INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Volumetric Video Streaming

DUARTE FIGUEIREDO MARQUES
Setembro de 2024

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Volumetric Video Streaming

Duarte Marques

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Graphic Systems and Multimedia

Advisor: Dr. Nuno Pereira

Porto, September 18, 2024

Statement of Integrity

I hereby declare having conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore the work presented in this document is original and authored by me, having not previously been used for any other end.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, September 18, 2024

Abstract

The evolution of digital media consumption has led to innovative ways to ingest its content. In this thesis, on the research and state-of-the-art topic, we explore the concept of volumetric video streaming, a type of media that can be seen from any position and angle in 6 degrees of freedom navigation, giving a total state of immersion.

The work starts with deep research into existing capture systems with an array of hundreds of cameras down to a single camera system from various fields (sports, meetings, music, fashion, health care, entertainment, and gaming). How the end-user can consume the content is also a topic of concern. Present devices, from head mount displays (HDMI), such as the Oculus Rift, to specialized volumetric displays, are analyzed, and examples are given to help understand how they work and their advantages and disadvantages. Finally, for the streaming aspect, as the volumetric content is vastly more extensive than traditional "flat" media, methods to arrange volumetric data, such as Point Clouds and Volumetric Meshes, are explored, as well as how to compress them.

The practical section of the thesis was focused on designing, implementing, and evaluating a volumetric video streaming system. Various potential solutions were explored and tested, leading to the development of an optimized system capable of capturing and analyzing real-time performance metrics. To validate the system, experiments were conducted across three scenarios: a static camera with no movement, a static camera with regular movement, and a moving camera with extreme movement. The captured data were analyzed to assess the system's stability and performance under different conditions. The findings highlighted key challenges and provided insights for future improvements, confirming the feasibility of the proposed system and offering a foundation for further research in volumetric video streaming.

Keywords: Volumetric Video, Volumetric Streaming, Video Capture

Resumo

A evolução do consumo de conteúdo digital levou a fortes inovações de consumo do mesmo. Nesta tese, exploramos o streaming de vídeos volumétricos, um tipo de média que pode ser visto de qualquer posição e ângulo com uma navegação de 6 graus de liberdade, proporcionando um estado total de imersão. A análise começa com uma pesquisa aprofundada nos sistemas de captura existentes, desde um conjunto de centenas de câmeras até um sistema de uma câmera apenas, focados em diversos campos (desportos, reuniões, música, moda, saúde, entretenimento e jogos). Onde o utilizador pode consumir o conteúdo também é uma preocupação do tópico. Dispositivos atuais, desde HMDs (como o Oculus Rift) até displays volumétricos especializados são analisados e exemplos dados para entender como funcionam, bem como suas vantagens e desvantagens. Finalmente, para o streaming, visto que o conteúdo volumétrico é consideravelmente maior do que os meios tradicionais "2D", são explorados métodos para organizar dados volumétricos, como Point Clouds e Volumetric meshes e como comprimir estes tipos de formatos.

A outra secção da tese centra-se no design, implementação e avaliação de um sistema de streaming de vídeo volumétrico. Foram exploradas e testadas várias potenciais soluções, o que levou ao desenvolvimento de um sistema otimizado, capaz de capturar e analisar métricas de desempenho em tempo real. Para validar o sistema, foram realizados testes em três cenários: uma câmara estática sem movimento, uma câmara estática com movimento regular, e uma câmara em movimento com movimento extremo. Os dados capturados foram analisados para avaliar a estabilidade e o desempenho do sistema em diferentes condições. Os resultados destacaram os principais desafios e forneceram informações para futuras melhorias, confirmando a viabilidade do sistema proposto e oferecendo uma base para pesquisas futuras no streaming de vídeo volumétrico.

Contents

List of Figures

List of Tables

List of Acronyms

2D	Two Dimensions.
3D	Three Dimensions.
3DoF	Three Degrees of Freedom.
6DoF	Six Degrees of Freedom.
ALU	Arithmetic logic unit.
API	Application Programming Interface.
AR	Augmented Reality.
BPS	Bits per second.
CPU	Central Processing Unit.
CSV	Comma-Separated Values.
CUDA	Compute Unified Device Architecture.
FOV	Field of View.
FPS	Frames per Second.
FVV	Free Viewpoint Video.
G-PCC	Geometry-Based Point Cloud Compression.
GBPS	Gigabytes per Second.
GPU	Graphics Processing Unit.
HMD	Head-Mount Display.
MB	Megabytes.
MPEG	Moving Picture Experts Group.
MPEG-PCC	MPEG Point Cloud Compression.
ntp	Network Time Protocol.
OpenGL	Open Graphics Library.
POV	Point of View.
PTCL	Point Cloud.
PTP	Precision Time Protocol.
SDK	Software Development Kit.
V-PCC	Video-Based Point Cloud Compression.
VMAF	Video Multi-method Assessment Fusion.

VOXEL	Volumetric Pixel.
VR	Virtual Reality.

Chapter 1

Introduction

This first chapter will provide a context for the topic, concisely define volumetric videos, and explore their initial challenges. We will then define this thesis's main question and how to answer it within smaller individual objectives. Finally, we will explain the work plan for achieving all the proposed goals.

1.1 Context

What if the lines between the physical and digital realms blur? Imagine videos that capture what's in front of the camera and the entire 3D space around it. Step into a world where movies are no longer confined to flat screens and where your favorite characters come to life in three dimensions right before your eyes. This is the captivating realm of volumetric videos, a groundbreaking technology reshaping how we experience visual content. With volumetric videos, the boundaries of traditional video are shattered, paving the way for a new era of storytelling and interactive experiences.

Volumetric videos, also referred to as Free Viewpoint Video (FVV), are three-dimensional (3D) media that can be used for Virtual Reality (VR) and Augmented Reality (AR) content [1]. It allows the creation of a unique immersive experience that can have multiple applications. Education, employee training, and marketing are only some fields that could benefit from it, allowing users to learn more realistic material [2].

However, this tool also comes with challenges. Volumetric video capture uses complex setups with its multi-view capture system that, depending on the number of cameras and sensors used, can add significant complexity [1]. Furthermore, the data needed to produce a realistic volumetric model video is significantly greater than that required for capturing a traditional flat video or image [3]. The processing power needed to generate such content is another crucial factor contributing to the complexity of this technology, thus becoming an obstacle to becoming more widely used by the general public.

Thus, the preliminary question, which we will refine during the thesis work, is: "What are the workflow and requirements for practical volumetric capture and streaming?". To answer this question and grasp the concept of volumetric videos, we can explore a recent technology developed and launched by Intel in 2018.

They launched a platform, True View, that allowed viewers to watch a game from every angle, creating an immersive game experience. This technology relied on multiple cameras arranged around the stadium (figure 1.1) and required high-performance servers to store, synchronize, analyze, and render terabytes of volumetric data. This data was then stored as voxels (three-dimensional pixels) needed to form a point-cloud representation. Available to

watch on several platforms, such as television, VR headsets, computers, and mobile phones, this project set a significant mark in volumetric video capture [4].

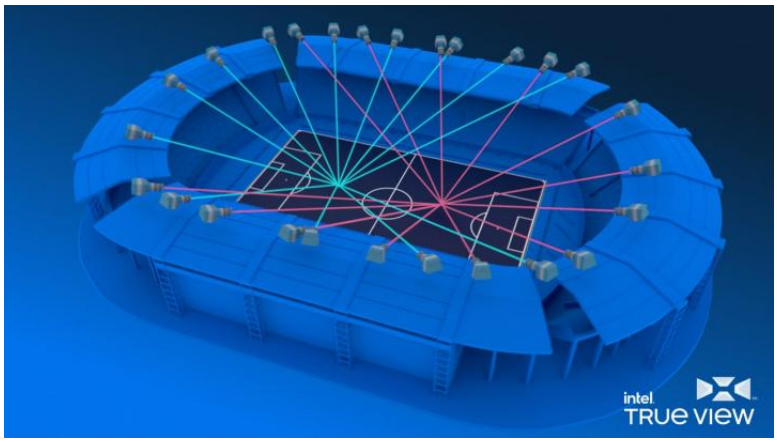

Figure 1.1: Cameras array for the Intel True View platform

Apart from systems similar to the one above that use an array of generic cameras, there are also cameras with depth sensors that can create a volumetric video with a single device. An example of that is the ZED 2 (figure 2.9), a versatile stereo camera for spatial perception developed and commercialized by Stereolabs [5]. It can detect and track objects with spatial context and uses neural networks to reproduce human vision. A further breakdown of this device can be found in subsection 2.4.

On the streaming topic, this type of video implicates a series of tasks, namely sending, decoding, and rendering. The speed of these tasks and the data transmission are bounded by the network capacity and the performance of the display device for decoding and rendering the volumetric data. Traditional streaming techniques for conventional flat videos do not directly apply to volumetric ones because of the fundamental difference between the 3D frames, i.e., the lack of regularity of data format and the large data size. As shown in the study in [6], uncompressed volumetric video can have a single frame with the size of 4 megabytes (MB) to more than 15MB, and a whole stream will require from 1 gigabyte per second (GBPS) to 3.6GBPS.

1.2 Objectives

To answer the preliminary question stated above in 1.1, some objectives were formulated:

- **Defining volumetric video:** This is the first step to achieving the goal. To that end, a study on the main concepts of this type of media will be conducted, as well as how it distinguishes itself from conventional "flat" content and eventually understanding the key differences between volumetric and 3D videos. Exploring its current applications and use cases is a must to understand this technology and where its thriving fields are located, such as sports, music, fashion, and entertainment.
- **Studying the known techniques:** There are multiple ways to capture, stream, and display volumetric videos. Studying the existing capture systems and cameras to record volumetric content will help structure a test plan. The methods of live streaming it on the web and its behavior are also of concern. Finally, unique volumetric displays are required for users to consume volumetric content.

- **Design and build a system to perform streaming tests:** With all the state-of-the-art research, we will require a system to evaluate the technology. The system must be able to run a functional live stream of volumetric content. We will also need to consume this stream and construct the volumetric space.
- **Evaluate the videos:** To get valid readable results, we must correctly evaluate the system under different techniques and conditions. A streaming pipeline includes many factors, including capturing systems, data formats, and compression. To get valid results, a testing system with different scenarios is mandatory.

1.3 Project Approach

The first stage of this thesis involved extensive research on volumetric video streaming. This entailed an in-depth study of all state-of-the-art technologies and methodologies to understand the topic comprehensively. Specifically, the research focused on the intricacies of capturing, streaming, and displaying volumetric content, which formed the foundation for the subsequent stages of the project. It is important to note that state-of-the-art was the target of change due to continuous research on the topic.

The next phase involved designing and developing a volumetric capture system tailored to perform the necessary research tests and experiments. This process was particularly challenging, requiring meticulous attention to detail to ensure the system could support the project's experimental needs. The design had to be functional and flexible enough to accommodate various test scenarios and modifications.

The project's most labor-intensive and technically demanding part was testing different streaming pipelines and evaluating their performance. This phase explored various frames per second (FPS), bandwidth requirements, and bit-rate usage. A difficulty lay in identifying technologies that met the specific requirements for this project while also providing valid and readable metrics for evaluation.

Finally, the thesis concludes with a comprehensive analysis of the findings, identifying an ideal pipeline for volumetric video streaming. This conclusion will summarize the results of the experiments and tests and provide insights into the most effective approaches and technologies for handling volumetric content. This work, while challenging, has contributed valuable knowledge to the field and will serve as a reference for future research and development efforts in volumetric video streaming.

1.4 Structure

In this section, we present an overview of the thesis structure, outlining the content and purpose of each chapter to provide a roadmap for the reader.

Chapter 1 discusses the project's context and outlines the research problem, objectives, and scope. This chapter sets the stage for the entire thesis by justifying the research. It introduces the main investigation questions and hypotheses and explains the significance of the study. Additionally, we review relevant literature to frame the research within the existing body of knowledge and identify gaps that this thesis aims to address.

Chapter 2 starts by detailing the research methodologies used throughout the project. Afterward, we will offer a background on volumetric video streams. A generic "What is a

volumetric video" starts the literature review, passing into examples of uses of this technology in various fields. The chapter comprehensively compares existing theories, models, and frameworks related to volumetric video streams. We critically analyze previous studies and highlight their contributions, limitations, and implications. This chapter positions our research within the broader academic discourse, demonstrating how it builds on and diverges from prior work. By comparing and contrasting various approaches, we establish a theoretical foundation that informs the design and methodology of our project.

Chapter 3 details and discusses the choices for designing and implementing the project. This chapter includes the initial design requirements and constraints, system architecture, technology stack, and the implementation process. We explain the motivation behind key decisions and describe the development environment, tools, and techniques used. This chapter also covers the iterative refinement of the design based on testing, highlighting the challenges faced and the solutions adopted.

Chapter 4 presents all of our research's test results. It includes a detailed analysis of the collected data accompanied by tables, figures, and charts to illustrate key results. We interpret the data in the context of the research questions and hypotheses, highlighting significant patterns, trends, and insights. This chapter aims to clearly and concisely present the empirical evidence gathered during the study.

In Chapter 5, we conclude the thesis by summarizing the main findings and contributions of the research. We reflect on the research objectives and assess whether they have been achieved. This chapter also outlines the study's limitations and suggests directions for future research. We emphasize the value and relevance of our work by highlighting its broader implications and potential applications.

1.5 Ethical Considerations

This document was prepared with a commitment to ethical considerations, aiming to uphold academic principles of honesty and integrity. The following section discusses the most pertinent ethical considerations.

- **Generative artificial intelligence (AI):** Sections of this thesis have been rewritten with the help of AI tools for correctness, clarity, and engagement purposes. Only the tools' built-in capabilities were used, and all outputs were validated.
- **Data privacy and sensitive information:** As this project focuses on an experimental system, there isn't any relevant sensitive data to be disclosed.

Chapter 2

State of the Art

This second chapter will handle research on every end of volumetric video streaming. The research methodology is the first section explored. What were the author's methods of researching the available data? Then, an in-depth definition of volumetric video is given alongside a streaming pipeline with all the required steps. To enhance the reader's knowledge of the topic, relevant use cases are analyzed with their respective systems to capture the content. After that, different capture and display systems will be explored. Lastly, streaming is investigated, including the formats in which volumetric video can be stored and how to compress it.

2.1 Research Methodology

The methods used when searching for materials about the topic can be grouped into three parts: Researching, Filtering, and Choosing.

When researching topics, Content Analysis was used alongside Snowball Sampling. This first studies documents and communication articles in various formats (text, pictures, audio, and video) [7]. It is best used and understood with a broad family of techniques. According to Klaus Krippendorff, six questions must be addressed in every content analysis [8]:

1. Which data is analyzed?
2. How is the data defined?
3. From what population is the data from?
4. What is the relevant context?
5. What are the boundaries of the analysis?
6. What is to be measured?

Snowball Sampling's primary goal is to gather as many relevant samples as possible. This method takes a starting individual and is asked to name k different individuals. Each named individual is then asked to name k more and so forth [9]. In the scope of this thesis, the first individual was a series of documents that best could answer the research questions. From there, those document's references were analyzed, thus creating a snowball effect.

Some preliminary filters were applied to filter the information gathered above. As volumetric video is a recent technology, newer documents should have more relevant information. Their citations could be considered a good practice as more citations usually mean a more appropriate study. However, these two should and were only used as initial filter techniques.

For more in-depth filtering, questions 1 and 4 of the Klaus Krippendorff list[8] were also considered.

With the documents and filter applied, the chosen documents were the ones that best answered the research questions, using the remainder of the questions.

2.2 Volumetric Video

As previously stated, volumetric video is a 3D media that can be used for VR and AR. It consists of video capture with multiple cameras and sensors to record a subject and create a full-volume recording rather than a flat image. Through post-production, this captured data becomes a volumetric video, which is viewable from any point of view (POV) [10].

A key aspect of volumetric video is its capability to freely move in any direction, thus making it truly three-dimensional. They consist of elements such as voxels (volumetric pixels) or 3D meshes (polygons), achieving six degrees of freedom (6DoF). This means that the POV can be freely changed by the position (X, Y, Z) and by the rotation (yaw, pitch, roll) [11]. On the other hand, regular and 360° video content is constructed for the flat or curved 2D experience. They consist of only 2D pixels, thus considered only to have three degrees of freedom (3DoF). They also don't support POV freedom, as the viewer's transnational position is permanently fixed, meaning the user can only rotate through yaw, pitch, or roll. An example comparing both degrees can be seen in figure 2.1.

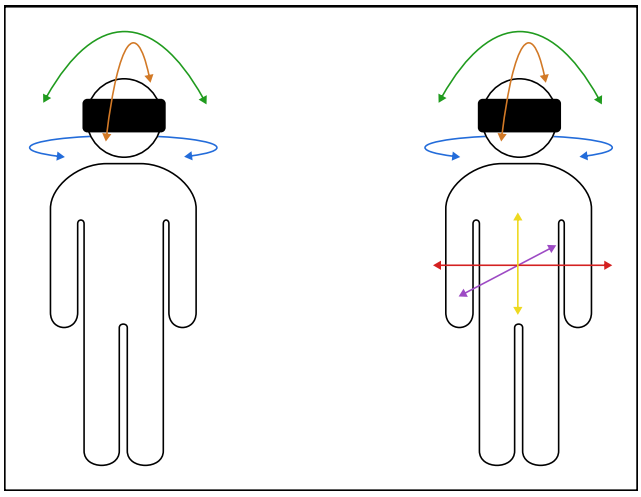

Figure 2.1: 3DoF (left) and 6DoF (right) comparison

Volumetric video can be broken down into a 3-step pipeline as shown in figure 2.2. The first step, capture, can be achieved with a system of cameras or a single one if it has an incorporated depth sensor. For streaming, the previously recorded content must be in a suitable video format and encoded before the transmission. Finally, there are several options for displaying the volumetric video.

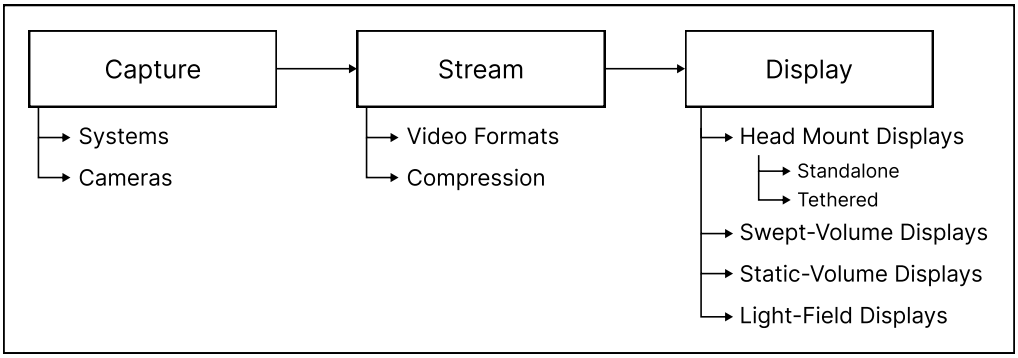

Figure 2.2: Volumetric Video Streaming Pipeline

2.3 Applications Examples

This section will explore some fields in which volumetric video is used. As already explored in section 1.1, Intel True view is an example of its use in sports. Another example, specifically from NBA, is described below in section 2.3.1. It also explores real applications for regular meetings, video calls, music, and fashion.

2.3.1 NBA Volumetric Video Broadcast

On March 16, 2022, NBA broadcasted a game in a 3D format, resorting to volumetric video. The viewers can view the entire game as if they were there through ESPNews or ESPN Plus platforms. This format, denominated "NBA CourtView" or "Netaverse", was accomplished by capturing the scene using 110 Canon technology cameras around the court. This technique lets the operator change angles on the fly. It also allows the camera seemingly free 360-degree movement as if it were a drone cam flying around the stadium [12]. Although this project seemed to have some glitches, it displayed a realistic future use for volumetric video capture and streaming.

2.3.2 Google's Project Starline

In 2021, Google first introduced Project Starline, a project that conveyed a new way for people to communicate with each other. This project aimed to have people meet in a video conference-like style, wherein the participants would be presented with a realistic, 3D representation of the person they are communicating with (figure 2.3) instead of the conventional 2D flat image, allowing for a more engaging communicational experience [13]. In May 2023, Google introduced an updated prototype for the project, involving less complexity in the volumetric capture of the subject to reproduce its 3D model. This new prototype only requires a few standard cameras, going "from the size of a restaurant booth to a flat-screen TV", highlighting the remarkable progress in volumetric video capture technology [14].

2.3.3 Radiohead - House of Cards

The music video for the single "House of Cards" by Radiohead was produced in partnership with Google, without any camera or light. To capture the 3D footage, they used a structured light scanner from Geometric Informatics [15] for the close-up shots of the singers and a Velodyne LIDAR[16] scanner for the landscapes. The LIDAR uses 64 lasers, shooting 900

Figure 2.3: Google's Project Starline Live Render

times a minute to capture larger scenes [17]. It then creates an XYZ point cloud of data rendered and read by 3D software [18] as seen in figure 2.4.

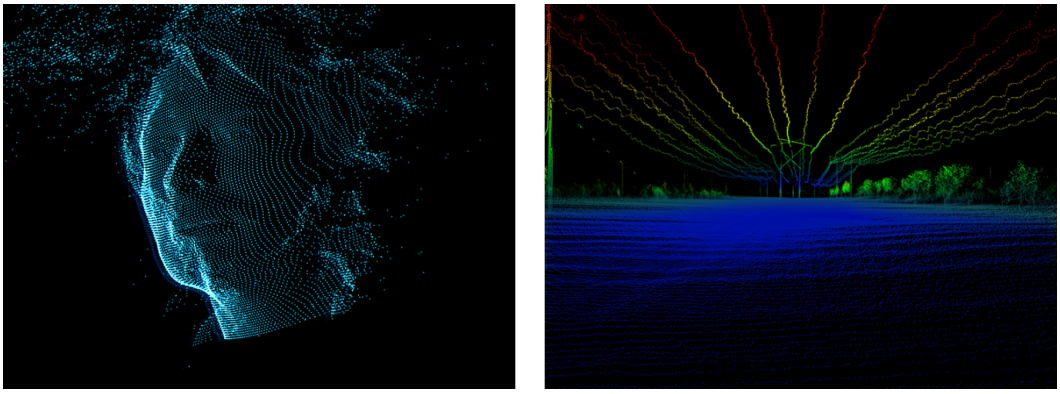

Figure 2.4: Radiohead vocalist Thom Yorke (left) and power lines (right) from video clip

2.3.4 Fashion Innovation Agency

This agency collaborated with Portsmouth University's Centre for Creative and Immersive Extended Reality (CCIXR) to create a 3D fashion film using volumetric capture (figure 2.5). This approach used a 4DViews Holosys system with 32 4K resolution cameras to capture high-detail volumetric video [19], up to 60 frames per second. The resulting footage enables designers to create more engaging, interactive, and accessible fashion experiences showcasing the garments while accurately representing movement, texture, and shape [20].

2.4 Capture

Nowadays, many technologies and diverse system setups exist to capture volumetric videos. These systems are generally very complex, requiring precise positioning, synchronization, and calibration of the hardware components to capture data accurately. Some of the previously mentioned examples use one of two methods. The first includes a dome setup, distributing

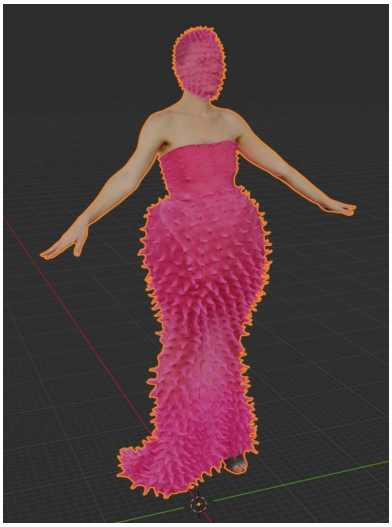

Figure 2.5: FIA x Portsmouth CCIXR capture model

many cameras in a dome-like structure to surround the intended subject, capturing its geometrical information. For example, we have the HOLOSYS volumetric video capture system as referred to in subsection 2.3.4. This system uses 32 4K cameras recording 60 frames per second, as seen below in figure 2.6. Its recording and storage capacity is 110 minutes, at 30 frames per second (FPS) for the first and up to 30 hours of volumetric data for the latter. On December 18, Woodkid used this system to record a song's live performance and broadcast it on German television [21].

Figure 2.6: HOLOSYS System Setup

Canon, a well-known camera manufacturer, developed another system consisting of a studio with 8x8x3.50 meters of capture volume with a green screen background, as seen in figure 2.7. It has 100 specialized 4K cameras recording at 60 frames per second. It can also capture up to 10 people simultaneously, get an immediate view of the captured images as free-viewpoint video, and explore the camera path due to high-speed processing of extensive

volume data [22].

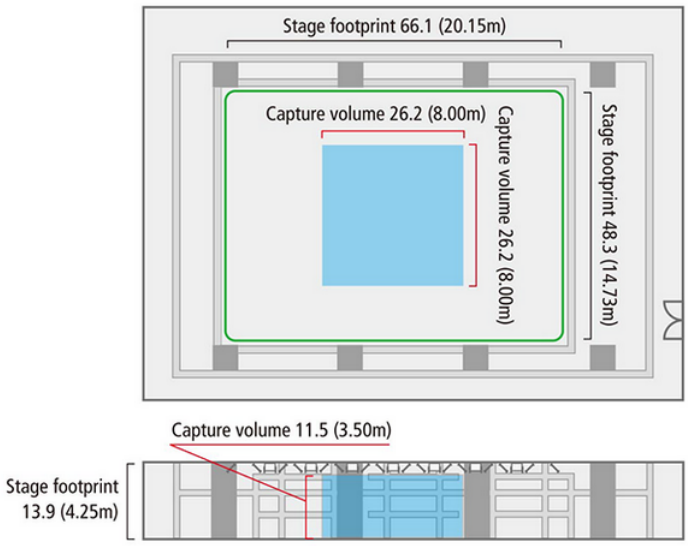

Figure 2.7: Canon Studio top and side views

A more detailed example of a system set up to capture volumetric video would be the one described in [1], where it aims to reduce the complexity of the capture and streaming process using the Intel RealSense technology. In this study, it was implemented a set of 4 modules, designated as "eyes" (figure 2.8), each composed of an RGB-D (red, green, blue, and depth) Intel RealSense D415 sensor and an Intel NUC processing unit. These modules were then connected to a LAN switch by Ethernet cables, which the "client" was also connected to. The processing units in the acquisition modules aim to alleviate the computational burden associated with compression and pre-processing tasks, which aims to minimize complexity on the receiving client's end.

Figure 2.8: Capture system referenced in [1]

There is also specialized hardware for capturing 3D media where one or more cameras are placed in front of the subject with sensors to estimate the scene's depth, which can create

a 3D visualization of the scene in post-production. However, it may encounter limitations in viewpoint selection and can face challenges related to occlusion [1].

Taking the ZED camera in figure 2.9 (left) and exploring it further, it is designed to replicate how human vision works. Using its two “eyes” and triangulation, the ZED provides a three-dimensional understanding of the observed scene, allowing your application to become space and motion-aware. Its key features range from depth sensing to plane detection. ZED offers different resolutions, from VGA to 2K and 15 to 100 FPS. It has a gyroscope, accelerometer, magnetometer, barometer, and temperature sensors. To interact with ZED camera modules, Stereolabs provides a software development kit (SDK) [23].

Another example of such hardware could be the Azure Kinect technology (figure 2.9, center), the successor of the Microsoft Kinect camera. It is constituted by a 1-MP depth sensor with wide and narrow field-of-view (FOV) options and 12-MP RGB video for an additional color stream aligned to the depth stream. It has a built-in 7-microphone array for far-field speech and sound capture that helps optimize and build advanced computer vision and speech models. Accelerometer and gyroscope for sensor orientation and spatial tracking are also on the Kinect, and it has an external sync pin to synchronize sensor streams from multiple Kinect devices easily [24].

A less user-friendly camera is the LiDAR, which stands for Light Detection and Ranging. It works by emitting a large amount of narrow beams of near-infrared light with circular/elliptical cross-sections, which reflect off of objects in their trajectories and return to the detector of the LiDAR sensor. Due to the characteristics of its short wavelength, it can detect even the most miniature objects and create exact 3D models of them. Figure 2.9 (right) shows an example of this device.

Figure 2.9: Cameras examples. ZED 2 (left), Azure Kinect (center) and Velodyne LiDAR (right).

2.5 Display

Volumetric videos are becoming increasingly available across various platforms and applications. The most common are the VR/AR head-mount displays (HMDs). This technology allows total or partial user immersion through a multisensory computer-generated 3D virtual representation of real environments. Based on the concept of the Reality-Virtuality Continuum, which classifies all modern technologies of "realities" (Augmented Reality, Mixed Reality, etc.) according to the degree of "virtuality", VR technology, in general, has the most significant degree of virtuality. This means that VR provides the highest level of immersion compared to other technologies [25].

Modern VR HMDs can be classified into standalone and tethered [25]. The key aspect differentiating these displays is that standalone VR contains all required hardware for computing, processing, and displaying on the device without needing to physically connect to any external hardware. They usually have integrated graphics processing units (GPU) and embedded

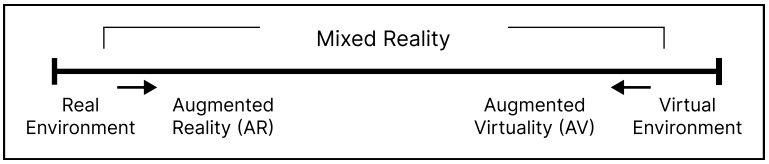

Figure 2.10: Reality-Virtuality Continuum

sensors to calculate the exact position and orientation in space to determine the correct user perspective. This characteristic makes standalone HMDs much more mobile-oriented, yet it also has drawbacks. Firstly, the device's processing power limits the performance. Increasingly demanding applications may turn these devices obsolete. Their battery life is also a concern as the need to keep recharging the device negatively impacts the user's immersive experience [25].

Tethered displays only have a viewing screen and sensors; all processing power is handled on a separate computer. This results in performance that is only limited by the external hardware. The main compromise of these displays is the need for constant physical connection to another device, reducing the user's immersion and comfort level.

Additionally, it is worth noting that newer hybrid devices, such as the Oculus Quest developed by Meta, support both standalone and tethered architectures, allowing users to utilize the device in their preferred way.

A study conducted in [25] compares five tethered HMDs regarding their display, tracking capabilities, controller functions, and ergonomic aspects. In the following table, we only show 3 of them (table 2.1).

Table 2.1: Comparative Analysis of different HMDs.

Device	**Oculus Rift S**	**HTC Vive Cosmos**	**Valve Index**
Display Type	LCD	LCD	LCD
Resolution (per eye)	1280x1440	1440x1700	1440x1600
Field of View	90º	110º	130º
Pixel Density (pixel/degree)	11.63	13.09	11.07
Refresh Rate	80Hz	80/90/120/144Hz	90Hz
Tracking Type	Positional	Positional	Positional
Method	Inside-out Markless	Inside-out Markless	Inside-out Marker-based
Technology	Camera-based (5x)	Camera-based (6x)	IR laser emitting beacons (2x)
Movement Type	6DoF	6DoF	6DoF
Weight	561g	645g	809g
Additional Features	AI-Assisted tracking	Addons	Finger tracking

There also are different types of display that don't require an HMD. Starting with Swept-Volume displays, rotating emissive or reflective screens, including illuminated spinning paddles, spinning LEDs, or translating projection surfaces, are commonly used. As an example, we can take a look at Voxon VX1, which, as stated on the product technical document [26], is technically described as a "swept surface volumetric display" and is powered by the Voxon Photonics Engine. This engine comprises an ultra-high-speed digital projection system, a central processing unit (CPU), a volumetric graphics engine, and a reciprocating light diffuser. Volumetric images are created by projecting slices of light at 4,000 FPS onto a moving screen to diffuse at precisely the correct position in physical space. Through the persistence

of vision, the human eye blends the slices, and the result is an actual three-dimensional digital object that can be viewed in the same way as one would view a real object, from any angle, and without special goggles or glasses.

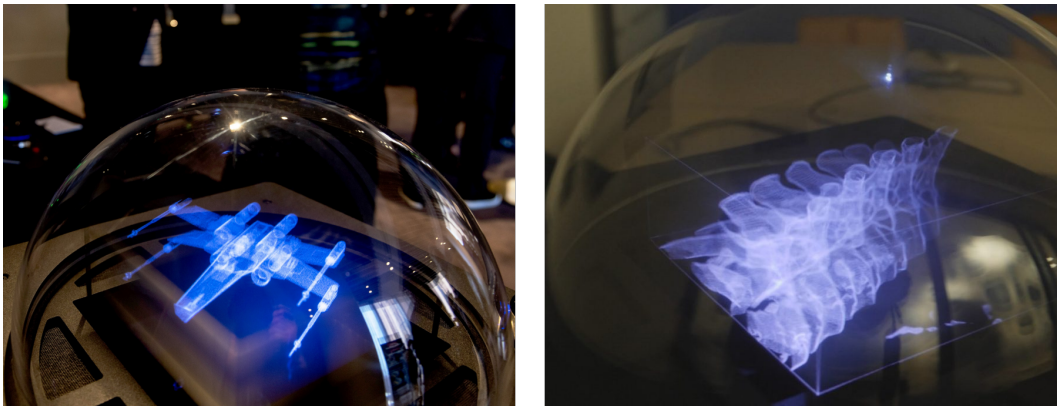

Figure 2.11: Voxon VX1 render examples. A Star Wars X-Wing spaceship (left) and a human spine (right)

Another type is the static-volume displays, which form images by upconversion in nonlinear gases or solids or by projecting onto several diffusing planes [27]. Looking at the research conducted by Rochester University [28], a device was built consisting of a glass box surrounding an airtight glass sphere about the size of a globe that heats to approximately 70 degrees Celsius (158 degrees Fahrenheit). The sphere contains cesium vapor, a silvery-gold metal good at emitting light. Two laser beams with invisible wavelengths are crossed in the sphere. Where both lasers illuminate the laser beams cross, cesium atoms and are excited into an especially high energy state. When these atoms decay, they emit sky-blue light in all directions, as seen in figure 2.12.

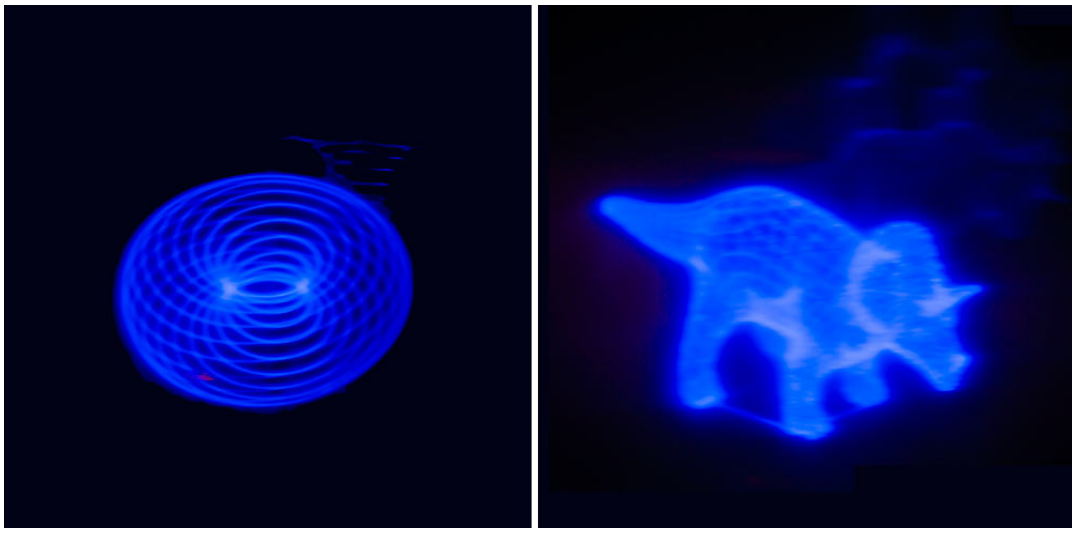

Figure 2.12: Examples of images displayed in true 3-D (University photo / J. Adam Fenster)

One exciting example is the Looking Glass Factory Light-Field display. To produce volumetric content, Looking Glass provides up to 100 discrete views of a 3D scene and presents them over a view cone roughly 58° wide. This arrangement of views tricks the visual perception

system into seeing 3D objects in two significant ways: By changing the user's aspect on the scene as they move their head around it (parallax) and by presenting different perspectives to each eye (stereo vision). We can easily detect the "magic" behind this display in the animation shown in [29]. It is possible to notice the different frames for each angle by looking at it. In figure 2.13 (right), it is possible to see the frame number (green number to the right of the screen). This makes it so that as your head moves around, you don't discretely hop from one view to the next but instead cross-fade amongst many, making the visual experience more fluid and natural. The only downside is the introduction of blur.

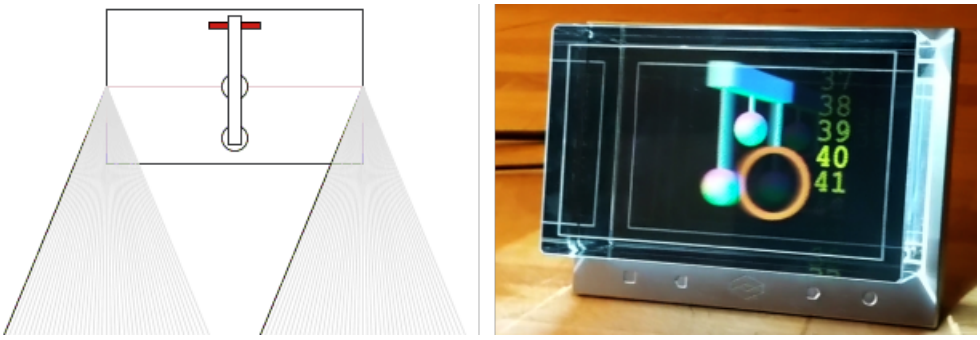

Figure 2.13: Looking Glass angles (left) and frame on an animation (right)

A final example is a 3D laptop. The technology, developed by the manufacturer Dimenco, is called Spatial Vision. It uses eye-tracking cameras on the top bezel of the computer that can precisely follow your face to create a perfect 3D image for the user without any HMD. Moreover, recent models even have a third camera connected to the laptop that can track the use of a stylus pen, allowing interaction with the volumetric objects on the scene [30].

Figure 2.14: Dimenco 3D Laptop with Stylus pen

2.6 Streaming

The transmission of volumetric media requires an immense amount of data to accurately represent the depth and realism of the content. As a result, volumetric video streaming becomes more complex than traditional video streaming.

2.6.1 Data Formats

After capturing an object, individual, or scene, the content must be stored in a specific form to be later represented. The most common form for volumetric videos is Point Clouds (PtCl). They represent the data using a set of individual points in space with attributes such as coordinates, color, and intensity representing the subject or scene. This method has become famous for being more accessible to process and manipulate. However, because it lacks information on the surfaces of the captured scene, it can lead to a less realistic representation [3]. Another method for transmitting volumetric content is through meshes consisting of vertices and edges. It faces that interconnected form triangles or polygons, allowing for a more accurate and detailed representation of the scene. Hence, polygonal meshes must preserve the connectivity information, which brings some limitations during compression and transmission, leading to higher bandwidth requirements, as shown in the study in [31]. It concludes that textured meshes provide the best visualization and are more advantageous for high-bitrate bandwidth (e.g., over 50 Mbps) applications. In contrast, the point clouds are well-suited for applications with limited bandwidth (e.g., below 20 Mbps) [31].

2.6.2 Compression

When discussing the streaming of volumetric video, it is essential to acknowledge that it involves large amounts of data to faithfully represent 3D objects and scenes. To minimize this factor, various methods exist for compressing and transmitting the 3D data from one end to another. Most existing ones resort to octree-based or k-dimensional (k-d) tree-based compression. The first is a tree data structure in which each internal node has precisely eight children, each recursively subdividing it into another eight octants. The latter is a binary tree on which every node is a k-dimensional point. Every non-leaf node recursively splits the plane into two parts, known as half-spaces. The left sub-tree of that node represents points to the left of this plane, and the right sub-tree represents points to the right of the hyperplane. Examples of these methods can be seen in figure 2.15.

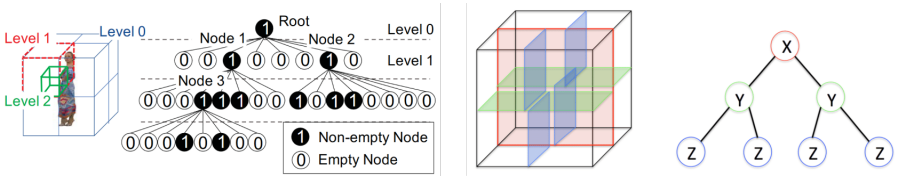

Figure 2.15: Octree (left) and k-d (right) data structures

A study in [32] compares different point cloud compression and decompression performance of open-source libraries. They used Draco from Google (k-d), Point Cloud Library (octree), and Limited Error Point Cloud Compression. The results show that the highest bit rate among the used videos is around 180 Mbps, even after compression.

Another compression method was developed by the Moving Picture Experts Group (MPEG) using PtCl, called MPEG Point Cloud Compression (MPEG-PCC). Due to the wide range of applications for volumetric content, MPEG-PCC standardization created three categories of PtCl. Static (many details, millions to billions of points), dynamic (fewer point locations, with temporal information), and dynamically acquired (millions to billions of points, colors, surface normal and reflective properties attributes) [33]. In early 2020, MPEG published the final PCC standards consisting of two solutions classes. Video-based (V-PCC) is appropriate

for point sets with a relatively uniform distribution of points, and Geometry-based (G-PCC) is appropriate for more sparse distributions. The same study shown in 2.6.1 also concludes that V-PCC is more effective than G-PCC compression [31].

2.7 Evaluation

Video quality is a critical aspect of multimedia services and applications, encompassing various methodologies to ensure that video content meets the desired visual and auditory fidelity standards. Different techniques and frameworks used in video evaluation have their applications and advancements in the field. These techniques can be qualified as subjective, when there is only human intervention in the evaluation process, or objective, when only precise mathematic calculations are used to get a final result. There can also be a blend of those, resulting in a method that combines the key aspects of both techniques. Figure 2.16 shows a diagram showing different known evaluation techniques and their category.

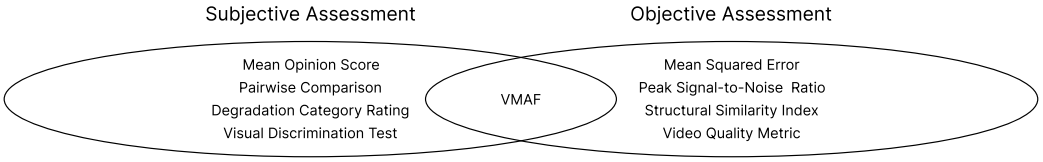

Figure 2.16: Evaluation Techniques Diagram

About subjective video quality assessment. It involves human viewers to evaluate the video content. Standard techniques include Mean Opinion Score, where viewers rate video quality on a predefined scale (for example, from one to five), and pairwise comparison, where viewers compare two video sequences and select the one they perceive to be higher quality [34]. Despite its accuracy, subjective assessment is time-consuming, expensive, and not scalable for large datasets.

Another approach is objective assessment, which uses algorithms to automatically evaluate the quality of a video without human intervention, which is ideal for large-scale applications. As stated in [35], the simplest and most widely used full-reference quality metric is the mean squared error MSE), computed by averaging the squared intensity differences of distorted and reference image pixels, along with the related quantity of peak signal-to-noise ratio. These are appealing because they are simple to calculate, have precise physical meanings, and are mathematically optimal.

Finally, and probably the best strategy, hybrid models are used. They combine subjective and objective assessments to leverage the strengths of both approaches. A prominent and presumably the most popular is Netfix's video Multi-method Assessment Fusion (VMAF). It integrates human vision modeling and machine learning to predict subjective quality scores. At its core, VMAF operates by fusing various quality metrics through a machine-learning model trained on a comprehensive database of video clips. These clips are annotated with subjective quality scores from human observers. The algorithm evaluates multiple aspects of video quality, including spatial detail, temporal smoothness, and visual artifacts, and synthesizes these evaluations to produce a single VMAF score [36]. This score ranges from 0 to 100, with higher values indicating better-perceived quality.

The table below (table 2.2) lists some of the known techniques to evaluate videos with a small description associated with each.

Table 2.2: Video evaluation techniques.

Technique	Type	Description
Mean Opinion Score (MOS)	Subjective	Evaluators rate the video quality on a scale (e.g., 1-5) where one is poor and five is excellent. The average score from all evaluators provides the MOS, offering an overall measure of perceived video quality.
Pairwise Comparison (PC)	Subjective	Participants compare two videos side-by-side and choose the better quality. This method directly compares two versions of the same video or different encoding techniques.
Degradation Category Rating (DCR)	Subjective	Viewers are shown a reference video followed by a test video with potential degradation. They rate the difference in quality, focusing on the perceived amount of degradation.
Visual Discrimination Test (VDT)	Subjective	This technique assesses how well viewers can distinguish between different levels of video quality. Evaluators indicate whether they perceive a difference between a reference and a test video.
Mean Squared Error (MSE)	Objective	MSE computes the squared differences between corresponding pixel values of the reference and distorted videos. It sums up all these squared differences and then takes the average.
Peak Signal-to-Noise Ratio (PSNR)	Objective	Measure the ratio between the maximum possible signal power and the noise introduced by compression or transmission. Higher PSNR values indicate better quality but may not always correlate well with human perception.
Structural Similarity Index (SSIM)	Objective	Evaluate the structural similarity between the reference and distorted videos by comparing luminance, contrast, and structure. SSIM is designed to better reflect human perception compared to PSNR.
Video Quality Metric (VQM)	Objective	Measures video quality degradation by analyzing features like blurring, noise, and color distortion. VQM is widely used in the telecommunications industry and often correlates well with subjective evaluations.
Video Multi-Method Assessment Fusion (VMAF)	Mixed	A machine-learning-based metric combine several quality metrics like SSIM and temporal motion analysis to predict video quality as humans perceive it. It is widely adopted in streaming services.

Chapter 3

Design & Implementation

This third chapter discusses the choices for designing and implementing the project. This section serves as a comprehensive guide throughout the different stages of development, highlighting the methodologies, technologies, and strategies used to achieve the project objectives. It also describes the code base that serves as a pillar of the system. This dive explains the algorithms and metrics used to evaluate the system. By examining the theoretical and practical considerations, we aim to provide a clear and coherent narrative of the design process.

3.1 Design

3.1.1 Requirements

To start, we need to design a system that can ingest the feed from a camera with depth sensors and stream it in real time through a local network. The same system must then capture that signal and build the volumetric scene. While this is happening, to evaluate the quality of the video, it is also required to record both feeds and calculate metrics. An example of a system with these requirements would resemble the diagram in figure 3.1.

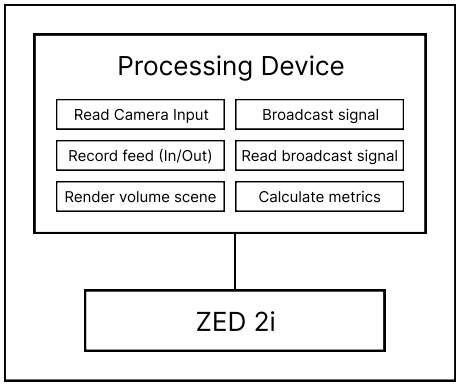

Figure 3.1: First Architecture Diagram

In this approach, the processing device is responsible for every task within the project (decode camera, sender, and receive signal). This leads to an increased workload on the device that may lead to performance issues, as this project proved. Due to hardware limitations (specifications can be seen in section 3.3), this system was discarded, and a new one started to be designed. The new system split the processing and computing duties into two devices (figure 3.2).

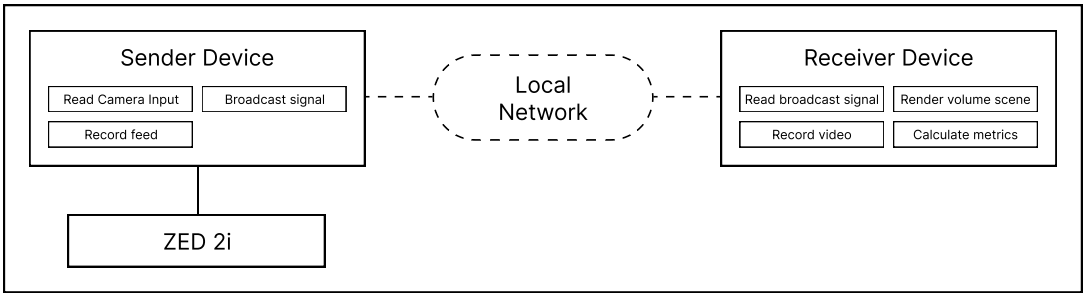

Figure 3.2: Second Architecture Diagram

Another required design is related to the metrics we will use to evaluate the system. The most basic but still relevant are the FPS, the frequency at which consecutive images are captured or displayed, and stream latency, the time it takes for a packet of data to travel from a source to a destination. Higher FPS and lower latency result in a more stable and smoother broadcast. Bandwidth is also a valid and essential metric for evaluating our system. As explored in 2.6, volumetric content requires much more data than traditional videos. Knowing that it's relevant to keep track of the bandwidth (data transfer capacity of a network in bits per second, Bps) usage, we can compare which system settings require more network capacity. The final metric is related to the quality of the video itself. Comparing the volumetric video before and after the broadcast is pertinent to understanding the amount of corrosion that affects the original video during the broadcast.

A final variable for the system is related to the possible scenarios in which a volumetric video can be broadcast. It could be streaming a still image or with low movement, a presentation, such as live news or an interview, where there are short movements, or, in the extreme case, streaming a sports event where all the players are in constant movement, and the need for higher reception is required.

3.1.2 Software

To apply all the requirements and answer the questions above, we started by opting for a ZED camera by Stereolabs. One of the reasons for this choice was the ease of access to a camera, as the supervisor loaned it. This camera comes with a dedicated SDK (Software Development Kit) designed to leverage the capabilities of their cameras. Those are known for their high-resolution depth sensing spatial awareness and subsequent ease of use. Its key features include an end-to-end spatial perception platform for human-like sensing capabilities to achieve the best accuracy in all use cases. Reduce development time with their comprehensive, ready-to-use hardware and software. User-friendly, intuitive integrations, and well-documented APIs. To use this SDK, it's mandatory to have an NVidea GPU (any GPU from another manufacturer will not be able to run it). Figure 3.3 shows a functional diagram of this SDK.

Another requirement is to install CUDA (Compute Unified Device Architecture). CUDA is a parallel computing platform and application programming interface (API) model created by NVIDIA [37]. It provides a development environment that allows for the creation of high-performance, GPU-accelerated applications. It was introduced to allow developers to use GPUs' massive processing power for non-graphics tasks. Its architecture includes a unified shader pipeline, which allows a program intending to perform general-purpose computations to marshal every arithmetic logic unit (ALU) on the chip. Because NVIDIA intended this

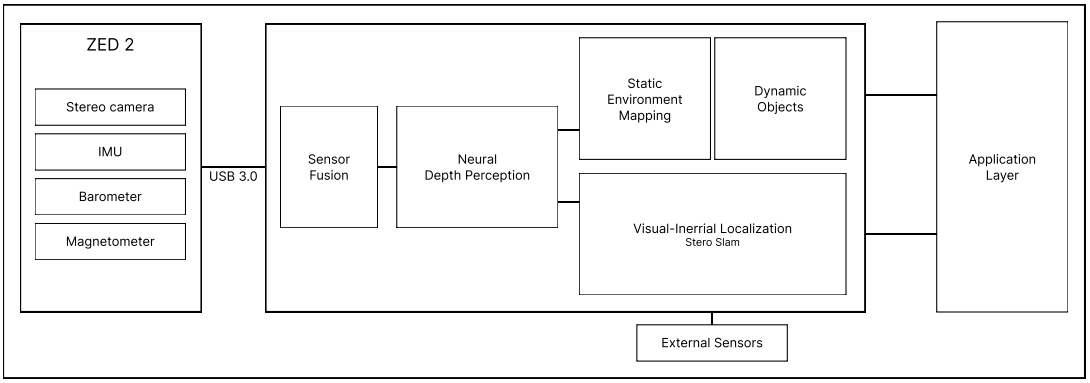

Figure 3.3: Functional ZED SDK

new family of graphics processors for general-purpose computing, these ALUs were built to comply with IEEE requirements for single-precision floating-point arithmetic.

Stereolabs recommends that users use the Open Graphics Library (OpenGL) to render a view field for the volumetric content. It provides a comprehensive and versatile API for rendering 2D and 3D vector graphics. Since its inception in early 1992 by Silicon Graphics Inc., OpenGL has been instrumental in advancing computer graphics hardware and software, influencing various industries, including gaming, virtual reality, and scientific visualization [38].

To backbone all of these technologies, we opted to code in C++ as it is super compatible with ZED, OpenGL, and CUDA. C++ is a well-known, robust, high-performance programming language used in various fields, from systems programming and game development to real-time simulations and high-frequency trading. From the book of its creator, Bjarne Stroustrup: "C++ programming language provides a model of memory and computation that closely matches that of most computers. (...) Thus, C++ supports styles of programming that rely on fairly direct manipulation of hardware resources to deliver a high degree of efficiency, (...)" [39].

3.1.3 Evaluation Metrics

Regarding evaluation metrics, latency is probably one of the most crucial factors when talking about live streaming. To calculate the difference between sender and receiver, we first tried to use Precision Time Protocol (PTP), an IEEE standard [40], which is a method to synchronize the clock of multiple devices on an Ethernet network. This is achieved by setting one device as the master clock and having all other devices synchronize and periodically adjust to the master clock. The devices handle synchronization automatically once the master and slave devices are set. Unfortunately, we could not go with this approach due to limitations as it requires hardware timestamps from the GPU, and the current one of the computers does not provide them. As an alternative, we decided to use the Network Time Protocol (NTP), an internet protocol that synchronizes with computer click timer sources in a network.

FPS variation is also significant as their primary purpose is to verify the variation between the camera and senders (where the FPS are defined). By analyzing its values, we may differentiate scenarios with more or less movement.

Another used metric is bandwidth consumption. Bandwidth specifically refers to the capacity at which a network can transmit data. For example, if the bandwidth of a network is 100 Mbps, it implies that the network cannot transmit data faster than 100 Mbps in any given case. What we want to calculate and evaluate here is the bandwidth required for our volumetric video stream. To calculate it, we used InterFace STATistics (Ifstat for short). This tool provides a method to report network interface activity [41]. For the last metric, we compared video records between sender and receiver devices using the previously explored VMAF (section 2.7).

The table below (table 3.1) describes all used metrics with a small description resuming this topic.

Table 3.1: Metrics used to evaluate the system.

Metric	**Description**
FPS	Determines how smoothly video or media plays. Higher FPS leads to smoother motion, improving the viewing experience, while lower FPS can cause stuttering or lag.
Latency	Measures the delay before data is transmitted or displayed. Lower latency ensures quicker response times, critical for real-time applications like streaming. High latency can cause noticeable delays, affecting the user experience by making media feel sluggish or unresponsive.
Bandwidth	Amount of data that can be transmitted over a network in a given time. Higher bandwidth allows for faster and higher-quality media streaming, as it supports larger data transfers. Insufficient bandwidth can lead to buffering, lower video quality, and interruptions in media playback.
VMAF	Measures video quality by combining human perception with computational models. It's important because it accurately represents how viewers perceive video quality, helping optimize compression and streaming without sacrificing visual experience. Higher VMAF scores mean better-perceived video quality.

3.2 Implementation

In the previous chapter, we explored the technologies used to answer all requirements of this project, and with that, we can now proceed to explain the architecture used. The diagram 3.2 shows that the system comprises three devices. The ZED 2i camera, the sending and receiving computers (sending/receiving the live stream feed). The sender device has the ZED camera connected via a USB cable, processes its input, and sends its signal on a live feed through an IP address over the local network. The receiver device, in turn, captures and reads the feed. After processing its signal, it can represent the depth map of the camera POV. In the subsection below (3.2.1), we take a deeper dive into all the processes of the sender device.

3.2.1 Sender

The sender device connects to the ZED camera, which will handle the live streaming feed. It must set its starting parameters (resolution, codec, fps, bitrate, and port) to get the camera to run. A function to parse the running arguments was created to handle these parameters. It receives the number of arguments, its array, streaming parameters, initial parameters, and a record boolean flag. From there, it loops through all elements and checks if their values match any of the existing ones. The ones available can be seen in the list below.

- H264/H265, sets the streaming codec to the selected one
- HD2K/HD1080/HD720/VGA, sets the camera resolution
- -b <number>, sets the stream bitrate
- -p <number>, sets the IP address port
- -fps <number>, sets the camera recording FPS
- -r, records the camera capture

The initial parameters and streaming parameters are updated on each match. For parameters requiring a number input, an extra verification step is needed to ensure that there is a number to be processed and to skip the value of the next iteration.

Having all the arguments parsed and loaded into the camera, we can open it and start the live stream. We check the return message from opening the ZED and the live stream. If any of those do not match a success state, the runtime is aborted due to errors. Then, if the record flag (-r) was set on input, we start the recording process. Here, we take the same approach as before. Listen to the return message when enabling the recording and check if it's a success, only adding the output file name.

A usage example of this part of the project can be seen below, using all available flags:

```
/sender H265 HD1080 -b 8000 -p 5200 -fps 30 -r
```

3.2.2 Receiver

Things get interesting on the receiver. As you may have understood, the sender device does nothing extraordinary. It just captures the camera and streams it. On this project side, we consume the live feed, build a viewer, and display the depth map. The first step is similar to the previous part. We start by parsing the running command values. The approach was identical to the senders when parsing the arguments. We check for the recording flag and the resolution. We used this code base to test all the camera specifications and parameters. As it is possible to see, we can receive the camera input in three different forms. We can ingest a previously recorded scene in our code as an SVO file (Stereolabs file format for a recording). As a stream, with or without specifying a port (by default, zed uses port 30000) or directly from the camera, thus checking the camera resolution parameter.

After having all the arguments loaded into the camera and starting it, we go through some steps to calculate the resolution of the GLViewer window. For that, we grab the resolution sent by the live feed and calculate the image aspect ratio *(width/height)* and the minimum possible resolution for the GL viewer. This is needed because we still want an acceptable viewport for the screen with a lower resolution. This step ensures a minimum of 720 pixels width.

Then, we can grab the CUDA stream, which is not to be confused with the stream from the ZED. This one allows efficient operations that execute in issue order on the GPU. All previous calculations are injected into the GL Viewer initialization to load the viewing port. The last step before going through the main loop of retrieving the volumetric images is to create an output file for our evaluation metrics, which will be explored below in 3.2.3. We make a CSV (comma-separated values) file to store the transmission FPS and latency.

For the main loop, which runs while the camera is connected and the viewer (GL Viewer) is available, we use the ZED *grab* function to grab the current frame and see its returned state to check if any error occurred. If any did, we just went straight to the next frame, and so on. If no errors appear, we use the native ZED function that allows us to retrieve the measurement of the depth map and display it on the viewer. The only other operation in the main loop is the calculations for the evaluation metrics. We grab the current FPS directly from the ZED and get the latency by calculating the difference between the current frame timestamp and the timestamp of the same frame when the senders send it. Finally, we save these values into the output file, and when we close the viewer, we stop the recording, clear the point cloud, and close the camera.

3.2.3 Evaluation Metrics

At this point, we are already successfully streaming a volumetric video. We need to evaluate our system using some metrics to continue this experiment. From what we have seen in subsection 3.2.2, we capture the camera's current fps and calculate its latency in the receiver main loop. Both these values are essential for our metrics, the latency in particular. For the FPS, ZED has a built-in function that allows the user to get the current frame FPS values. We can use it and record it into an output file. We want to ensure minimal delay between the senders and receiver for the latency so it feels live. To calculate, as stated above in 3.2.2, we use both senders and receiver capture timestamps and subtract them. However, this is only trustworthy if both devices have precisely the same timestamp. We achieved that by applying the mentioned NTP (see 3.1). We have developed a small bash script to calculate the bandwidth using the *ifstat* tool, which prints network usage statistics and saves its logs into a file. The function receives a *-t* flag to add a timestamp to each entry. By setting its value to 0.25, we capture the bandwidth usage every quarter of a second.

As stated above, we needed to use the record flag (-r) on both runners to get both recordings. This originates an SVO file with all the camera information. To compare them, we need two things first. Sync the videos, ensuring their content is identical, and convert them into media player files (avi, mp4). Here, we read the distorted SVO and print the first and last frame timestamps. With them, we can run a command within the Zed SDK that allows cutting SVO files by referencing two timestamps. The command below is an example of this. It returns a new SVO with all the content between those two timestamps.

```
ZED_SVO_Editor –cut svo1.svo –start–timestamp 100000 –end–timestamp 200000 output.svo"
```

To convert them, we can run a code by Stereolabs on their GitHub [42]. It runs by ingesting an SVO file and exporting it in a left image RGB and right image depth sense in an avi format (example on figure 3.4). We can use these exported video files to evaluate the quality of a VMAF. The training models were downloaded from the official VMAF GitHub page [30]. The most basic usage of the command is as follows:

```
ffmpeg –i distorted.mp4 –i original.mp4 –filter_complex libvmaf –f null –
[libvmaf @ 0x1b5b700] VMAF score: 99.055347
```

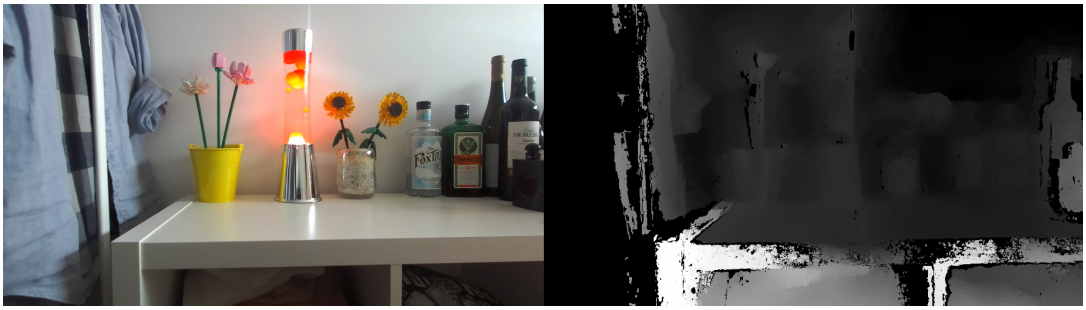

Figure 3.4: Export Recording

3.3 Configurations

As stated throughout the project, volumetric videos require much processing power. We have noticed this since, on the first approach, we tried to run the project using a single device responsible for capturing the camera, live streaming it, receiving it, and displaying the volumetric video. As a result, the computer could not run the entire project length. To fix this, we opted to split the responsibilities into two devices. One is to capture the camera signal and live stream it (senders), and the other is to receive and display it (receiver). The processing power required for the first one is much lower than that required for the second. Due to this and the considerable gap in terms of hardware, the most powerful got dedicated to receiving the live stream and building the volumetric video can be seen in table 3.2.

Table 3.2: Devices hardware specifications.

Device	**CPU**	**GPU**	**RAM**
Sender	Intel(R) Core(TM) i7-7700HQ	NVIDIA GeForce GTX 1060	8GB
Receiver	12th Gen Intel(R) Core(TM) i5-12500H	NVIDIA GeForce RTX 4060 GB	16 GB

On the network aspect, both devices were connected via Ethernet with an average connection speed of 60MB/s download and 20MB/s upload.

Different movement levels impact the Quality of Service (QoS) metrics such as latency, jitter, and packet loss. By testing across a spectrum of motion, we can better understand how these factors vary with content complexity, which is critical for maintaining a consistent viewing experience. By setting up three different scenarios with different levels of movement, we can analyze the results of the varying evaluation metrics. The first scenario has no movement, which provides a baseline for the lowest possible settings and helps evaluate video compression efficiency under static conditions. One with low movement allows us to see how the system handles minor changes. This is useful for changes and maintaining video quality without excessive bandwidth consumption. Finally, a scenario with constant movement. High movement scenarios are typically the most challenging for video compression algorithms, requiring more data to be encoded and transmitted to maintain video quality. This scenario helps evaluate the system's performance under maximum stress, including latency, buffering, and image quality degradation. Movement also impacts the network and server resources. By testing with different levels of motion, you can gauge the system's capability to handle data spikes and continuous high load. These scenarios provide a comprehensive view of how your video broadcasting system performs under different conditions, helping you

identify potential bottlenecks, optimize settings, and ensure a high-quality viewing experience across diverse content types.

Chapter 4

Results

This chapter presents, analyzes, and discusses all the results for the three streaming scenarios. These three scenarios each focus on a different real-world application. The first is in a static environment with as minimal movement as possible. The second resembles a typical conversation between two persons in which the movement is small and constant. The third's main objective is to simulate a high-movement scenario where the camera constantly moves in every direction.

To focus on collecting metrics of interest, some settings remained constant throughout all project scenes in the experiments, namely codec and bitrate (table 4.1). This helped to control variances and focus on the most critical metrics. The first was set to H264, an industry standard, and the latter to 10,000 bits per second (bps). The rest of them followed the same pattern for all scenarios. The decision behind this was again to contain the variance in the results. The resolutions used were 720p and 1080p, each for a pair of tests. The FPS used ranged from 15 to 60. These settings were defined on the sender's device, which controls the camera settings, while the receiver's main function is to ingest the streaming feed and build the volumetric scene.

Table 4.1: Constant settings for all scenarios.

Senario	**Codec**	**Bitrate**
01	H264	10.000
02	H264	10.000
03	H264	10.000

Another important note is regarding the score of the VMAF algorithm. As stated in 3.2.3, the exported files can have either Depth and RGB or full RGB. Only depth was used in the first approach. However, due to the low percentage results, the full RGB was also used, thus resulting in two VMAF results for each test.

4.1 Scenario 1: Static Scene

In the first scenario, the camera was set still facing a sideboard with a lava lamp placed on top to show as little movement as possible.

Figure 4.1 shows a screenshot of the first scenario with a straight POV. The same timeframe can be seen in figure 4.2 with an applied rotation and translation. The pyramid at the bottom right of the figure shows the initial camera position.

Even though we are evaluating a no-movement scenario, the project is aimed at videos, so minimal movement is necessary. The table below shows the settings used for the four tests regarding the first scenario (table 4.2). Some values in this and the following tables are colored in light gray. These are to be identified as constant values throughout the project.

Figures 4.3, 4.4, and 4.5 displayed the FPS, latency, and bandwidth variations, respectively. The table 4.3 shows the calculated results for this scenario.

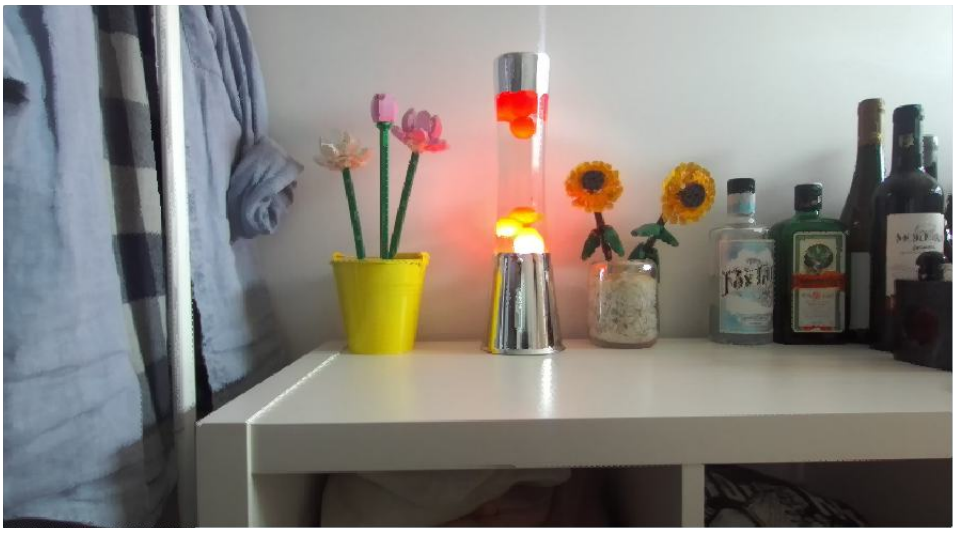

Figure 4.1: Scene 1 front-view screenshot

Figure 4.2: Scene 1 side-view screenshot

Table 4.2: Test settings for scenario 1.

Id	Movement	Resolution	FPS	Codec	Bitrate
01	None	HD1080	15	H264	10.000
02	None	HD1080	30	H264	10.000
03	None	HD720	30	H264	10.000
04	None	HD720	60	H264	10.000

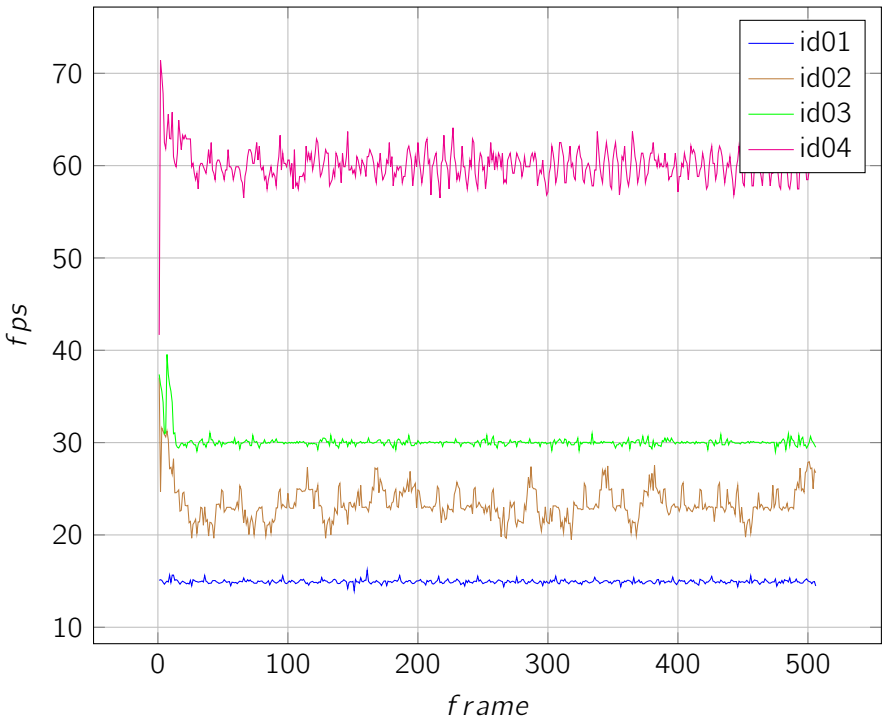

Figure 4.3: FPS variation for streaming scenario 1.

Analyzing the first set of results regarding the still camera scenario, the FPS variance was more meaningful on tests id02 and id04. On test id02, the FPS was set to 30, yet its average values were significantly lower (23,41 average FPS). This was because the resolution used significantly impacted performance, hence the variation in this metric. The other test had a slight variance (id04), but the average FPS was still as expected (around 60 FPS).

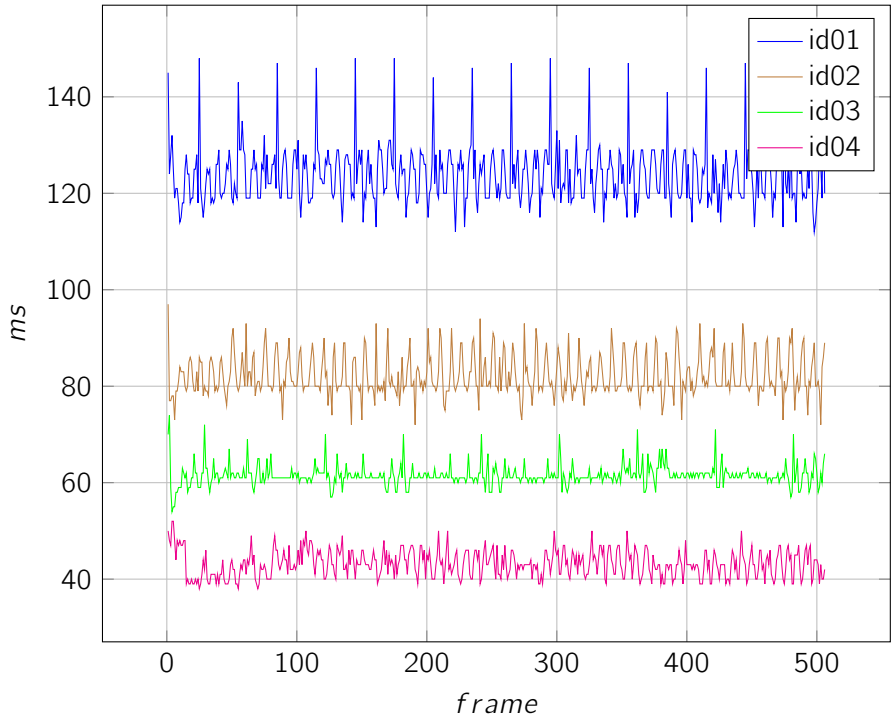

Figure 4.4: Latency variation for streaming scenario 1.

Regarding the latency, an interesting phenomenon occurred. The lower the FPS input, the higher the latency the test got. This was due to the fact that lower frame rates increase latency since the time it takes for the next frame to be presented increases [43]. It can also be observed that the variance also increases with the increase of FPS. On test id04, the difference between the lowest and rights values is 10ms, while on test id01, this difference rises to around 30 ms.

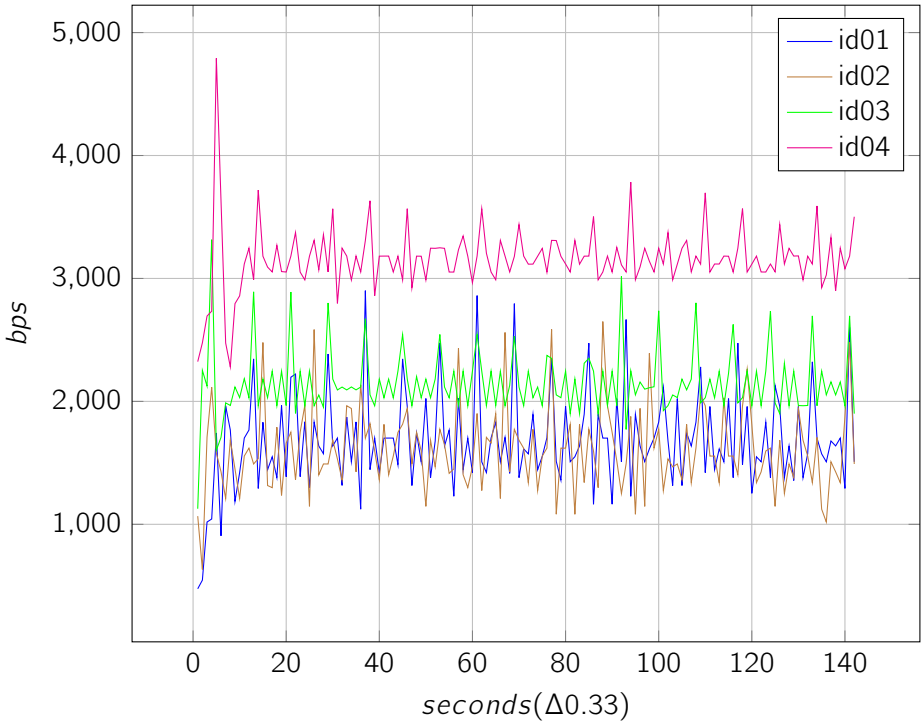

Figure 4.5: Bandwidth variation for streaming scenario 1.

The bandwidth's takeaway is that combining higher FPS values and higher resolution increases its value. On tests id01 and id02, where the resolution is the same at 1080p, even though the FPS doubled from the first test to the second (15 to 30 FPS), the bandwidth variance is very similar, with averages of 1681,72 and 1608,60, respectively. On id04, with the increase of FPS to 60, we can see that the values almost doubled to an average of 3157,46.

Table 4.3: Calculated results for scenario 1.

Id	AVG. FPS	AVG. Latency (ms)	AVG. Bandwidth (bps)	VMAF Depth	VMAF RGB
01	14.93+-0.2	123.88+-6.1	1681.72+-396.4	30.26 %	93.32 %
02	23.41+-1.9	82.21+-4.2	1608.60+-336.9	22.10 %	87.60 %
03	30.11+-0.9	61.50+-2.2	2154.1+-263.0	21.36 %	91.88 %
04	60.15+-1.9	43.18+-2.8	3157.46+-233.9	16.39 %	93.95 %

Regarding the VMAF depth results, the higher the FPS values, the lower the similarity percentage. This is because the depth map gains a considerable amount of noise on still parts of the video, which, to the consumer, are almost unnoticeable. However, objective calculations have a significant impact on precise video quality. The RGB values' similarity values were all around 90%, meaning that only a 10% distortion occurred from the original camera capture to the volumetric construction on the receiver's device.

4.2 Scenario 2: Low Movement Scene

In the second scenario, the camera is facing a person with small, constant movement, mimicking a person's normal conversation (figures 4.6 and 4.7). Once again, the settings used are described in table 4.4, variance results on figures 4.8, 4.9, 4.10, and calculated results on table 4.5.

Figure 4.6: Scene 1 front-view screenshot

Figure 4.7: Scene 2 side-view screenshot

Table 4.4: Test settings for scenario 2.

Id	Movement	Resolution	FPS	Codec	Bitrate
05	Low	HD1080	15	H264	10.000
06	Low	HD1080	30	H264	10.000
07	Low	HD720	30	H264	10.000
08	Low	HD720	60	H264	10.000

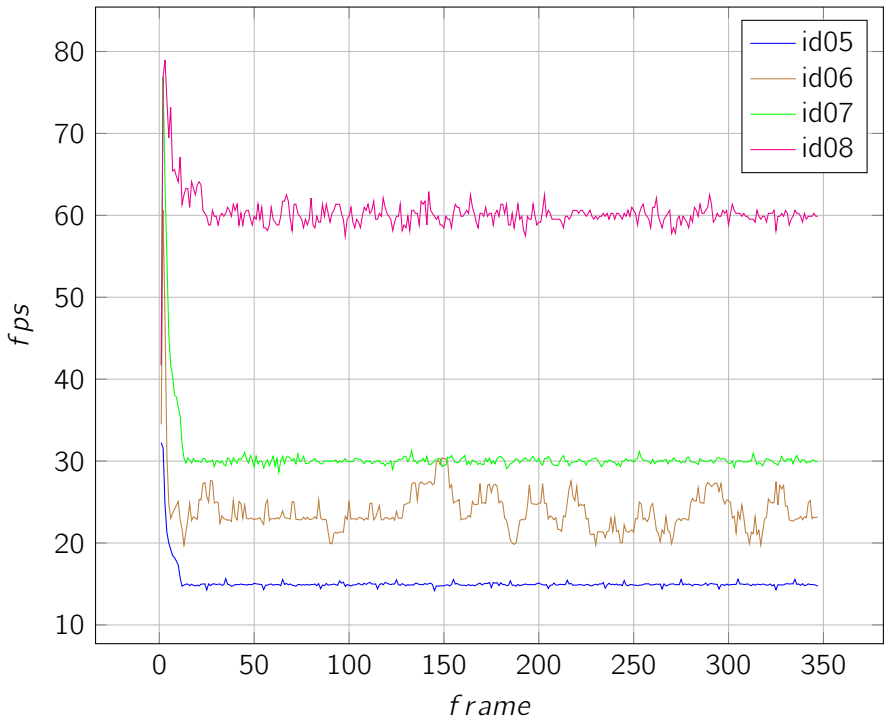

Figure 4.8: FPS variation for streaming scenario 2.

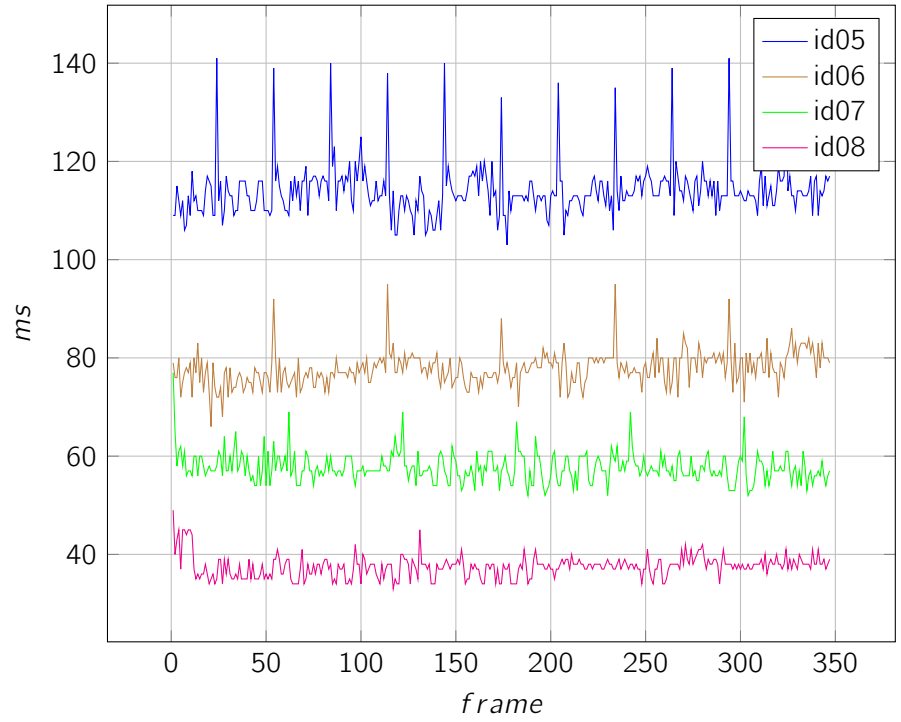

Figure 4.9: Latency variation for streaming scenario 2.

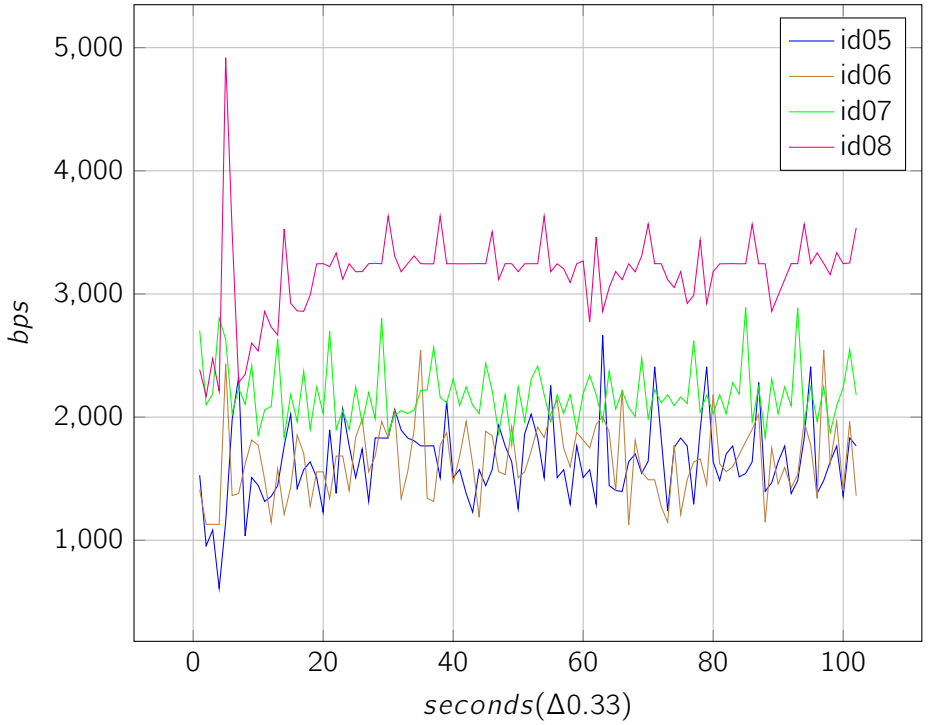

Figure 4.10: Bandwidth variation for streaming scenario 2.

The results obtained for this scenario closely mirrored those observed in the first one. The variation in FPS was consistent with the values defined during the setup phase, except for test 06, which again deviated similarly to its counterpart in the first scenario.

Table 4.5: Calculated results for scenario 2.

Id	AVG. FPS	AVG. Latency (ms)	AVG. Bandwidth (bps)	VMAF Depth	VMAF RGB
05	15.13+-1.5	114.11+-5.5	1642.08+-326.9	47.05 %	92.43 %
06	24.02+-3.2	77.82+-3.4	1662.78+-296.5	44.80 %	90.26 %
07	30.54+-3.7	57.64+-2.9	2175.72+-222.0	42.62 %	89.80 %
08	60.31+-2.4	37.49+-2.2	3202.29+-284.4	48.25 %	93.22 %

A noteworthy observation was the latency behavior across different FPS settings. As more FPS were introduced into the test environment, the average latency decreased, reinforcing the earlier conclusion that higher FPS correlates with reduced latency. This finding is crucial, emphasizing the trade-off between FPS and latency and highlighting the need for a balanced approach when configuring real-time systems for optimal performance.

Regarding bandwidth utilization, the tests id05 and id06 demonstrated a nearly identical variance, maintaining the pattern observed in the previous scenario. This consistency suggests a predictable bandwidth performance under similar conditions. However, test id08 stood out, showing a significant increase in bandwidth usage. This spike could indicate specific network or system behaviors under increased load, pointing to potential bottlenecks or areas for optimization in future system iterations.

The VMAF results revealed a marked improvement compared to the first scenario, particularly in the depth component. The VMAF values, derived by comparing the original content to the distorted output using the VMAF tool, hovered around 45%, indicating a moderate quality degradation level in the processed video. When analyzing the RGB component, the system maintained a consistent performance, with scores remaining around 90%, similar to those observed in the previous scenario. This stability in RGB calculations confirms the system's robustness in preserving color fidelity, even under varying conditions.

4.3 Scenario 3: High Movement Scene

The screenshots in the third and final scenario can be seen in figures 4.11 and 4.12. The settings used can be seen in table 4.6, variation tables on 4.13, 4.14, 4.15, and calculated values on table 4.7. This scenario was designed to test the system's limits. It features a scene with intense movement, with the camera continuously shifting in multiple directions.

Table 4.6: Test settings for scenario 3.

Id	Movement	Resolution	FPS	Codec	Bitrate
09	Constant	HD1080	15	H264	10.000
10	Constant	HD1080	30	H264	10.000
11	Constant	HD720	30	H264	10.000
12	Constant	HD720	60	H264	10.000

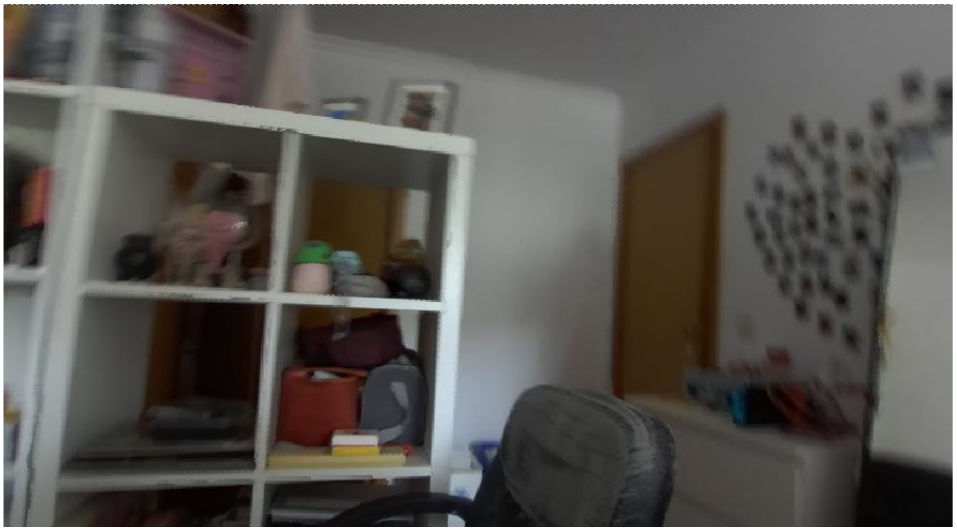

Figure 4.11: Scene 3 front-view screenshot

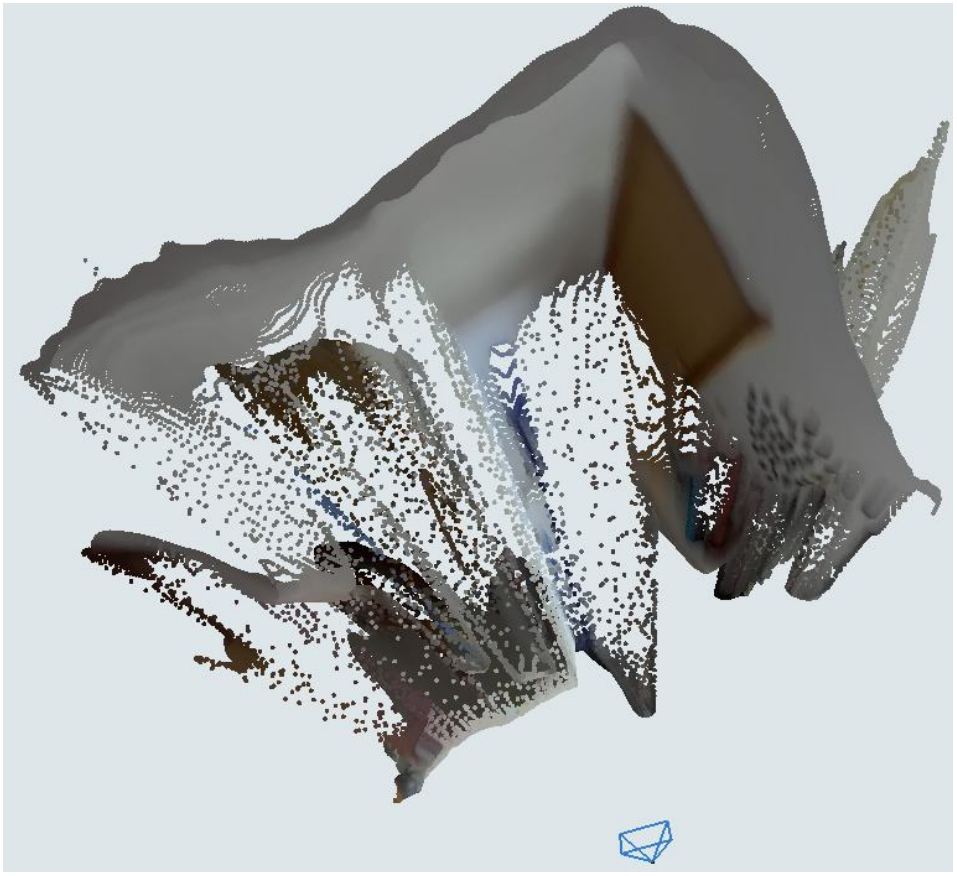

Figure 4.12: Scene 3 side-view screenshot

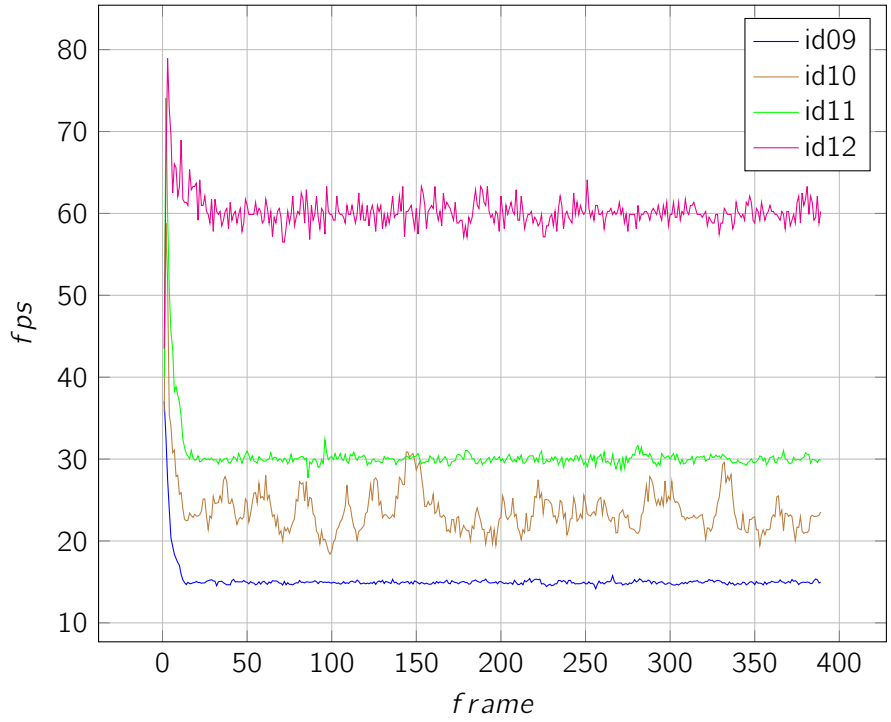

Figure 4.13: FPS variation for streaming scenario 3.

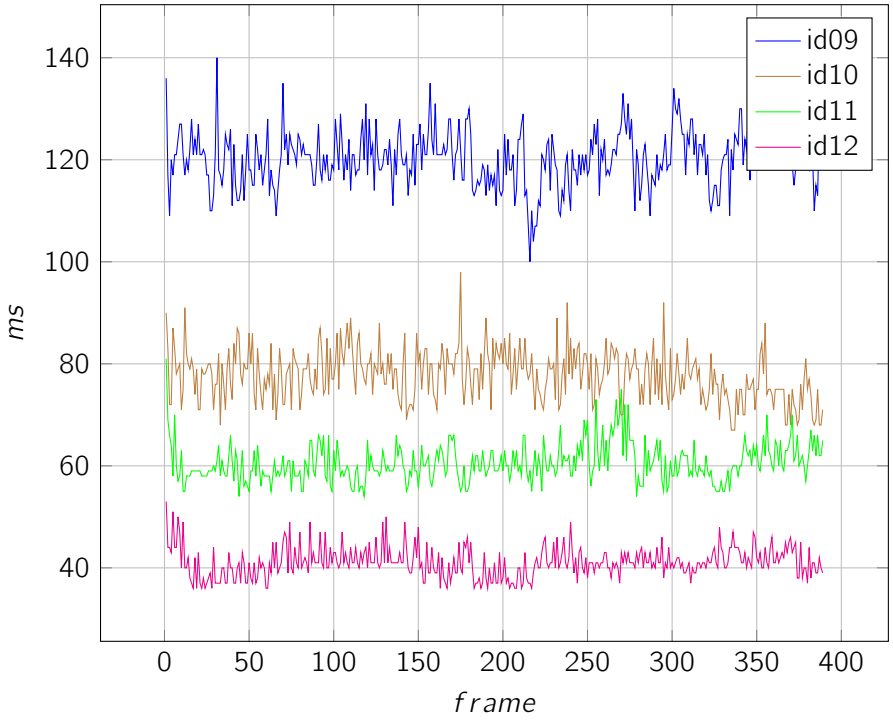

Figure 4.14: Latency variation for streaming scenario 3.

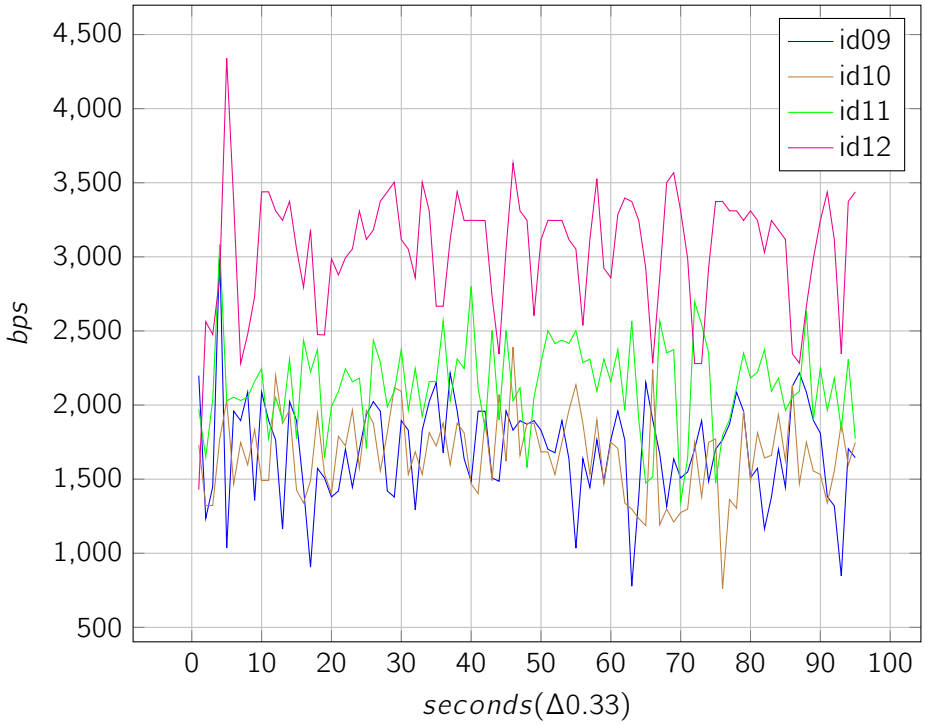

Figure 4.15: FPS variation for streaming scenario 3.

Table 4.7: Calculated results for scenario 3.

Id	AVG. FPS	AVG. Latency (ms)	AVG. Bandwidth (bps)	VMAF Depth	VMAF RGB
09	15.15+-1.7	120.43+-5.8	1719.83+-341.5	14.58 %	85.22 %
10	23.82+-3.4	77.73+-5.0	1666.85+-227.3	23.97 %	91.46 %
11	30.45+-3.2	60.57+-3.7	2142.28+-304.6	34.32 %	99.78 %
12	60.23+-2.1	41.08+-3.0	3072.78+-411.0	25.41 %	93.58 %

The results from this high-movement scenario once again exhibited variations, though the averages remained close to those observed in the first two tests. Notably, the VMAF scores reflected a drop in depth quality compared to the second scenario, indicating the increased challenge of maintaining depth accuracy under extreme conditions. However, the RGB quality remained consistently high, with scores around 95% across all four test cases, demonstrating the system's robustness in preserving visual fidelity despite the demanding circumstances.

This chapter's results highlight the performance and effectiveness of the tested volumetric video streaming pipelines. Key metrics such as FPS, latency, bandwidth usage, and VMAF scores were analyzed to evaluate each pipeline's overall quality and efficiency. These results provide valuable insights into which specific settings are most suitable for different streaming scenarios and offer a foundation for further optimization and refinement. The next chapter conducts further concluding remarks.

Chapter 5

Conclusion

Volumetric video technology is advancing rapidly, with continuous improvements in content capture using newer and more accessible cameras and displays. These advancements and ongoing developments in streaming technologies significantly contribute to refining open-source algorithms to compress volumetric data. With its ability to deliver immersive, multi-dimensional experiences, volumetric video streaming is poised to revolutionize various fields, including entertainment, sports, communication, gaming, and interactive storytelling.

The experiments conducted in this project provided crucial insights into the current state of volumetric video streaming and its potential. These experiments demonstrated that even entry-level hardware, such as the ZED 2i camera, can manage basic volumetric video streaming, even in scenarios involving significant movement. This finding underscores the technology's accessibility, suggesting that high-quality volumetric streaming is becoming increasingly feasible without prohibitively expensive equipment.

Through this research, we have comprehensively understood the hardware and software requirements for stable volumetric video streaming across different scenarios. The findings offer valuable guidelines for optimizing future systems, ensuring they are robust enough to handle the varying demands of different streaming conditions.

This project's chosen and built system is robust and reflective of real-world use cases. Its choice was driven by its ability to balance the high computational demands of volumetric content with the need for efficient streaming over diverse network conditions. It accounts for factors such as bandwidth variability, the scalability of content delivery, and the quality of user experience, all of which are critical in ensuring the system's relevance to industry applications. By replicating these key conditions, the setup provides a comprehensive framework for testing and analyzing performance and offers valuable insights into potential optimizations and future development in volumetric video streaming.

Based on the experiments and analyses conducted in this thesis, we can outline minimal hardware and software requirements for achieving stable volumetric video streaming. First, a camera with depth sensors is crucial for the system. A device such as the ZED 2i can capture high-resolution depth and RGB data. The device receiving the stream feed needs at least 16GB of RAM to ensure smooth operations and avoid bottlenecks during broadcasts. A powerful GPU is recommended as this component will handle building the volumetric scene. Once again, it is essential to note that an NVIDIA GPU is required to use any ZED camera. The RAM needed for the sending device is substantially smaller; 8 GB is recommended, but 4GB can run a smooth live stream in specific scenarios.

Regarding network connection speeds, an average home connection with around 100MB/S download and upload speeds can support this technology. The camera's 10,000 bps bitrate

is more than enough for an average stream. If its use is for high-movement scenarios, an increase in this value would be needed. The resolution would not have much impact when comparing HD720 to HD1080, so any of these values are valid. Finally, depending on the use case, the FPS has quite a visual difference. From 15 FPS to 60 FPS is a valid choice, but, as shown in the results chapter (4), the FPS did not have much impact on the performance, meaning higher values will have better video quality while not cutting on performance.

In summary, this thesis explored the critical hardware and software requirements for stable volumetric video streaming across various scenarios. The insights gained from the experiments serve as a foundation for optimizing future systems. As technology evolves, ongoing research and development will be essential to harness the potential of volumetric video streaming, ultimately making it a mainstream technology capable of transforming how we experience media and communication.

5.1 Recommendations & Future Works

Concerning hardware, investing in more powerful GPUs can significantly enhance performance, especially in high-movement scenarios. RAM on sending and receiving computers can improve system stability and data handling capabilities.

Implementing more sophisticated adaptive bitrate streaming techniques can ensure consistent performance even under varying network conditions by dynamically adjusting the quality of the volumetric video stream in real time. These advanced techniques assess network bandwidth, device capabilities, and user behavior to deliver the best possible quality without overloading the network or causing interruptions. By integrating adaptive bitrate algorithms, the system can intelligently switch between different resolution levels, reducing the bitrate when network conditions degrade and increasing it when more bandwidth is available. This minimizes buffering and latency while maintaining an optimal balance between quality and performance. These techniques allow the system to accommodate a broader range of devices, from high-end hardware capable of handling full-resolution volumetric content to lower-end devices that may only support lower-quality streams.

Another valuable extension would be implementing a multi-camera system using the Stereolabs SDK to enhance the captured scene.

On the project at hand, where only one camera was used, there were many dark or blind spots. One reason behind this is the limited camera FOV. Another is that the camera can't capture what's behind any intercepted object. Multiple cameras could improve this by expanding the FOV and capturing more accurate volumetric representations, particularly for large or dynamic scenes. However, integrating multiple cameras introduces several challenges that need to be addressed for efficient performance. Ensuring temporal synchronization between multiple cameras is critical to avoid discrepancies in the captured data. Misalignments can occur in the depth of information without precise synchronization, leading to inconsistent volumetric reconstructions and degraded user experiences. Each device must be precisely calibrated in position and orientation to create a unified coordinate system. Misalignment or calibration drift over time can result in gaps or overlaps in the 3D data, affecting the accuracy of the volumetric output. Finally, the real-time processing required to merge data from multiple cameras increases the computational burden. Therefore, the system must be optimized to process incoming data.

Bibliography

[1] Vladimiros Sterzentsenko et al. "A low-cost, flexible and portable volumetric capturing system". In: *2018 14th international conference on signal-image technology & internet-based systems (SITIS)*. IEEE. 2018, pp. 200–207.

[2] Cathy Hackl. *What Is Volumetric Video And Why It Matters To The Enterprise*. https://www.forbes.com/sites/cathyhackl/2020/09/27/what-is-volumetric-video--why-it-matters-to-the-enterprise/. Accessed: 2023-11-24. 2020.

[3] Irene Viola and Pablo Cesar. "Volumetric video streaming". In: *Immersive Video Technologies* 425 (2022).

[4] F. Yeung et al. "DELIVERING OBJECT-BASED IMMERSIVE MEDIA EXPERIENCES IN SPORTS". In: *ITU Journal* 3.1 (2020), pp. 1–8.

[5] *Stereolabs, ZED 2*. https://www.stereolabs.com/products/zed-2. Accessed: 2023-12-14.

[6] Kyungjin Lee et al. "GROOT: A Real-Time Streaming System of High-Fidelity Volumetric Videos". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. MobiCom '20. London, United Kingdom: Association for Computing Machinery, 2020. isbn: 9781450370851. doi: 10.1145/3372224.3419214. url: https://doi.org/10.1145/3372224.3419214.

[7] Alan Bryman and Emma Bell. *Business research methods*. Cambridge: Oxford University Press, 2011.

[8] James W Drisko and Tina Maschi. *Content analysis*. Pocket Guide to Social Work Re, 2016.

[9] Leo A. Goodman. "Snowball Sampling". In: *The Annals of Mathematical Statistics* 32.1 (1961), pp. 148–170. issn: 00034851. url: http://www.jstor.org/stable/2237615 (visited on 12/07/2023).

[10] ARCTURUS. *What is Volumetric Video: A Begginer's Guide*. https://arcturus.studio/blog/what-is-volumetric-video. Accessed: 2023-11-23. 2023.

[11] Kaiyuan Hu et al. "Understanding User Behavior in Volumetric Video Watching: Dataset, Analysis and Prediction". In: *Proceedings of the 31st ACM International Conference on Multimedia*. MM '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 1108–1116. isbn: 9798400701085. doi: 10.1145/3581783.3613810. url: https://doi.org/10.1145/3581783.3613810.

[12] Cal Jeffrey. *The NBA aired an entire game using volumetric video tech that renders players in 3D in real-time*. https://www.techspot.com/news/93823-nba-aired-entire-game-using-volumetric-video-tech.html. Accessed: 2023-11-30. 2022.

[13] Bavor Clay. *Project Starline: Feel like you're there, together*. https://blog.google/technology/research/project-starline/. Accessed: 2023-12-01. 2021.

[14] Andrew Nartker. *A first look at Project Starline's new, simpler prototype*. https://blog.google/technology/research/project-starline-prototype/. Accessed: 2023-12-01. 2023.

[15] *Geometric Informatics Website*. https://www.geometricinformatics.com/. Accessed: 2023-12-06.

[16] *Velodyne Lidar Website*. https://velodynelidar.com/. Accessed: 2023-12-06.
[17] NME. *Radiohead make new video – without cameras*. https://www.nme.com/news/music/radiohead-442-1335451. Accessed: 2023-12-06. 2008.
[18] Calley Nye. *Radiohead Partners With Google For Music Video Launch*. https://www.washingtonpost.com/wp-dyn/content/article/2008/07/14/AR2008071402460.html. Accessed: 2023-12-06. 2008.
[19] *4D Views Website*. https://www.4dviews.com/holosys. Accessed: 2023-12-06.
[20] Fashion Innovation Agency. *Entering A New Dimension: 3D Video Revolutionising Fashion Film*. https://www.fialondon.com/projects/a-new-dimension-3d-video-revolutionising-fashion-film/. Accessed: 2023-12-06. 2023.
[21] *4D Views Website*. https://www.4dviews.com/showreel/17. Accessed: 2023-12-14.
[22] *Canon Studio Features*. https://global.canon/en/vvs/features/studio.html. Accessed: 2023-12-14.
[23] *Stereolabs Docs: API Reference, Tutorials, and Integration*. https://www.stereolabs.com/docs. Accessed: 2023-12-14.
[24] *Azure Kinect DK*. https://azure.microsoft.com/en-us/products/kinect-dk/. Accessed: 2024-01-04.
[25] Vladislav Angelov et al. "Modern Virtual Reality Headsets". In: *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. 2020, pp. 1–5. doi: 10.1109/HORA49412.2020.9152604.
[26] *The Voxon VX1- Technical Description*. https://shorturl.at/bkIMP. Accessed: 2024-01-02.
[27] Daniel Smalley et al. "Volumetric displays: turning 3-D inside-out". In: *Opt. Photonics News* 29.6 (2018), pp. 26–33.
[28] Lindsey Valich. *Researchers use lasers to display 'true' 3-D objects*. https://www.rochester.edu/newscenter/researchers-use-lasers-display-true-3-d-objects/. Accessed: 2024-01-02. 2017.
[29] *Looking Glass Documentation*. https://docs.lookingglassfactory.com/keyconcepts/how-it-works. Accessed: 2024-01-03.
[30] Luke Larsen. *This 3D laptop screen was the coolest thing I experienced at CES 2023*. https://www.digitaltrends.com/computing/3d-laptop-screen-was-best-thing-from-ces-2023/. Accessed: 2024-01-05. 2023.
[31] Emin Zerman et al. "Textured Mesh vs Coloured Point Cloud: A Subjective Study for Volumetric Video Compression". In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. 2020, pp. 1–6. doi: 10.1109/QoMEX48832.2020.9123137.
[32] Bo Han, Yu Liu, and Feng Qian. "ViVo: Visibility-Aware Mobile Volumetric Video Streaming". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. MobiCom '20. London, United Kingdom: Association for Computing Machinery, 2020. isbn: 9781450370851. doi: 10.1145/3372224.3380888. url: https://doi.org/10.1145/3372224.3380888.
[33] MPEG. *Introduction to the MPEG-PCC project*. https://mpeg-pcc.org/. Accessed: 2024-01-06.
[34] Felix Mercer Moss et al. "On the optimal presentation duration for subjective video quality assessment". In: *IEEE Transactions on Circuits and Systems for Video Technology* 26.11 (2015), pp. 1977–1987.

[35] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. doi: 10.1109/TIP.2003.819861.

[36] Netflix. "VMAF: The Journey Continues". In: *Medium* (2018). url: https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12.

[37] NVIDIA. *What is CUDA?* https://nvidia.custhelp.com/app/answers/detail/a_id/2132/what-is-cuda. Accessed: 2024-06-05.

[38] OpenGL. *OpenGL Overview*. https://www.khronos.org/opengl/. Accessed: 2024-07-01.

[39] Bjarne Stroustrup. "An overview of the C++ programming language". In: *Handbook of object technology* (1999), p. 72.

[40] IEEESA. *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. https://standards.ieee.org/ieee/1588/6825/. Accessed: 2024-07-01.

[41] Gaël Roualland. *Github - ifstat*. https://github.com/matttbe/ifstat. Accessed: 2024-07-01.

[42] Stereoabs. *Github - Stereolabs export*. https://github.com/stereolabs/zed-sdk/tree/master/recording/export/svo/cpp. Accessed: 2024-07-02.

[43] Benjamin F Janzen and Robert J Teather. "Is 60 fps better than 30? The impact of frame rate and latency on moving target selection". In: *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 2014, pp. 1477–1482.