential that maps a member of the Lie algebra $\mathfrak{se}(3)$ to a member of the corresponding Lie group $\mathbb{SE}(3)$ [170].

$$\begin{aligned} \mathbf{u}^a &= \pi \left(K \cdot \exp(\hat{\xi}) \cdot \mathbf{T} \cdot \mathbf{p}(\mathbf{u}, \mathbf{D}_t) \right) \\ E_{rgb} &= \sum_{\mathbf{u} \in \Omega} \left(I(\mathbf{u}, \mathbf{C}_t) - I(\mathbf{u}^a, \mathbf{C}_{t-1}^a) \right)^2 \end{aligned} \tag{3.15}$$

Similarly, the color from the current frame C_t and the map model color estimate C_{t-1}^a is used to find the photometric error E_{rgb} (intensity difference) between pixels. To minimize the function in Eq. 4.7, the Gauss-Newton non-linear least-squares method is used from [170]. The process continues iteratively to generate the 3D reconstructed scene in real-time. Figure 3.5 shows two sample 3D reconstructed scenes.

(i) (ii)

Figure 3.5: Real-time 3D reconstruction of a living room and office space, captured in real-time.

3.2.1 Map Partitioning and Sampling

For brevity, the symbol $\mathscr{M}$ is used interchangeably for both, the real-time point-cloud and the global surfel map. The density of $\mathscr{M}$, especially at high resolutions, implies increased computational complexity and more graphical and time resources for streaming it in

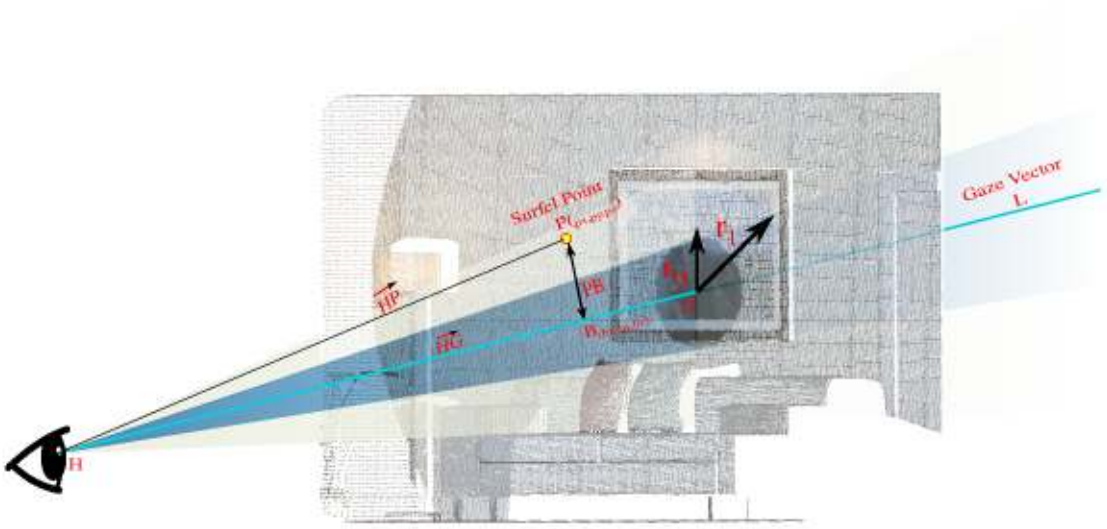


Figure 3.6: Visualization of map partitioning described in Section 3.2.1, This figure shows how the surfel point $\mathbf{P}(px, py, pz)$ is partitioned using the ray $\mathbf{L}$ casted from the point of origin $\mathbf{H}(hx, hy, hz)$.

immersive remote visualization system. The foveation model can be utilized here to reduce the data. By projecting the retinal fovea regions into it, $\mathscr{M}$ is partitioned into regions. It is then resampled to approximate the monotonically decreasing visual acuity in the foveation model, termed *foveated sampling*.

To partition $\mathscr{M}$ into regions, the discussion in sec. 3.1.3 is taken forward. The center of the eye gaze is used as a point of origin $\mathbf{H}(hx, hy, hz)$. To partition $\mathscr{M}$ into $\mathscr{M}_n$ regions $\forall n \in \{0...N\}$, for each of the N retinal regions, a ray is cast from $\mathbf{H}(hx, hy, hz)$. This ray, i.e., the *gaze vector* $\mathbf{L} \in \mathbb{R}^3$ is extended up to last point of intersection $\mathbf{G}(gx, gy, gz)$ with the surfel map. The foveation regions are now structured around $\mathbf{L}$. Section 3.1.3 describes the 2D foveation approach with concentric circles. With 3D data, the concentric regions are conical volumes, with their apex at $\mathbf{H}(hx, hy, hz)$ (Fig. 3.6), with increasing radii away from $\mathbf{H}$. Algorithm 1, which is implemented in CUDA in the GPU for faster processing, details how the radii are calculated based on d^{vi} for each surfel. To assign each surfel in $\mathscr{M}$ to a particular region $\mathscr{M}_n$, the shortest distance, i.e., the perpendicular distance between the surfel and $\mathbf{L}$ is used. As shown in Fig. 3.6, the shortest distance from the surfel $\mathbf{P}(px, py, pz)$ to the ray $\mathbf{L}$ is the perpendicular $\mathbf{PB} \perp \mathbf{L}$, where $\mathbf{B}(bx, by, bz)$ is a point on $\mathbf{L}$. $\|PB\|$ can be obtained using the the projection of $\vec{HP}$ on $\mathbf{L}$, i.e., the cross product of $\vec{HP}$ and $\vec{HG}$, normalized to the length of $\vec{HG}$, refer Eq. (3.16). Algorithm 1 assigns surfel $\mathbf{P}$ to the region $\mathscr{M}_n$.

$$\|PB\|_{\mathbf{P}} = \frac{\|\vec{HP} \times \vec{HG}\|}{\|\vec{HG}\|} \tag{3.16}$$

Algorithm 1: Map partitioning and sampling algorithm

```
Input: M                                          /* map to be partitioned  */
       L                                          /* Gaze direction vector  */
       e_0...n                                    /* eccentricity angles    */
foreach surfel P_k in the map M do
    B ← proj_L^{P_k}                              /* projection of point P_k on L */
    d^{Vi} ← ||HB||                               /* distance between H and B */
    d ← PB⊥L                                      /* shortest distance      */
    for i=1 to max(e) do
        r_i ← tan(e_i) * d^{Vi}                   /* compute The radii r_i  */
    end
    /* put P_k into the maps M_0 ...M_n                                      */
    if d < r_0 then
        M_0 ← P_k;
    else if d > r_0 AND d ≤ r_1 then
        M_1 ← P_k;
    ⋮
    else
        M_n ← P_k
    end
end
```

3.2.2 Foveated Sampling

The partitioned global map, with region-assigned surfels, is converted into a PCL point-cloud data structure, $\mathcal{P}_n$ for each $\mathcal{M}_n$ region $\forall n \in \{0...N\}$. To implement the foveated sampling, the $\mathbb{R}^3$ space of each $\mathcal{P}_n$ region needs to be further partitioned into an axis-aligned regular grid of cubes as shown in Figure 3.7. This process of re-partitioning the regions is called *voxelization* [134] and the discrete grid elements are called *voxels*. After voxelization, the down-sampling of the point cloud follows the foveation model, in that, the voxels in the fovea region of the point cloud are the most dense, and this density progressively reduces for voxels in the peripheral regions.

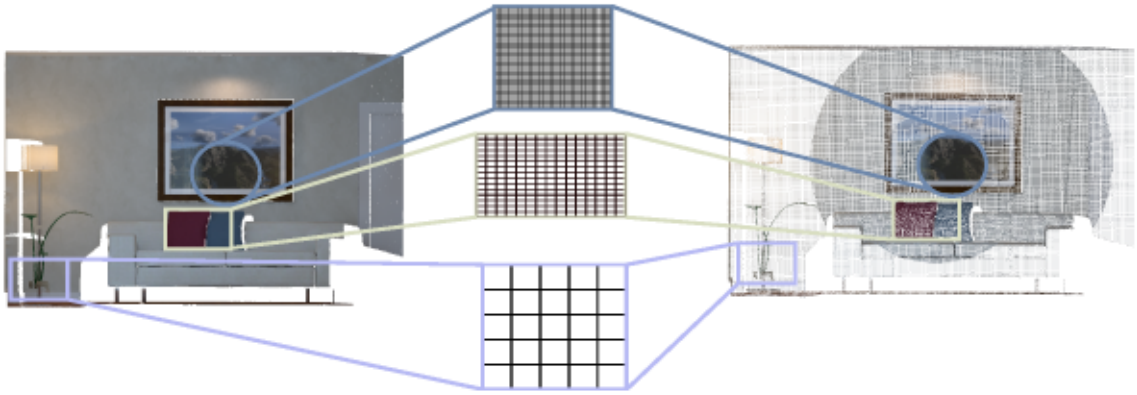


Figure 3.7: A 3D voxel grid defined by an edge length or voxel size v. The point-cloud inside each voxel is approximated by the centroid point of the point-cloud in that voxel.

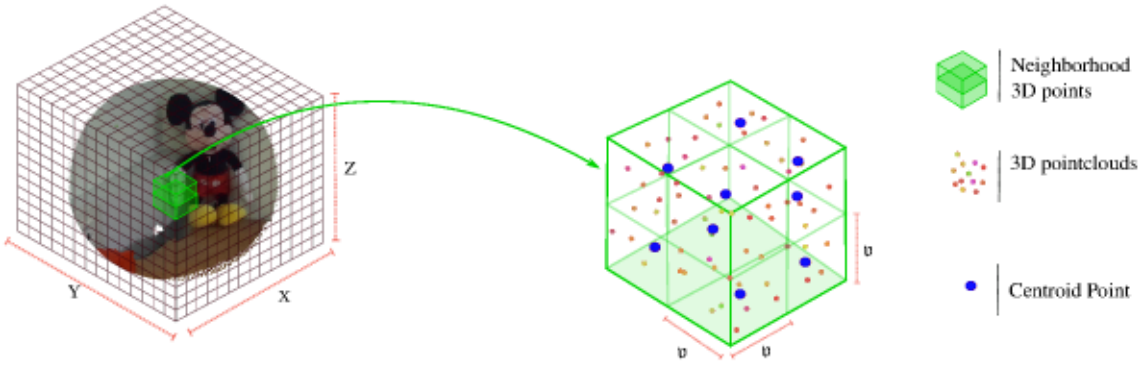


Figure 3.8: Foveated point cloud sampling. Three-layer voxel grids do the sampling process in this particular Figure. The foveated sampling process greatly reduces the number of points, latency, and overall graphics computation.

This voxelization and down-sampling is a three-step process:

1. Calculating the volume of the voxel grid in each region,

2. Calculating the voxel size, i.e., dimension, υ_n, for the voxelization in each region, and

3. Down-sampling the point-cloud inside each voxel for the region by approximating it with the 3D centroid point of the point-cloud.

Calculating the volume of the voxel grid for each region is done by simply calculating the point-cloud distribution for that region, $[(x_{n,min}, x_{n,max}), (y_{n,min}, y_{n,max}), (z_{n,min}, z_{n,max})]$. Calculating the voxel size, υ, is more involved. Here, the visual acuity discussion from sec. 3.1.2 is utilized.

Consider the voxelization of the central fovea region, $\mathcal{P}_0$. As noted earlier, the smallest angle a healthy human with a normal visual acuity of 20/20 can discern is 1 arcminute, i.e., 0.016667°. Following Eq. (3.2) therefore, MAR_0, which is the smallest resolvable angle, is 0.016667°. Using Eq. (3.9), the smallest resolvable object length on a virtual image can be calculated as,

$$\mathfrak{l} = d^{Vi} * \tan(MAR_0) \tag{3.17}$$

Eq. (4.10) itself could provide the optimum voxel size, υ. The important consideration here is the value of d^{Vi}. Since Fig. 3.4 is a 2D rendering, it shows a single d^{Vi} from the human eye to the virtual image. In a 3D scene such as the global map however, each point in the map would have an independent d^{Vi} value. Given that density of global maps can be in millions of points, using a d^{Vi} for each point would be impractical, also from

the point-of-view of voxelization. To overcome this issue, we take advantage of the partitioned point cloud and calculate one d^{Vi} value for the entire $\mathscr{P}_0$ region, approximated as the distance from the eye to the 3D centroid of point-cloud in the region, Eq. (4.11).

$$\mathfrak{pc}_0 = \frac{1}{N_{\mathscr{P}_0}} \left(\sum_{i=1}^{N_{\mathscr{P}_0}} x_i, \sum_{i=1}^{N_{\mathscr{P}_0}} y_i, \sum_{i=1}^{N_{\mathscr{P}_0}} z_i \right) \tag{3.18}$$

$$d_0^{Vi} = \mathbf{d}(eye, \mathfrak{pc}_0) \tag{3.19}$$

, where $N_{\mathscr{P}_0}$ is the number of points of the point cloud in the fovea region $\mathscr{P}_0$. Then, Eq. (4.10) is re-written as Eq. (4.13) to give the voxel size $\mathfrak{v}_0$ for the region.

$$\mathfrak{v}_0 = d_0^{Vi} * \tan(MAR_0) \tag{3.20}$$

Once the voxelization of region $\mathscr{P}_0$ is finalized, for the subsequent concentric regions from $\mathscr{P}_1$ to $\mathscr{P}_n$, the voxel sizes are correlated with the linear MAR relationship in Fig. 3.3. Using the voxel size for $\mathscr{P}_0$ as the base size, as the eccentricity angle of the regions increases, so do the voxel sizes, in proportion to the increasing MAR. This correlation is captured in Eq. (4.14), $\forall n \in \{1, ..., N\}$.

$$\begin{aligned} MAR_n &= m \cdot E_n + MAR_0 \\ \mathfrak{v}_n &= \frac{MAR_n}{MAR_{n-1}} * \mathfrak{v}_{n-1} \end{aligned} \tag{3.21}$$

The increasing voxel size away from the fovea region implies more and more points of the point-cloud in the corresponding regions are now accommodated within a single voxel. Therefore, when the down-sampling step is applied, the approximation of the point-cloud within a voxel is done over progressively dense voxels. For the down-sampling part, since the region $\mathscr{P}_0$ is the fovea region, which should have the highest visual acuity, it is left untouched, with the density of the point cloud in the region remaining the same as that determined by the incoming global map density, $\mathscr{M}_0$. The down-sampling in the subsequent peripheral regions is done by approximating the point-cloud within each voxel with its 3D centroid, using Eq. (4.15).

$$\mathfrak{pc}_n^v(x, y, z) = \frac{1}{N_{\mathscr{P}_n}^v} \left(\sum_{i=1}^{N_{\mathscr{P}_n}^v} x_i, \sum_{i=1}^{N_{\mathscr{P}_n}^v} y_i, \sum_{i=1}^{N_{\mathscr{P}_n}^v} z_i \right) \tag{3.22}$$

Here $N_{\mathscr{P}_n}^v$ is the number of points in voxel v of the region $\mathscr{P}_n$ ($\forall n \in \{1...N\}$). This process not only reduces the point-cloud data, but also maintains the shape characteristics of the point-cloud, allowing a more accurate approximation of the surface. Fig. 3.7 shows the sample voxel grids for the different regions, while Fig. 3.8 shows the centroid approximation of the point-cloud.

3.3 Immersive Remote Visualization Framework

The overall goal of this work is to design a remote visualization system using the human visual system with reduced latency and throughput requirements of 3D reconstructed scenes by maintaining a rich visual experience for the user. To that end, the proposed framework, termed as the *Foveated Rendering (FR)* framework, seen in Figure 3.1 and Figure 3.9, comprises a server-client architecture that encapsulates the foveation methodology detailed previously, and is divided into three major parts: the **user site**, the **remote site**, and a **packetization and communication network** between them. An RGB-D data acquisition module at the remote site passes the data to the real-time 3D reconstruction module implemented on a GPU. The control parameters module receives the eye-gaze pose data from the user site that is required to correctly project the foveation model into the point-cloud, and perform the foveated point-cloud sampling of the remote scene. The foveated point-cloud sampling and compression is the most computationally expensive system component, and the OpenMP multi-platform shared-memory parallel programming is used to achieve real-time performance. The foveated point-cloud is streamed to the user site and rendered in the VR HMD. The communication network allows real-time data exchange between the user site and the remote site, i.e., sending eye-gaze pose and direction, rendering the camera pose, along with the real-time foveated point-clouds. The schema 3.9 and 3.10 shows the detailed overview of the proposed FR framework and the main system compoments are described below.

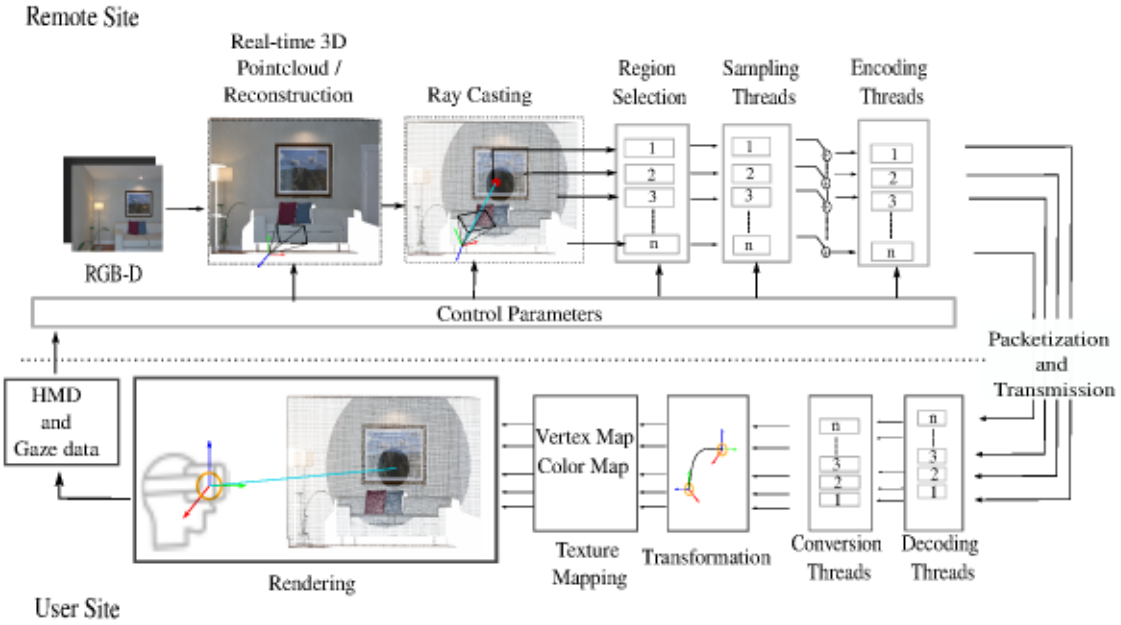


Figure 3.9: Schema shows an overview of the proposed *FR* framework. The server on the remote site performs the 3D data acquisition, scene reconstruction, foveated sampling, compression, and streaming of the point-cloud to the client. The client on the user site provides the eye-gaze tracking data, decodes the received point-cloud packets, and allows interaction.

3.3.1 User Site

The user site manages the: (1) decoding and rendering of the streamed point-cloud data, (2) tracking of the eye-gaze and HMD pose, and (3) real-time transfer of this information to the remote site.

To visualize and explore the incoming point-cloud, a VR-based interface is designed using the UE graphics engine on Windows 10. This creates the immersive remote visualization system environment for the user. As noted earlier, there is an interdependence between the user site and the remote site, in that, the eye-gaze data from the user site is required for the foveation model at the remote site. The foveated point-cloud is then streamed back to the user site to be rendered for visualization. Furthermore, the user site and the remote site are independent environments with their respective reference frames. It is therefore necessary to implement appropriate transformations among all the entities to ensure correct data exchange and conversion. As shown in Figure 3.10, the reference frames are as follows: UE world coordinate frame **U**, HMD coordinate frame **E**, and the gaze direction vector ${}^{E}\vec{D} \in R^3$ on **E**.

To calculate the correct gaze pose in **U**, transforming ${}^{E}\vec{D}$ from **E** $\rightarrow$ **U** is required, through the head pose ${}^{U}\mathbf{H}$ on **U**, as follows:

$$ {}^{U}\vec{D} = {}^{U}\mathbf{H} \cdot {}^{E}\vec{D} \tag{3.23} $$

This gaze direction ${}^{U}\vec{D}$, along with the head pose **H** are communicated to the remote site for further processing. Further, the received point-cloud from the remote site is visualized in UE and needs to be positioned based on the pose of the camera positioned at the remote site. At the remote site, the coordinate system of the camera pose, ${}^{O}\mathbf{P}$ is in OpenGL, **O**. The pose has to be transformed, using a change-of-basis matrix, into the UE coordinate system. UE uses a left-handed, z-up coordinate system, while the camera coordinates of OpenGL use a right-handed coordinate system, y-up. Eq. (3.24) provides the coordinate transformation formula, where **B** is the change-of-basis transformation matrix (see Appendix A).

$$ {}^{U}\mathbf{P} = \mathbf{B} \cdot {}^{O}\mathbf{P} \cdot \mathbf{B}^{-1} \tag{3.24} $$

At the user site, rendering the received real-time dynamic point-cloud data from the remote site requires a high speed large data transfer, as well as efficient and high quality visualization. To meet these requirements, the following modules were developed, as seen in Figure 3.9:

1. **Single / Parallel streamer:** The incoming point-cloud can be received as a single stream, combining all the foveated regions of the point-cloud, or as parallel separate streams of the regions. Single streams have the advantage of synchronized data, but can be very heavy in terms of bandwidth requirements. Parallel streams can help the network optimize the data transmission, reducing simultaneous bandwidth

requirement. However, parallel streams may also suffer from varying data transmission rates due to network delays and size differences. To address this issue, all streamed point-cloud regions are timestamped and a buffer resource module is created at the user site for software synchronization using the local clock synchronized with a central NTP server [1].

2. **A real-time point-cloud decoder:** that decompresses the data received at the user site. The decoding module includes the state-of-the-art point-cloud codec algorithm from [103] and uses the Boost ASIO over a TCP socket for data transfer.

3. **Conversion system:** Each decoded point-cloud region $\mathcal{P}_n(\forall n \in 1,...,N)$ has to be converted into a texture for visualization, where the reference frame of the received data has to be transformed into that of the user site, i.e., the UE graphics engine coordinate system.

4. **A rendering system:** The data should be transferred to the GPU and made accessible to the graphics engine shader for real-time rendering.

3.3.2 Remote Site

The remote site system consists of modules for acquisition, reconstruction, sampling, and streaming of 3D reconstructed maps, as shown on figure 3.9. The RGB-D data acquired will go through a software pipeline as described in detail in section 3.2 for real-time reconstruction. While real-time reconstruction is in progress, a parallel module will receive information about head pose ${}^{U}\mathbf{H}$ and the gaze direction ${}^{U}\vec{D}$ from the user site. The coordinate system of the head pose ${}^{U}\mathbf{H}$ and the gaze direction ${}^{U}\vec{D}$ has to be transformed from the UE frame, **U**, to the remote site OpenGL coordinate system, **O**, using a similar change-of-basis matrix, Eq. (3.25). Here: **Q** is the change-of-basis transformation matrix from UE to OpenGL. Figure 3.10 left, shows the reference frames of the remote site and they are described as follows: OpenGL world coordinate frame **O** and the camera frame **C**.

$$\begin{aligned} {}^{O}\mathbf{H} &= \mathbf{Q} \cdot {}^{U}\mathbf{H} \cdot \mathbf{Q}^{-1} \\ {}^{O}\vec{D} &= \mathbf{Q} \cdot {}^{U}\vec{D} \cdot \mathbf{Q}^{-1} \end{aligned} \tag{3.25}$$

As noted in section 3.2, in every frame the color image **C** and depth map **D** are registered into the map model $\mathcal{M}$ by estimating the global pose of the camera **P**. Therefore, the gaze direction vector ${}^{O}\vec{D}$ and the head pose ${}^{O}\mathbf{H}$ have to be transformed into the camera coordinate frame, in order to perform map partitioning and sampling.

$$\begin{aligned} {}^{C}\mathbf{H} &= \mathbf{P}^{-1} \cdot {}^{O}\mathbf{H} \\ {}^{C}\vec{D} &= \mathbf{P}^{-1} \cdot {}^{O}\vec{D} \end{aligned} \tag{3.26}$$

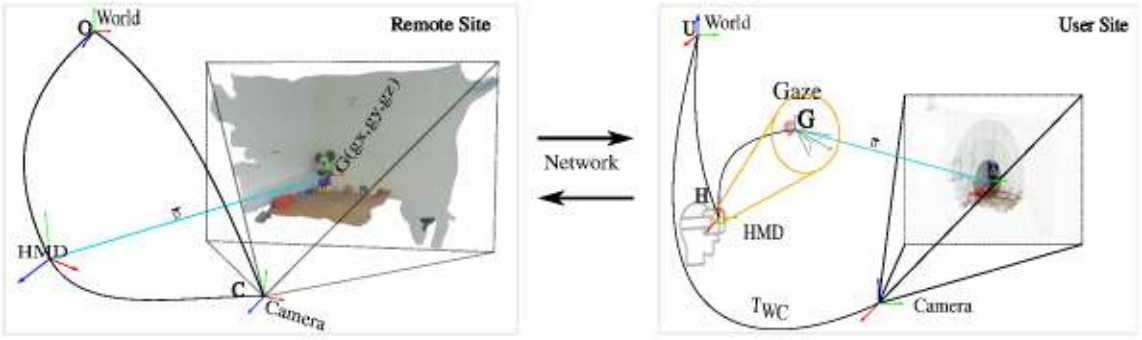


Figure 3.10: A schematic showing the coordinate system on the Remote and User sites.

The head pose in the camera frame is used as a point of gaze origin **H** (hx,hy,hz) and using the gaze direction vector, ${}^{C}\vec{D}$, a ray, i.e., the *gaze vector* **L** is projected in to the global surfel map.

3.3.3 Communication Network

Between the remote and user sites, a new point-cloud streaming pipeline was implemented using the Boost ASIO cross-platform C++ library for network and low-level I/O programming. This pipeline accounted for the throughput-intensive point-cloud data. The user site communicates with the remote site using TCP sockets. Further, the Robot Operating System (ROS) served as an additional connection to exchange lightweight data between the user VR interface and the remote site. This included the head pose ${}^{U}\mathbf{H}$, the gaze direction vector ${}^{U}\vec{D}$, and the pose of the remote camera ${}^{O}\mathbf{P}$, among other things. This part of the communication takes place through ROSbridge. [25].

3.4 Implementation

The FR framework provides a rendering *client* on the user site, while a streaming *server* provides the visualization data from the remote site. In order to implement the overall framework, the following software and hardware components for the data acquisition and visualization were used. Note that the FR framework could just as well be implemented on any other type of immersive remote visualization system hardware.

1. User site visualization hardware: The HTC Vive Pro Eye VR headset, which comes with gesture controllers and a tracking system, as well as a built-in Tobii Eye Tracking system. This HMD provides 2K resolution, with a 100° FOV and max. 90 Hz screen refresh rate. The Tobii system shows an accuracy of $0.5° - 1.1°$ with a trackable FOV of 110°. The PC had Windows 10 operating system with Nvidia GeForce GTX 1080 graphics card.

2. User site visualization software: The UE is used for rendering the visualization data. The immersive scene in UE renders the real-time point cloud feed and allows to

interact with the VR environment through the Tobii gaze tracker and the HMD gesture controllers.

3. Remote site acquisition hardware: The ZED stereoscopic camera and the Intel Realsense RGB-D camera were used for implementation. The remote site computing unit consisted of an MSI GE63 Raider laptop with Intel Core i7-8750H CPU @ 2.20 GHz, 12 cores and with an Nvidia GP104M Graphics card running Ubuntu Linux.

4. Communication hardware: A point-to-point direct Ethernet LAN connection was used between the server and the client, passing through the NightHawk Pro Gaming (SX10) 10 Gbit/s switch.

3.5 Experiment Design

The experiment design focused on the evaluation of the FR framework using online and acquired datasets, through defined experimental conditions and benchmarked against defined objective metrics.

3.5.1 Datasets

To allow a thorough evaluation of the framework implementation, the strategy was to use datasets that would help test it against benchmarks available in literature. For this, it was decided to use two known static world datasets from literature, available online. Further, to ensure that the complete FR framework pipeline is evaluated, two real world datasets were acquired using the remote site hardware described earlier and sample images are shown in Figure 3.11.

For the online datasets, the ICL-NUIM synthetic dataset was used, which provides benchmarking for RGB-D, Visual Odometry, and SLAM algorithms [51]. It consists of two different scenes, the living room (**LIV**) and the office room (**OFF**) provided with the ground truth. Being static worlds, these datasets are not enough to reflect the dynamics of a real world environment.

The two acquired real world datasets consisted of (i) a kitchen area (**KIT**) with arranged objects, e.g., microwave, utensils, glasses, etc., and (ii) a dynamic scene with a moving balloon (**BAL**), captured in a lab area (inspired by the TUM dynamic scene dataset [121]).

3.5.2 Experimental Conditions

The experimental analysis was performed to understand the impact of the FR framework on visual quality for the user as well as the changes in the computational and network performance of the system. Taking reference from the six foveation regions mentioned in Table 3.1, three test foveation conditions were created, each having a different combination of the regions projected into the point-clouds.

(a) Kitchen area (KIT)

(b) Balloon (BAL)

(c) Office room (OFF)

(d) Living room (LIV)

Figure 3.11: Sample frames from the 4 evaluation datasets.

- **F1:** The visual field of the point-cloud is divided into four regions based on the eccentricity angles, (i) Fovea (5°), (ii) Parafovea (8°), (iii) Perifovea (18°), and (iv) the rest of the point-cloud. The foveated sampling in the first three regions follows the strategy outlined in sec. 3.2.2 for progressive down-sampling. The 4^{th} region, i.e., the remaining point-cloud is sampled using the voxel sizes for the far peripheral region in Eq. (4.14). The hypothesis is that in this condition, the reduction in visual quality would be evident, but it would also offer the highest computational / network performance gain.

- **F2:** The visual field is divided into five regions - the Fovea, Parafovea, and Perifovea (as above), then the *near peripheral* region (up to 30°), and then the rest of the point-cloud. Here again, the foveated sampling strategy is as in the **F1** condition, with the addition of the near peripheral region before the rest of the point-cloud. This mapping is chosen as the middle point, an expected intuitive balance between the visual quality degradation and the performance gain.

- **F3:** The visual filed is divided into six regions - the Fovea, Parafovea, Perifovea, and near peripheral (as above), then the mid peripheral region (60°), and then the rest of the point-cloud in the far peripheral region. With this mapping, the visual quality reduction is expected to be the least likely to be detected, but it would offer the least computational / network performance gain, that could still sufficiently justify the use of the FR framework.

In addition to the three conditions above, two reference conditions are created to represent the two ends of the scale, no sampling to full sampling, to allow comparison

with the FR framework.

- **F0:** The point-cloud in the visual field is not divided into regions. The sampling strategy is applied across the whole point-cloud, using the voxel size of the far peripheral region in Eq. (4.14). This condition simulates the approach of uniformly down-sampling a point-cloud before streaming it to a remote user, to save bandwidth.
- **FREF:** The visual field is left untouched and the FR framework is not applied. On this condition there is no visual field division and foveated sampling, it keeps the acquired point-cloud as is and streams it to the user site.

3.5.3 Evaluation Metrics

The evaluation metrics utilized for the experiments help analyze the performance of the FR framework in terms of the benefits it provides as well as the costs it imposes, when implemented as part of an immersive remote visualization system system. The evaluation is therefore carried out both through a quantitative (objective) and subjective assessment. The quantitative metrics help objectively evaluate the FR framework in terms of:

1. The amount of data that can be reduced while streaming 3D reconstruction / point-cloud data from remote site.
2. The improvement, or otherwise, in the data transfer rate in streaming.
3. The improvement, or otherwise, in the end-to-end latency.
4. The effect of the data reduction on the quality of the visualization for the user.

The subjective metrics help understand the user experience for having the FR framework in an immersive remote visualization system setup at the user site.

3.5.3.1 Data reduction

The FR framework provides a method of reducing the number of points in a point-cloud, i.e., the overall density, while maintaining the highest density in the center of the gaze fixation. To evaluate this density reduction, a density estimation method proposed by [45] is used. The volumetric density of a point-cloud is computed by counting the number of neighbors $N^v_{\mathscr{P}}$ for each point $\mathfrak{p}$ in the point-cloud $\mathscr{P}$ that lie inside a spherical volume v, as seen in Eq. 3.27. $N_{\mathscr{P}}$ is the total number of points in point-cloud $\mathscr{P}$, and the spherical volume is based on a radius R, whose value is by averaging the leaf size across foveated regions.

Figure 3.12 illustrates a colored pointcloud and Figure 3.13 shows the concept of the volumetric density for a sample point-cloud in color scale. To analyse the reduction offered by the foveated conditions against the reference conditions, a density difference

$\mathbf{vd}^{df}$ in Eq. (3.28) is computed by subtracting the densities of the conditions under consideration. $\mathbf{vd}^{df}$ will show how much the density in particular conditions is changing due to the application of the FR framework.

$$\mathbf{vd}_{\mathfrak{p}} = \frac{N^{v}_{\mathscr{P}}}{\frac{4}{3} \cdot \pi \cdot R^3} \quad ...\forall \mathfrak{p} \in \mathscr{P} \tag{3.27}$$

$$\mathbf{vd}^{df}_{\mathfrak{p}} = \mathbf{vd}^{ref}_{\mathfrak{p}} - \mathbf{vd}^{test}_{\mathfrak{p}} \tag{3.28}$$

3.5.3.2 Data Transfer Rate

Any reduction in the amount of data for the point-cloud streaming would improve the overall network data transfer rate between the user and remote sites. To measure this rate,the network data packet analysis tool, Wireshark [136], was used.

Figure 3.12: Reference colored point cloud.

3.5.3.3 Latency

With the reduction in the data and the improvement in the data transfer rate, the FR framework can potentially reduce the end-to-end latency for the data transfer, where the acquisition and rendering are in physically separated environments. To understand the contributing factors in determining the proposed system's overall latency, it is decompose it into sub-modules.

As briefly described in section 3.3, the FR framework is composed of modules, e.g., acquisition, mapping, conversion, etc., both on the user and the remote sites. The latency analysis was done by measuring the latency for each of the modules: (1) on the remote site - from reading RGB-D images (log-read), ray-casting, converting the global map into PCL data structures (conversion), to sampling; and (2) on the user site, the decoding,

Figure 3.13: Volume density estimated (VD_{ref}) - in color scale.Point Density estimated on the reference point-cloud using CloudCompare; r refers to the sphere radius used for density estimation.

conversion, and rendering system modules. There are some hardware and software parts of the framework, which have their latencies specified: (1) the HTC Vive Pro Eye has a minimum declared eye tracker latency of around $8ms(120Hz)$; (2) The use of the ROS-bridge network to communicate this to the remote site, implies publishing it through the ROS gaze pose publisher at $10ms(100Hz)$.

3.5.3.4 Objective Quality Assessment

The FR framework facilitates reduction of data of the 3D reconstruction point-cloud. However, this can result in degradation of the visual quality when the point-cloud is rendered to the user. To understand this degradation, we propose to use the Peak Signal-to-Noise Ratio (Peak Signal-to-Noise Ratio (PSNR)) metric, drawing inspiration from the video compression research community.

The effect of the foveated sampling is to change the geometry, i.e., to distort the original point-cloud. To assess this, the *Point-to-Point* PSNR-based geometry quality metrics is frequently used as a measure of the distortion introduced by MPEG compression [2]. In the case of point-clouds, this metric is calculated as follows: (1) First the point-to-point symmetric root mean square (rms) distance is calculated between a pair of point-clouds, e.g., between point-clouds in the FREF and F1 conditions, using the shortest distance calculation[1]. The shortest distance calculation is done by measuring the shortest distance between every point in one point-cloud and its nearest corresponding point in the second point-cloud; (2) The PSNR is defined as a ratio of the diagonal distance of a bounding box

[1] "symmetric" implies the rms value is calculated in both directions, i.e., from FREF to F1 and vice versa.

of the overall point-cloud over the symmetric rms distance value as shown in Eq. (3.29).

$$d_{rms}(\mathcal{P}_1,\mathcal{P}_2) = \frac{1}{\sqrt{N_{\mathcal{P}_1}}}\sum_{i=1}^{N_{\mathcal{P}_1}} \|\mathfrak{p}^i_{\mathcal{P}_1} - \mathfrak{p}^i_{\mathcal{P}_2}\|_2$$

$$d_{sym}(\mathcal{P}_1,\mathcal{P}_2) = max(d_{rms}(\mathcal{P}_1,\mathcal{P}_2), d_{rms}(\mathcal{P}_2,\mathcal{P}_1)) \tag{3.29}$$

$$PSNR_{pp} = 10\cdot log_{10}\frac{\|max_{x,y,z}(\mathcal{P}_1)\|_2^2}{\Big(d_{sym}(\mathcal{P}_1,\mathcal{P}_2)\Big)^2}$$

$N_{\mathcal{P}_1}$ is the number of points in point-cloud $\mathcal{P}_1$. $\mathfrak{p}_{\mathcal{P}_1}$ represents a point in the point-cloud $\mathcal{P}_1$, while $\mathfrak{p}_{\mathcal{P}_2}$ is its closest corresponding point in $\mathcal{P}_2$.

In applications related to 3D reconstruction / point-clouds, the Point-to-Point metric can be sensitive to size differences and noise when calculating the peak signal estimation. For instance, the peak signal estimation for large and small size point-clouds, even when applied with the same amount of distortion, would be different, and it will produce a higher PSNR value for the large point-cloud and small value for the smaller size point cloud. This should not be the case since each point-cloud has the same distortions. In order to avoid this size and noise sensitivity, another PSNR metric, based on volumetric density, is proposed here. This metric utilized two different volumetric density values for the point-clouds under consideration: (1) its general volumetric density, as given by Eq. (3.27). Here too, a symmetric density difference is calculated for the point-clouds; and (2) its maximum volumetric density as the peak signal.

For the symmetric density difference calculation, for every point $\mathfrak{p}$ in the point-cloud $\mathcal{P}_1$, the closest corresponding point $\mathfrak{p}_{nn} \in \mathcal{P}_2$ is found. The vector $\mathbf{vd}_{\mathcal{P}_1}$ and $\mathbf{vd}_{\mathcal{P}_2}$ are then estimated using Eq. (3.27). For the peak signal, the volumetric density is calculated with the k-nearest neighbor approach, as seen in Eq. (3.30), to account for the distribution of the density across the point-cloud, and avoid any skew in the values due to sensor noise.

$$\begin{aligned}\mathbf{vd}_{\mathfrak{p}\in\mathcal{P}_1} &= \frac{1}{k}\sum_{i=1}^{k}\frac{N^v_{\mathcal{P}_1}}{\frac{4}{3}\cdot\pi\cdot R^3},\\ \mathbf{vd}^{max}_{\mathcal{P}_1} &= \max_{\forall\,\mathfrak{p}\in\mathcal{P}_1}\Big(\mathbf{vd}_{\mathfrak{p}}\Big)\end{aligned} \tag{3.30}$$

The value of k=10 was found experimentally, The density-based PSNR is calculated similar to the point-to-point metric discussed earlier, as a ratio of the maximum density of a reference point-cloud to the symmetric rms error in the general densities. Eq. (3.31)

provides the equations to be used.

$$\mathbf{vd}^{rms}(\mathscr{P}_1,\mathscr{P}_2) = \sqrt{\frac{1}{N_{\mathscr{P}_1}}\sum_{i=1}^{N_{\mathscr{P}_1}}\left[\mathbf{vd}^{i}_{\mathscr{P}_1} - \mathbf{vd}^{i}_{\mathscr{P}_2}\right]^2}$$

$$\mathbf{vd}^{sym}(\mathscr{P}_1,\mathscr{P}_2) = max\left(\mathbf{vd}^{rms}(\mathscr{P}_1,\mathscr{P}_2)\right), \mathbf{vd}^{rms}(\mathscr{P}_2,\mathscr{P}_1)) \tag{3.31}$$

$$PSNR_{\text{vd}} = 10 \cdot log_{10}\frac{(\mathbf{vd}^{max}_{\mathscr{P}_1})^2}{(\mathbf{vd}^{sym}(\mathscr{P}_1,\mathscr{P}_2))^2}$$

3.5.3.5 Subjective Quality Assessment

A user study is conducted to assess the user experience of the FR framework, especially from the point-of-view of immersive remote visualization system systems, in situations where there is a limited networking resources. The following two research questions were considered in the study (adapted from the research work [166]):

***Research Question 1**: Can subjects differentiate between scenes with varying graphical contexts, streamed with and without the proposed system?*

***Research Question 2**: How do different combinations of the foveated regions impact subjective quality perception?*

This questions are answered through the user study as discussed here. Although literature is replete with several methodologies for subjective quality assessment for images and videos, the same cannot be said for point-clouds. The most relevant methods were found in [111, 62] and the ITU-T P.919 [68] for 3D videos and graphics and FR. Taking inspiration from these, the Double Stimulus Impairment Scale (DSIS) was adopted for this user study [68]. The method involves conducting the user trials in short sessions, where the subjects are presented with the unimpaired condition, immediately followed by the impaired (altered) condition of the stimulus for a comparative judgement. In this case, the stimulus is the point-cloud (foveated vs. non-foveated). The subjects are then asked to rate the alteration in the second presented stimulus using the following 5-point scale [68], on whether it is:

- (5) imperceptible
- (4) perceptible, but not annoying
- (3) slightly annoying
- (2) annoying

- (1) very annoying

The hypothesis for the study is that "it is easy to tell that a scene is visually represented in a foveated way." The expectation is that hypothesis is *nullified*, i.e., subjects are NOT able to distinguish easily between the foveated and non-foveated representations. The study is conducted over multiple sessions, with at least a five-minute break in between to avoid fatigue. For initial training, subjects are allowed to familiarize themselves with the VR HMD and the use of the gesture controller device for interaction. For the trial itself, in each session, subjects are first presented with the **FREF** condition of a point-cloud, followed by a 3-second pause, with one of the altered conditions (**F0** - **F3**) following immediately after. The sequence among **F0** - **F3** is randomized across subjects. The subjects are asked to rate their assessment of the impairment on the 5-point scale within each session itself. The arithmetic mean opinion score (MOS) is calculated for each condition, averaged over all subjects.

3.5.3.6 Subjective Visual Search Assessment

As briefly described in section 2.2.5, The visual world is overwhelmingly rich — it contains far too much information to perceive at once, and the visual system's processing capacity is limited by the high metabolic cost of cortical computations. Given these limits, the human visual system needs mechanisms to optimally allocate processing resources according to task demands. Typically, It scan the scene to aim the fovea at a place (salient regions) that It want to process more deeply and shift to another item. Several studies has shown that targets presented near fixation point (fovea) are, in fact, found more efficiently than are targets presented at more peripheral locations. However when targets presented away from fovea accuracy reduces and increase search times and number of eye movements [174, 38]. It takes longer to find a target in the periphery because fewer cortical neurons are devoted to the analysis of peripheral visual information (cortical magnification) Section 2.2. The human visual system have evolved gaze-shifting mechanisms to overcome these limitations [95]. In addition, eye movements are also guided by information about the target, including basic features color, size, orientation, and shape. Adding distractors that share the same features with the target element affects the performance: leading to significant increases in response times during the search [38, 36, 115, 54].

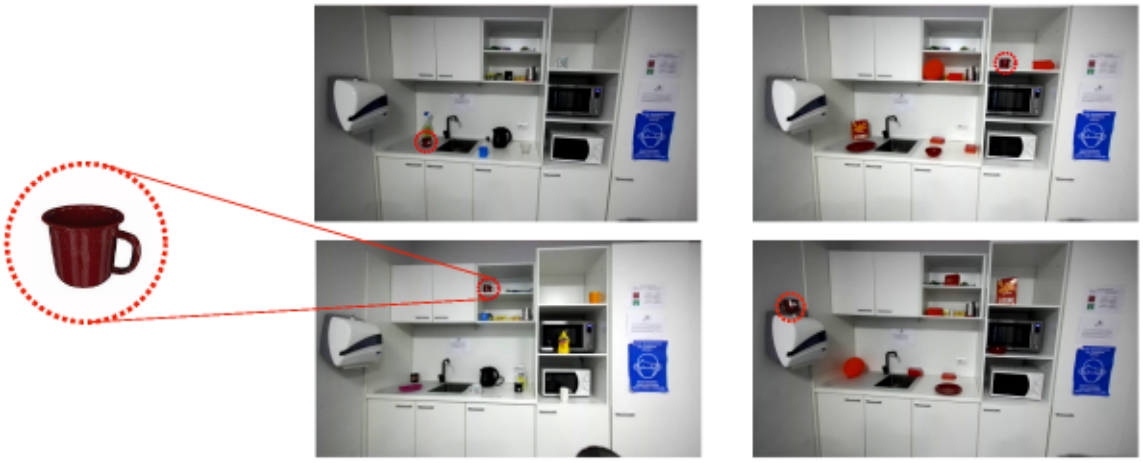
In applications such as search and rescue using remotely controlled robots, It is very important to act fast when possible survivors (targets) might be trapped amongst the debris. However, the environment could be destroyed and unstructured, which adds additional distractors to the target. Target locations should be assessed with time and bandwidth constraints in mind. Several methodologies has been proposed to measure the performance of visual search. Taking inspiration from a research work [115], A user study is conducted to assess attentional and distracter effects while using the FR framework. Especially from the point-of-view of immersive remote visualiation systems,

in situations where there is a limited networking. The experimental conditions from 3.5.2 are used by varying the target–distractor discriminability. The design of the experiment followed the typical flanker paradigm: a way of studying the cognitive processes involved in detecting and recognizing targets in the presence of distracting information [115]. As shown in figure 3.14, In one set of trial conditions, targets and distractors were shown in a different colour, thereby easy to discriminate **(high discriminability)** from each other and In another set of trial conditions, both targets and distractors had similar color, and thus were difficult to discriminate **(low discriminability)**: In total the experiment contain 10 conditions (**F0**, **F1**, **F2**, **F3**, **FREF** defined in section 3.5.2 for each high and low discriminability conditions). The stimulus set consists 10 objects: a fixation stimulus(water tap), a target Object (mug Fig. 3.14 a) and 8 distracting objects, there location and type was randomized to avoid learning effects and The conditions were randomized.

Subjects were presented with the trial conditions and asked to press a button to measure the time it takes to detect the target amongst its distractors (RT). Then, the subjects were first asked to estimate the distance between the target and one of the distracting objects randomly selected. Secondly, They were asked with the following statments to specify their level of agreement to questionaries with five points: (1) Strongly disagree; (2) Disagree; (3) Neither agree nor disagree; (4) Agree; (5) Strongly agree.

Statement 1: I had to focus my attention strongly to accomplish the task in this trial.

Statement 2: I succeeded in performing the task to my satisfaction.



(a) Target (b) High Discriminability (c) Low Discriminability

Figure 3.14: Sample frames from the Kitchen area (KIT) dataset: (a) Shows the Target, (b) The target and distractors were shown in a different colour therby having **high discriminability** and (c) shows the target and distractors have similar colour and it is difficult to discriminate **low discriminability**.

The following research question was considered in the study.

***Research Question**: How accurately an average subject would be able to find a target visual stimulus amongst other visual stimuli (distracters) when presented in different foveation conditions ?*

The **hypothesis** for this study is that "Search performance and reaction time (RT) metrics are improved when a target is presented with high discriminability and foveated way (as against non-foveated way). (Expectation: hypothesis is sustained, i.e., foveated representation improves the search accuracy and reduces latency when searching a target.)

3.5.3.7 Subjective Tracking Performance Assessment

The FR framework provides a method of reducing the number of points in a point cloud,i.e., the overall density, while maintaining the highest density in the center of the gaze fixation, and the point density on the peripheral field of view is low. This study aims to assess the effect of peripheral quality loss on improving accuracy and reducing latency when tracking moving objects.

Subjects were asked to follow (track) a moving balloon; the balloon trajectory and eye-gaze trajectory in 3D are used to calculate the trajectory error (RMSE). The errors are calculated by measuring the closest distance to the target trajectory. Figure 3.16 shows the error calculation for the i^{th} measurement point, where A_m is the gaze trajectory and T_k is the ground truth ballon trajectory. Maximum error is the highest value of err, and root-mean-square-error (RMSE) as shown in Figure 3.16 where N is the total number of measurement points.

The research question defined for this experiment is the following:

***Research Question**: How accurately subjects would be able to track/follow a dynamically moving object with and without the proposed system ?*

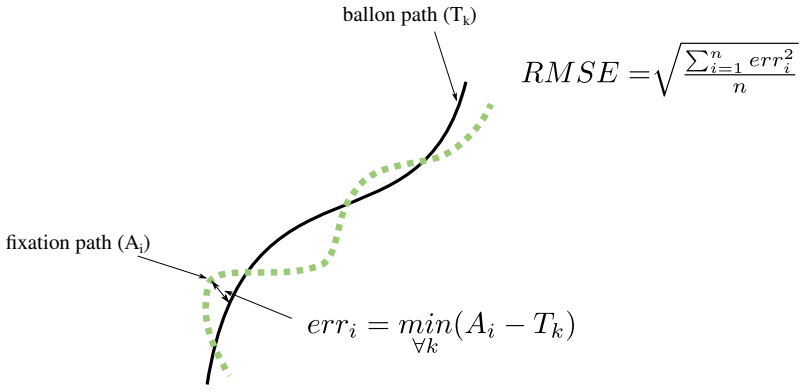


Figure 3.15: Evaluating the RMSE based on the target trajectories.

The **hypothesis** for this study is that "Accuracy and latency metrics are improved when a scene with moving objects is visually represented in a foveated way (as against non-foveated way). (Expectation: hypothesis is sustained, i.e., foveated representation improves the accuracy and reduces latency when tracking moving objects).

Figure 3.16: Sample frames from the Balloon (BAL) dataset: Top image is a pink balloon thrown from right to left and bottom image is a yellow ballon thrown from Left to right.

3.5.3.8 Participants

For all subjective studies 3.5.3.6, 3.5.3.5, 3.5.3.7, Twenty-five subjects (10 females and 15 males) participated in the subjective trials, aged 21 to 35 years. All subjects had a 20/20 or corrected vision, and the Eye tracker was calibrated for all subjects. Based on the ITU-T [68] recommendation, subjects were trained with the dataset **OFF** to get familiar with the test methodology and The dataset **LIV** was used for the experimental trial.

3.6 Results and Analysis

Following a recommendation by [14], 5 randomized HMD positions with varying distances to the center of the datasets were used for the objective metrics evaluation. For the raw point-cloud streaming, four hundred frames were tested for each HMD position from each dataset. The 3D reconstruction analysis was done on the **OFF** and **LIV** datasets, with the map allowed to grow up to 200 frames (for Latency and PSNR) and 100 seconds (for Bandwidth). Here too, the data was averaged over 5 randomized HMD positions.

3.6.1 Data Transfer Rate

Table 3.5 reports the average bandwidth required for raw point-cloud streaming and the relative percentage reduction in bandwidth as compared to the FREF condition. The mean bandwidth required for the F1 condition gives an average 61% reduction as compared with FREF. The numbers are similar for the F2 condition, an average 61% reduction, and F3 offers a lower 56% reduction. Statistical t-test analysis Table 3.2 showed that these reductions are significant at 95% CI (p-values $<<$ 0.05). Within the 3 conditions, although F1 is the most advantageous, the difference between the 3 reductions is not statistically significant (p-value = 0.3). On the other hand, as expected, the foveation conditions perform worse than the F0 condition, which offers the highest bandwidth reduction, up to 81%.

For the 3D reconstruction map streaming instead, the Fig. 3.17 shows how the bandwidth numbers increase over time, as the map grows, for all the conditions. Table 3.5 reports the average bandwidth values for streaming the **OFF** and **LIV** datasets in each

condition and the relative percentage reductions in the values as compared to the FREF condition. The foveation conditions offer reductions in the range of, on average: F0 = 75%, F1 = 82%, F2 = 77.9%, and F3 = 68.5%, against FREF, all statistically significant (p-values<<0.05): Table 3.3 shows more detail. Fig. 3.17 shows how the bandwidth numbers increase over time, as the map grows, for all the conditions for one sample trial. The bandwidth increase over time is much slower for the foveation conditions. Estimating the slopes of the trends through linear data fitting provides the following values: F0 = 0.0021, F1 = 0.0023, F2 = 0.0026, F3 = 0.0030, and FREF = 0.0247. The foveation conditions rise almost 10 times less steeply than the FREF condition.

Table 3.2: Independent-samples t-test data transfer rate (BW)- raw Point-cloud.

		mean				
Parameter - P_1	Parameter - P_2	P_1	P_2	t	p	Signif: Y/N
FREF	F0	1.16	0.46	12.14	0.00	Y
FREF	F1	1.16	0.79	5.56	0.00	Y
FREF	F2	1.16	0.87	4.27	0.00	Y
FREF	F3	1.16	0.93	3.46	0.00	Y
F0	F1	0.46	0.79	-5.14	0.00	Y
F0	F2	0.46	0.87	-6.16	0.00	Y
F0	F3	0.46	0.93	-7.60	0.00	Y
F1	F2	0.79	0.87	-1.04	0.30	N
F1	F3	0.79	0.93	-2.04	0.04	Y
F2	F3	0.87	0.93	-0.92	0.36	N

Table 3.3: Independent-samples t-test for BW - Global Map

		mean				
Parameter - P_1	Parameter - P_2	P_1	P_2	%age reducn	p	Signif: Y/N
FREF	F0	1.606	0.183	87.261	0.00	Y
FREF	F1	1.606	0.303	78.069	0.00	Y
FREF	F2	1.606	0.371	72.894	0.00	Y
FREF	F3	1.606	0.398	71.472	0.00	Y
		using %age reduction				
F0	F1	87.261	78.069	9.192	0.00	Y
F0	F2	87.261	72.894	14.366	0.00	Y
F0	F3	87.261	71.472	15.789	0.00	Y
F1	F2	78.069	72.894	5.175	0.00	Y
F1	F3	78.069	71.472	6.598	0.00	Y
F2	F3	72.894	71.472	1.423	0.42	N

3.6.2 Data Reduction

To compare the data reduction of the proposed system, the spacing between points (density) estimated for each condition and then the difference between the reference and test conditions is estimated, the density difference is summarized in the Table 3.7. On average,

Table 3.4: Compressed bandwidth (MBytes/sec; top row) and latency (ms; bottom row) for real-time raw point-cloud streaming

	KIT	BAL	OFF	LIV	Reduction(%)
					20 40 60 80 100
F0	0.52	0.78	0.49	0.25	
	187	198	196	190	
F1	0.91	0.80	1.02	0.50	
	226	226	224	223	
F2	0.95	0.97	1.03	0.55	
	229	235	240	242	
F3	1.01	1.03	1.22	0.74	
	253	256	256	257	
FREF	1.67	1.82	1.78	1.32	
	557	562	622	618	

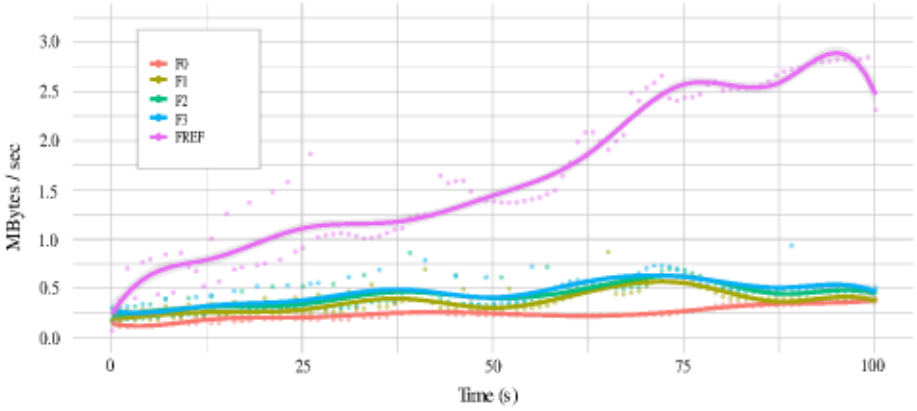


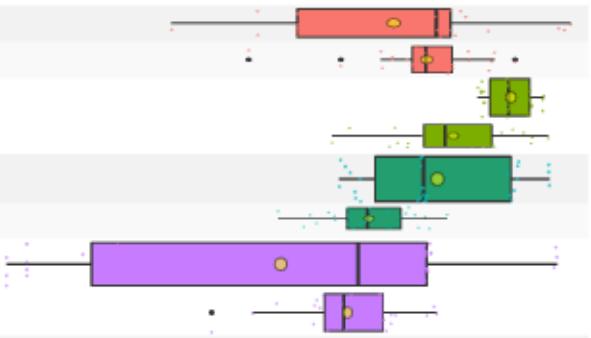
Figure 3.17: Compressed bandwidth for 3D reconstructed map.

the proposed system achieved density reduction between 30% and 65 % using the test condition (F3) and using the test condition (F2), a density reduction between 50% and 88% is achieved. The highest density reduction is achieved using the proposed system is the test condition **F1**, which is between 76% and 90%. The volumetric density for 3D reconstruction is summarized in the Table 3.7. On average, The proposed system achieved density reduction between 68% and 73% using the test condition F3 and using the test condition F2, a density reduction between 70% and 72% is achieved. The highest density reduction is achieved using the proposed system is the test condition F1, which is between 72% and 73% for **OFF** and **LIV** data sets respectively.

The density difference results for each test and reference conditions on the dataset **LIV** is shown in Figure 3.18. The highest density difference can be seen around the peripheral

Table 3.5: Compressed Bandwidth (MBytes/sec; top row) and Latency (ms; bottom row) for real-time streaming of 3D reconstruction.

	OFF	LIV	Reduction(%)
			60 70 80
F0	0.61	0.35	
	290	294	
F1	0.33	0.30	
	286	309	
F2	0.39	0.37	
	351	359	
F3	0.88	0.39	
	360	379	
FREF	2.09	1.41	
	1376	1429	



regions. In comparison, approximately zero density differences are localized around the fixation point of the data sets. The mean and standard deviation statistic difference between the reference (**FREF**) and the test conditions on a radius of r = 0.019809 is also listed on Table 3.7.

Table 3.6: Mean density difference between the reference (**FREF**) and the test conditions on a radius of r = 0.019809.

	KIT	BAL	OFF	LIV
F0	53871	53975	170815	37146
F1	114239	114343	291458	97514
F2	175892	175691	375668	102121
F3	421338	347388	394238	363250
FREF	1187134	488030	731807	904479

3.6.3 Latency Reduction

The mean latency values in percentages for the real-time point-cloud streaming are illustrated in Table. 3.5 and Table 3.19. As is seen, the foveation conditions offer between 60% (F3) and 67% (F1) speedup over the FREF condition. However, they also perform worse than the F0 condition, between 20% and 35% slower. The speedups (or slowdowns) are statistically significant, p-values << 0.05: more detail statistical analysis are shown on table 3.10.

Likewise, with the global map streaming, seen in Fig. 3.20, The latency numbers from Table 3.5 show that the foveation conditions offer between 73% (F3), 75% (F2), and 79% (F1) speedup over the FREF condition, all statistically significant, p-values<<0.05.

Table 3.7: Mean volume density for all the conditions on r = 0.019809 on 3D reconstructed datasets.

		OFF	LIV
F0	Mean	16933	17650
	Std	5726	6902
F1	Mean	82157	152605
	Std	155063	226672
F2	Mean	92796	157122
	Std	156011	174596
F3	Mean	95393	159524
	Std	138125	209928
FREF	Mean	494293	575867
	Std	298683	261057

Table 3.8: Mean number of points per frame(N_{pt}) for each experimental conditions

	KIT	BAL	OFF	LIV	Reduction(%)
F0	70094	38941	17519	19840	
F1	59083	44416	50463	49132	
F2	69788	65032	53899	68618	
F3	70900	66307	55664	72775	
FREF	252672	252672	307200	307200	

A more detailed system component level evaluation is seen in Table 3.9. The most time-consuming elements are related to data conversion, encoding and decoding. As expected, the numbers show an upward trend from F0 to FREF. However, it is noted that this trend in not linear - latency increases at a greater rate with increasing resolution.

Figure 3.20 shows that the speedups get more significant as the map grows. This shows the added advantage of FR, where the latency increase is much slower, reducing the negative impact on presence in *IRV*. The slope trends for the latency values are similar to those for the bandwidth: F0 = 1.5979, F1 = 1.6332, F2 = 1.9617, F3 = 2.1280, and FREF = 8.4582. The foveation conditions rise around 4.5 times less steeply than the FREF condition. Fig. 3.22a, 3.22b, 3.21a and 3.21b shows detail latency graph for each conditions.

3.6.4 PSNR Metric

Figure 3.23 illustrates the volumetric density based PSNR metric, that helps objectively discriminate among the test conditions in terms of the costs they impose on the visual quality. In all cases, the F0 PSNR is significantly worse (p-value << 0.05), which negates the bandwidth and latency advantages it offers. For the raw point-cloud, the foveation

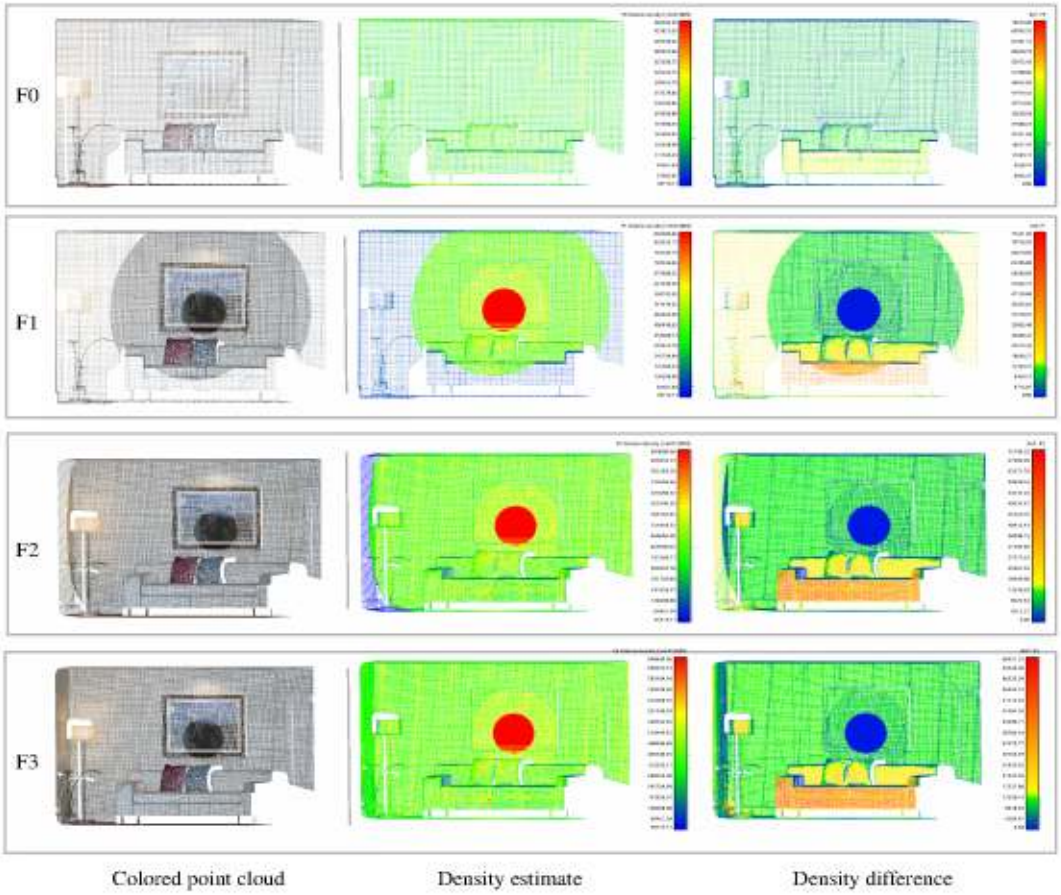


Figure 3.18: The density difference analysis between the reference and test conditions - (a), (d), (g), (j) shows the foveated point cloud for each condition **F0, F1, F2, F3, FREF** respectively - (b), (e), (h), (k) show the density estimate for each test conditions in color scale mode, and (c), (f), (i), and (l) shows the density difference between the reference and the test conditions.

conditions offer progressively better PSNR values, averaging 69.5 dB (F1), 70 dB (F2), and 71.6 dB (F3), over the 4 datasets. The F3 PSNR is significantly better than F1 (p-value << 0.05), but not significantly better than F2 (p-value = 0.64) : Table 3.12 shows more.

For the 3D reconstruction, Figure 3.24 illustrates the VD based PSNR metric, that helps objectively discriminate among the test conditions in terms of the costs they impose on the visual quality. The PSNR values average 68 dB (F3), 66.5 dB (F2), and 64 dB (F1). The F0 condition is worst performing (60.5 dB). Each condition shows a statistically significant improvement over the previous condition (p-values<<0.05)

3.6.5 Quality Of Experience (Quality of experience (QoE))

Figure 3.25 shows the MOS, averaged over the 24 subjects. It is seen that all three foveation conditions have their MOS > 3. For the F1 and F2 conditions, the foveation is certainly perceptible, but it may not hinder the users' experience, since the perceived degradation is only 'slightly annoying' (F1) or 'not annoying'. With an MOS > 4, the F3

Figure 3.19: Comparison of averaged measured system latency per frame on the evaluated datasets.

	Component	KIT					BAL					OFF					LIV				
		F_0	F_1	F_2	F_3	F_{REF}	F_0	F_1	F_2	F_3	F_{REF}	F_0	F_1	F_2	F_3	F_{REF}	F_0	F_1	F_2	F_3	F_{REF}
Remote	Log-read	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	Partitioning	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
	Conversion	46	59	43	51	56	54	53	47	46	56	41	50	56	56	63	40	47	52	52	63
	Sampling	21	22	23	25	0	30	21	22	22	0	28	23	25	25	0	27	21	23	23	0
	Encoding	60	88	114	111	319	104	73	109	119	396	37	88	111	109	401	28	75	121	112	408
User	Decoding	25	36	46	47	121	47	38	59	58	158	17	31	36	38	137	20	41	52	56	142
	Conversion	0.7	1.3	1.8	2	7	1	1.2	1.6	1.6	6	0.5	1	1	1	7	0.4	1	2	2	7
	Rendering	18	18	18	18	18	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
Total(ms)		175	228	250	258	525	254	204	257	265	634	141	211	247	247	626	236	299	316	368	638

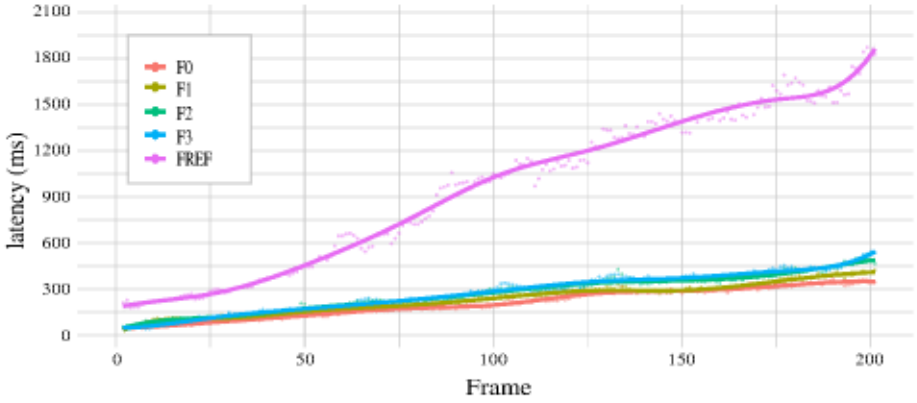
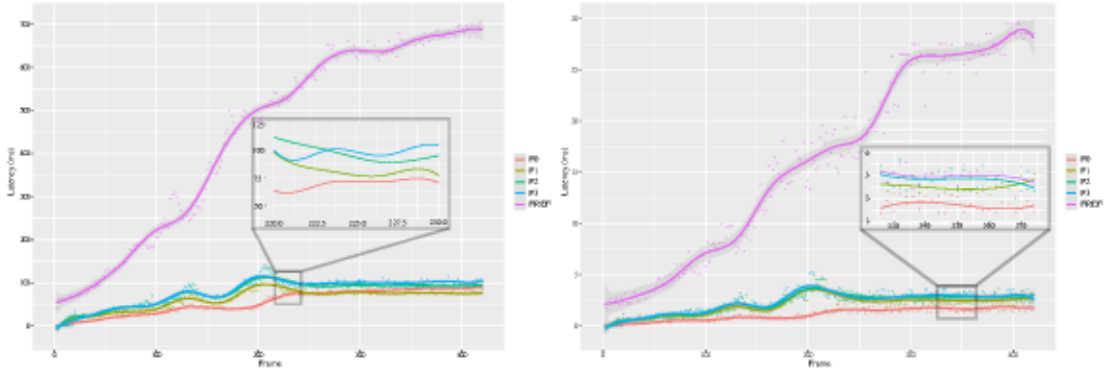


Figure 3.20: Per frame end-to-end latency for 3D reconstructed map.



(a) Decoding time.

(b) Conversion time.

Figure 3.21: Per-frame decoding and conversion time in the user site.

condition shows that subjects are not able to easily perceive the degradation, and even if they do, it is 'not annoying'. The F0 condition has an MOS < 3, implying the degradation can be annoying for subjects, which further negates the benefits it offers on the other metrics.

Table 3.9: Comparison of averaged component Latency per frame on the evaluated datasets. (Gray rows are remote site values)

Component	OFF					LIV				
	F_0	F_1	F_2	F_3	F_{REF}	F_0	F_1	F_2	F_3	F_{REF}
Log-read	4	4	4	4	4	4	4	4	4	4
Partitioning	0.0	0.3	0.3	0.3	0.0	0.0	0.3	0.3	0.3	0.0
Conversion	141	150	156	156	284	112	103	106	192	297
Sampling	88	53	57	65	0	121	87	63	73	0
Encoding	137	118	131	139	611	104	122	133	136	812
Decoding	117	131	136	138	550	98	114	136	142	592
Conversion	4	4.3	4.7	6	24	3.2	4.9	4.6	5.2	30
Rendering	14	14	14	14	14	14	14	14	14	14
Total(ms)	505	484	503	522	1487	397	449	461	566	1750

Table 3.10: Independent-samples t-test for latency on raw Point-cloud.

		mean				
Parameter - P_1	Parameter - P_2	P_1	P_2	t	p	Signif: Y/N
FREF	F0	605.25	171.25	73.98	0.00	Y
FREF	F1	605.25	213.42	74.95	0.00	Y
FREF	F2	605.25	59.35	59.35	0.00	Y
FREF	F3	605.25	57.63	57.63	0.00	Y
F0	F1	171.25	213.42	-9.23	0.00	Y
F0	F2	171.25	258.08	-17.05	0.00	Y
F0	F3	171.25	254.81	-15.90	0.00	Y
F1	F2	213.42	258.08	-9.76	0.00	Y
F1	F3	213.42	254.81	-8.71	0.00	Y
F2	F3	258.08	254.81	0.62	0.54	N

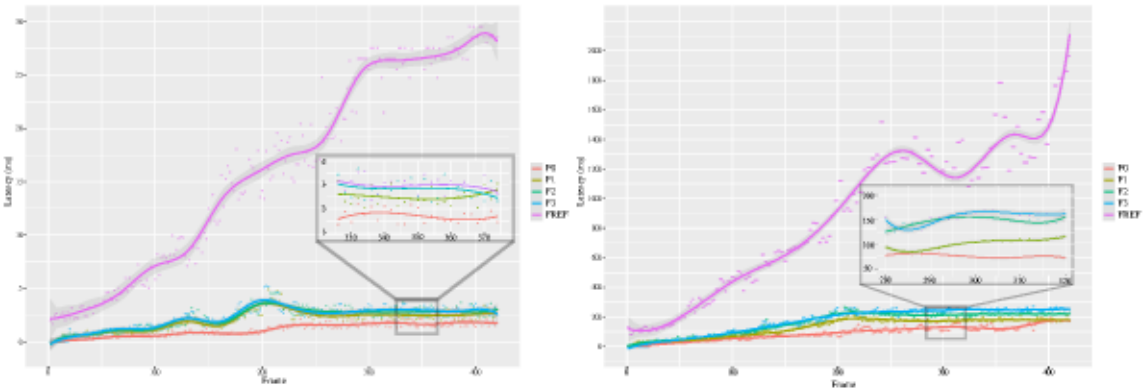
3.6.6 Visual Search Assessment

The mean Reaction time (RT) of correct response trials was calculated per condition. The data was treated to remove outliers; data beyond the 5^{th} percentile and 95^{th} percentile threshold) for each column was removed using the MATLAB "quantile" function. The analysis considers both the high discrimnablity and low discrimnablity data together, The remaining data has about 42 points in each column. However, this data is NOT normally distributed; instead, it is log-normally distributed. The log of each column was taken, and then the Two-way Students' T-test was used to compare the means (log) of the data distributions.

As shown in Figure 3.26, the mean and standard deviation of participants in condition **F0** gives a mean of ($\mu = 1219.6, \sigma = 883$), showing that participants were very slow and the large standard deviation indicates that the RT in this condition is farther away from the mean. The mean and standard deviation for the F1 condition is ($\mu = 512, \sigma = 198$), for F2 ($\mu = 442.4, \sigma = 283$), F3 ($\mu = 288.9, \sigma = 138.2$), and FREF ($\mu = 248.8, \sigma = 120.8$). The

Table 3.11: Independent-samples t-test for Latency - Global Map.

		mean				
Parameter - P_1	Parameter - P_2	P_1	P_2	%age reducn	p	Signif: Y/N
FREF	F0	953.84	207.29	76.69	0.00	Y
FREF	F1	953.84	234.02	72.48	0.00	Y
FREF	F2	953.84	270.47	68.88	0.00	Y
FREF	F3	953.84	275.66	69.22	0.00	Y
using %age reduction						
F0	F1	76.69	72.48	4.21	0.00	Y
F0	F2	76.69	68.88	7.81	0.00	Y
F0	F3	76.69	69.22	7.47	0.00	Y
F1	F2	72.48	68.88	3.60	0.00	Y
F1	F3	72.48	69.22	3.26	0.00	Y
F2	F3	68.89	69.22	-0.34	0.59	N



(a) Conversion time. (b) Encoding time.

Figure 3.22: Per-frame conversion and decoding time in the user site.

Table 3.12: Independent-samples t-test for PSNR RAW - pointclouds.

		mean				
Parameter - P_1	Parameter - P_2	P_1	P_2	t	p	Signif: Y/N
F0	F1	66.48	67.98	-5.80	0.00	Y
F0	F2	66.48	70.02	-12.71	0.00	Y
F0	F3	66.48	70.54	-3.62	0.00	Y
F1	F2	67.98	70.02	-8.22	0.00	Y
F1	F3	67.98	70.54	-2.30	0.02	Y
F2	F3	70.02	70.54	-0.47	0.64	N

Statistical analysis based on Two-way Students' T-test, which compares the means of the data distributions in Table 3.13 showed that, the comparisons that are NOT statistically significant are between F1 and F2 ($p-value = 0.0507$) and the test between F3 and FREF (p-value = 0.1817). All the other differences in mean are significant ($p-values << 0.05$).

To assess whether the distance between the target and selected randomly selected object was affected by the foveated conditions. Subjects were asked to estimate a distance

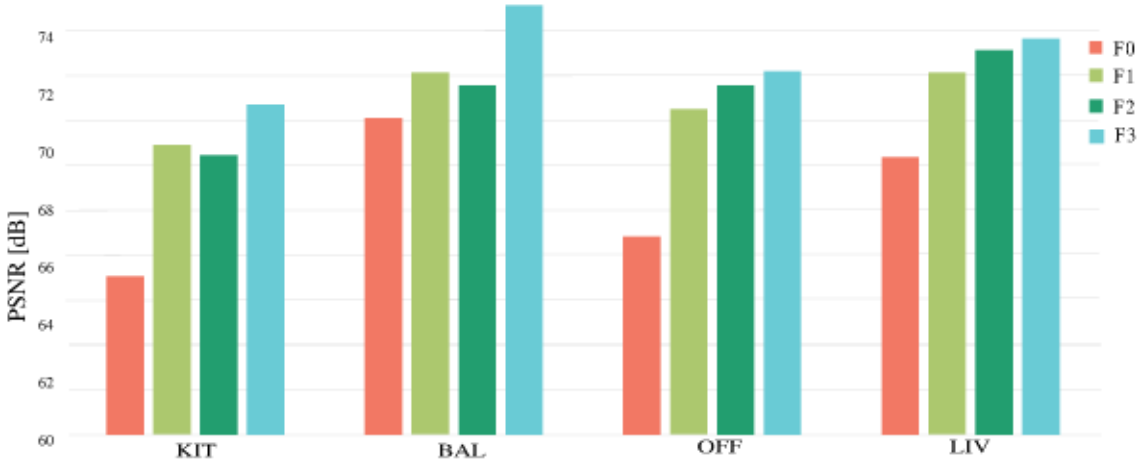


Figure 3.23: Volumetric density based PSNR for the raw point-cloud and the 3D reconstructed map streaming.

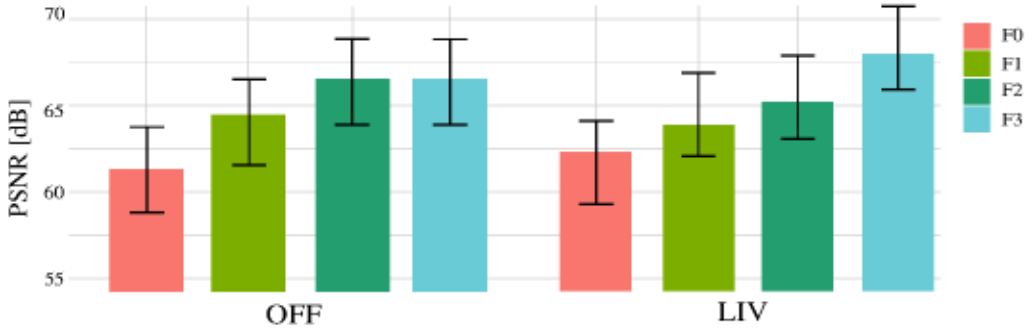


Figure 3.24: Volumetric density based PSNR for 3D reconstruction of the **OFF** and **LIV** data-sets.

Table 3.13: Visual search reaction time assessment overall p-values.

	F0	**F1**	**F2**	**F3**	**FREF**
F0		0.0001	<< 0	0.0253	<< 0
F1			0.0507	<< 0	<< 0
F2				0.0004	<< 0
F3					0.1817
FREF					

given reference distances, and they were asked to click on the reference distances. The distance estimation error was combined for high and low discreminablity conditions. Figure 3.14 shows distance estimation errors in detail.

To assess whether the immersive remote visualiation system provided with the foveated condition was affecting subjects, they were asked to answer with a 5 point Likert scale : (1) Strongly disagree; (2) Disagree; (3) Neither agree nor disagree; (4) Agree; (5) Strongly agree the following two statments:

Statement 1: I had to focus my attention strongly to accomplish the task in this trial.

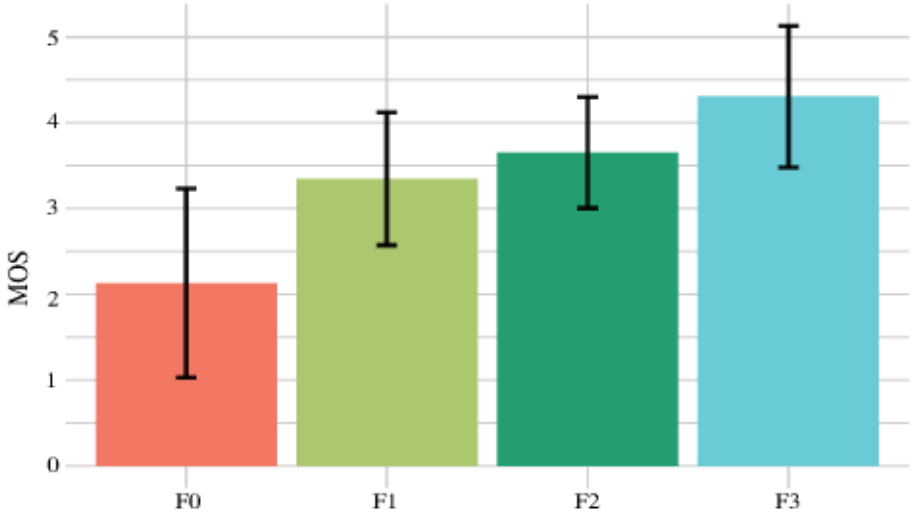


Figure 3.25: Mean MOS for the QoE metric against the experimental conditions (**F0**,**F1**,**F2**,**F3**). Error bars show the standard deviation.

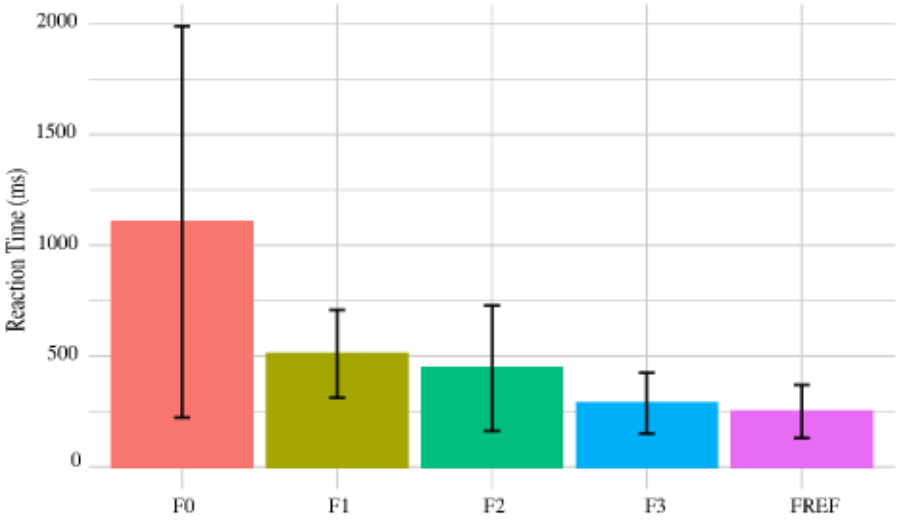


Figure 3.26: Reaction time as a function of experimental conditions (**F0**,**F1**,**F2**,**F3** and **FREF**). Error bars represent standard deviation of the RT.

Statement 2: I succeeded in performing the task to my satisfaction.

Figure 3.27, shows the mean and median response for Statement 1, results of the user studies revealed that participants in F1 condition had to focus strongly compared to the other three conditions, Moreover, a statistical analysis based on Wilcoxon Ranksum Test, that compares medians of the data distributions revealed a statistically significant in its difference of median value when compared to all the other conditions ($p - values << 0.0$), seen in Table.3.15. A statistical analysis between F1 and F3 showed their is a statistically significant difference ($p - values << 0.0129$). Also between F1 and FREF ($p - value < 0.0253$). but no significant effect between F1 vs F2, F2 vs.F3, F2 vs. FREF and F3 vs. FREF.

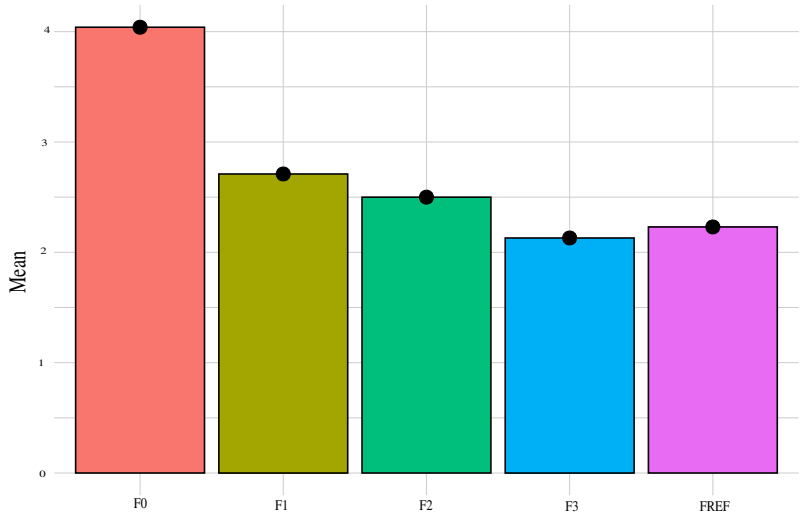


Figure 3.27: Participants response for statement 1 in both High and Low discriminability experiments.

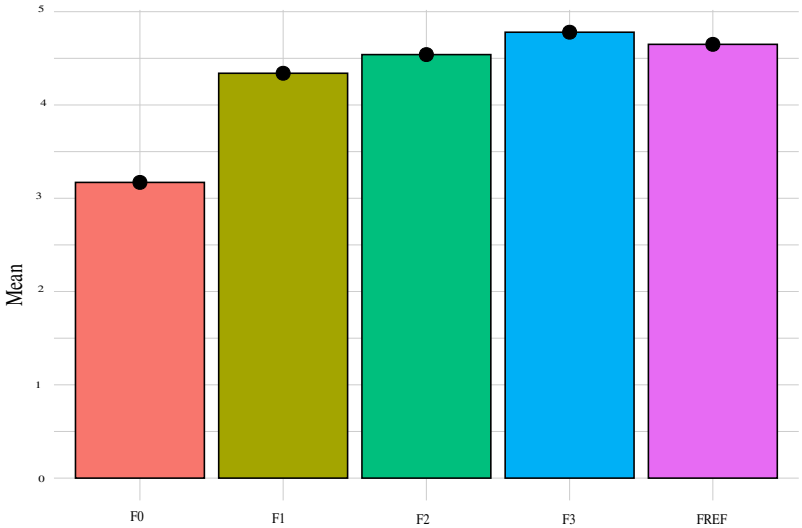


Figure 3.28: Participants response for statement 2 in both High and Low discriminability experiments.

For the success rates in statement 2, results of the user studies, as seen in Fig. 3.28 reviled that participants in F0 condition responded neither agree nor disagree with there performance compared to the other conditions. Specifically, the average response for this experiment is three and a statistical analysis with a Wilcoxon Ranksum Test in Table. 3.17 for this condition is statistically significant ($p - values << 0.0$) in its difference of median value when compared to all the other conditions. Participants response on the other comparisons in average is more than four, that indicates participants are satisfied with there performance and a statistical analysis between F1 vs. F3, F1 vs. FREF and F2 vs. F3 is statistically significant. but, all the other comparisons do not show statistically significant difference.

Table 3.14: Distance estimation errors for both High and Low discriminability experiments.

	F0	F1	F2	F3	FREF
Overall Errors	11	5	15	16	7
Errors by "1" step	7	4	14	16	4
Errors by "2" steps	4	1	1	0	3
Errors - under estimation	8	4	14	6	6
Errors - over estimation	3	1	1	10	1

Table 3.15: Wilcoxon Ranksum Test for Statement 1 for both High and Low discriminability experiments .

	F0	F1	F2	F3	FREF
F0		<<	<< 0	<< 0	<< 0
F1			0.2097	0.0129	0.0253
F2				0.0653	0.1111
F3					0.5195
FREF					

Table 3.16: Wilcoxon Ranksum Test for Statement 2 for both High and Low discriminability experiments .

	F0	F1	F2	F3	FREF
F0		<< 0	<< 0	<< 0	<< 0
F1			0.1106	0.0012	0.0166
F2				0.0277	0.1587
F3					0.8079
FREF					

3.6.7 Visual Tracking Assessment

For the visual tracking assessment, the mean and standard deviation of the RMSE is analyzed; the analysis considers both pink and yellow balloon data together for reputability. The data was organized in columns, one for each condition - F0, F1, F2, F3, and FREF - each column had 5106 points and was treated to remove outliers: The data beyond the 5^{th} percentile and 95^{th} percentile threshold (using the MATLAB "quantile" function) for each column was removed. After removing the outliers, The remaining data has about 4595 points in each column. However, this data is NOT normally distributed - instead is "log-normally distributed". The log of each column was taken, and then the Two-way Students' T-test was used to compare the means (of the log) of the data distributions.

As seen in Fig. 3.29, The user study results revealed a smaller error for condition F2 compared to the other conditions. Mainly, The mean, standard deviation for the F2 condition is ($\mu = 0.1498, \sigma = 0.1074$) and the Two-way Students' T-test showed that it is statistically significant ($p - values << 0.0$). In other words, this implies the F2 condition is a optimal condition between quality and latency requirements: latency increases from

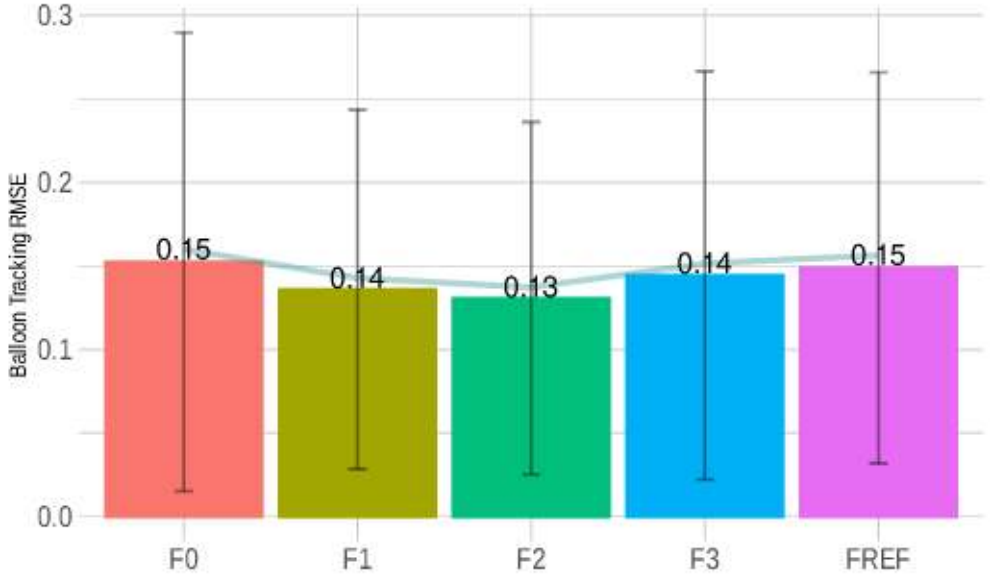


Figure 3.29: Balloon tracking RMSE mean and standard deviation. Error bars represent standard deviation of the RT.

F0 to FREF and the Quality gets better from F0 to FREF, with the cost of bandwidth and latency. All the other conditions are statistically significant, except the Two-way Students' T-test between F1 vs. F3 ($p-values << 0.08$) and between F0 vs. FREF as expected, Subjects found hard to track the balloon while they are in the F0 condition, which offers the lowest visual quality and It was hard to track the balloon while they are using the FREF condition as it have latency.

Table 3.17: Two-way Students' T-test on balloon Tracking experiment.

	F0	**F1**	**F2**	**F3**	**FREF**
F0		0.0001	$<< 0$	0.0253	0.1451
F1			0.0098	0.0865	$<< 0$
F2				$<< 0$	$<< 0$
F3					0.002
FREF					

The users' study regarding their focus and performance while doing the experiment was also analyzed here. The mean and median response for Statement 1 revealed that participants in the F1 condition had to focus strongly compared to the other three conditions. Moreover, a statistical analysis based on Wilcoxon Ranksum Test revealed a statistically significant difference in its median value when compared to all the other conditions ($p-values << 0.0$). But, All the other conditions do not show any statistically significant

difference when compared among each other. For the success rates in statement 2, results of the user studies indicate all participants succeeded in performing the task, and the statistical test revealed none of the conditions show any statistically significant difference when compared against the others.

3.7 Discussion

The four metrics analysed here offer a cost-benefit understanding of the tested conditions, i.e., the benefits in bandwidth and latency vs. the costs in PSNR and QoE. For instance, the F0 condition as expected, offers the most benefit for bandwidth and latency, but the costs in PSNR and QoE are the highest. Whereas the FREF condition is the ideal in terms of PSNR and QoE, the overall analysis demonstrates that the foveated conditions together provide the optimal cost-benefit ratio, as compared to both F0 and FREF conditions. The perceived degradations are seen to not significantly impact QoE. A deeper analysis shows that the F3 condition performs significantly better in the benefit metrics, while its costs are also not significantly worse than FREF. As expected, the F1 condition falls at the lower end within the 3 conditions, but still offers significantly higher benefit on latency and bandwidth. The F2 condition offers a good cost-benefit compromise between the two conditions. Here, the flexibility of the FR framework offers a key advantage. Real-time usage requirements and a user-selectable approach can allow users the choice among the three conditions, being able to switch among them as required. The FR framework contributes to the state-of-the-art in enhancing the 3D scene visualization experience in immersive remote visualization system.

3.7.1 Real-World Use Case

To demonstrate a real-world application with the FR framework, the remote inspection use case is discussed here. Such tasks generally involve a remotely controlled vehicle with on-board cameras and sensors, and have hard constraints on the networking infrastructure and bandwidth availability. Figure 3.30 shows the original and foveated point-clouds of a captured real-world 3D scene side-by-side, where high latency and bandwidth statistics for the original point-cloud are noted. In such cases, the FR framework can be switched on to substantially reduce the stress on the network resources, without significantly impacting user experience, and helping maintain presence in immersive remote visualization system.

3.8 Conclusions and Future Work

This chapter presented a novel FR based visualization pipeline that utilizes the acuity fall-off in human visual systems. The approach facilitates the processing, transmission, and rendering of dense point-clouds / 3D reconstructed scenes while simultaneously

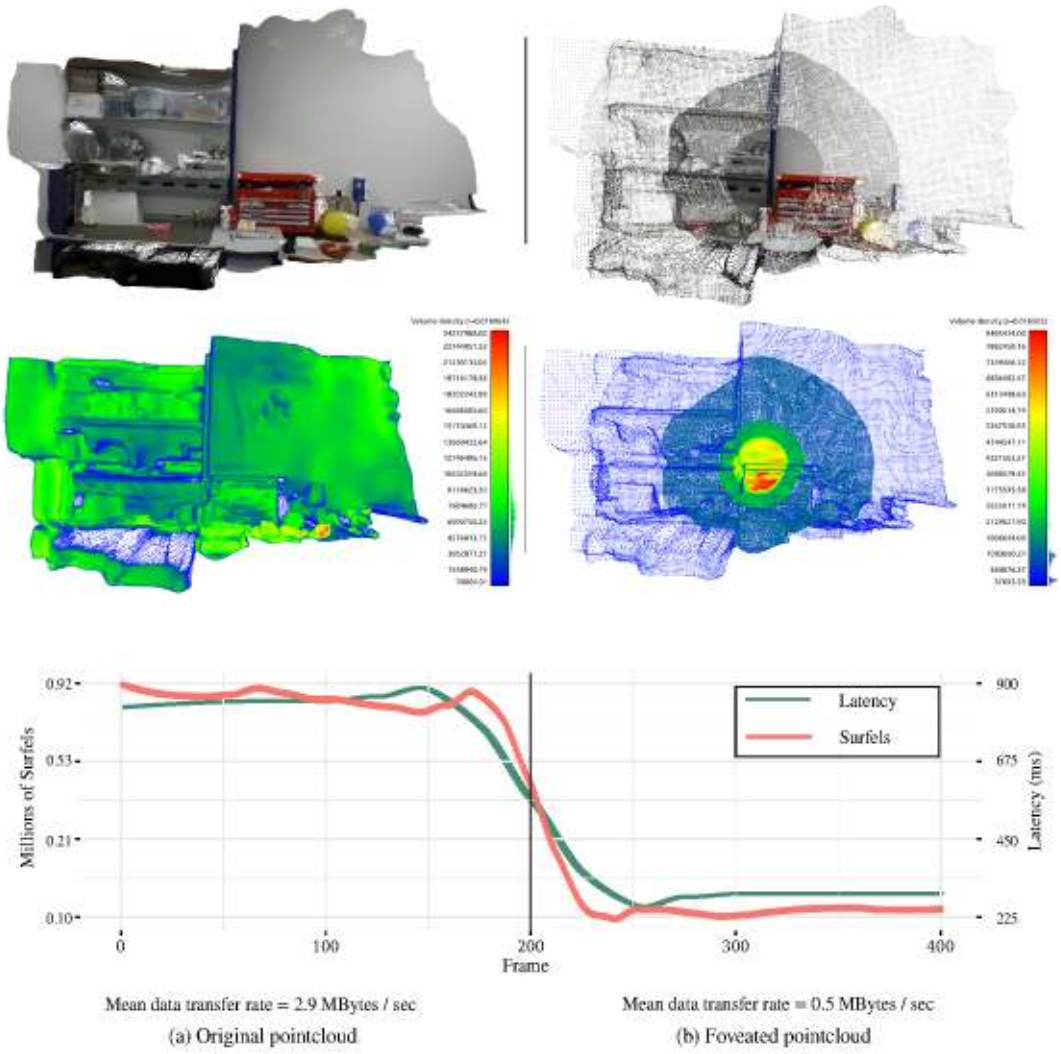


Figure 3.30: Remote inspection use case example. The FR framework dramatically reduces the bandwidth and latency in the system.

reducing throughput requirements and maintaining a rich visual experience for the user in applications such as immersive remote telepresence. The main contribution of this paper shows that by incorporating user eye-tracking data, remotely acquired real-time 3D data can be represented to the user in a *foveated* way, which not only helps reduce the bandwidth and latency, but also maintains the user visual quality experience. Validation experiments demonstrated significant reductions in latency and throughput, higher than 60% in both. The novel PSNR metric allowed to discriminate among the foveated conditions for objective visual quality assessment, showing the overall advantages of the framework. Preliminary user trials demonstrated that FR of real-time 3D perception data in VR does not have a significant negative impact on visual quality experience. Figure 3.31 shows the benefit cost ratio compared against the the experimental conditions. F2 condition is the optimum, or it gives the maximum benefit in terms of quality and transmission rate. Furthermore, the proposed FR framework provides the ability to choose

among different conditions (settings) based on task and user requirements.

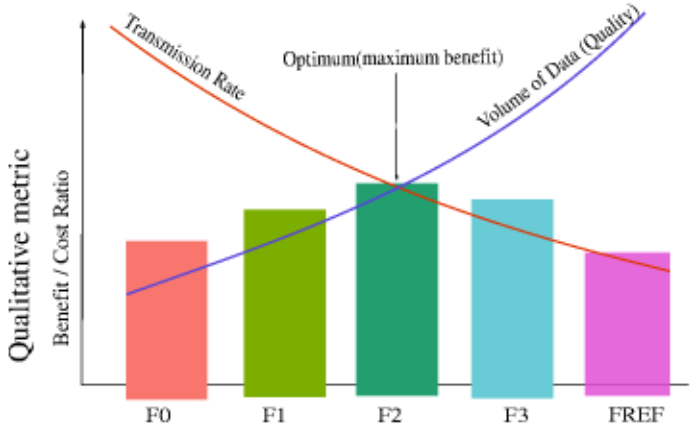


Figure 3.31: Benefit-cost ratio against different conditions.

Future investigations will include the analysis of the limitations in the approach, e.g., effects of discontinuities at region boundaries and the over-sampling in the peripheral regions. A comprehensive user study will help situate the FR framework in terms of usability and user experience in real-world environments.

4

Gaze Contingent Object-Level Remote - Immersive Visualization Framework

"We often engage the defense mechanism of tunnel vision, just to keep ourselves focused on our daily lives. This makes us terribly jaded in our perception of what is really around us."
(Vera Nazarian)

"If ever there was a more perfect picture of love, it was the silhouette of this couple standing at the window with the full moon behind them in a star filled sky." (Jason Medina)

As briefly discussed in section 2.1.2, vision is a sense that remote users are heavily dependent upon while using immersive remote visualization systems. Various applications, such as working with remotely teleoperated robots for disaster response applications, require operators to give attention or sustained focus on their task fully, processing all inputs and filtering relevant information to execute the appropriate action. This intense use of perceptual and cognitive skills may lead to mental and physical workload, which may cause fatal hazards and cause negative implications on task performance.

A considerable emphasis has to be given to human attention mechanisms because it has its limits as a resource when provided with multiple information. On the other side, It can be seen as a selection process and could be used to draw engagement to specific information from the dynamic spatio-temporal scene — it filters out information that is not needed and frees resources for the task at hand. However, It is prone to errors that can be raised by limitations in the sensory system, which leads to an inability to notice significant visual changes.

This chapter proposed a remote visualization system by exploiting the benefit and limitation of human attention mechanisms to facilitate the processing, streaming, and rendering of 3D data to a remote user, thereby reducing the amount of data transmitted. The acquired remote scene can contain several physical objects of various types (e.g., people, vehicles) interacting with each other or their environment. A scene understanding

mechanism and user's gaze point are used to draw the user's engagement to a specific place in the scene. The proposed system can detect, extract semantics, select and stream important visual information to the user.

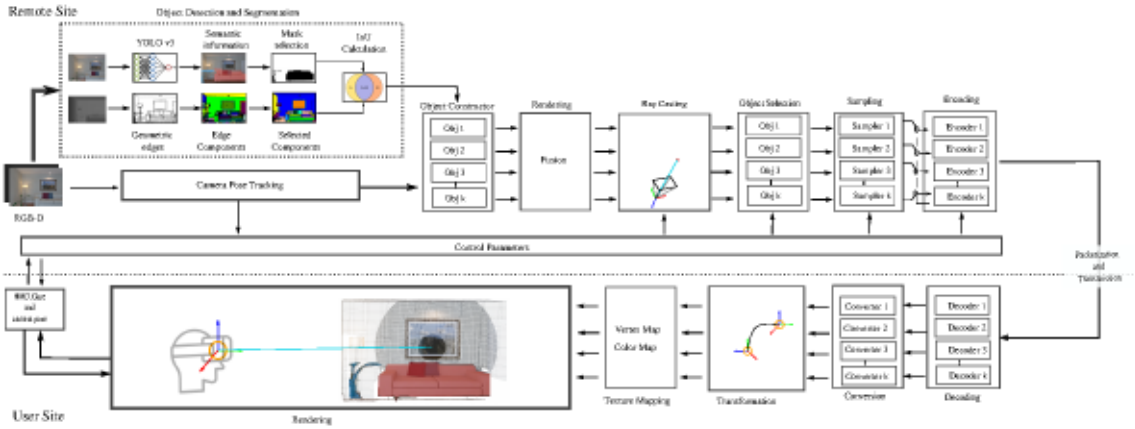


Figure 4.1: A schematic showing an overview of the proposed object-level remote immersive visualization framework.

4.1 System Overview

The proposed framework enables real-time processing, transmission, and rendering of dense point-clouds / 3D reconstructed scenes in immersive remote visualization, while maintaining a rich visual experience. Figure 4.1 shows the proposed framework, It comprises of a server-client architecture. It is divided into three major parts: the remote site, user site and communication network between them.

- **Remote site:** The remote site consists of the following modules:(1) acquisition, (2) object detection and segmentation, (3) Object level 3D reconstruction, (4) map partitioning, (5) foveated sampling, (6) encoding, and (7) streaming, as shown in Fig. 4.1. The implementation of the remote site encoder and streaming strategy is detailed in Section 3.3.2.

- **User site:** The user site manages the: (1) decoding, conversion, and texture rendering of the streamed 3D data, (2) tracking of the eye-gaze and HMD pose, and (3) real-time transfer of gaze and pose information to the remote site. A VR-based interface is designed using the UE graphics development environment on Windows 10, which serves as the immersive environment for the user. A point-cloud decoder and a conversion system to transfer the textures to the UE GPU shaders is implemented. A detail description of of this system can be found in Section 3.3.1.

- **Communication network:** A custom point-cloud and data packetization and streaming pipeline was implemented using the Boost ASIO cross-platform C++ library

for the communication network and communication network described in detail in Section 3.3.3 is used here.

4.2 Object Detection and Segmentation

Visual information provided to a user can contain many physical objects of various types (e.g., people, vehicles...). Inspired by the human visual perception and attention mechanisms, object detection and segmentation technique was proposed to extract the most relevant information from the input RGB-D signal for further processing. This section carefully investigates how the human vision consciously perceives objects in the environment. Moreover, how it retrieves what these objects are from memory in daily lives to construct a coherent view of reality. This section proposes two strategies for extracting information about the scene: semantic and geometric instance detection and segmentation strategies.

4.2.1 Semantic Instance Detection and Segmentation

Semantic scene understanding and segmentation strategies aim to identify target objects in the input RGB data and determine the categories and position information to achieve machine vision understanding. Computers get trained or instructed to perceive and understand just like HVS by following human brain functionalities. Numerous approaches have been proposed to mimic the HVS; one of the first vision-based object-based detectors was proposed by Viola and Jones [163]. This work mainly focused on detecting faces in an image using harr-like features and Adaboost feature classification. Following this work, researchers used object detection by manually extracting feature models such as HOG (histogram of oriented gradient), SIFT (scale-invariant feature transform), Haar (Haar-like features), and other classic algorithms along with Support Vector Machines (SVMs) based classifiers [94, 104, 40, 73, 89, 113].

With the evolution of neural networks, inspired by simplified models of neurons in the brain recent algorithms can extract scene information robustly in the presence of different objects in the entire scene. The most famous examples are R-CNN (region-based convolutional neural networks) [46] and the YOLO (you only look once) [129]. R-CNN is a two-stage detection algorithm; The first stage identifies a subset of regions in an image that might contain an object. It uses selective search to identify several bounding-box object region candidates and then independently extracts features from each region for classification. YOLO instead looks at parts of the image with high probabilities of containing the object. The algorithm uses a single convolutional network to predict the bounding boxes and the class probabilities.

These two techniques focus primarily on performance over speed instead a research work by Bolya et al., [10] proposed a real-time instance segmentation framework using one-stage object detectors which performs instance segmentation by breaking into two

parallel tasks. The framework is trained on Microsoft's Common Objects in Context dataset (COCO), the most popular object detection dataset [91]. The dataset contains 300,000 segmented images with 80 different categories of objects with exact location labels. This segmentation framework gives a simple, fully convolutional model for real-time ($> 30fps$) instance segmentation which is higher than the current state of the art works. For this reason, the semantic segmentation pipeline of the proposed framework is based on this work.

(a) Color Image (b) Masks with Label and confidence (c) Mask 1 (d) Mask 2

Figure 4.2: Semantic segmentation pipeline outputs. (a) Input color image, (b) semantic information overlayed on the input color image, (c) and (d) shows the output masks.

As shown in Figure 4.1, the semantic segmentation pipeline takes the input RGB image $\mathbf{C_t}$ and It gives as an out put a semantic information: object masks $\mathbf{m_t}$: $\Omega \rightarrow \mathbb{R}$ and for all detected masks $\forall m_t \in \mathbf{C_t}$ it gives a confidence $\mathbf{p_t} \in \{0, 100\}$, bounding boxes $\mathbf{b_t} : \Omega \rightarrow \mathbb{N}^4$ and class IDs $\mathbf{I_t} \in \{0, 100\}$. In the following step, object masks $\mathbf{m_t}$ with more than 50 % confidence score $\mathbf{p_t}$ value are selected and this value is used as a threshold to filter out false positives and ensure that a predicted bounding box and masks have a certain minimum score.

While this system provides good semantic information such as masks, class categories, and exact location, it has three limitations:

1. Limited number of object categories - It doesn't detect objects outside the 80 predefined categories.

2. Object masks are not flawless – It does not always perform an excellent accurate segmentation at the edge or border of objects.

3. The pipeline runs in parallel with the rendering system; the prediction time needs to meet the real-time requirements of the rendering system

Thus to overcome all this three limitations, this chapter proposes a geometry-based object segmentation pipeline in parallel.

4.2.2 Geometric Instance Detection and Segmentation

Perceiving a visual scene needs a complex understanding of color, shading, shape, motion, texture, and context information, as described at the beginning of this section, making it challenging to understand these pieces of information simultaneously. It is natural then

to choose a strategy of dividing the problem of perception into parts, and this strategy is known as the *minima rule* from the cognitive science theory [59]. Figure 4.3 shows how the outlines in this figure are easily recognized without color, shading, motion, texture, or context: The figure contains only shape and that of a restricted type, namely silhouettes [59]. Thus, in many cases, shape alone permits object recognition. Indeed we can recognize thousands of objects entirely by their shapes.

Figure 4.3: Easily recognized silhouettes, from [59].

The geometric instance detection and segmentation pipeline proposed here is based on this idea of *minima rule*. The minima rule states that human perception usually divides a surface into parts along the concave discontinuity of the tangent plane. Naturally, a visually consistent region has a consistent concavity, convexity, or curvature. I.e., the surface of the 3D shape can be defined as a collection of the planar, concave and convex regions. Research work in [179], [78] and [156] showed that locally concave regions in the input depth map are more likely to correspond to object boundaries, and locally convex regions are likely to belong to an object and should not become segment boundaries. A concave-convex relation is defined to improve the algorithm's performance through the computation of the angle between surface normals in this work.

The input depth map $\mathbf{D}$ can reveal two properties about edge boundaries in the scene:

1. Edge boundaries tend to have significant depth measurement differences around borders.

2. Sudden changes of the normal surface direction around edges.

An algorithm that segments objects from the depth measurement is implemented using these properties. As described in section 3.2, the input depth map $\mathbf{D}$ is associated with a vertex map $\mathbf{v}$ and a normal map $\mathbf{n}$. Edges can be detected through the computation of the angle between surface normals of adjacent points as follows:

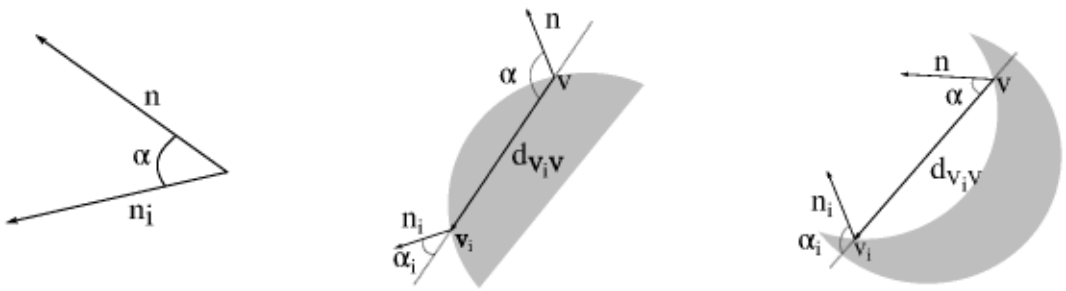
$$cos(\angle(\mathbf{n}, \mathbf{n}_i)) = \mathbf{n}.\mathbf{n}_i \tag{4.1}$$

Figure 4.4 shows the concave-convex relation between neighboring vertices $\mathbf{v}$ and $\mathbf{v_i}$, the unit connection distance vector is defined as $\mathbf{d}_{\mathbf{v_i v}} = \frac{(\mathbf{v_i}-\mathbf{v})}{|\mathbf{v_i}-\mathbf{v}|}$. The cosine of the angle α can be calculated using the identity for the dot products $\hat{a}.\hat{b} = |\hat{a}|.\left|\hat{b}\right|.cos(\alpha)$ with $\alpha = \angle(\hat{a}, \hat{b})$. When the angle α_i is smaller than angle α the connection line of the two vertices on passes through the interior part of the shape as shown in Figure 4.4 b and the two vertices are convexly connected. This can be derived as follows:

$$\alpha_i < \alpha \Rightarrow cos\alpha_i - cos\alpha > 0 \leftrightarrow \mathbf{n}_i.\mathbf{d}_{\mathbf{v_i v}} - \mathbf{n}.\mathbf{d}_{\mathbf{v_i v}} > 0, \tag{4.2}$$

Similarly, For concave connection the relationship can be defined when the angle α is greater than the angle α_i and as shown in Figure 4.4 c the connection line of the two vertices are on the outside and this can be derived as follows:

$$\alpha_i > \alpha \Rightarrow cos\alpha_i - cos\alpha < 0 \leftrightarrow \mathbf{n}_i.\mathbf{d}_{\mathbf{v_i v}} - \mathbf{n}.\mathbf{d}_{\mathbf{v_i v}} < 0, \tag{4.3}$$



(a) Angle between normal vectors (b) Convex Relation (b) Concave Relation

Figure 4.4: A schematic showing the convexity and concavity between vertices by comparing normal angles.

Using this definition of concave-convex relation and similar to the research work [22] and [131] concave boundaries are defined by computing the dot product between the normal at $\mathbf{v}$ in a local neighborhood $\mathcal{N}$ and a penalty term for concave regions ϕ_i is defined as follows:

$$\phi_i = \min_{i \in \mathcal{N}} \begin{cases} 1, & \text{if } (\mathbf{n}_i.\mathbf{d}_{\mathbf{v_i v}} - \mathbf{n}.\mathbf{d}_{\mathbf{v_i v}}) > 0, \\ (\mathbf{n}_i.\mathbf{n}), & \text{otherwise} \end{cases} \tag{4.4}$$

ϕ_i will have a higher value 1 when the $\mathbf{n}_i$ and $\mathbf{n}$ are on a convex surface and it will take the dot product when it is on a concave surface. This operation is performed for all neighbors $\mathcal{N}$ take the minimum.

The distance between the neighboring vertex $\mathbf{v}$ to vertex $\mathbf{v_i}$ defined as $\mathbf{d}_{\mathbf{v_i v}}$ is used as another cue for depth borders, a depth discontinuity term ϕ_d in a local neighbourhood $\mathcal{N}$ is defined as follows:

$$\phi_d = \max_{i \in \mathcal{N}} |(\mathbf{v_i} - \mathbf{v}).\mathbf{n}| \tag{4.5}$$

As shown in Figure 4.5, The geometric segmentation pipeline takes the input depth image $\mathbf{D_t}$ from the ICL-NUIM 3.5.1 dataset as an input and computes the geometric edges Figure 4.5(b), We applied the Connected Components Labeling algorithm from [9] to convert the extracted edges into a symbolic one, in which all pixels of the same object (connected component) are given the same geometric labels $\mathbf{L_t} : \Omega \rightarrow \{\mathbf{0}..\mathbf{N}\}$, where N is the number of extracted edge components Figure 4.5 (c). Each detected geometric component have a label $\mathfrak{L}_t$, bounding boxes $\mathbf{b_t} : \Omega \rightarrow \mathbb{N}^4$ and area (in pixels) of the component $\mathbf{a_t} : \Omega \rightarrow \mathbb{N}$. In the following step, geometric components $\mathbf{L_t}$ with small total area are filtered out as a noise and larger components are kept as a geometric masks.

(a) Depth Image (b) Edges (c) Edge Components (c) Selected Components

Figure 4.5: Detected components from a depth map.

Advantage of geometry-based segmentation systems is that they produce accurate object boundaries, their weakness is that they typically result in over-segmentations and they do not convey any semantic information.

4.2.3 Mask Merging

The semantic segmentation pipeline gives up to 80 different mask categories maximum. In contrast, geometric segmentation could give more. Masks from both pipelines can overlap; for this reason, the pairwise 2D overlap (Intersection over Union) between the semantic and geometric masks is computed. Whenever the Intersection over Union (IOU) is higher than a threshold of 50 %, the geometric mask is merged with the semantic mask, and a new object instance is created. Otherwise, the geometric masks will be considered as a new object, and a new object instance will be created. An illustration of this merging process can be seen in Figure 4.6.

4.3 Multiple Object SLAM

For each object of **N** instance formed in section 4.2.3, their surfels are generated using the parameters in section 3.2 and each has independent surfel map $\mathscr{M}_m \forall_m \in 0..N$ represented using an unordered list of surfels [124], where each surfel $\mathscr{M}_n$ has a position $\mathbf{p} \in \mathbb{R}^3$, a normal $\mathbf{n} \in \mathbb{R}^3$, a color $\mathbf{c} \in \mathbb{R}^3$, a weight $w \in \mathbb{R}$, a radius $r \in \mathbb{R}$, an initialization timestamp t_0, and a current timestamp t. The camera intrinsic matrix $\mathbf{K}$ is defined by: (i) the focal lengths f_x and f_y in the direction of the camera's $x-$ and $y-$ axes, (ii) a principal point in the image (c_x, c_y), and (iii) the radial and tangential distortion coefficients k_1, k_2 and p_1, p_2

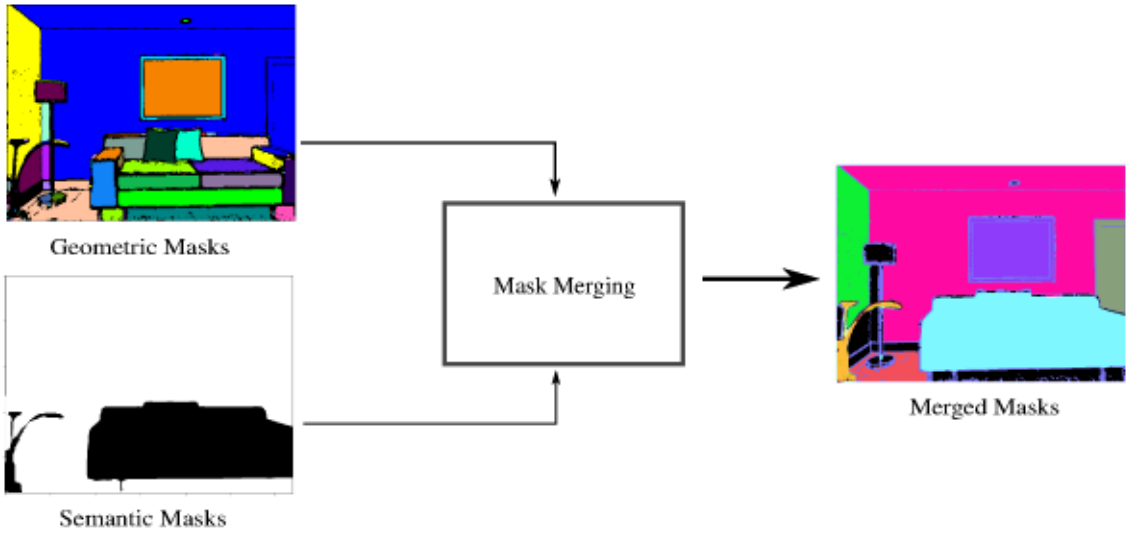


Figure 4.6: Semantic and geometric mask merging.

respectively. The domain of the image space in the incoming RGB-D frame is defined as $\Omega \subset \mathbb{N}^2$, with the color image $\mathbf{C}$ having pixel color $\mathbf{c} : \Omega \rightarrow \mathbb{N}^3$, and the depth map $\mathbf{D}$ having pixel depth $d : \Omega \rightarrow \mathbb{R}$.

Given $\mathbf{K}$, the 3D back-projection of a pixel $\mathbf{u}_i = [x_i, y_i]^T \in \Omega$ for a given segmented depth value $d(\mathbf{u}_i) \in \mathbf{D}$ is defined as $\mathbf{p}_i(\mathbf{u}_i, d(\mathbf{u}_i)) = \mathbf{K}^{-1} [\mathbf{u}_i, 1]^T d(\mathbf{u}_i)$. Over all pixels $\mathbf{u}_i$, this converts the RGB-D frame into a 3D map model. Further, the perspective projection of the 3D point $\mathbf{p}(x, y, z)$ is defined as $\mathbf{u} = \pi(\mathbf{Kp})$, where $\pi(\mathbf{p}) = [x/z, y/z]^T$ denotes the dehomogenization operation. The intensity value of the pixel $\mathbf{u} \in \Omega$ in the color image $\mathbf{C}$ with color $\mathbf{c}(\mathbf{u}) = [c_1, c_2, c_3]^T$ is defined as $I(\mathbf{u}, \mathbf{C}) = (c_1 + c_2 + c_3)/3$.

At each time step t, the segmented $\mathbf{C}_t$ and $\mathbf{D}_t$ are registered into their corresponding map model $\mathcal{M}_m$ by estimating the global pose of the camera $\mathbf{P}_t$, with rotation $\mathbf{R}_t \in \mathbb{SO}(3)$ and translation $\mathbf{t}_t \in \mathbb{SE}(3)$ with respect to the previous pose estimate $\mathbf{P}_{t-1}$. This registration provides the relative change from t to $t-1$.

$$\mathbf{P}_t = \begin{bmatrix} & \mathbf{R}_t & & \mathbf{t}_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{SE}(3) \tag{4.6}$$

The alignment between the current color $\mathbf{C}_t$ and depth map $\mathbf{D}_t$ with those of the active map model from the previous pose estimates, is achieved by minimizing a joint tracking error E_{track}, composed of the geometric and photometric error functions E_{icp} and E_{rgb} respectively.

$$E_m^{track} = E_m^{icp} + w_m^{rgb} E_m^{rgb} \tag{4.7}$$

The weight w_{rgb} is empirically set to 0.1 reflecting the difference in units between the two error terms [169]. The geometric error function estimates the back-projection error

from the current depth map $\mathbf{D}_t$ to the model depth map from $t-1$.

$$E_m^{icp} = \sum_i \left(\left(\mathbf{v^i} - \left(\exp(\hat{\xi}_m) \cdot \mathbf{T} \cdot \mathbf{v_t^i} \right) \right) \cdot \mathbf{n^i} \right)^2 \tag{4.8}$$

Here $\mathbf{v_t^i}$ is the back-projection of the i^{th} vertex in $\mathbf{D}_t$. $\mathbf{v}^i$ and $\mathbf{n}^i$ represent the corresponding vertex and normal in the model depth map from $t-1$. $\mathbf{T}$ is the current estimation of the transformation of the camera pose from $t-1$ to t, and $\exp(\xi)$ is the matrix exponential that maps a member of the Lie algebra $\mathfrak{se}(3)$ to a member of the corresponding Lie group $\mathbb{SE}(3)$ [170].

$$\begin{aligned} \mathbf{u}^a &= \pi \left(K \cdot \exp(\hat{\xi_m}) \cdot \mathbf{T} \cdot \mathbf{p}(\mathbf{u}, \mathbf{D}_t) \right) \\ E_m^{rgb} &= \sum_{\mathbf{u} \in \Omega} (I(\mathbf{u}, \mathbf{C}_t) - I(\mathbf{u}^a, \mathbf{C}_{t-1}^a))^2 \end{aligned} \tag{4.9}$$

Similarly, the color from the current frame C_t and the map model color estimate C_{t-1}^a is used to find the photometric error (intensity difference) between pixels. To minimize the function in Eq. 4.7, the Gauss-Newton non-linear least-squares method is used from [170].

4.4 Foveated Partitioning and Sampling

As introduced in Section 2.2, Table 3.1, and Figure 3.2-A, the HVS has the highest visual acuity at the center of the visual field and reduces monotonically towards the periphery based on the distribution of the photoreceptors. This distribution is defined as *Foveation* in section 3.1.1. Section 3.1.2 briefly describes the most popular way of determining Visual acuity quantitatively in terms of the *minimum angle of resolution* (MAR, measured in arcminutes), which is the smallest angle at which two objects in the visual scene are perceived as separate by the human eye. The relationship between MAR and eccentricity can be approximated as a linear model, Eq. 3.2. Given this definition, a map object selection and partitioning concept is defined in this section.

4.4.0.1 Object Selection and Map Partitioning

The map partitioning concept introduced in section 3.2.1 is extended here for each independent map $\mathcal{M}_m$, This symbol is used interchangeably for the live map and the global 3D reconstruction map for each independent model. Applying the foveation model to each segmented object map $\mathcal{M}_m$ implies projecting the retinal fovea regions into each object to partition it into concentric conical regions $\mathcal{M}_m^s$. After projecting the retinal fovea regions, the foveated regions $\mathcal{M}_m^s$ is then resampled to approximate the monotonically decreasing visual acuity in the foveation model.

Algorithm 2: Object Selection and Map Partitioning Algorithm

Input: $\mathcal{M}_{0...N}$ /* independent Maps to be partitioned */
$\mathbf{L}$ /* Gaze direction vector */
$e_{0\cdots n}$ /* Eccentricity angles */
foreach $\mathcal{M}_m \; \forall_m \in 0..N$ **do**
  foreach *surfel* P_i *in the map* $\mathcal{M}_m$ **do**
    $\mathbf{B} \leftarrow \mathbf{proj}_{\mathbf{L}}^{P_i}$ /* projection of P_i on L */
    $d^{vi} \leftarrow \|\vec{HB}\|$ /* dist. between H and B */
    $\boldsymbol{d} \leftarrow \mathbf{PB} \perp \mathbf{L}$ /* shortest distance */
    for *j=1* **to** *max(e)* **do**
      $r_j \leftarrow \tan\left(e_j\right) * d^{vi}$ /* calc. radii r_j */
    end
    /* put P_i into the maps $\mathcal{M}_m^0 \; \ldots \mathcal{M}_m^s$ */
    if $\boldsymbol{d} < r_0$ ***AND*** *IsForground* **then**
      $\mathcal{M}_m^0 \leftarrow \mathbf{P}_i$;
    else if $\boldsymbol{d} > r_0$ ***AND*** $\boldsymbol{d} \leq r_1$ **then**
      $\mathcal{M}_m^1 \leftarrow \mathbf{P}_i$;
    ⋮
    else
      $\mathcal{M}_m^s \leftarrow \mathbf{P}_i$
    end
  end
end

4.4.0.2 Object Map Sampling

The partitioned regions for each object map $\mathcal{M}_m^s$is down-sampled according to the acuity drop-off. Each independent map $\mathcal{M}_m^s$ is converted into a PCL point-cloud data structure, $\mathcal{P}_n$ for each $\mathcal{M}_n^s$ region $\forall n \in \{0...N\}$. Then, to implement the sampling, the $\mathbb{R}^3$ space of each $\mathcal{P}_n$ region needs to be further partitioned into an axis-aligned regular grid of cubes as shown in Fig. 3.6. This process of re-partitioning the regions is defined in section 3.2.2 and named as *voxelization* [134] and the discrete grid elements are called *voxels*.

This voxelization and down-sampling is a three-step process: (1) calculating the volume of the voxel grid in each region, which is the point-cloud distribution along x-, y-, and z-axes; (2) calculating the voxel size, i.e., dimension, $\mathfrak{v}_n$, for the voxelization in each region, and (3) down-sampling by approximating the point-cloud inside each voxel by its 3D centroid point.

For the voxel size, $\mathfrak{v}$, consider the voxelization of the central fovea region, $\mathcal{P}_0$. The smallest angle a healthy human with a normal visual acuity of 20/20 can discern is 1 arcminute, i.e., 0.016667°. In Eq. (3.2) therefore, $MAR_0 = 0.016667°$. Eq. 4.10 calculates the smallest visually resolvable object length.

$$\mathfrak{l} = d^{vi} * \tan(MAR_0) \tag{4.10}$$

The important consideration here is the value of d^{vi}, which is the distance to the image along the gaze vector $\mathbf{L}$. In Alg. 1, a d^{vi} value for each surfel is calculated. In contrast,

here in order to down-sample the region based on the voxelization, we calculate one d^{vi} value for the entire $\mathcal{P}_0$ region, approximated as the distance from $\mathbf{H}(hx, hy, hz)$ to the 3D centroid of the point-cloud in the region, Eq. (4.11).

$$\mathfrak{pc}_0 = \frac{1}{N_{\mathcal{P}_0}} \left(\sum_{i=1}^{N_{\mathcal{P}_0}} x_i, \sum_{i=1}^{N_{\mathcal{P}_0}} y_i, \sum_{i=1}^{N_{\mathcal{P}_0}} z_i \right) \tag{4.11}$$

$$d_0^{vi} = \mathbf{d}(\mathbf{H}, \mathfrak{pc}_0) \tag{4.12}$$

where $N_{\mathcal{P}_0}$ is the number of point cloud points in $\mathcal{P}_0$, and $\mathbf{H}$ is the eye gaze origin. Then, Eq. (4.10) is re-written as Eq. (4.13) to give the voxel size $\mathfrak{v}_0$ for the region.

$$\mathfrak{v}_0 = d_0^{vi} * \tan(MAR_0) \tag{4.13}$$

Once the voxelization of region $\mathcal{P}_0$ is finalized, for the subsequent concentric regions from $\mathcal{P}_1$ to $\mathcal{P}_n$, the voxel sizes are correlated through the linear MAR relationship. Eq. (4.14) shows that as the eccentricity angle of the regions increases, so do the voxel sizes.

$$\begin{aligned} MAR_n &= m \cdot E_n + MAR_0 \\ \mathfrak{v}_n &= \frac{MAR_n}{MAR_{n-1}} * \mathfrak{v}_{n-1} \end{aligned} \tag{4.14}$$

The increasing voxel size away from the fovea region implies more and more surfels of the point-cloud of the corresponding regions are now accommodated within each single voxel of that region. Therefore, when the down-sampling step is applied, the approximation of the point-cloud within a voxel is done over progressively dense voxels. For the down-sampling part, the region $\mathcal{P}_0$ being the fovea region is left untouched so its density is the same as the incoming global map density, i.e., the resolution set for the RGB-D camera. The down-sampling in the subsequent regions is done by approximating the point-cloud within each voxel with its 3D centroid, using Eq. (4.15).

$$\mathfrak{pc}_n^v(x, y, z) = \frac{1}{N_{\mathcal{P}_n}^v} \left(\sum_{i=1}^{N_{\mathcal{P}_n}^v} x_i, \sum_{i=1}^{N_{\mathcal{P}_n}^v} y_i, \sum_{i=1}^{N_{\mathcal{P}_n}^v} z_i \right) \tag{4.15}$$

Here $N_{\mathcal{P}_n}^v$ is the number of points in voxel v of the region $\mathcal{P}_n$ ($\forall n \in \{1...N\}$). Figure 3.6 shows the centroid approximation of the point-cloud, while Fig. 3.8 shows the sample voxel grids for the different regions.

4.5 Experiment Design And Evaluation Metrics

The proposed technique exploits the acuity fall-off and attention mechanisms to facilitate the processing, streaming, and rendering of 3D data to a remote user at the object level, thereby reducing the amount of data transmitted, introducing new use cases, and

streaming. The experiment design focused on evaluating the proposed system based on the following use-cases and applications:

- **Priority objects visualization**: Object masks with their id can be selected and streamed with different quality and streaming speed priorities; for example, semantically segmented objects have semantic information such as masks, class categories, and their exact location. While geometrically segmented objects do not have this information. The viewer may start visualizing the semantically segmented objects with high priority and geometrically segmented objects with low priority to save computational and bandwidth resources, which helps speed up processing at the user and remote side to meet real-time requirements.

- **Objects filtering:** Since all the objects in the scene are visualized at the object level, viewers may choose to not stream objects with low interest for example background objects. In addition, the remote site encoder can avoid processing and stream such objects to free computational and bandwidth resources.

- **Objects of interest:** An object of great interest can be selected and visualized only by assigning algorithmic and computational resources.

- **Background visualization:** Background objects can be visualized based on the requirements of the use case. Undesirable foreground objects can be removed, and only background objects can be visualized independently of other objects.

The experiment uses the two synthetic dataset of a living room environment and an office room (**OFF**), (**LIV**), seen in Fig. 3.11 and discussed in section 3.5.1.

4.5.1 Experimental Conditions

The experiment and evaluation phase is divided in two sets: with and without foveated partitioning and sampling detailed in section 4.4.

4.5.1.1 Experiment Condition Set One

In the first set of the experiment, three test conditions without applying foveated partitioning and sampling were created as follows:

- **SEMA:** This test condition only consists semantically segmented regions using the technique proposed in section 4.2.1. The hypothesis is that the reduction in visual quality would be evident in this condition. However, it would also offer the highest computational /network performance gain by filtering out unknown objects and keeping only relevant or interesting objects based on semantic segmentation information.

- **GEOM:** All segmented regions using geometry-based segmentation from section 4.2.2 are used. This test condition is chosen to test the geometry-based segmentation, and it could give more regions than the semantic segmentation pipeline. This condition can act as a middle point and expected intuitive balance between the visual quality degradation and the performance gain.
- **MERG:** Point-cloud in this condition are from semantic and geometric mask merging in section 4.2.3. All the geometric masks are merged with the semantic masks with this test condition. The visual quality reduction is expected to be the least likely to be detected. However, it would offer the least computational / network performance gain that could still sufficiently justify the use of the proposed system.

In addition to the three conditions above, one additional condition is created to represent the reference , i.e., no-sampling.

- **REF:** The raw point-cloud is left untouched and the proposed system is not applied. This condition is used as the base reference condition, against which all the other conditions SEMA, GEOM and MERG conditions are compared.

4.5.1.2 Experiment Condition Set Two

In the second set of the experiment, three test conditions by applying foveated partitioning and sampling on the point cloud were created by dividing the visual field into six regions - the Fovea, Parafovea, Perifovea, and near peripheral (as above) mid-peripheral region (60 °), and then the rest of the point-cloud in the far peripheral region. The visual quality reduction is expected to be the least likely to be detected, but it would offer computational / network performance gain by additionally applying the foveated partitioning and sampling technique.

- **SEMA_FOV:** This test condition only consists semantically segmented regions by applying the foveated partitioning and sampling technique.
- **GEOM_FOV:** All segmented regions using geometry-based segmentation with foveated partitioning and sampling.
- **MERG_FOV:** Point-cloud in this condition are from semantic and geometric mask merged with the foveated partitioning and sampling applied.

These second set of the experimental conditions are compared with one additional reference condition

- **FOV:** This condition applies the foveated partitioning and sampling technique on the raw pointcloud without any segmentation technique. The visual field is divided into six regions - the Fovea, Parafovea, Perifovea, and near peripheral (as above) mid-peripheral region (60 °), and then the rest of the point-cloud in the far peripheral region.

4.5.2 Evaluation Metrics

The evaluation metrics utilized for the experiments help analyze the performance of the FR framework in terms of the benefits it provides and the costs it imposes when implemented as part of an immersive remote visualization system. Therefore, the evaluation is conducted in a quantitative (objective) assessment to evaluate the algorithmic and computational performance in terms of:

- **Data Transfer Rate:** The improvement, or otherwise, in the data transfer rate in streaming.
- **Latency Reduction:** The improvement, or otherwise, in the end-to-end latency.

4.6 Results and Analysis

Similar to the procedures section 3.6 followed, five randomized HMD positions with varying distances to the center of the datasets were used to evaluate the objective metrics. Two hundred frames were tested for each HMD position from each dataset for each experimental condition. The 3D reconstruction analysis was done on the OFF and LIV datasets, and the data is averaged over 5 randomized HMD positions.

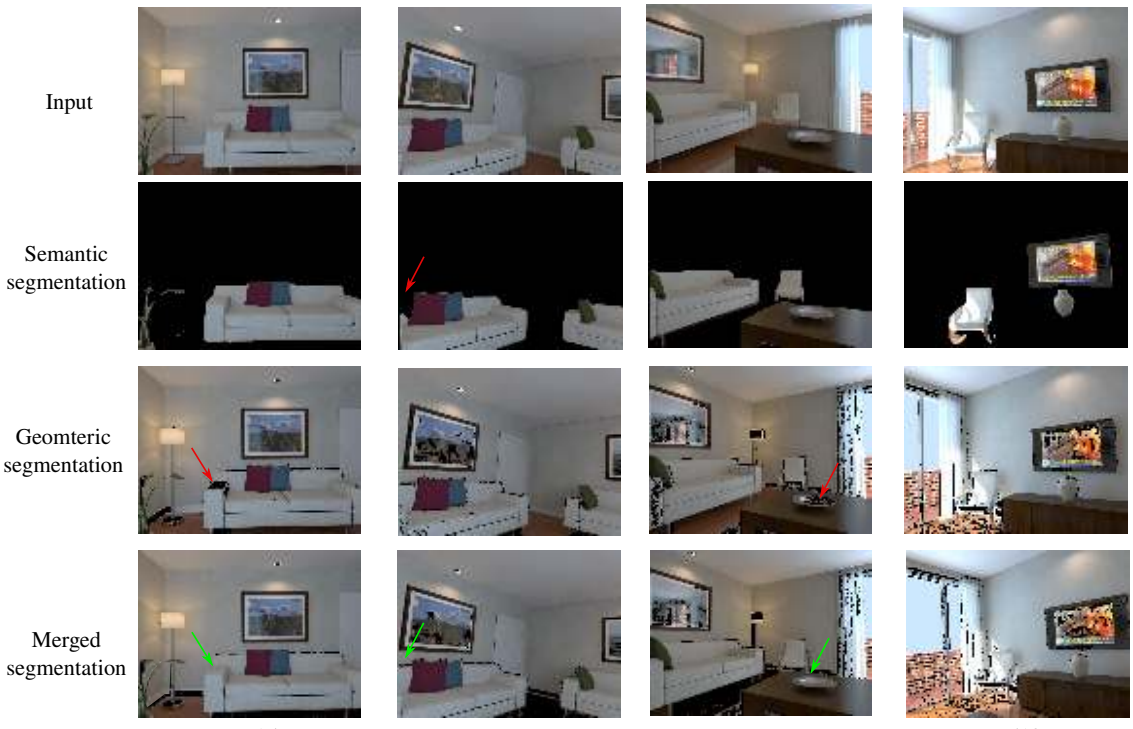


Figure 4.7: The figure shows the comparison of segmentation between semantic, geometric, and merged segmentation for **LIV** data set. The red arrow indicates missing regions, and the green arrow indicates missing regions completed by merging masks.

A visual comparison of the LIV and OFF segmentation output is shown in Fig 4.7 and Fig. 4.8. Both figures compare the semantic, geometric, and merged segmentation outputs. As expected object masks are not flawless, both segmentation techniques does not always perform an excellent accurate segmentation at the edge or border of objects. The red arrow indicates the missing or over segmented regions and blue arrow indicates regions completed by merging segmentation using both techniques. By calculating the Intersection over Union (IOU) between semantic and geometric segmentation mask, if it is higher than a threshold of 50 %, the geometric mask is merged with the semantic mask, and missing regions are completed.

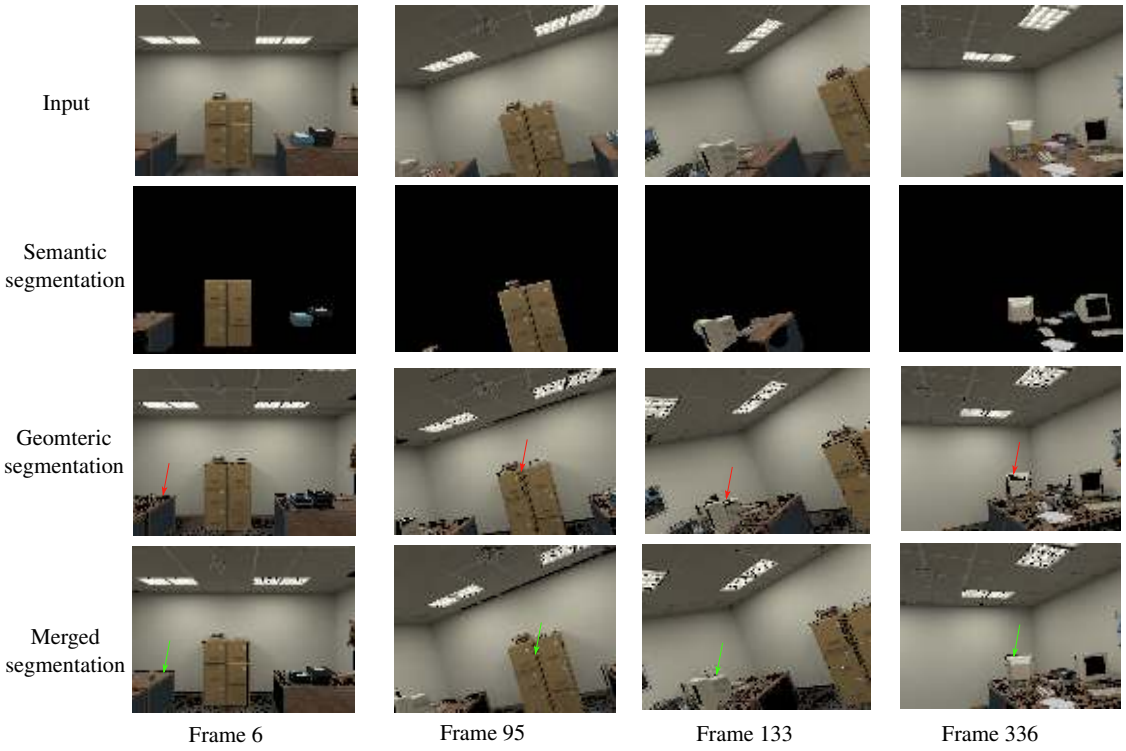


Figure 4.8: The figure shows the comparison of segmentation between semantic, geometric, and merged segmentation for **OFF** data set. The red arrow indicates missing regions, and the green arrow indicates missing regions completed by merging masks.

4.6.1 Data Transfer Rate

Table 4.1 and Fig.4.9 reports the average bandwidth required for **LIV** dataset point-cloud streaming in the first phase of the experiment without applying the foveation techniques and the relative percentage reduction in bandwidth as compared to the **REF** condition. The mean bandwidth required for the **SEMA** condition gives an average 49% reduction as compared with **REF** and the statistical t-test analysis showed the reduction is significant (p-values<<0.00). The relative bandwidth reduction for **GEOM** condition is in average 38% reduction and statistical t-test analysis showed statistically significant compared to

the **REF**. Similarly the numbers for **MERG** condition are similar with the **GEOM** condition and statistically significant. Within the 3 conditions (**SEMA**, **GEOM** and **MERG**), although **SEMA** is the most advantageous, the difference between the 3 reductions is not statistically significant (p-value >0.2).

Table 4.1: Two-way students' T-test on bandwidth reduction on **LIV** RAW - point cloud.

	P_1	P_2	μ_P_1	μ_P_2	Diff.	t	df_error	p
1	REF	SEMA	2.12	0.78	1.34	8.10	115.00	0.00
2	REF	GEOM	2.12	0.92	1.20	5.90	119.00	0.00
3	REF	MERG	2.12	0.90	1.23	6.06	149.00	0.00
4	SEMA	GEOM	0.78	0.92	-0.14	-1.07	190.00	0.29
5	SEMA	MERG	0.78	0.90	-0.12	-0.96	220.00	0.34
6	GEOM	MERG	0.92	0.90	0.02	0.16	224.00	0.88

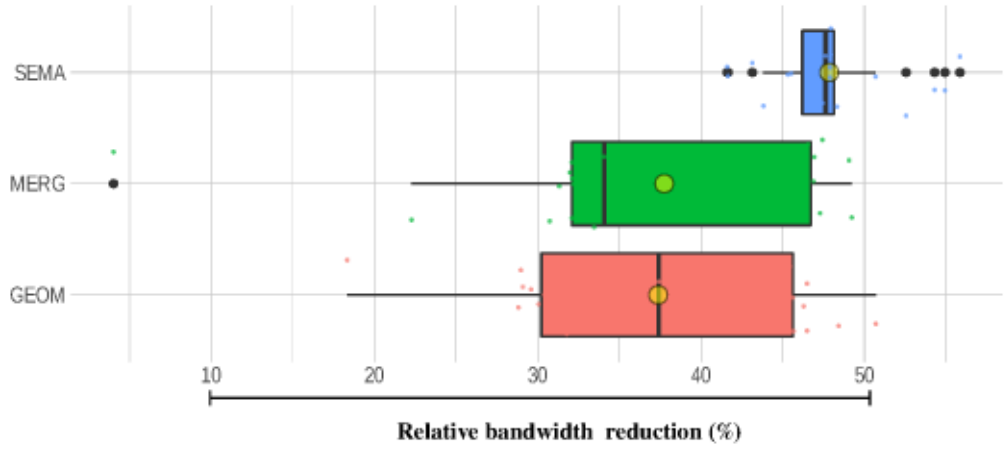


Figure 4.9: The figure shows relative bandwidth reduction in percentage for semantic, geometric, and merged segmentation for **LIV** dataset.

For the second phase of the experiment by applying foveated partitioning and sampling on segmented point cloud for **LIV** dataset, Table 4.2 and Fig. 4.10 shows how the relative band-width percentage improved by applying foveation as compared with the **FOV** condition. The relative bandwidth reduction follow the same trend here: **SEMA_FOV** - 76%, **GEOM_FOV** - 47% and **MERG_FOV** - 44%, as compared to **FOV** condition. All are statistically significant (p-values<<0.00) except the statistical test between **GEOM_FOV** and **MERG_FOV**.

The relative bandwidth reduction for data-set **OFF** in the experimental condition set one are reported in Table 4.3 and Fig.4.11 as compared to the **REF** condition. The mean bandwidth required for the **SEMA** condition gives an average 58% reduction as compared with **REF** and the statistical t-test analysis showed the reduction is significant

Table 4.2: Two-way students' T-test on bandwidth reduction on foveated **LIV** RAW - pointcloud.

	Parameter - P_1	Parameter - P_2	μ-P_1	μ-P_2	Diff	t	df_error	p
1	FOV	SEMA_FOV	1.72	0.27	1.46	22.09	102.00	0.00
2	FOV	GEOM_FOV	1.72	0.60	1.13	8.25	102.00	0.00
3	FOV	MERG_FOV	1.72	0.62	1.10	7.75	102.00	0.00
4	SEMA_FOV	GEOM_FOV	0.27	0.60	-0.33	-4.48	166.00	0.00
5	SEMA_FOV	MERG_FOV	0.27	0.62	-0.21	-4.69	166.00	0.00
6	GEOM_FOV	MERG_FOV	0.60	0.62	0.16	-0.28	166.00	0.78

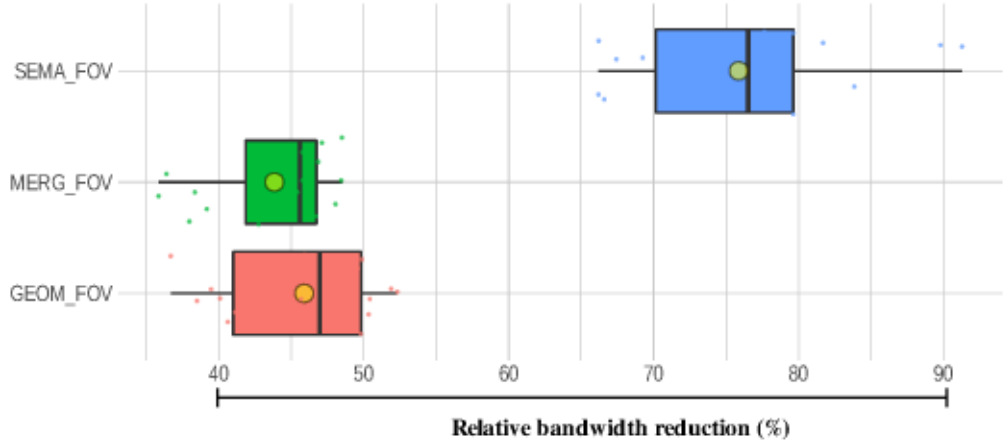


Figure 4.10: The figure shows relative bandwidth reduction in percentage for semantic, geometric, and merged segmentation on foveated **LIV** dataset.

(p-values<<0.00). Geometrical segmented condition **GEOM** reported a relative bandwidth reduction of 56% in average and the statistical t-test analysis showed statistically significant compared to the **REF**. When segmened Objects merged in **MERG** condition the relative bandwidth is 46% in average and it is statistically significant. Within the 3 conditions (**SEMA**, **GEOM** and **MERG**), although **SEMA** is the most advantageous, the difference between the 3 reductions is not statistically significant (p-value >0.1).

Table 4.3: Two-way students' T-test on bandwidth reduction on **OFF** RAW - pointcloud in experiment condition set one.

	Parameter - P_1	Parameter - P_2	μ-P_1	μ-P_2	Diff	t	df_error	p
1	REF	SEMA	1.97	0.56	1.41	9.79	89.00	0.00
2	REF	GEOM	1.97	0.63	1.34	8.38	91.00	0.00
3	REF	MERG	1.97	0.72	1.25	6.67	87.00	0.00
4	SEMA	GEOM	0.56	0.63	-0.07	-0.67	148.00	0.50
5	SEMA	MERG	0.56	0.72	-0.16	-1.40	144.00	0.16
6	GEOM	MERG	0.63	0.72	-0.09	-0.76	146.00	0.45

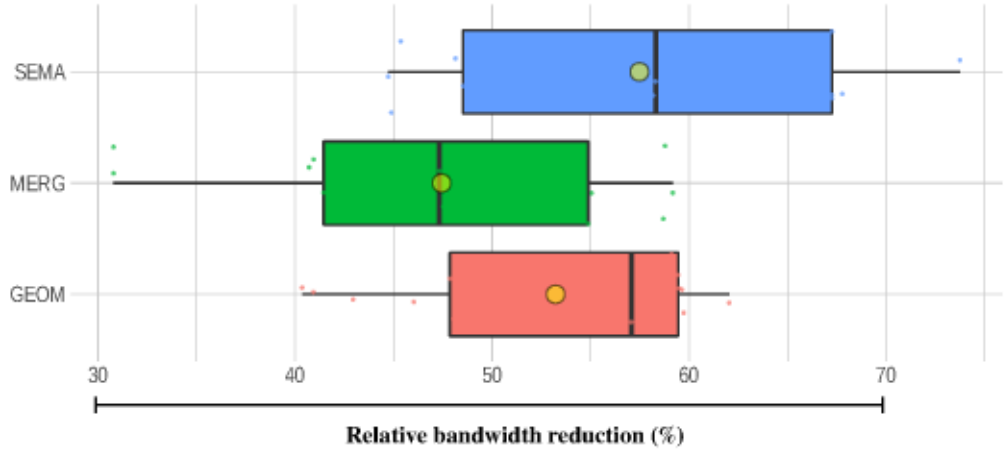


Figure 4.11: The figure shows relative bandwidth reduction in percentage for semantic, geometric, and merged segmentation for **OFF** dataset in experiment condition set one.

For the second phase of the experiment by applying foveated partitioning and sampling on segmented point cloud for **OFF** dataset, the relative band-width percentage improvement by applying foveation as compared with the **FOV** condition is shown in Table 4.6 and Fig. 4.12. The relative bandwidth reduction follow the same trend here: **SEMA_FOV** - 79%, **GEOM_FOV** - 63% and **MERG_FOV** - 39%, as compared to **FOV** condition. All are statistically significant (p-values<<0.00) except the statistical test between **GEOM_FOV** and **MERG_FOV**.

Table 4.4: Two-way students' T-test on bandwidth reduction on Foveated OFF RAW - pointcloud in experiment condition set two.

	Parameter - P_1	Parameter - P_2	μ-P_1	μ-P_2	Diff	t	df_error	p
1	FOV	SEMA_FOV	1.14	0.24	0.91	10.13	104.00	0.00
2	FOV	GEOM_FOV	1.14	0.50	0.64	5.23	148.00	0.00
3	FOV	MERG_FOV	1.14	0.75	0.39	2.14	98.00	0.04
4	SEMA_FOV	GEOM_FOV	0.24	0.50	-0.14	-4.24	214.00	0.00
5	SEMA_FOV	MERG_FOV	0.24	0.75	-0.33	-5.62	164.00	0.00
6	GEOM_FOV	MERG_FOV	0.50	0.75	-0.07	-2.80	208.00	0.01

4.6.2 Latency Reduction

Relative latency improvement statistics results in percentages for the real-time point-cloud streaming are illustrated in Table. 4.5 and Fig. 4.13 as compared with the **REF** condition for **OFF** dataset in experiment condition set one. The results presented for condition **SEMA** demonstrate more than 80 % relative latency improvement and two-way student's T-tests were used to analyse the statistically significance with the other

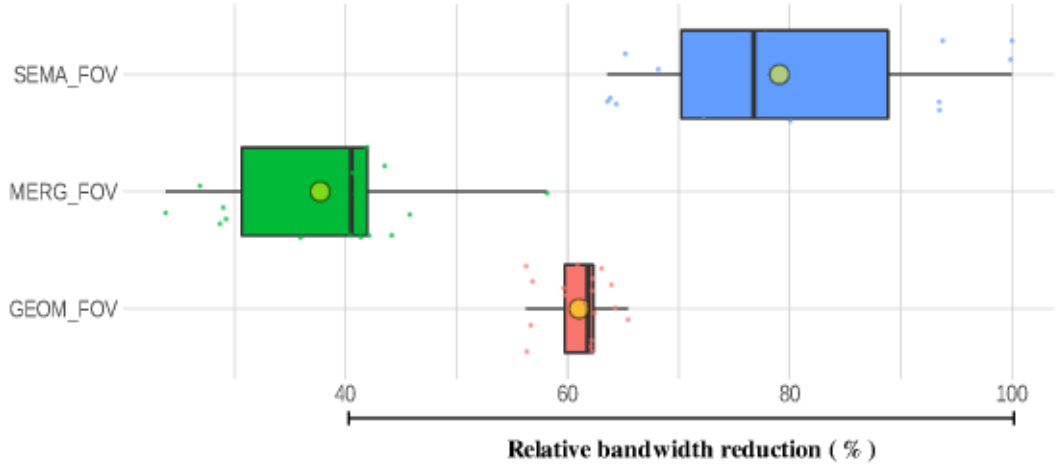


Figure 4.12: The figure shows relative bandwidth reduction in percentage for semantic, geometric, and merged segmentation for **OFF** dataset in experiment condition set two.

conditions, the test results are significant ($p << 0.00$). The relative latency improvement for conditions **GEOM** and **MERG** is also highlighted in the figure and table, the mean difference 34 and 28 milliseconds and there was a significant difference as compared to the **FOV**. but, no significant differences were found between **GEOM** and **MERG**.

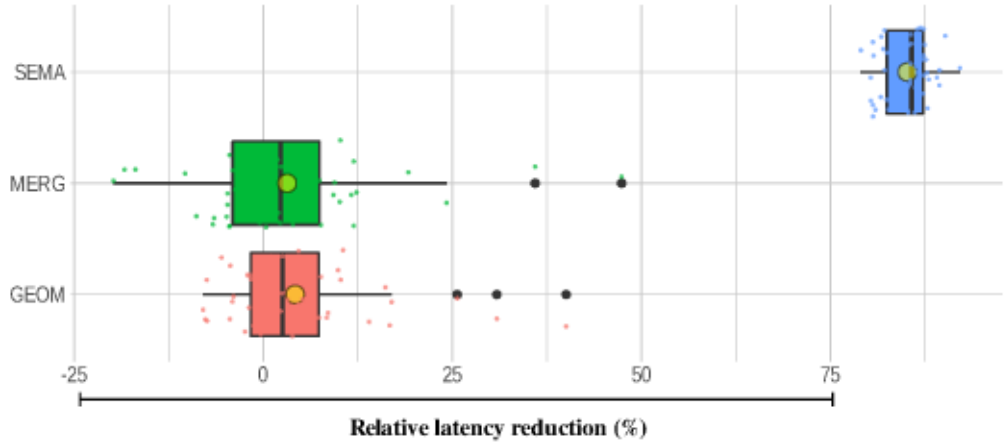


Figure 4.13: The figure shows relative latency reduction in percentage for semantic, geometric, and merged segmentation for **OFF** dataset in experiment condition set one.

Turning now to the experimental evidence for the second phase of the experiment when foveation is applied, the relative latency improvement is highlighted in Table. 4.6 and Fig. 4.14. The mean latency improvement for condition **SEMA_FOV** is around 142 milliseconds, which is around 82 % latency improvement. This result is significant at the $p << 0.05$ when compared with all other conditions. From the 4.6 and Fig. 4.14, It can be seen that the mean latency difference is around 7 milliseconds for experimental conditions

Table 4.5: Two-way students' T-test on latency reduction on **OFF** RAW -pointcloud in experiment condition set one.

	Parameter - P_1	Parameter - P_2	μ-P_1	μ-P_2	Diff	t	p
1	REF	SEMA	639.49	94.03	545.46	63.29	0.00
2	REF	GEOM	639.49	605.31	34.18	3.47	0.00
3	REF	MERG	639.49	610.92	28.56	3.04	0.00
4	SEMA	GEOM	94.03	605.31	-511.28	-90.97	0.00
5	SEMA	MERG	94.03	610.92	-516.89	-107.08	0.00
6	GEOM	MERG	605.31	610.92	-5.62	-0.83	0.41

GEOM_FOV and **MERG_FOV** and a very small amount of improvement and this results are significant when compared with **FOV** condition. but, no significant differences were found between **MERG_FOV** and **GEOM_FOV**.

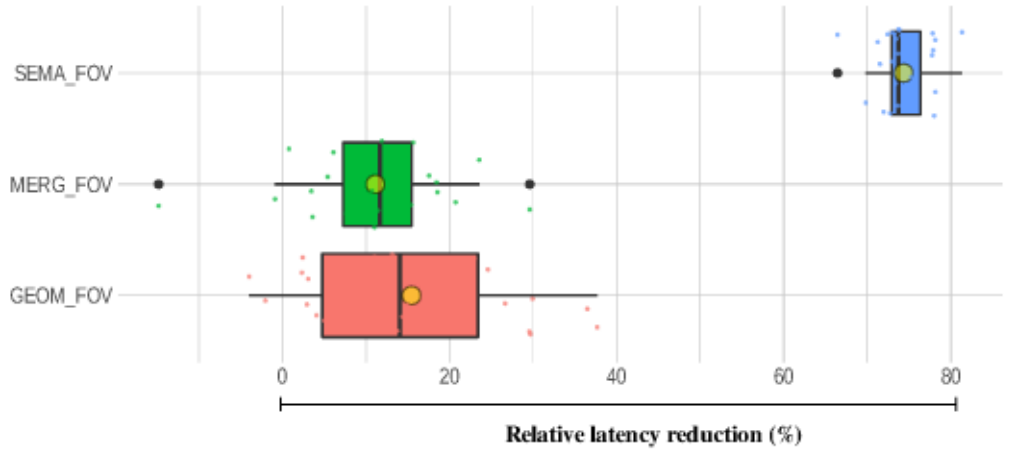


Figure 4.14: The figure shows relative latency reduction in percentage for semantic, geometric, and merged segmentation for **OFF** dataset in experiment condition set two.

Table 4.6: Two-way students' T-test on latency reduction on **OFF** RAW -pointcloud in experiment condition set two.

	Parameter - P_1	Parameter - P_2	μ-P_1	μ-P_2	Diff	t	df_error	p
1	FOV	SEMA_FOV	191.10	48.93	142.17	67.46	56.00	0.00
2	FOV	GEOM_FOV	191.10	162.17	28.93	7.46	56.00	0.00
3	FOV	MERG_FOV	191.10	170.52	20.58	7	56.00	0.00
4	SEMA_FOV	GEOM_FOV	48.93	162.17	-113.24	-32.65	56.00	0.00
5	SEMA_FOV	MERG_FOV	48.93	170.52	-121.59	-42.58	56.00	0.00
6	GEOM_FOV	MERG_FOV	162.17	170.52	-8.35	-1.93	56.00	0.06

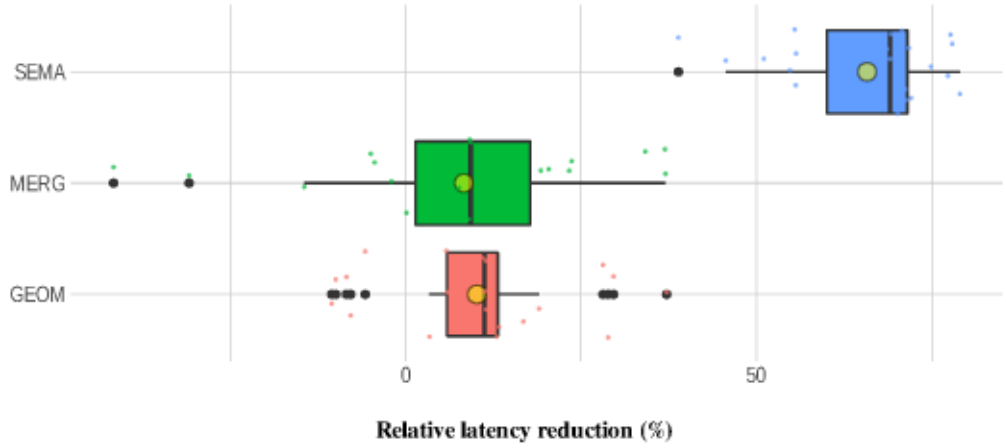


Figure 4.15: The figure shows relative latency reduction in percentage for semantic, geometric, and merged segmentation for **LIV** dataset in experiment condition set one.

Table 4.7: Two-way students' T-test on latency reduction on **LIV** RAW -pointcloud in experiment condition set one.

	Parameter - P_1	Parameter - P_2	μ-P_1	μ-P_2	Diff	t	p
1	FOV	SEMA_FOV	686.83	223.15	463.68	25.35	0.00
2	FOV	GEOM_FOV	686.83	590.14	96.69	5.44	0.00
3	FOV	MERG_FOV	686.83	602.23	84.61	3.47	0.00
4	SEMA_FOV	GEOM_FOV	223.15	590.14	-366.99	-24.97	0.00
5	SEMA_FOV	MERG_FOV	223.15	602.23	-379.07	-17.05	0.00
6	GEOM_FOV	MERG_FOV	590.14	602.23	-12.08	-0.55	0.58

Table 4.8: Two-way students' T-test on latency reduction on **LIV** RAW -pointcloud in experiment condition set two.

	Parameter - P_1	Parameter - P_2	$\mu - P_1$	$\mu - P_2$	Diff..	t	p
1	FOV	SEMA_FOV	303.92	157.48	146.44	14.32	0.00
2	FOV	GEOM_FOV	303.92	185.01	118.91	13.29	0.00
3	FOV	MERG_FOV	303.92	219.05	84.87	9.84	0.00
4	SEMA_FOV	GEOM_FOV	157.48	185.01	-12.72	-3.69	0.00
5	SEMA_FOV	MERG_FOV	157.48	219.05	-47.52	-8.69	0.00
6	GEOM_FOV	MERG_FOV	185.01	219.05	-24.01	-6.73	0.00

4.7 Discussion and Conclusions

This chapter has presented an approach to building a novel object-level visualization pipeline that allows streaming multiple object-level 3D reconstructed/ raw point-clouds. In contrast to Chapter 3 that utilizes the acuity fall-off in HVS to sample and transmit dense point-clouds / 3D reconstructed scenes, this chapter proposed to use systems that are inspired by the human attention mechanisms to detect, extract semantic, select and

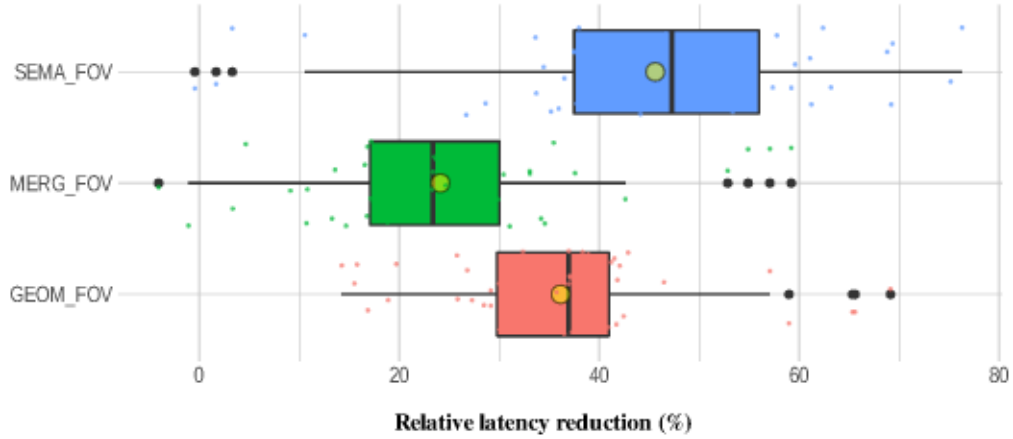


Figure 4.16: The figure shows relative latency reduction in percentage for semantic, geometric, and merged segmentation for **LIV** dataset in experiment condition set two.

stream important visual information to the user from remotely captured 3D data. A neural network based real-time instance segmentation technique is integrated to the pipeline. The segmentation technique provided semantic information such as masks, class categories, and exact location. However, It doesn't detect objects outside the 80 predefined categories and It does not always perform an excellent accurate segmentation at the edge or border of objects. To overcome this limitations a geometric based object detection and segmentation strategy using the depth map proposed and this strategy produce accurate object boundaries, their weakness is that they typically result in over-segmentation and they do not convey any semantic information. To take the advantages from both strategies segmented objects are combined and eventually the geometry of each segmented object is tracked, reconstructed and streamed to the user site, thereby reducing the amount and the speed of transmitting 3D reconstructed data at object level. This improvements was shown by designing experimental studies to understand the cost-benefits of the proposed framework. i.e., the benefits in bandwidth and latency vs. quality. The results from the two metrics analyzed showed that **SEMA** and **SEMA_FOV** conditions, as expected, offer the most benefit for bandwidth and latency. Whereas the **MERG**, **MERG_FOV**, **GEOM** and **GEOM_FO** conditions have a closer bandwidth and latency requirements like with the reference conditions.

5

Everything must come to an end

"The world is full of magic things, patiently waiting for our senses to grow sharper." (William Butler Yeats)

5.1 Conclusion

A close understanding of relevant theoretical foundations to remote visualization systems and technological and human factors that should be known when designing immersive interfaces has tremendous potential to improve the quality, the speed of communication, and the perception of remote environments. In order to develop better and more efficient remote visualization systems, it is also important to understand how to exploit the HVS or use the potential of human visual perception. Such understanding will help us to enhance the quality and speed of remote visualization systems by facilitating the processing, transmission, buffering, and rendering of remotely reconstructed environments while simultaneously reducing throughput and latency requirements. This thesis showed some of the potential and limitations of the HVS and how to exploit them for remote visualization. In this final chapter, we summarize the significant contributions of the results of this work and discuss future work.

5.2 Achieved results

In the rest of this section, we provide a summary of the research questions that were addressed in this Thesis:

***Main Research Question**: How can we support a real-time immersive remote visualization system for remote Teleoperation and Telepresence applications with state-of-the-art*

streaming rates?

This research question has been addressed throughout this thesis by developing a remote visualization system in Chapters 3 and 4. It showed that by exploiting the HVS features, remotely acquired real-time 3D data can be visualized to the user efficiently, which helps to reduce the bandwidth and latency and does not significantly impact the QoE.

Research Question 1: *What are the state-of-the-art immersive remote visualization systems for telepresence and teleoperation systems, and What are the technological, perceptual, and cognitive constraints in designing such systems?*

This research question has been addressed by performing a detailed literature review and developing experimental setups to understand the current state-of-the-art remote visualization systems for remote telepresence and teleoperation applications. It looked in detail at technological factors related to the field of view, camera orientation and viewpoints, degraded depth perception, time delays, and motion. In addition, It looked into cognitive and perceptual limitations; the Cognitive limitations looked into mental processes involved in gaining knowledge and comprehension, and the perceptual factors looked different visual cues that are processed when interacting with real or virtual environments.

Research Question 2: *What are the advantages and limitations of the HVS, and How can it be exploited in designing immersive remote visualization systems?*

This research question was thoroughly addressed in Chapters 3 and 4; these two chapters presented the essential physiological and perceptual foundation and features of the HVS that can be used to design efficient visualization systems and proposed models to describe these. Chapter 3, looked into the characteristics of the human eye, specifically on the distribution of the photoreceptors in the retina: cones and rods. The cone density is highest in the central region of the retina and reduces monotonically to a reasonably even density into the peripheral retina region and proposed a model to quantitatively represent it in terms of the MAR. Chapter 4 looked into the human attention mechanisms: specifically to select and to draw engagement to specific information from the dynamic spatio-temporal environment. The peripheral vision provides low-resolution cues to guide the eye movements so that the central vision visits all the interesting and crucial parts of the visual field. The chapter used this information to detect, segment, and assign semantic labels to interesting and crucial parts of the scene and used the acuity model proposed in chapter 3 to facilitate the remote visualization.

Research Question 3: *How do we design an improved remote visualization system with*

reduced latency and throughput requirements using the HVS compared to the current state-of-the-art techniques?

In the design of remote visualization systems inspired by human vision, user studies and experiments are fundamental tools to understand the limitations and potentials of the proposed system. The thesis designed and proposed different experiments and evaluation metrics. For applications such as remote inspection, search and rescue, and high-quality visualization, carefully designed experiments are proposed. The thesis evaluated and compared different experimental conditions and latency and throughput requirements in these experiments. It also proposed a novel volumetric point-cloud density based PSNR metric to evaluate the quality of the proposed approaches.

5.3 Future development

In this final section, The thesis presents a list of possible improvements and continual developments that need to be addressed in future work or by other researchers. Although this section could have listed many more important future developments, the lists presented here are those that we consider a great challenge to an immersive remote visualization, hoping that this will serve as a research agenda for future work.

- **Improvement to Discontinuities at Region Boundaries:** The proposed foveated rendering-based visualization pipeline in Chapter 3 uses the photoreceptors density distribution in HVS to facilitate the processing, transmission, buffering, and rendering in VR of dense point clouds / 3D reconstructed scenes. The chapter proposed to approximate the retina as being formed of discrete concentric regions, which helps to process and stream point-cloud in order to meet perceptual and performance requirements. However, discrete concentric regions create discontinuities at region boundaries, creating aliasing artifacts. In general, if real-time requirement is not of high priority, These artifacts can be fixed by creating more regions and taking more samples to counteract artifacts. For specific scenarios that require a high-quality visualization and interactivity such as, the visual search experiment in section 3.5.3.6.

- **Development of Foveated and Object based Point Cloud compression:** Developing a point-cloud based remote visualization system requires a real-time processing, streaming, and high-quality representations. The foveated rendering-based visualization pipeline proposed in Chapter 3 has the potential to substantially reduce point cloud resolution in the visual periphery, with hardly noticeable perceptual quality degradations. This can be a key idea to create a compression ratio that will have a low compression ratio at the fixation point and progressively increasing compression ratios towards the periphery. Similarly, knowledge about the remote scene that can contain several physical objects of various types (e.g., people, vehicles) can

be used to allocate different compression ratios to the objects of interest, and users can personalize to render streamed content.

- **Gaze Direction and View Port Prediction:** One of the key objectives of this thesis was to propose a strategy to create a novel remote visualization system that satisfies speed, throughput, and visual quality requirements in real time. The thesis demonstrated how to balance visual quality degradation and the performance gain to understand the impact through experiments. Thus, to find a perfect balance between quality and performance gain, it is necessary to develop new approaches that allow a remote site point cloud streamer to deliver predicted viewpoints using saliency, history of head orientation, and fixation points to predict future head and gaze position.

- **Availability of Specific Benchmarks and Datasets:** Although this thesis used a dataset from ICL-NUIM synthetic dataset and introduced the kitchen area (**KIT**) and a dynamic scene a moving balloon (**BAL**) benchmark dataset (section 3.5.1.) We believe that more benchmark dataset that can consider different dynamic behavior of remote scenes is highly needed for future evaluation.

Bibliography

[1] [Accessed: 25-Oct-2021]. Feb. 2014. URL: http://www.ntp.org/index.html (cit. on p. 60).

[2] M. 3DG and Requirements. *Call for Proposals for Point Cloud Compression V2*. Tech. rep. Hobart, AU: MPEG 3DG and Requirements, 2017 (cit. on p. 66).

[3] T. Akenine-Mller, E. Haines, and N. Hoffman. *Real-Time Rendering, Fourth Edition*. 4th. USA: A. K. Peters, Ltd., 2018. ISBN: 0134997832 (cit. on pp. 36, 38–41).

[4] D. Barnes et al. "Driven to distraction: Self-supervised distractor learning for robust monocular visual odometry in urban environments". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1894–1900 (cit. on p. 35).

[5] I. A. Bârsan et al. "Robust dense mapping for large-scale dynamic environments". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 7510–7517 (cit. on p. 35).

[6] A. K. Bejczy. "Virtual Reality in Robotics". In: *1996 IEEE Conference on Emerging Technologies and Factory Automation. ETFA '96*. 1996, 7–15 vol.1 (cit. on p. 42).

[7] B. Bescos et al. "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4076–4083 (cit. on p. 35).

[8] T. Blascheck et al. "State-of-the-Art of Visualization for Eye Tracking Data." In: *EuroVis (STARs)*. 2014 (cit. on p. 29).

[9] F. Bolelli, S. Allegretti, and C. Grana. "One DAG to rule them all". In: (2021) (cit. on p. 95).

[10] D. Bolya et al. "Yolact++: Better real-time instance segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* (2020) (cit. on p. 91).

[11] R. T. Born, A. R. Trott, and T. S. Hartmann. "Cortical magnification plus cortical plasticity equals vision?" In: *Vision Research* 111 (2015), pp. 161–169. ISSN: 18785646. DOI: 10.1016/j.visres.2014.10.002. URL: http://dx.doi.org/10.1016/j.visres.2014.10.002 (cit. on pp. 27, 48).

[12] W. H. Bosking, M. S. Beauchamp, and D. Yoshor. "Electrical stimulation of visual cortex: relevance for the development of visual cortical prosthetics". In: *Annual review of vision science* 3 (2017), pp. 141–166 (cit. on p. 24).

[13] D. A. Bowman et al. *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004. ISBN: 0201758679 (cit. on pp. 10–15, 20).

[14] V. Bruder et al. "On evaluating runtime performance of interactive visualizations". In: *IEEE transactions on visualization and computer graphics* 26.9 (2019), pp. 2848–2862 (cit. on p. 72).

[15] V. Bruder et al. "Voronoi-Based Foveated Volume Rendering". In: *EuroVis (Short Papers)*. 2019, pp. 67–71 (cit. on pp. 26, 42, 49).

[16] M. Carfagni et al. "Metrological and critical characterization of the Intel D415 stereo depth camera". In: *Sensors* 19.3 (2019), p. 489 (cit. on p. 31).

[17] A. Charlton. *What is Foveated Rendering? Explaining the VR technology key to lifelike realism*. [Accessed: 05-Sep-2021]. May 2021. URL: https://www.gearbrain.com/vr-foveated-rendering-explained-2652950180.html (cit. on p. 42).

[18] J. Y. C. Chen, E. C. Haas, and M. J. Barnes. "Human Performance Issues and User Interface Design for Teleoperated Robots". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (Nov. 2007), pp. 1231–1245 (cit. on pp. 2, 43).

[19] J. Chen, R. V. N. Oden, and J. O Merritt. "Utility of stereoscopic displays for indirect-vision driving and robot teleoperation". In: *Ergonomics* 57 (Jan. 2014), pp. 12–22. DOI: 10.1080/00140139.2013.859739 (cit. on p. 19).

[20] J. Y. C. Chen, E. C. Haas, and M. J. Barnes. "Human Performance Issues and User Interface Design for Teleoperated Robots". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37 (2007), pp. 1231–1245 (cit. on pp. 19, 20).

[21] A. Cheng, U. T. Eysel, and T. R. Vidyasagar. "The role of the magnocellular pathway in serial deployment of visual attention". In: *European Journal of Neuroscience* 20.8 (2004), pp. 2188–2192 (cit. on p. 24).

[22] S. Christoph Stein et al. "Object partitioning using local convexity". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 304–311 (cit. on p. 94).

[23] C. E. Connor, H. E. Egeth, and S. Yantis. "Visual attention: bottom-up versus top-down". In: *Current biology* 14.19 (2004), R850–R852 (cit. on p. 27).

[24] A. Cowey and E. T. Rolls. "Human Cortical Magnification Factor and its Relation to Visual Acuity". In: *Experimental Brain Research* 21 (1974), pp. 447–454 (cit. on pp. 25, 26, 48).

[25] C. Crick et al. "ROSbridge: ROS for non-ROS users". In: *Robotics Research*. Ed. by H. Christensen and O. Khatib. Springer, 2017, pp. 493–504 (cit. on p. 61).

[26] J. E. Cutting. "How the eye measures reality and virtual reality". In: *Behavior Research Methods, Instruments, & Computers* 29.1 (1997), pp. 27–36 (cit. on p. 15).

[27] J. E. Cutting. "Reconceiving perceptual space." In: (2003) (cit. on p. 15).

[28] A. Dai et al. "BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration". In: *ACM Transactions on Graphics* 36.3 (June 2017), pp. 1–18 (cit. on p. 34).

[29] J. Dai, K. He, and J. Sun. "Instance-aware semantic segmentation via multi-task network cascades". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3150–3158 (cit. on p. 36).

[30] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo. *Time-of-flight cameras and Microsoft KinectTM*. Springer Science & Business Media, 2012 (cit. on p. 31).

[31] P. Daniel and D. Whitteridge. "The representation of the visual field on the cerebral cortex in monkeys". In: *The Journal of physiology* 159.2 (1961), pp. 203–221 (cit. on p. 25).

[32] Davison. "Real-time simultaneous localisation and mapping with a single camera". In: *Proceedings Ninth IEEE International Conference on Computer Vision*. 2003, 1403–1410 vol.2. DOI: 10.1109/ICCV.2003.1238654 (cit. on p. 32).

[33] E. Dima et al. "View Position Impact on QoE in an Immersive Telepresence System for Remote Operation". In: *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2019, pp. 1–3 (cit. on p. 42).

[34] D. Drascic. "An investigation of monoscopic and stereoscopic video for teleoperation." In: (1993) (cit. on p. 18).

[35] J. L. Drury et al. "Changing shape: Improving situation awareness for a polymorphic robot". In: *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. ACM. 2006, pp. 72–79 (cit. on p. 19).

[36] J. Duncan and G. W. Humphreys. "Visual search and stimulus similarity." In: *Psychological review* 96.3 (1989), p. 433 (cit. on p. 69).

[37] E. Eade and T. Drummond. "Unified loop closing and recovery for real time monocular SLAM." In: *BMVC*. Vol. 13. Citeseer. 2008, p. 136 (cit. on p. 32).

[38] M. P. Eckstein. "Visual search: A retrospective". In: *Journal of vision* 11.5 (2011), pp. 14–14 (cit. on p. 69).

[39] M. R. Endsley. "Toward a theory of situation awareness in dynamic systems". In: *Human factors* 37.1 (1995), pp. 32–64 (cit. on p. 9).

[40] M. Enzweiler and D. M. Gavrila. "Monocular pedestrian detection: Survey and experiments". In: *IEEE transactions on pattern analysis and machine intelligence* 31.12 (2008), pp. 2179–2195 (cit. on p. 91).

[41] A. J. Fairchild et al. "A Mixed Reality Telepresence System for Collaborative Space Operation". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27 (2017), pp. 814–827 (cit. on p. 43).

[42] J. M. Findlay, I. D. Gilchrist, et al. *Active vision: The psychology of looking and seeing*. 37. Oxford University Press, 2003 (cit. on p. 27).

[43] P. M. Fitts and M. I. Posner. "Human performance." In: (1967) (cit. on pp. 19, 20).

[44] P. M. Fitts, R. E. Jones, and J. L. Milton. "Eye Movements of Aircraft Pilots during Instrument-landing Approaches". In: *Aeronautical Engineering Review* 9.2 (1949), pp. 24–29 (cit. on p. 28).

[45] D. Girardeau-Montaut. "Cloudcompare-open source project". In: *OpenSource Project* 588 (2011) (cit. on p. 64).

[46] R. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587 (cit. on p. 91).

[47] E. B. Goldstein and J. Brockmole. *Sensation and perception*. Cengage Learning, 2016 (cit. on pp. 22, 25, 27).

[48] J. Gordon and I. Abramov. "Color vision in the peripheral retina. II. Hue and saturation". In: *JOSA* 67.2 (1977), pp. 202–207 (cit. on p. 23).

[49] B. Guenter et al. "Foveated 3D Graphics". In: *ACM Transactions on Graphics (TOG)* 31.6 (2012), pp. 1–10 (cit. on pp. 25, 26, 42, 48, 49).

[50] R. Hachiuma et al. "DetectFusion: Detecting and Segmenting Both Known and Unknown Dynamic Objects in Real-time SLAM". In: *arXiv preprint arXiv:1907.09127* (2019) (cit. on p. 35).

[51] A. Handa et al. "A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM". In: *IEEE Intl. Conf. on Robotics and Automation, ICRA*. Hong Kong, China, May 2014, pp. 1524–1531 (cit. on p. 62).

[52] T. Hansen, L. Pracejus, and K. R. Gegenfurtner. "Color perception in the intermediate periphery of the visual field". In: *Journal of vision* 9.4 (2009), pp. 26–26 (cit. on p. 23).

[53] H. Hartridge and L. C. Thomson. "Methods of Investigating Eye Movements". In: *The British Journal of Ophthalmology* 32.9 (1948), p. 581 (cit. on p. 28).

[54] M. Harvey et al. "Manual responses and saccades in chronic and recovered hemispatial neglect: a study using visual search". In: *Neuropsychologia* 40.7 (2002), pp. 705–717 (cit. on p. 69).

[55] K. He et al. "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969 (cit. on p. 35).

[56] J. Hegdé and D. C. Van Essen. "Selectivity for complex shapes in primate visual area V2". In: *Journal of Neuroscience* 20.5 (2000), RC61–RC61 (cit. on p. 25).

[57] A. Hendrickson. "Organization of the Adult Primate Fovea". In: *Macular Degeneration*. Ed. by P. L. Penfold and J. M. Provis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1–23. ISBN: 978-3-540-26977-9. DOI: 10.1007/3-540-26977-0_1. URL: https://doi.org/10.1007/3-540-26977-0_1 (cit. on p. 22).

[58] T. K. Heok and D. Daman. "A review on level of detail". In: *Proceedings. International Conference on Computer Graphics, Imaging and Visualization, 2004. CGIV 2004.* 2004, pp. 70–75. DOI: 10.1109/CGIV.2004.1323963 (cit. on p. 41).

[59] D. D. Hoffman and M. Singh. "Salience of visual parts". In: *Cognition* 63.1 (1997), pp. 29–78 (cit. on p. 93).

[60] J. C. Horton and W. F. Hoyt. "The representation of the visual field in human striate cortex: a revision of the classic Holmes map". In: *Archives of ophthalmology* 109.6 (1991), pp. 816–824 (cit. on p. 22).

[61] E. J. Horvitz and J. Lengyel. "Perception, attention, and resources: A decision-theoretic approach to graphics rendering". In: *arXiv preprint arXiv:1302.1547* (2013) (cit. on p. 42).

[62] C.-F. Hsu et al. "Is Foveated Rendering Perceivable in Virtual Reality?: Exploring the Efficiency and Consistency of Quality Assessment Methods". In: *25th ACM Intl. Conf. on Multimedia*. Oct. 2017, pp. 55–63. DOI: 10.1145/3123266.3123434 (cit. on p. 68).

[63] D. H. Hubel and T. N. Wiesel. "Receptive fields and functional architecture of monkey striate cortex". In: *The Journal of physiology* 195.1 (1968), pp. 215–243 (cit. on p. 25).

[64] E. Huey. *The Psychology and Pedagogy of Reading: With a Review of the History of Reading and Writing and of Methods, Texts, and Hygiene in Reading*. M.I.T. Press paperback series. M.I.T. Press, 1968. ISBN: 9780262580106. URL: https://books.google.com.gh/books?id=NnTlswEACAAJ (cit. on p. 28).

[65] T. Huyen et al. "Impacts of Retina-Related Zones on Quality Perception of Omnidirectional Image". In: *IEEE Access* 7 (Jan. 2019), pp. 166997–167009. DOI: 10.1109/ACCESS.2019.2953983 (cit. on p. 50).

[66] J. Hyönä. "Foveal and Parafoveal Processing during Reading". In: *The Oxford Handbook of Eye Movements*. Ed. by S. P. Liversedge, I. Gilchrist, and S. Everling. Oxford University Press, 2011 (cit. on p. 23).

[67] I. Iehisa et al. "Factors affecting depth perception and comparison of depth perception measured by the three-rods test in monocular and binocular vision". In: *Heliyon* 6.9 (2020), e04904 (cit. on p. 12).

[68] International Telecommunication Union. *Recommendation ITU-T P.919: Subjective test methodologies for 360° video on head-mounted displays*. ITU, 2020 (cit. on pp. 68, 72).

[69] Y. Ishiguro and J. Rekimoto. "Peripheral vision annotation: noninterference information presentation method for mobile augmented reality". In: *Proceedings of the 2nd Augmented Human International Conference*. 2011, pp. 1–5 (cit. on p. 23).

[70] S. Izadi et al. "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*. 2011, pp. 559–568 (cit. on p. 33).

[71] M. Jaimez et al. "Fast odometry and scene flow from RGB-D cameras based on geometric clustering". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 3992–3999 (cit. on p. 35).

[72] Jean-Marc Denis. *From Product Design to Virtual Reality | by Jean-Marc Denis | Google Design | Medium*. Oct. 2015. URL: https://medium.com/google-design/from-product-design-to-virtual-reality-be46fa793e9b#.2w5m9hq6n (visited on 01/25/2022) (cit. on p. 17).

[73] M. Kachouane et al. "HOG based fast human detection". In: *2012 24th International Conference on Microelectronics (ICM)*. IEEE. 2012, pp. 1–4 (cit. on p. 91).

[74] O. Kähler et al. "Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices". In: *IEEE Transactions on Visualization and Computer Graphics* 22.11 (2015) (cit. on p. 34).

[75] M. Kamezaki et al. "A Basic Framework of Virtual Reality Simulator for Advancing Disaster Response Work Using Teleoperated Work Machines". In: *Journal of Robotics and Mechatronics* 26.4 (2014), pp. 486–495. DOI: 10.20965/jrm.2014.p0486 (cit. on pp. 2, 43).

[76] G. Kamsickas. "Future combat systems (FCS) concept and technology development (CTD) phase—Unmanned combat demonstration—Final report". In: *Boeing Company, Seattle, WA, Tech. Rep. D786–1006102* (2003) (cit. on p. 20).

[77] X. Kang et al. "Object-Level Semantic Map Construction for Dynamic Scenes". In: *Applied Sciences* 11.2 (2021), p. 645 (cit. on p. 35).

[78] A. Karpathy, S. Miller, and L. Fei-Fei. "Object discovery in 3d scenes via shape analysis". In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 2088–2095 (cit. on p. 93).

[79] P. M. Kebria et al. "Robust Adaptive Control Scheme for Teleoperation Systems With Delay and Uncertainties". In: *IEEE transactions on cybernetics* (2019) (cit. on p. 20).

[80] M. Keller et al. "Real-Time 3D Reconstruction in Dynamic Scenes Using Point-Based Fusion". In: *2013 International Conference on 3D Vision - 3DV 2013*. 2013, pp. 1–8. DOI: 10.1109/3DV.2013.9 (cit. on p. 34).

[81] M. Keller et al. "Real-time 3D Reconstruction in Dynamic Scenes using Point-based Fusion". In: *2013 International Conference on 3D Vision-3DV 2013*. IEEE. 2013, pp. 1–8 (cit. on p. 33).

[82] L. Keselman et al. "Intel(R) RealSense(TM) Stereoscopic Depth Cameras". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 1267–1276. DOI: 10.1109/CVPRW.2017.167 (cit. on p. 29).

[83] J. Kessenich, G. Sellers, and D. Shreiner. *OpenGL Programming Guide: The official guide to learning OpenGL, version 4.5 with SPIR-V*. Addison-Wesley Professional, 2016 (cit. on p. 36).

[84] B. Keyes et al. "Camera placement and multi-camera fusion for remote robot operation". In: *Proceedings of the IEEE international workshop on safety, security and rescue robotics*. National Institute of Standards and Technology Gaithersburg, MD. 2006, pp. 22–24 (cit. on pp. 18, 19).

[85] M. Q. Khan and S. Lee. "Gaze and Eye Tracking: Techniques and Applications in ADAS". In: *Sensors* 19.24 (2019), p. 5540 (cit. on p. 28).

[86] G. Klein and D. Murray. "Parallel Tracking and Mapping for Small AR Workspaces". In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2007, pp. 225–234. DOI: 10.1109/ISMAR.2007.4538852 (cit. on p. 32).

[87] D. Krupke et al. "Prototyping of Immersive HRI Scenarios". In: *Proc. 20th Intl. Conf. on CLAWAR 2017*. Oct. 2017, pp. 537–544 (cit. on p. 43).

[88] S. Laine and T. Karras. "Efficient sparse voxel octrees". In: *IEEE Transactions on Visualization and Computer Graphics* 17.8 (2010), pp. 1048–1059 (cit. on p. 41).

[89] Q. Li and X. Wang. *Image classification based on SIFT and SVM. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)(pp. 762-765)*. 2018 (cit. on p. 91).

[90] S. Lichiardopol. "A survey on teleoperation". In: *Technische Universitat Eindhoven, DCT report* 20 (2007), pp. 40–60 (cit. on p. 10).

[91] T.-Y. Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755 (cit. on p. 92).

[92] P. Lindstrom and G. Turk. "Image-driven simplification". In: *ACM Transactions on Graphics (ToG)* 19.3 (2000), pp. 204–241 (cit. on p. 41).

[93] J. I. Lipton, A. J. Fay, and D. Rus. "Baxter's Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing". In: *IEEE Robotics and Automation Letters* 3.1 (2018), pp. 179–186 (cit. on p. 43).

[94] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110 (cit. on p. 91).

[95] C. J. Ludwig, J. R. Davies, and M. P. Eckstein. "Foveal analysis and peripheral selection during active visual sampling". In: *Proceedings of the National Academy of Sciences* 111.2 (2014), E291–E299 (cit. on p. 69).

[96] D. Luebke et al. *Perceptually driven simplification using gaze-directed rendering*. Tech. rep. Tech. Rep. CS-2000-04, Department of Computer Science, University of ..., 2000 (cit. on p. 42).

[97] B. Luo et al. "Parallax360: Stereoscopic 360 scene representation for head-motion parallax". In: *IEEE transactions on Visualization and Computer Graphics* 24.4 (2018), pp. 1545–1553 (cit. on pp. 42, 43).

[98] T. Madl, B. J. Baars, and S. Franklin. "The timing of the cognitive cycle". In: *PloS one* 6.4 (2011), e14803 (cit. on p. 20).

[99] A. Maimone and H. Fuchs. "Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras". In: *10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE. 2011, pp. 137–146 (cit. on p. 43).

[100] P. Majaranta and A. Bulling. "Eye Tracking and Eye-Based Human–Computer Interaction". In: *Advances in Physiological Computing*. Ed. by S. H. Fairclough and K. Gilleade. London: Springer London, 2014, pp. 39–65. ISBN: 978-1-4471-6392-3. DOI: 10.1007/978-1-4471-6392-3_3. URL: https://doi.org/10.1007/978-1-4471-6392-3_3 (cit. on p. 28).

[101] J. McDonald et al. "Real-time 6-DOF multi-session visual SLAM over large-scale environments". In: *Robotics and Autonomous Systems* 61.10 (2013). Selected Papers from the 5th European Conference on Mobile Robots (ECMR 2011), pp. 1144–1158. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2012.08.008. URL: https://www.sciencedirect.com/science/article/pii/S0921889012001406 (cit. on pp. 32, 33).

[102] M. Meehan et al. "Effect of Latency on Presence in Stressful Virtual Environments". In: *IEEE Virtual Reality, 2003. Proceedings.* 2003, pp. 141–148 (cit. on p. 2).

[103] R. Mekuria, K. Blom, and P. Cesar. "Design, implementation, and evaluation of a point cloud codec for tele-immersive video". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.4 (2016), pp. 828–842 (cit. on p. 60).

[104] S. Messelodi, C. M. Modena, and G. Cattoni. "Vision-based bicycle/motorcycle classification". In: *Pattern recognition letters* 28.13 (2007), pp. 1719–1726 (cit. on p. 91).

[105] O. Miksik and V. Vineet. "Live Reconstruction of Large-Scale Dynamic Outdoor Worlds". In: *arXiv preprint arXiv:1903.06708* (2019) (cit. on p. 35).

[106] P. Milgram and J. Ballantyne. "Real World Teleoperation via Virtual Environment Modeling". In: *Intl. Conf. on Artificial Reality and Tele-existence (ICAT'97)*. Dec. 1997 (cit. on p. 42).

[107] A. Mossel and M. Kröter. "Streaming and Exploration of Dynamically Changing Dense 3D Reconstructions in Immersive Virtual Reality". In: *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2016, pp. 43–48 (cit. on p. 43).

[108] A. Naceri et al. "The *Vicarios* Virtual Reality Interface for Remote Robotic Teleoperation". In: *Journal of Intelligent & Robotic Systems* 101.80 (2021) (cit. on pp. 6, 9, 43, 44).

[109] G. Narita et al. "PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things". In: *arXiv preprint arXiv:1903.01177* (2019) (cit. on p. 35).

[110] E. National Academies of Sciences and Medicine. *Learning from the Science of Cognition and Perception for Decision Making: Proceedings of a Workshop*. Ed. by S. J. Debad. Washington, DC: The National Academies Press, 2018. ISBN: 978-0-309-47634-8. DOI: 10.17226/25118. URL: https://www.nap.edu/catalog/25118/learning-from-the-science-of-cognition-and-perception-for-decision-making (cit. on p. 11).

[111] Y. Nehmé et al. "Comparison of Subjective Methods, with and without Explicit Reference, for Quality Assessment of 3D Graphics". In: *ACM Symposium on Applied Perception 2019*. SAP '19. Barcelona, Spain: Association for Computing Machinery, 2019. ISBN: 9781450368902. DOI: 10.1145/3343036.3352493. URL: https://doi.org/10.1145/3343036.3352493 (cit. on p. 68).

[112] R. A. Newcombe, D. Fox, and S. M. Seitz. "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 343–352 (cit. on p. 35).

[113] T. Nguyen et al. "Object detection using scale invariant feature transform". In: *Genetic and evolutionary computing*. Springer, 2014, pp. 65–72 (cit. on p. 91).

[114] F. Nilsson. *Upper body ergonomics in virtual reality: An ergonomic assessment of the arms and neck in virtual environments*. 2017 (cit. on p. 16).

[115] B. Olk et al. "Measuring visual search and distraction in immersive virtual reality". In: *Royal Society open science* 5.5 (2018), p. 172331 (cit. on pp. 69, 70).

[116] O. Olmos, C. D. Wickens, and A. Chudy. "Tactical displays for combat awareness: An examination of dimensionality and frame of reference concepts and the application of cognitive engineering". In: *The International Journal of Aviation Psychology* 10.3 (2000), pp. 247–271 (cit. on p. 19).

[117] C. R. Olson. "Object-based vision and attention in primates". In: *Current opinion in neurobiology* 11.2 (2001), pp. 171–179 (cit. on p. 27).

[118] L. E. Ortiz, E. V. Cabrera, and L. M. Gonçalves. "Depth data error modeling of the ZED 3D vision sensor from stereolabs". In: *ELCVIA: electronic letters on computer vision and image analysis* 17.1 (2018), pp. 0001–15 (cit. on pp. 29, 31, 32).

[119] S. Orts-Escolano et al. "Holoportation: Virtual 3D Teleportation in Real-Time". In: *29th Annual Symposium on User Interface Software and Technology (UIST)*. Tokyo, Japan: Association for Computing Machinery, 2016, pp. 741–754. ISBN: 9781450341899 (cit. on pp. 2, 43).

[120] E. Oyama et al. "Robots for Telexistence and Telepresence: from Science Fiction to Reality". In: () (cit. on p. 10).

[121] E. Palazzolo et al. "ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals". In: *IEEE/RSJ IROS*. 2019. URL: https://www.ipb.uni-bonn.de/pdfs/palazzolo2019iros.pdf (cit. on p. 62).

[122] C. Papadopoulos and A. E. Kaufman. "Acuity-driven gigapixel visualization". In: *IEEE transactions on visualization and computer graphics* 19.12 (2013), pp. 2886–2895 (cit. on p. 42).

[123] L. Peppoloni et al. "Immersive ROS-integrated Framework for Robot Teleoperation". In: *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. 2015, pp. 177–178 (cit. on p. 43).

[124] H. Pfister et al. "Surfels: Surface elements as rendering primitives". In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000, pp. 335–342 (cit. on pp. 41, 52, 95).

[125] S. Pheasant and C. M. Haslegrave. *Bodyspace: Anthropometry, ergonomics and the design of work*. CRC press, 2018 (cit. on pp. 16, 17).

[126] J. Pokorny. "Steady and pulsed pedestals, the how and why of post-receptoral pathway separation". In: *Journal of Vision* 11.5 (2011), pp. 7–7 (cit. on p. 24).

[127] M. Quigley et al. "ROS: an open-source Robot Operating System". In: *IEEE ICRA Workshop on Open Source Software*. IEEE. 2009 (cit. on p. 43).

[128] N. Quinn et al. "The clinical relevance of visualising the peripheral retina". In: *Progress in Retinal and Eye Research* 68 (2019), pp. 83–109. ISSN: 1350-9462. DOI: https://doi.org/10.1016/j.preteyeres.2018.10.001. URL: https://www.sciencedirect.com/science/ARTICLE/pii/S1350946218300399 (cit. on p. 23).

[129] J. Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788 (cit. on p. 91).

[130] E. Rosen et al. "Mixed Reality as a Bidirectional Communication Interface for Human-Robot Interaction". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 11431–11438. DOI: 10.1109/IROS45743.2020.9340822 (cit. on pp. 2, 43).

[131] M. Runz, M. Buffier, and L. Agapito. "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects". In: *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2018, pp. 10–20 (cit. on pp. 35, 94).

[132] M. Rünz and L. Agapito. "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 4471–4478 (cit. on p. 35).

[133] S. Rusinkiewicz and M. Levoy. "QSplat: A multiresolution point rendering system for large meshes". In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000, pp. 343–352 (cit. on p. 41).

[134] R. B. Rusu and S. Cousins. "3D is here: Point Cloud Library (PCL)". In: *2011 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2011, pp. 1–4 (cit. on pp. 43, 55, 98).

[135] M. Sainz and R. Pajarola. "Point-based rendering techniques". In: *Computers & Graphics* 28.6 (2004), pp. 869–879 (cit. on p. 41).

[136] C. Sanders. *Practical packet analysis: Using Wireshark to solve real-world network problems*. No Starch Press, 2017 (cit. on p. 65).

[137] S. P. Schipani et al. *Quantification of cognitive process degradation while mobile, attributable to the environmental stressors endurance, vibration, and noise*. Tech. rep. ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD, 1998 (cit. on pp. 18, 20).

[138] T. Schöps, T. Sattler, and M. Pollefeys. "SurfelMeshing: Online Surfel-Based Mesh Reconstruction". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.10 (2020), pp. 2494–2507 (cit. on pp. 33, 34).

[139] E. R. Schotter, B. Angele, and K. Rayner. "Parafoveal processing in reading". In: *Attention, Perception, & Psychophysics* 74.1 (2012), pp. 5–35 (cit. on p. 23).

[140] R. K. Scoggins, R. Machiraju, and R. J. Moorhead. *Enabling level-of-detail matching for exterior scene synthesis*. IEEE, 2000 (cit. on p. 41).

[141] R. Scona et al. "StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1–9 (cit. on p. 35).

[142] W. R. Sherman and A. B. Craig. "Chapter 3 - The Human in the Loop". In: *Understanding Virtual Reality (Second Edition)*. Ed. by W. R. Sherman and A. B. Craig. Second Edition. The Morgan Kaufmann Series in Computer Graphics. Boston: Morgan Kaufmann, 2018, pp. 108–188. ISBN: 978-0-12-800965-9. DOI: https://doi.org/10.1016/B978-0-12-800965-9.00003-9. URL: https://www.sciencedirect.com/science/ARTICLE/pii/B9780128009659000039 (cit. on p. 25).

[143] D. J. Simons and R. A. Rensink. "Change blindness: past, present, and future". In: *Trends in Cognitive Sciences* 9.1 (2005), pp. 16–20. ISSN: 1364-6613. DOI: https://doi.org/10.1016/j.tics.2004.11.006. URL: http://www.sciencedirect.com/science/article/pii/S1364661304002931 (cit. on p. 12).

[144] M. J. Simpson. "Mini-review: Far peripheral vision". In: *Vision Research* 140 (2017), pp. 96–105. ISSN: 0042-6989. DOI: https://doi.org/10.1016/j.visres.2017.08.001. URL: https://www.sciencedirect.com/science/ARTICLE/pii/S0042698917301657 (cit. on p. 23).

[145] C. C. Smyth. "Indirect vision driving with fixed flat panel displays for near unity, wide, and extended fields of camera view". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 44. 36. SAGE Publications Sage CA: Los Angeles, CA. 2000, pp. 541–544 (cit. on p. 19).

[146] R. Snowden et al. *Basic vision: an introduction to visual perception*. Oxford University Press, 2012 (cit. on p. 24).

[147] J.-P. Stauffert, F. Niebling, and M. E. Latoschik. "Latency and Cybersickness: Impact, Causes, and Measures. A Review". In: *Frontiers in Virtual Reality* 1 (2020), p. 31. ISSN: 2673-4192 (cit. on p. 2).

[148] N. Stein et al. "A comparison of eye tracking latencies among several commercial head-mounted displays". In: *i-Perception* 12.1 (2021), p. 2041669520983338 (cit. on p. 28).

[149] M. Stengel et al. "Adaptive Image-space Sampling for Gaze-contingent Real-time Rendering". In: *Computer Graphics Forum* 35.4 (2016), pp. 129–139 (cit. on p. 42).

[150] J. Stenning. *Direct3D Rendering Cookbook*. Packt Publishing Ltd, 2014 (cit. on p. 36).

[151] P. Stotko et al. "A VR System for Immersive Teleoperation and Live Exploration with a Mobile Robot". In: *IEEE/RSJ IROS*. Nov. 2019, pp. 3630–3637. DOI: 10.1109/IROS.2012.6386012 (cit. on pp. 2, 43).

[152] H. Strasburger, I. Rentschler, and M. Jüttner. "Peripheral Vision and Pattern Recognition: A Review". In: *Journal of Vision* 11.5 (2011), pp. 13–13 (cit. on pp. 23, 25, 26, 48, 49).

[153] M. Strecke and J. Stückler. "EM-Fusion: Dynamic Object-Level SLAM with Probabilistic Data Association". In: *arXiv preprint arXiv:1904.11781* (2019) (cit. on p. 35).

[154] S. Suzuki and R. Suda. "A vision system with wide field of view and collision alarms for teleoperation of mobile robots". In: *ROBOMECH Journal* 1.1 (Sept. 2014), p. 8. ISSN: 2197-4225. DOI: 10.1186/s40648-014-0008-5. URL: https://doi.org/10.1186/s40648-014-0008-5 (cit. on p. 19).

[155] S. Tachi et al. "Tele-existence (I): Design and evaluation of a visual display with sensation of presence". In: *Theory and Practice of Robots and Manipulators*. Springer, 1985, pp. 245–254 (cit. on p. 10).

[156] K. Tateno, F. Tombari, and N. Navab. "Real-time and scalable incremental segmentation on dense slam". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 4465–4472 (cit. on p. 93).

[157] B. W. Tatler et al. "Yarbus, eye movements, and vision". In: *i-Perception* 1.1 (2010), pp. 7–27 (cit. on p. 28).

[158] Y. Tefera et al. "Towards Foveated Rendering For Immersive Remote Telerobotics". In: *The International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions at HRI*. 2022 (cit. on p. 6).

[159] M. Theofanidis et al. "VARM: Using Virtual Reality to Program Robotic Manipulators". In: *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA 2017*. 2017, pp. 215–221 (cit. on p. 43).

[160] P. Trebuňa, M. Mizerák, and L. Rosocha. "3D Scanning—Technology and Reconstruction". In: *Acta Simulatio* 4 (2018), pp. 1–6 (cit. on p. 29).

[161] J. B. Van Erp and P. Padmos. "Image parameters for driving with indirect viewing systems". In: *Ergonomics* 46.15 (2003), pp. 1471–1499 (cit. on p. 19).

[162] O. Vila et al. "A Method to Compensate for the Errors Caused by Temperature in Structured-Light 3D Cameras". In: *Sensors* 21.6 (2021). ISSN: 1424-8220. DOI: 10.3390/s21062073. URL: https://www.mdpi.com/1424-8220/21/6/2073 (cit. on p. 30).

[163] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. Ieee. 2001, pp. I–I (cit. on p. 91).

[164] J. Wang, M. Lewis, and S. Hughes. "Gravity-referenced attitude display for teleoperation of mobile robots". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 48. 23. SAGE Publications Sage CA: Los Angeles, CA. 2004, pp. 2662–2666 (cit. on pp. 18, 19).

[165] M. Weier. "Perception-driven rendering: techniques for the efficient visualization of 3D scenes including view-and gaze-contingent approaches". In: (2019) (cit. on pp. 41, 42).

[166] M. Weier et al. "Foveated Real-Time Ray Tracing for Head-Mounted Displays". In: *Computer Graphics Forum* 35 (Oct. 2016), pp. 289–298. DOI: 10.1111/cgf.13026 (cit. on p. 68).

[167] M. Weinmann et al. "Immersive VR-based Live Telepresence for Remote Collaboration and Teleoperation". In: *Wissenschaftlich-Technische Jahrestagung der DGPF*. Stuttgart, Germany, 2020, pp. 391–399 (cit. on p. 43).

[168] F. W. Weymouth. "Visual Sensory Units and the Minimal Angle of Resolution". In: *American Journal of Ophthalmology* 46.1 (1958), pp. 102–113 (cit. on pp. 25, 26, 48, 49).

[169] T. Whelan et al. "Real-time Large scale Dense RGB-D SLAM with Volumetric Fusion". In: *International Journal of Robotics Research (IJRR)* 34.4-5 (2015), pp. 598–626 (cit. on pp. 53, 96).

[170] T. Whelan et al. "ElasticFusion: Dense SLAM without a Pose Graph". In: *Robotics: Science and Systems*. 2015 (cit. on pp. 33, 52, 53, 97).

[171] D. Whitney et al. "ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots". In: *Proc. IEEE IROS 2018*. Oct. 2018, pp. 5018–5025 (cit. on p. 43).

[172] C. D. Wickens and K. S. Seidler. "Information Access in a Dual-Task Context: Testing a Model of Optimal Strategy Selection". In: *Journal of Experimental Psychology: Applied* 3.3 (1997), pp. 196–215. ISSN: 1076898X. DOI: 10.1037/1076-898X.3.3.196. URL: https://doi.org/10.1037//1076-898X.3.3.196 (cit. on p. 11).

[173] N. Williams et al. "Perceptually guided simplification of lit, textured meshes". In: *Proceedings of the 2003 symposium on Interactive 3D graphics*. 2003, pp. 113–121 (cit. on p. 42).

[174] J. M. Wolfe, P. O'Neill, and S. C. Bennett. "Why are there eccentricity effects in visual search? Visual and attentional hypotheses". In: *Perception & psychophysics* 60.1 (1998), pp. 140–156 (cit. on p. 69).

[175] D. D. Woods et al. "Envisioning human-robot coordination in future operations". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 34.2 (2004), pp. 210–218 (cit. on p. 17).

[176] B. Xu et al. "Mid-fusion: Octree-based object-level multi-instance dynamic SLAM". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 5231–5237 (cit. on pp. 35, 36).

[177] X. Xu et al. "A comparison of koniocellular, magnocellular and parvocellular receptive field properties in the lateral geniculate nucleus of the owl monkey (Aotus trivirgatus)". In: *The Journal of physiology* 531.1 (2001), pp. 203–218 (cit. on p. 24).

[178] G. Yang et al. "Keep Healthcare Workers Safe: Application of Teleoperated Robot in Isolation Ward for COVID-19 Prevention and Control". In: *Chinese Journal of Mechanical Engineering* 33.47 (2020). DOI: https://doi.org/10.1186/s10033-020-00464-0 (cit. on pp. 1, 9).

[179] X. Yang et al. "Developing a semantic-driven hybrid segmentation method for point clouds of 3D shapes". In: *IEEE Access* 8 (2020), pp. 40861–40880 (cit. on p. 93).

[180] A. L. Yarbus. *Eye Movements and Vision*. Germany: Springer Berlin/Heidelberg, 1967 (cit. on p. 28).

[181] Z. Zhang. "Microsoft Kinect Sensor and Its Effect". In: *IEEE MultiMedia* 19.2 (2012), pp. 4–10. DOI: 10.1109/MMUL.2012.24 (cit. on p. 29).

[182] L. Zhaoping. "The V1 hypothesis-creating a bottom-up saliency map for preattentive selection and segregation". In: *Understanding vision: theory, models, and data* (2014), pp. 189–314 (cit. on p. 28).

[183] M. Zollhöfer et al. "State of the Art on 3D Reconstruction with RGB-D Cameras". In: *Computer Graphics Forum* 37.2 (2018), pp. 625–652 (cit. on p. 29).

A

Change of basis

This section provides the coordinate transformation process. Consider an nxn Unreal matrix M_u and think of it as the standard representation of a transformation $TM_u : \mathbf{R^n} \rightarrow \mathbf{R^n}$. If we pick a different basis $v1,...,vn$ of R^n , what matrix C represents TM_u with respect to Opengl's Coordinates M_g. Looking at OpenGl from Unreal's point of view we could say that:

- $+X_g$ corresponds to $+Y_u$
- $+Y_g$ corresponds to $-Z_u$
- $+Z_g$ corresponds to $+X_u$

This translates in the following matrix (columns "represent" X, Y, Z):

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

The rotation matrix from Unreal to Opengl can be retrieved by the by using the formula defined in eq A.2:

$$M_g = CM_uC^{-1} \tag{A.2}$$