PRESENCE

Funded by the European Union

A toolset for hyper-realistic and XR-based human-human and human-machine interactions, PRESENCE

Grant Agreement nº 101135025

HE Call identifier: HORIZON-CL4-2023-HUMAN-01-CNECT
Topic: HORIZON-CL4-2023-HUMAN-01-21

Type of action: HORIZON Research and Innovation Actions

D2.1 Volumetric and holoconferencing technologies report I

DISSEMINATION LEVEL

☒	**PU**	**Public**
	SEN	*Confidential, only for members of the consortium (including the Commission Services)*

Grant Agreement nº:	Project Acronym:	Project title:
101135025	PRESENCE	A toolset for hyper-realistic and XR-based human-human and human-machine interactions

Lead Beneficiary:	Document version:
I2CAT	V1.1

Work package:
WP2 - Multi user Holoconferencing

Deliverable title:
D2.1 Volumetric and holoconferencing technologies report I

Start date of the project:	Contractual delivery date:	Actual delivery date:
1st of January 2024	28 of February 2025	27 of February 2025

Editor(s):
Diego González Morín (i2CAT)

LIST OF CONTRIBUTORS

PARTNER	CONTRIBUTOR
I2CAT	Leonel Toledo, Mario Montagud
CERTH	Dimitris Zarpalas, Theofilos Tsoris, Eleana Almaloglou, Aris Cheimariotis, Dimitrios Pattas
RAYTRIX	Arne Erdmann

LIST OF REVIEWERS

PARTNER	REVIEWER/S
VECTION	Ivan Volpi

CHANGE HISTORY

VERSION	DATE	PARTNERS	DESCRIPTION/COMMENTS
V0.1	06-01-2025	i2CAT	Include Table of Contents and Skeleton
V0.2	30-01-2025	i2CAT	First draft of sections 1 and 2
V0.3	03-02-2025	i2CAT	Included draft of T4.3
V0.4	03-02-2025	CERTH	Included draft of T4.1
V0.5	03-02-2025	Raytrix	Included draft of T4.2
V0.6	05-02-2025	i2CAT	Included draft of T4.4
V0.7	07-02-2025	CERTH	Updated draft of T4.2
V0.8	07-02-2025	CERTH	Updated draft of T4.1
V0.9	11-02-2025	i2CAT	First draft of sections 5-8
V0.10	24-02-2025	VECTION	Provided feedback after review
V1.1	24-02-2025	i2CAT	Incorporated feedback and finalize report
V1.1	27-02-2025	i2CAT	Formatting, indexes and cross-references fixing, final version submitted to the EC

Executive summary

This deliverable presents the progress and key developments of Work Package 2 (WP2) – Multi-User Holoconferencing within the PRESENCE project. WP2 focuses on the research, design, and implementation of a real-time holoportation system that enables photorealistic, volumetric 3D human reconstruction for immersive Extended Reality (XR) communication. The document outlines the objectives, tasks, architecture, and status of WP2, detailing the advancements made in multi-camera capture, real-time 3D reconstruction, volumetric compression, and scalable holoconferencing solutions.

The deliverable highlights progress in the development of a multi-camera light-field capture system, enabling real-time volumetric video acquisition with enhanced depth accuracy. Key improvements in 3D reconstruction techniques—leveraging mesh-based and point-cloud representations—have significantly optimized the system's real-time performance. In addition, advanced volumetric video compression algorithms have been explored to maintain high visual quality while reducing bandwidth requirements for real-time transmission. To support multi-user holoportation, the work has also focused on scalability enablers, including distributed processing, edge/cloud computing, and network optimizations over 5G/6G infrastructures.

This document details the technical milestones achieved in T2.1 (Capturing), T2.2 (Reconstruction), T2.3 (Compression & Streaming), and T2.4 (Scalability), along with key performance indicators (KPIs) and current challenges. The findings contribute to the broader goals of PRESENCE by bridging the gap between volumetric holoportation research and real-world multi-user XR applications

Table of contents

List of Tables

List of Figures

1. Introduction

The concept of presence is central to the development of immersive technologies, particularly in Extended Reality (XR) environments. It encompasses several interrelated psychophysical aspects, such as plausibility (the illusion that virtual events are real), co-presence (the sensation of being with others), and place illusion (the feeling of being physically present in a virtual environment). Enhancing these aspects is crucial for improving XR experiences, yet current technologies have not yet achieved the levels of presence necessary to bring us closer to the ultimate goal of virtual reality: to be anywhere, doing anything, together with others, regardless of our physical locations. The **PRESENCE** project aims to address these challenges by pushing the boundaries of immersive technologies and improving presence in physical-digital worlds through the development of new tools and methodologies.

PRESENCE will tackle three main challenges in its research:

1. The creation of realistic visual interactions among remote individuals via holoportation, leveraging live volumetric capturing, compression, and optimization techniques.
2. The development of novel haptic systems to provide realistic touch sensations in multi-user remote environments, enhancing the sense of physical presence.
3. The generation of intelligent avatars and agents to enable natural social interactions among both virtual humans and AI agents.

This work package (WP2), led by i2CAT, focuses on developing the technological underpinnings necessary for photorealistic, multi-user holographic interactions. The primary goal is to bridge the gap between simple holoportation and more complex, interactive experiences that allow for true multi-user, scalable holoconferencing. The deliverable outlines the mid-term progress in the areas of volumetric capturing, reconstruction, compression, and optimization techniques critical for achieving real-time holoportation systems. In particular, WP2 focuses on the following technologies:

- **Volumetric Capturing and Data Acquisition**: WP2 focuses on advancing holoportation technologies by developing a high-performance volumetric capturing pipeline. This involves enhancing light-field technology for real-time 3D data acquisition, using multiple synchronized cameras to create high-resolution volumetric video. The goal is to make the system more affordable and scalable, enabling broader adoption of XR communication.
- **Volumetric Representations and Reconstruction**: Once high-quality 3D data is captured, the next step is creating photorealistic, full-body 3D models for seamless integration into XR environments. The project is working on a real-time 4D reconstruction system that maintains high frame rates and resolution, addressing challenges like multi-camera synchronization, noise, and occlusions. The approach also includes using past and future frames to improve current frame quality for smoother, more lifelike user representations.
- **Holoportation Compression and Streaming**: For remote holoportation, the project is developing efficient compression and transmission methods to handle volumetric video in real-time, minimizing latency while preserving visual quality. These advancements will be used to build a pipeline for remote holoportation that will be tested and demonstrated in XR applications.

- **Multi-User Holoconferencing and Scalability**: WP2 is expanding holoportation to multi-user experiences, enabling multiple remote participants to interact in an XR environment. The team is focusing on optimizing scalability and performance through edge and cloud computing, including leveraging 5G and 6G technologies. This will allow for large-scale, distributed holoportation suitable for both social and professional use cases.

1.1. Purpose, scope, and structure of the document

The purpose of this deliverable is to provide a comprehensive overview of the work carried out within WP2 of the PRESENCE project, focusing on the development of a real-time holoportation system. This document outlines the progress made in volumetric capturing, 3D data acquisition, volumetric video reconstruction, and holoportation compression and optimization techniques. It details the methods and innovations that contribute to advancing immersive communication experiences in XR environments, with an emphasis on scalability and multi-user interactions.

This deliverable covers the intermediate progress made towards achieving the goals of WP2, specifically in the areas of:

- **Volumetric capturing and data acquisition**: Advancements in real-time 3D data acquisition and the integration of light-field technology.
- **Volumetric representations and reconstruction**: Methods for converting volumetric data into photorealistic 3D models for XR applications.
- **Holoportation compression and streaming**: Development of compression and streaming techniques essential for real-time volumetric video transmission.
- **Multi-user holoconferencing and scalability**: Exploration of optimization strategies and technologies to support multiple remote users in XR environments.

This document is intended to provide stakeholders and partners with detailed insights into the progress, methodologies, and challenges encountered in WP2. The results from this work will contribute directly to the testing and implementation of user-oriented demonstrators in WP1, and will provide valuable insights into the feasibility of holoportation in XR environments. Additionally, the results will inform the creation of holoportation APIs for integration into future XR applications and multi-user scenarios (WP5).

The document is structured as follows:

Section 1 - Introduction: Outlines the purpose, scope, and structure of the document, providing an overview of WP2's objectives and key aspects of the holoportation system being developed.

Section 2 - WP2 - Multi user Holoconferencing: Describes the objectives and KPIs of WP2, the tasks involved, and the overall architecture of the holoportation system.

Section 3 - Related Work**:** Reviews relevant literature and prior work that informs the development of WP2 and holoportation technologies.

Section 4 - Status: Provides a detailed overview of the progress made in WP2, broken down by individual tasks (T2.1 to T2.4), including key achievements, KPIs status, and any deviations with mitigation plans.

Section 5 - Preliminary Evaluation: Discusses the initial evaluation of the developments made so far, focusing on performance and feasibility within the WP2 framework.

Section 6 - Outlook: Highlights future plans for experiments, publications, and pilots that will help advance the work of WP2 and holoportation technologies.

Section 7 - Abbreviations and definitions: Lists abbreviations and definitions relevant to the document for clarity and ease of understanding.

Section 8 - References: Provides a comprehensive list of all references cited throughout the document.

This structure allows for a clear understanding of the work accomplished thus far, while providing a roadmap for the continued development of immersive communication tools.

2. WP2 - Multi user Holoconferencing

2.1. Objectives and KPIs

WP2 is dedicated to advancing holoportation technology, enabling real-time, photorealistic communication within XR environments. The work focuses on providing intuitive and realistic user experiences that bring real humans into virtual spaces, fostering meaningful and interactive communication. The goals of WP2 are centered around the development of a comprehensive holoportation system that supports multi-user scenarios, integrates with other technologies, and maintains high levels of user satisfaction. WP2 solutions focus on enabling hyper-realistic, real-time volumetric communication for remote participants. This involves the digitization of humans and their seamless integration into interactive virtual environments, empowering multiple users to interact in a shared XR space. The particular target goals of WP2 can be summarized as:

- **Realistic User Experiences**: The goal is to develop a novel holoportation system that captures and transfers humans into virtual worlds with a high degree of realism. This will offer a more intuitive and engaging form of remote communication compared to traditional avatars.
- **Real-Time, Multi-User Communication**: WP2 aims to facilitate live, multi-user volumetric communication by employing light-field-based 3D data acquisition techniques, allowing multiple users to connect and interact simultaneously in high-quality, real-time environments.
- **Integration with Other Technologies**: The project seeks to explore and exploit synergies with cutting-edge technologies such as 5G/6G and edge/cloud computing to enhance the scalability and performance of the holoportation system.
- **API Development**: To ensure broad usability, the holoportation system will be designed to integrate easily with other systems, facilitating its inclusion in diverse XR applications.

These objectives directly contribute to enhancing the realism and usability of remote communications, especially in social and professional contexts.

WP2 has established several KPIs to track the progress and success of the project in delivering the stated objectives. These KPIs ensure that the system meets the required technical standards and provides a high-quality user experience.

- **KPI 2.1: Real-time Holoportation System with Photo-realistic Quality:** Deliver a real-time holoportation system that achieves users' acceptance levels of ≥4 on a 5-point scale in subjective tests using validated questionnaires (e.g., IPQ, Mon224).
- **KPI 2.2: Multi-User Support for Real-time Communication:** Develop a fully functional holoportation system that can support at least 6 users in a real-time multi-party XR communication scenario.
- **KPI 2.3: Holoportation APIs for Integration**: Deliver a set of holoportation APIs for integration with the two other pillars (use case scenarios), as well as additional XR scenarios.

2.2. Tasks

WP2 is organized into four interrelated tasks that collectively form the core framework for achieving the goals of real-time holoportation, volumetric capturing, and multi-user interactions in XR environments. Each task contributes distinct components of the overall system, from the initial data acquisition through to scalable multi-user holoportation. The tasks are structured to address key technical challenges while ensuring integration and compatibility across all components of the system.

Task 1: 3D Data Acquisition and Volumetric Capturing

The first task of WP2 focuses on the development of a state-of-the-art 3D data acquisition and volumetric video capturing system. The aim is to improve upon existing light-field technology to enable real-time volumetric capturing using a multi-camera setup. This task addresses several challenges related to the acquisition of high-quality 3D data, with an emphasis on real-time capabilities for holoportation.

- **Real-time Light-Field Capturing**: Traditional volumetric capture often uses single-camera systems that operate offline, which limits their ability to capture dynamic interactions in real-time. This task advances the concept by using multiple synchronized light-field cameras to capture detailed, high-resolution volumetric video. The key challenge is to integrate the data from these multiple cameras seamlessly, ensuring that the reconstructed 3D model is accurate and lifelike.
- **Cost Reduction and Scalability**: One of the primary goals is to reduce the costs associated with light-field 3D data acquisition systems, making them more affordable and accessible for large-scale deployment. By optimizing the use of hardware and software, this task aims to create a scalable system that can be implemented in various XR applications and used by multiple users simultaneously.
- **System Integration**: The data captured by the multi-camera system will serve as input for subsequent tasks, particularly those focused on volumetric reconstruction and holoportation. The

results of this task will provide the necessary foundation for WP2's later developments in 3D reconstruction, compression, and streaming.

Task 2: Volumetric Representations and Reconstruction

Following the acquisition of high-quality 3D data, the next task in WP2 focuses on transforming the captured data into photorealistic, full-body 3D models that can be seamlessly integrated into an XR environment. This task addresses the complex problem of volumetric video reconstruction and aims to achieve a high level of realism and interactivity for remote participants in holoportation and holoconferencing.

- **Real-time 4D Reconstruction**: This task aims to develop a real-time 4D reconstruction system that takes the captured volumetric data and converts it into fully reconstructed, photorealistic 3D models of human participants. A significant challenge lies in maintaining both high resolution and frame rates, particularly when handling large-scale volumetric data in real time. To achieve this, novel reconstruction algorithms are being explored, which not only take into account the 3D geometry of the captured scenes but also the temporal dimension, ensuring that each frame is consistent with the movement of the participant.
- **Novel Representation Methods**: The task will also focus on creating innovative methods for representing volumetric video. Light-field technology enables multiple viewpoints of the same surface, providing an unprecedented level of depth and realism. The task will explore how to fully exploit these capabilities, ensuring that the reconstructed models can be rendered in real-time from different angles, with varying lighting conditions, and with high visual fidelity.
- **Challenges and Solutions**: Volumetric reconstruction involves overcoming several significant challenges. These include multi-camera synchronization, handling noisy input data, and addressing occlusions where parts of the body or objects may be blocked from view. To tackle these issues, advanced algorithms for calibration, noise reduction, and frame interpolation will be developed.

Task 3: Holoportation Pipeline: Volumetric Compression and Streaming

In this task, WP2 addresses the critical challenge of transmitting volumetric video in real-time, ensuring that the high-quality 3D data captured and reconstructed in the previous tasks can be streamed effectively to remote users. The task focuses on developing a compression and transmission pipeline that preserves the visual quality of the volumetric video while minimizing latency.

- **Real-time Compression**: Volumetric video typically involves large amounts of data due to the high resolution and complexity of 3D models. Compressing this data without sacrificing visual fidelity is crucial for ensuring smooth holoportation experiences. This task will explore novel compression algorithms designed specifically for volumetric video, balancing compression ratios and image quality.
- **Transmission Systems**: Once the volumetric video is compressed, it must be transmitted to remote users in real time. This task will develop transmission protocols that minimize latency and ensure that the data is delivered smoothly across networks, even under variable network

conditions. The goal is to support high-quality, lag-free holoportation, even when users are located in different geographic locations.

- **System Integration and Testing**: The developed compression and streaming pipeline will be integrated into the overall holoportation system. It will be tested under real-world conditions to ensure that it can support remote holoportation and provide high-quality video streaming for users in diverse settings. This task will also feed into user evaluations and demonstrators, allowing real-world testing of the technology.

Task 4: Multi-user and Scalable Holoconferencing

The final task in WP2 focuses on enabling multi-user holoportation, allowing several participants to interact in a shared XR space. This task expands on the previous work by developing systems and protocols that ensure scalability, enabling holoportation for large groups of users, and optimizing the system for multi-user interactions.

- **Multi-Session and Multi-User Optimization**: A key challenge in holoconferencing is managing multiple participants simultaneously while maintaining high quality and low latency. This task will explore techniques for session management and multi-user optimization, ensuring that each participant has an immersive and responsive experience, regardless of the number of concurrent users.
- **Scalability with Edge and Cloud Computing**: To support large-scale multi-user interactions, the task will leverage edge and cloud computing resources, particularly through the use of 5G and future 6G networks. By distributing the computational load across cloud and edge nodes, the system can scale dynamically to accommodate more users without compromising performance.
- **Integration into XR Applications**: The solutions developed in this task will be integrated into XR applications, enabling real-time, interactive holoportation for multiple participants. The multi-user holoconferencing system will be tested in various use cases, including both professional and social environments, to ensure its versatility and effectiveness.

Together, these tasks form the backbone of WP2, driving the development of a robust and scalable holoportation and holoconferencing system. The integration of real-time 3D data acquisition, advanced volumetric reconstruction, high-performance compression and streaming, and multi-user scalability ensures that WP2 will contribute significantly to the PRESENCE project's overarching goals of enhancing virtual communication and collaboration in XR environments.

2.3. Architecture

Figure 1 shows a schematic diagram of WP2 tasks and internal components. Each task and component will be described in detail in their respective part in Section 4.

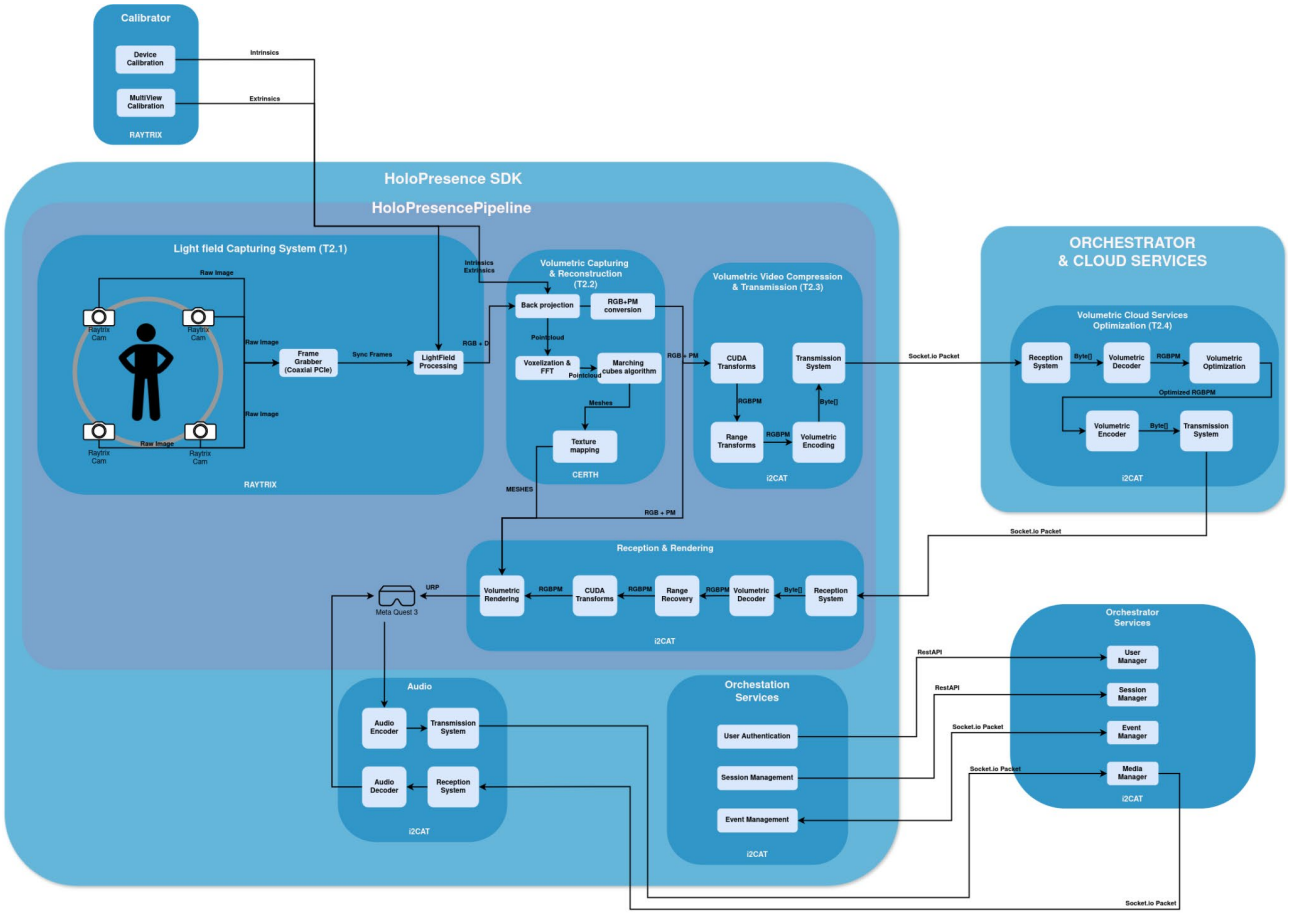


Figure 1: WP system architecture

3. Related Work

Multi-camera capture systems are pivotal in achieving real-time 3D reconstruction, enabling the creation of detailed and dynamic models of complex scenes. In the realm of telepresence and remote collaboration, multi-camera systems have been developed to model participants in real-time 3D. These systems capture and process data from multiple viewpoints, allowing for immersive interactions in virtual environments. In one of the very first examples of volumetric telepresence systems [Ref. 1] the authors utilize multiple cameras to capture dynamic scenes, reconstructing 3D models that can be streamed and visualized in real-time, thereby enhancing the sense of presence in remote communications. More recent systems, such as the one presented in [Ref. 2], present a full real-time reconstruction pipeline based on a multi RGB-D camera setup. The same setup was later used to build a full end to end immersive communication system [Ref. 3]. Similarly, the authors in [Ref. 4] present their holoconference system for social VR applications, based on low end RGB-D cameras for their real-time volumetric reconstruction solution.

All relevant examples of immersive communication systems rely on inexpensive multi RGB-D camera setups which have limited resolution and depth accuracy, hindering the potential reconstruction quality. A potential solution to overcome this limitation is the usage of light field cameras which can infer depth information with much higher resolution. Light field technology [Ref. 5] captures the intensity and direction of light rays in a scene, providing a comprehensive

representation of the visual environment. This approach enables post-capture adjustments such as refocusing and perspective shifts, offering significant advantages over traditional imaging methods. The concept of the light field was first introduced in 1936 and has gained prominence in computer graphics with advancements in computing power and network bandwidth.

Several manufacturers have developed light field cameras to leverage this technology. Wooptix[1] is focused on developing consumer level light field cameras, which are still in the prototyping phase. On the other hand, Raytrix[2] offers high end 3D light field cameras designed for high-resolution, real-time 3D capture. In the realm of 3D reconstruction, light field cameras have been utilized to enhance the accuracy and efficiency of capturing three-dimensional information. A novel structure-from-motion framework [Ref. 6] using light field cameras has been proposed for reliable 3D object reconstruction. This method involves moving a light field camera around an object and registering multiple shots to create a comprehensive 3D model. Additionally, compact light field photography techniques have been developed [Ref. 7] to achieve versatile 3D vision, enabling high-speed and accurate three-dimensional sensing over extensive distance ranges.

Once the capture system is ready, the next step is to develop an accurate and real-time human reconstruction solution. There are multiple solutions in the state of the art. The most efficient solutions rely on point cloud 3D reconstruction from RGB-D data, such as the systems presented in [Ref. 1][Ref. 2][Ref. 4]. This type of solution, while simple and hardware-efficient, is constrained in terms of reconstruction quality and transmission requirements. The overall quality can be enhanced by inferring mesh data from the RGB-D information. Recent advancements in real-time 3D human reconstruction using RGB-D cameras have led to the development of several notable mesh-based methods. Fast Body Net (FBN)[Ref. 8] is a lightweight system that utilizes a single RGB-D camera as input, employing a deep-learning network to generate a human model end-to-end, facilitating real-time applications while addressing computational challenges associated with 3D human reconstruction. Similarly, Real-Time Non-Rigid Reconstruction [Ref. 9] focuses on capturing complex human deformations using an RGB-D camera, integrating depth and color information within a volumetric framework to produce coherent 3D meshes in real time, even under highly dynamic conditions. Another notable approach, TexMesh [Ref. 10], reconstructs detailed human meshes with high-resolution textures from RGB-D video by leveraging both geometric and photometric data, ensuring spatiotemporally consistent reconstructions suitable for free-viewpoint rendering while adapting to real-world sequences in a self-supervised manner. Lastly, Accurate Neural Implicit Model (ANIM) [Ref. 11] reconstructs arbitrary 3D human shapes from single-view RGB-D images with high accuracy by leveraging multi-resolution pixel-aligned and voxel-aligned features while integrating depth supervision to enhance surface detail and mitigate ambiguities.

Beyond point clouds and meshes, various techniques have been developed to enhance 3D human reconstruction, such as Gaussian Splatting [Ref. 12]. While the original Gaussian Splatting optimization algorithm is not suitable for real-time applications, there are other relevant solutions, such as [Ref. 13], which are able to infer Gaussian splats for real-time human reconstruction.

One key component of real-time immersive communication systems is the transmission: volumetric data is heavy and can be hard to efficiently transmit in real-time. Real-time mesh and point cloud

[1] https://wooptix.com/
[2] https://raytrix.de/

compression is crucial for efficiently handling 3D data in applications such as VR, AR, and autonomous systems. Mesh compression techniques, including local 2D parameterizations and variable-length algorithms [Ref. 14], enable fast storage and decompression, making real-time applications feasible (ResearchGate). Another relevant mesh compression algorithm is Meshlet [Ref. 15] which is capable of compressing mesh data into smaller units called meshlets, optimizes GPU performance and provides rapid decompression, making it suitable for real-time applications. For point cloud data, frameworks like PointCompress3D [Ref. 16] address the challenges of compressing high-resolution LiDAR data while maintaining accuracy, enabling real-time processing and streaming. Additionally, novel methods propose a full end to end codec for Point Cloud compression and transmission as [Ref. 17].

Even in end-to-end systems with highly optimized resource usage and compression ratios, it is crucial to deploy scalability enablers that facilitate many-to-many immersive communication. One significant challenge is the need for ultra-low-latency transmission, substantial bandwidth and computational demands of holographic data [Ref. 18]. To address these challenges, several innovative solutions have been proposed. The integration of Mobile Edge Computing (MEC) with 5G networks has been explored [Ref. 19] to reduce latency and distribute computational workloads more efficiently. Additionally, the development of Multipoint Control Units (MCUs) tailored for volumetric video has been introduced to manage multi-user holographic meetings effectively [Ref. 20]. These MCUs are designed to handle the complexities of volumetric data, ensuring that multiple participants can engage in holographic conferences without compromising the quality or performance of the system.

4. Status

This section provides an overview of the progress made in WP2 over the past months, detailing the developments, key achievements, and challenges encountered in each task. It outlines the advancements in 3D data acquisition, volumetric reconstruction, compression, and multi-user scalability, highlighting the steps taken toward achieving real-time, photorealistic holoportation. Additionally, the section includes an assessment of the current status of key performance indicators (KPIs) and, where applicable, discusses any deviations from initial plans along with mitigation strategies.

4.1. T2.1 - 3D Data Acquisition and Volumetric Capturing

The goal of this task is to develop a real-time 3D data acquisition system capable of capturing high-quality volumetric representations of users for holoportation. To achieve this, we explore the use of light-field cameras due to their ability to passively capture high-resolution 3D information with low latency. Unlike traditional cameras, they capture both spatial and angular light information, enabling the generation of detailed 3D models from a single viewpoint. This capability makes them highly suitable for XR applications, reducing the complexity of multi-camera setups. However, integrating light-field technology into a real-time holoportation system presents challenges, including synchronization, calibration, and efficient data processing, which are being actively addressed within this task. In general, light-field technology has distinct characteristics that significantly differentiate it from regular pinhole cameras

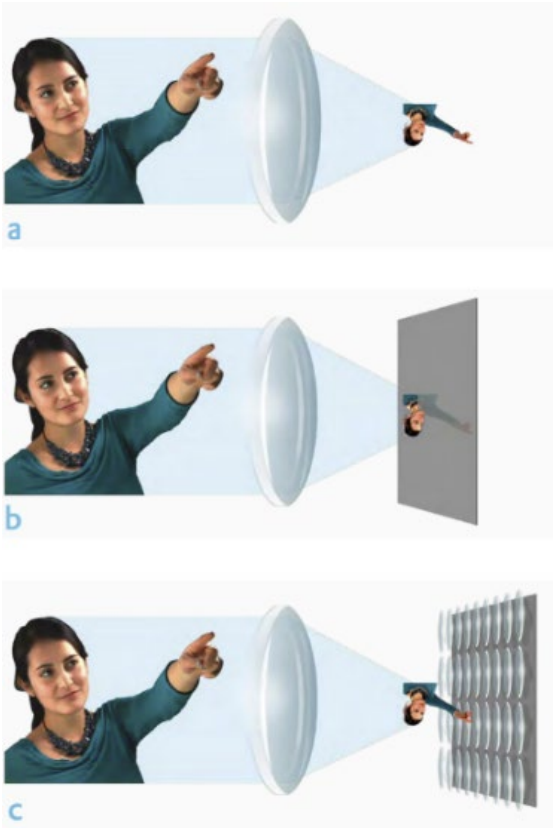


Figure 2: Simplified representation of Light-field cameras main characteristics

a) Camera lenses depict 3D: Contrary to our everyday experience with conventional 2D cameras, lenses inherently collect light rays from multiple directions, allowing them to capture a three-dimensional scene representation. Each incoming ray carries crucial information about intensity and direction, making depth perception possible.

b) 2D camera: In traditional 2D cameras, depth information is lost when the three-dimensional scene is projected onto a flat image sensor. The only indirect depth cues come from blurring, which results from parts of the scene being out of focus due to their offset from the sensor's focal plane.

c) Light-field cameras: A light field camera preserves depth information by placing the scene before a micro-lens array. Acting like thousands of tiny cameras, these micro-lenses capture the light field, encoding the direction and intensity of incoming light rays as shown in Figure 3. This wealth of data enables depth reconstruction and full 3D modeling from a single image.

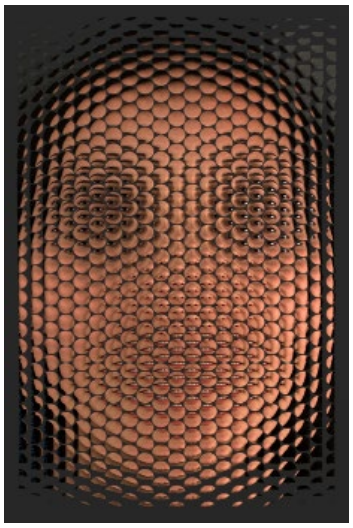


Figure 3: Representation of a micro-lens array (left), and a color image with 3D reconstruction obtained using Raytrix cameras (right)

Figure 4: Multi-camera setup built with Raytrix R32 cameras and an edge computing system

The main goal of WP2 is to achieve high-quality, full-body human reconstructions. Therefore, the capture setup must incorporate multiple cameras to cover all possible viewpoints of the subject. We developed a multi-camera light-field setup optimized for real-time 3D video capture and reconstruction. This system features four R32 light-field cameras arranged in a 3.5×3.5m square around a 2×2×2m capture area, ensuring comprehensive coverage for accurate volumetric reconstruction, as shown in Figure 4. It is connected to an edge computing system that processes and reconstructs the light-field data. The system synchronizes and generates four RGB+D (color + depth) streams, which are then forwarded to T2.2's volumetric reconstruction pipeline.

Following discussions with WP4 (Virtual Humans), the original symmetrical camera placement, with four cameras equally spaced at 90-degree intervals, was modified to enhance facial expression capture. The updated configuration now places three cameras at 120-degree intervals, while the fourth camera is dedicated to capturing the user's face in high detail, as shown in Figure 5. This adjustment ensures that even subtle facial expressions remain recognizable while providing sufficient body language detail for realistic 3D reconstruction.

At the time of writing, two initial camera sets were provided to i2CAT and CERTH early in the project to facilitate integration and testing. These early deployments enabled valuable insights into system performance and integration challenges. Based on the lessons learned, a second design iteration has been developed, and a set of updated cameras is available as of M12. This updated design significantly improves the capture quality.

The Volumetric Capturer relies on high-bandwidth, low-latency connectivity to enable real-time data transmission and 3D reconstruction. Each of the four R32 light-field cameras is connected to a PCIe frame grabber card in the edge computing unit via CoaXPress (CXP)-12, a high-speed serial communication standard optimized for imaging applications. A timing FPGA on the frame grabber card ensures precise image capture synchronization across all cameras. The cameras transmit raw light-field data over single coaxial cables, which support distances of up to 30 meters while maintaining full bandwidth (12.5 Gbps per channel).

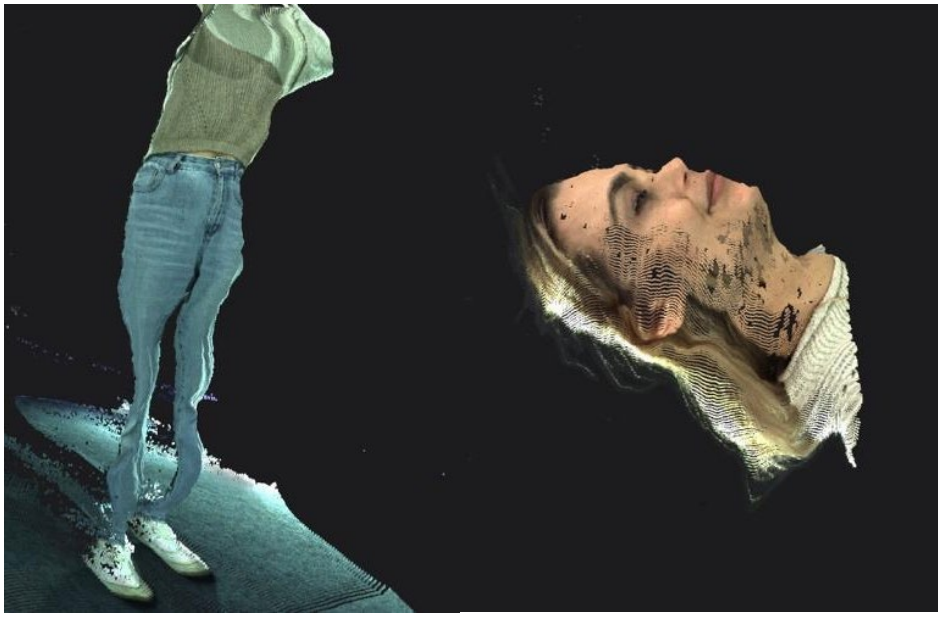


Figure 5: Screenshot of Raytrix's demo application presented during the General Assembly on January 22, 2025. The left image displays the frontal body camera output, while the right image shows the face camera output.

The edge unit is ideally equipped with one NVIDIA HPC GPU per camera, such as the NVIDIA RTX 4090, though multiple cameras may share a single GPU. To manage computational workloads efficiently, the light-field processing system spawns Image Processing Units (IPUs) that convert raw light-field data into synchronized RGB-D (color + depth) streams. These streams undergo intrinsic calibration (ensuring accurate depth and scaling) and extrinsic calibration (registering all cameras into a global coordinate system). The calibrated data is then fused into a single volumetric stream within T2.2's volumetric reconstruction pipeline, preparing it for further processing and delivery to XR applications (T2.3). The capture setup was successfully showcased during the third General Assembly on January 22, 2025. The demo setup included a custom application for easily testing and showing different capturing configurations, as shown in Figure 6.

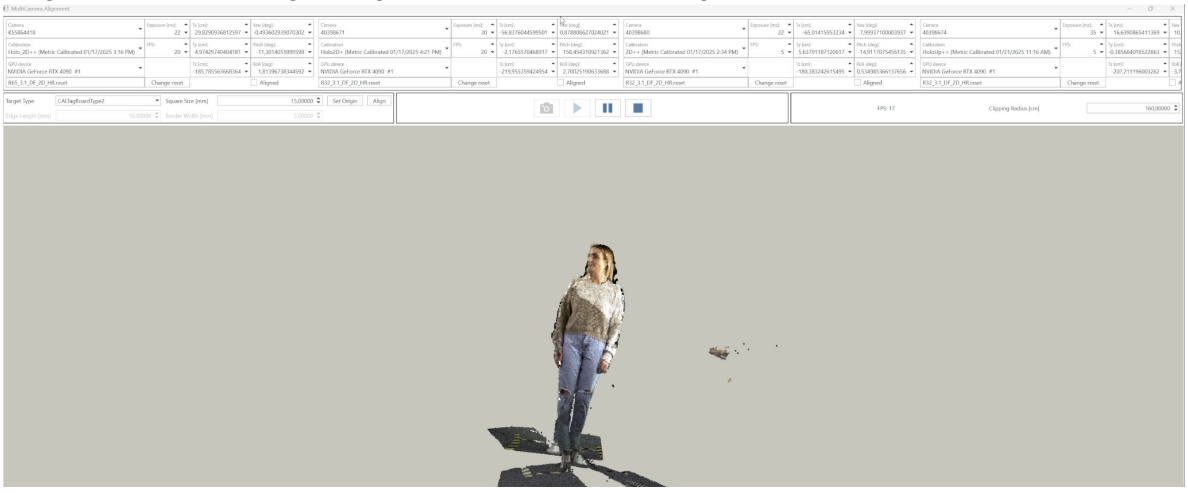


Figure 6: Screenshot of Raytrix's demo application during the January 22, 2025 General Assembly demonstration, for a 4 cameras setup

This document includes a datasheet for the R32 3D light-field camera as Annex A.

4.1.1. Development and Key Achievements

This deliverable does not aim to give a comprehensive introduction to the optical engineering of light-field cameras, but it will provide a brief overview to contextualize the developments and achievements of Task 2.1, which can be summarized as:

New Camera Model and Intrinsic Calibration

Wide-angle lenses are crucial for holoportation capture setups because they provide a larger field of view (FoV) in confined spaces. This allows the entire human subject to be captured without needing prohibitively long camera distances. However, these lenses also bring several challenges that Task 2.1 aims to address.

Wide-angle lenses must bend light more aggressively to cover a larger FoV, necessitating a stronger curvature of the lens elements or using materials with a higher refractive index. Both approaches can lead to optical errors known as aberrations. Additionally, the rear lens element is often positioned closer to the image sensor, further amplifying these aberrations for the same reason.

Typically, wide-angle lenses feature larger lens elements, especially in the front. As these elements grow, stricter manufacturing tolerances are required to ensure consistent optical performance. Even with high production quality, minor defects can have a more significant impact due to the increased variations in the optical path across the lens.

Even if a wide-angle lens is well-corrected like its longer focal length counterparts, its shorter focal length will amplify the impact of any aberration. In optical systems, the relationship between the object distance (d_o), the image distance (d_i), and the focal length (f) is given by the thin lens equation:

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i}$$

If f is small, small errors in the image space, can translate to larger errors in the object space, as shown in Figure 7. As discussed above, light-field cameras measure depth in image space.

To understand why, we can also rearrange the thin lens equation to express the image distance in terms of the object distance:

$$d_i = \frac{f\, d_o}{d_o - f} \quad f \ll d_o \rightarrow \frac{f\, d_o}{d_o} = f$$

This formula shows that when the focal length f is small (as in wide-angle lenses) and significantly smaller than the object distance d_o, the image distance d_i will asymptotically approach f and become increasingly insensitive to d_o. This has the counterintuitive effect that any aberration in image space is large compared to the change of image space distance Δd_i and thus gets magnified in the object space. In contrast, lenses with a longer focal length exhibit a weaker sensitivity to changes in d_i meaning that the same magnitude of errors in image space results in smaller errors in object space.

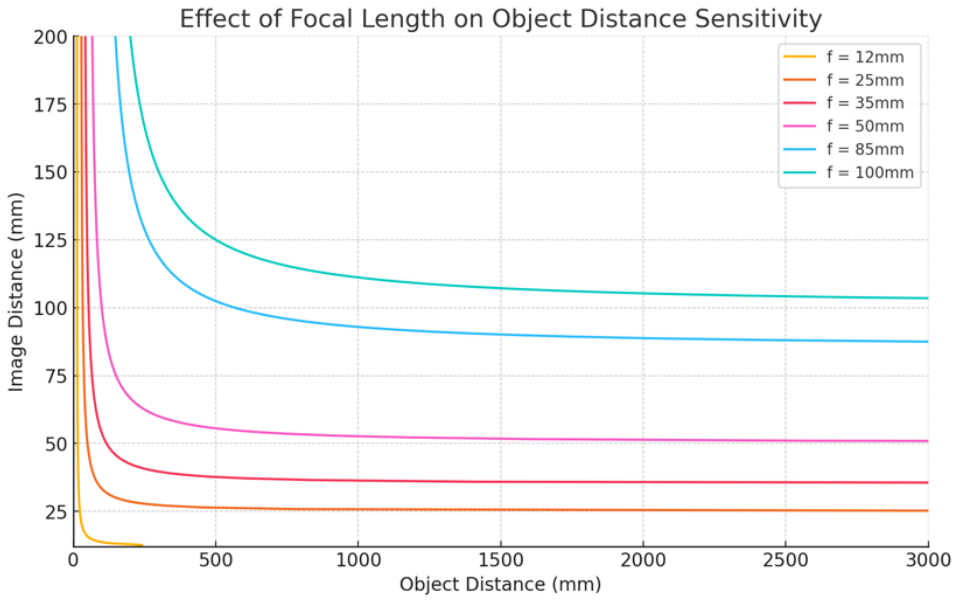
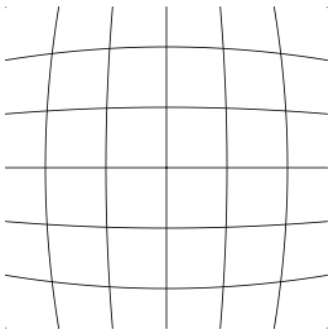


Figure 7: Relationship between the object distance (in mm) and the image distance (in mm)

Conventional correction methods based on 2D imaging do not easily adapt to light-field 3D imaging, as aberrations gain a dependence on depth. In T2.1, we developed a new camera model and an intrinsic calibration method to correct the most significant aberrations in wide-angle 3D light-field imaging. A summary of these aberrations, effects and potential corrections is described in Table 1.

Aberration	**Effect in 2D Images**	**Effect in 3D Light-Field Images**	**Corrected by state-of-the-art camera model?**	**Corrected by PRESENCE camera model?**
Distortion	 Straight lines appear curved.	Lateral positions (X, Y) become increasingly distorted toward the edges of the frame. If uncorrected, extrinsic calibration is impossible.	Yes	Yes

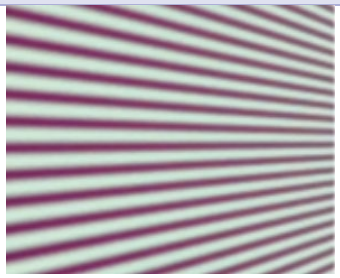
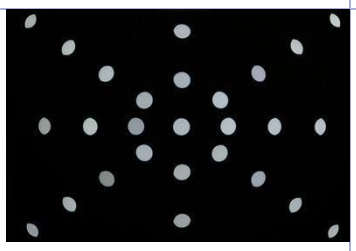
Aberration	Effect in 2D Images	Effect in 3D Light-Field Images	Corrected by state-of-the-art camera model?	Corrected by PRESENCE camera model?
Chromatic Aberration	 Color fringing at high-contrast edges.	Different wavelengths refract at different angles, causing depth (Z) to become color-dependent. Intrinsic and extrinsic calibration are only valid for one wavelength if uncorrected.	No	No
Mechanical Vignetting	 Off-center bokeh appears as cat-eye shapes	Light rays entering at an angle are partially blocked by the lens aperture or barrel. This introduces partial occlusion of ray bundles, leading to depth (Z) distortions. If uncorrected, extrinsic calibration is impossible.	Partially	Yes
Field Curvature	Image sharpness decreases toward the edges	The lens maps image space onto a curved surface, causing depth (Z) distortions. If uncorrected, extrinsic calibration is impossible.	Yes	Yes
Astigmatism	Horizontal and vertical lines blur differently	Depth (Z) varies depending on the orientation of a line (horizontal vs. vertical). This effect is noticeable in high-contrast calibration targets, making intrinsic calibration difficult.	No	Yes

Table 1: Description of observed aberrations, their effect on the quality and implemented corrective solutions.

Although their individual contributions are marginal, these corrections collectively lead to a significant improvement in calibration accuracy. As Figure 8 and Figure 9 illustrate, the average root-mean-squared (RMS) error of the body camera calibration compared to ground truth data using the state-of-the-art model is approximately 45 mm. The new calibration method reduces this error to 10.8 mm for the worst-performing body camera and 8.3 mm for the best in the set.

One focus of T2.1 moving forward will be to incorporate higher-order wavefront errors into this correction model, particularly those corrections that address lenslet misalignment issues stemming from the lens manufacturing process.

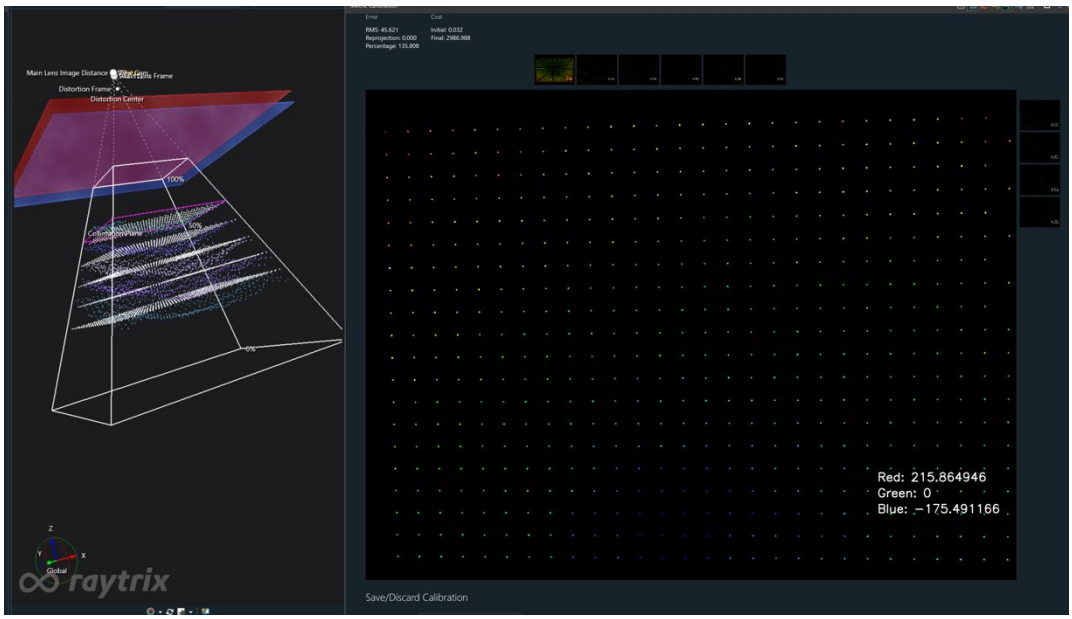


Figure 8: Calibration results of the state-of-the-art model for an R32 body camera. Left: Calibrated model points (colored dots) vs. ground truth (white dots). Right: Color-coded residuals of the calibration. The average RMS error over the measurement volume is 45.6 mm.

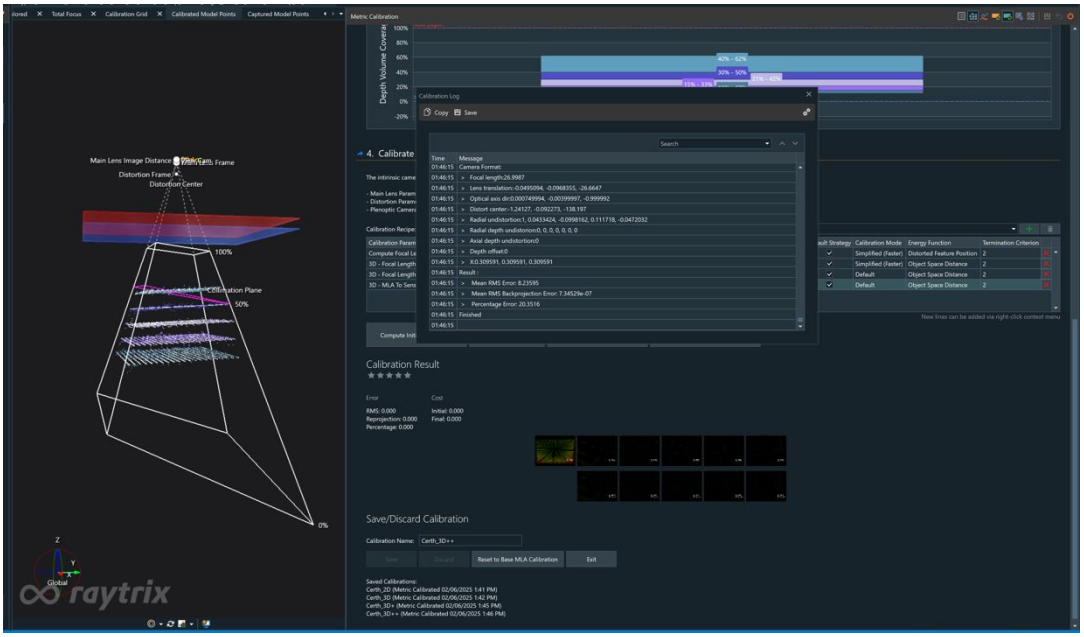


Figure 9: Calibration results of the new model for an R32 body camera. Left: Calibrated model points (colored dots) vs. ground truth (white dots). Right: Color-coded residuals of the calibration. The average RMS error over the measurement volume is 8.3 mm.

New Chromatic Aberration Method

Table 1 shows that chromatic aberration remains uncorrected even in the new calibration model. Instead, as part of T2.1, this correction has been integrated into the light-field depth algorithm. Depending on computational resource availability, two approaches can be chosen:

Fast Processing: To optimize speed, depth calculation can be performed using a single color channel, effectively bypassing the impact of chromatic aberration on depth estimation. The resulting depth data is combined with a fully corrected, full-color 2D texture.

High-Quality Processing: If sufficient computational resources are available, depth calculation can be performed separately for each color channel. These depth maps are then recombined using full 3D chromatic aberration correction, producing denser and higher-resolution depth maps at the cost of increased computational demands.

New Matching Algorithm

The depth algorithm has been further modified to address the challenges of capturing humans for holoportation. Like stereo vision, Light-field cameras are passive 3D sensors that capture angular and spatial information of light rays without emitting any signals, relying solely on ambient light. They estimate depth by analyzing local contrast and matching features across multiple viewpoints within the captured light field.

A light-field version of the ADCensus algorithm [Ref. 21] has been implemented as part of T2.1. It is a method that enhances depth perception by combining Adaptive Support Weights (AD) and the Census Transform (Census). This approach makes it particularly well-suited for scenarios where traditional matching struggles, such as low-texture surfaces, illumination changes, and depth discontinuities—all common issues when capturing human subjects.

ADCensus improves matching through a structured approach:

1. **Cost Computation**: The Census Transform converts local intensity patterns into robust binary descriptors, making depth estimation more resistant to illumination changes and low-contrast surfaces.
2. **Cost Aggregation**: Adaptive Support Weights dynamically assign different importance to neighboring pixels based on their similarity, helping preserve depth discontinuities and ensuring better matching on complex surfaces.
3. **Disparity Optimization & Refinement**: Refine the disparity map, reducing errors caused by reflective surfaces and varying textures.

Humans present unique challenges in matching due to clothing textures, skin reflectivity, and free-standing posture. ADCensus addresses these issues effectively:

Low-Contrast Clothing & Makeup: Many traditional algorithms struggle with dark or smooth clothing, but ADCensus leverages the Census Transform to extract structural information rather than relying on intensity values alone, ensuring better feature matching even in textureless regions.

Oily or Sweaty Skin Reflections: Specular highlights caused by light reflections can disrupt matching, but the Adaptive Support Weights help minimize their impact by prioritizing more reliable neighboring pixels.

Edge Preservation for Free-Standing Subjects: Since humans are often captured in open environments rather than against structured backgrounds, preserving edge details is critical. The ADCensus approach, particularly with adaptive weighting, ensures better depth continuity at object boundaries than traditional correlation-based methods.

New Depth Estimation Algorithm

T2.1 is developing a new depth estimation algorithm to improve the performance of light field cameras in wide-angle capture scenarios. This algorithm leverages the consistency condition of light fields, which has been shown to enhance depth estimation quality. However, as of now, the algorithm is still under development and is not yet fully capable of real-time processing; it currently operates at approximately 20 frames per second on an NVIDIA RTX 4090.

New Light-Field Camera Design & Design Tool

The resolution of a light field camera can be described as the product of a pinhole array and a (micro)lens. The conventional design philosophy of light field cameras aims to maximize resolution, achieving this when the (micro)lens resolution peaks while redundant imaging is minimized [Ref. 22].

The first generation of light field cameras in PRESENCE followed this approach. However, real-world testing quickly revealed a drawback: while the resolution was exceptionally high at the optimum, it dropped off significantly, creating challenges even for the ADCensus matcher, in low-contrast near-field areas. As a result, the design was revised.

To address this issue, larger microlenses were chosen to slow the increase in redundancy. While this would typically introduce more aberrations, the new calibration model effectively compensates for them. Additionally, the microlens focus was adjusted to counteract resolution loss caused by increased redundancy, prioritizing sustained resolution over peak resolution.

Since these effects are difficult to capture using conventional optical design tools like Zemax, a Blender-based simulator was developed to generate synthetic data and evaluate design performance.

CapturerDLL

As shown in Figure 10, the capturer pipeline must be seamlessly integrated with the reconstructor pipeline developed as part of task T2.2.

Following WP5's discussions on the personas of developers and users expected to utilize PRESENCE's tools, it became clear that expertise in GPU-based programming or managing multiple high-bandwidth streams with synchronization requirements cannot be assumed, even when fully functional example code is provided.

To address this, CERTH (T2.2) and Raytrix (T2.1) are collaborating on wrapping the light-field tools into a user-friendly and easily integrable capturer.dll, designed to mimic the behavior of more established sensors, such as the Microsoft Kinect. One of the key challenges is balancing ease of use with high performance. However, at the time of writing, the first version of capturer.dll is available,

and its API has been defined, which is attached as Annex B: *Capturer DLL API Specifications* to this document.

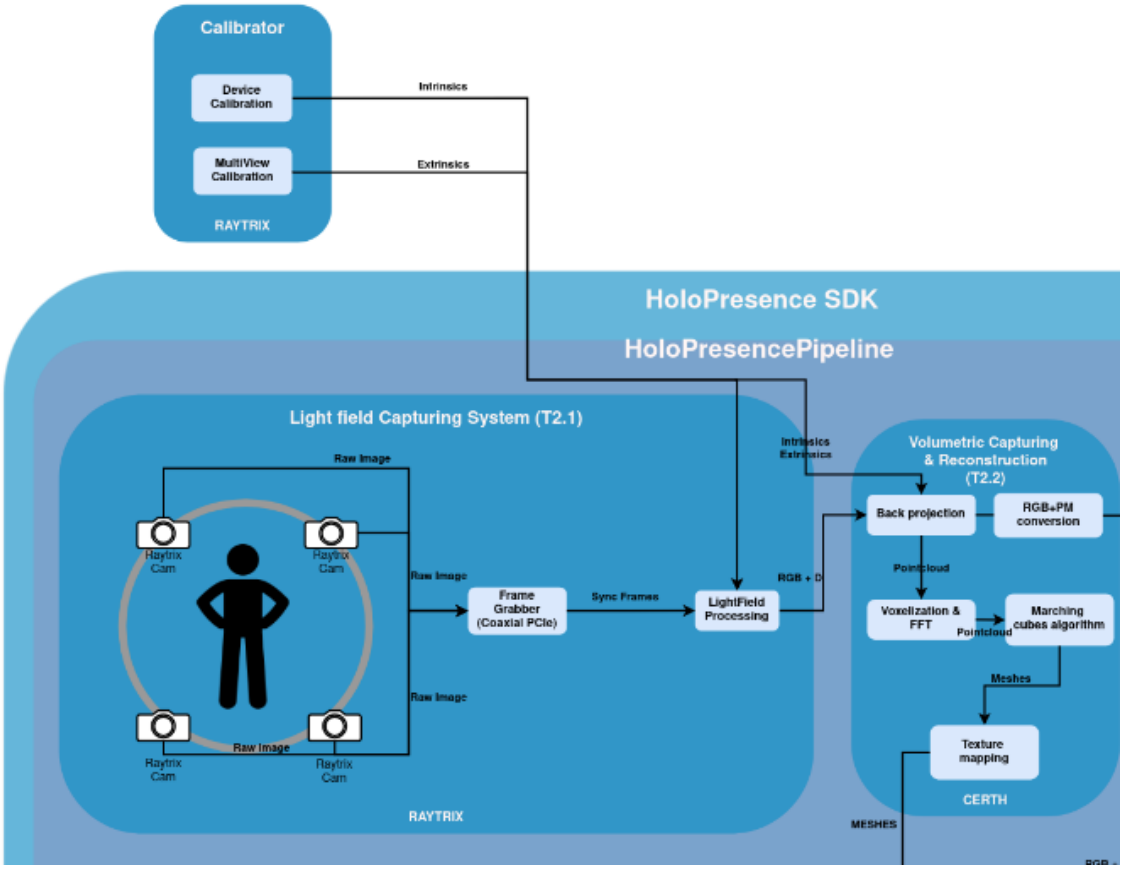


Figure 10: Diagram showing the interconnection between tasks T2.2 and T2.1

4.1.2. KPI Status

The KPIs relevant to T2.1 are KPI 2.1 and KPI 2.3:

- KPI 2.1 requires delivering a real-time holoportation system with photo-realistic quality, achieving user acceptance levels of ≥4 on a 5-point scale in subjective tests using validated questionnaires. While user acceptance will be evaluated later in the project, the real-time performance and photo-realistic quality aspects can already be assessed. As of writing, two R32 light-field cameras have been successfully demonstrated to perform light-field processing on a single NVIDIA RTX 4090 GPU at 30 fps while maintaining at least HD resolution. An example of the current quality we can achieve in our setup is shown in Figure 11.
- KPI 2.3 requires delivering one set of holoportation APIs to integrate holoportation with the two other pillars in the two demonstrators and in additional scenarios. T2.1's contributions to KPI 2.3 include the previously mentioned capturer.dll, the first version of which is now available.

Figure 11: Screenshot of a multi-camera head reconstruction using our light field capture setup.

4.2. T2.2 - Volumetric Representations

After the multiple light-field cameras of the system have captured different aspects of the user's scene (T2.1), the captured data is propagated to a Reconstructor module in order to be processed and to create a unified 3D volumetric model of the observed scene. There are two major types of volumetric representations: point clouds and meshes. In this mid-term report, the major focus is on the production of point clouds and their propagation further on to the compression module (T2.3) which compresses the point clouds for transmission. The production of meshes, which will be forwarded to the rendering module of the pipeline, will be covered extensively in a later stage of the development of the work package, although substantial preparatory work has already been done, which is reported in the *Volumetric Representations* part of the Developments & key achievements section below. In general, the development and integration goals of T2.2 are depicted in the schematic architecture shown in Figure 12.

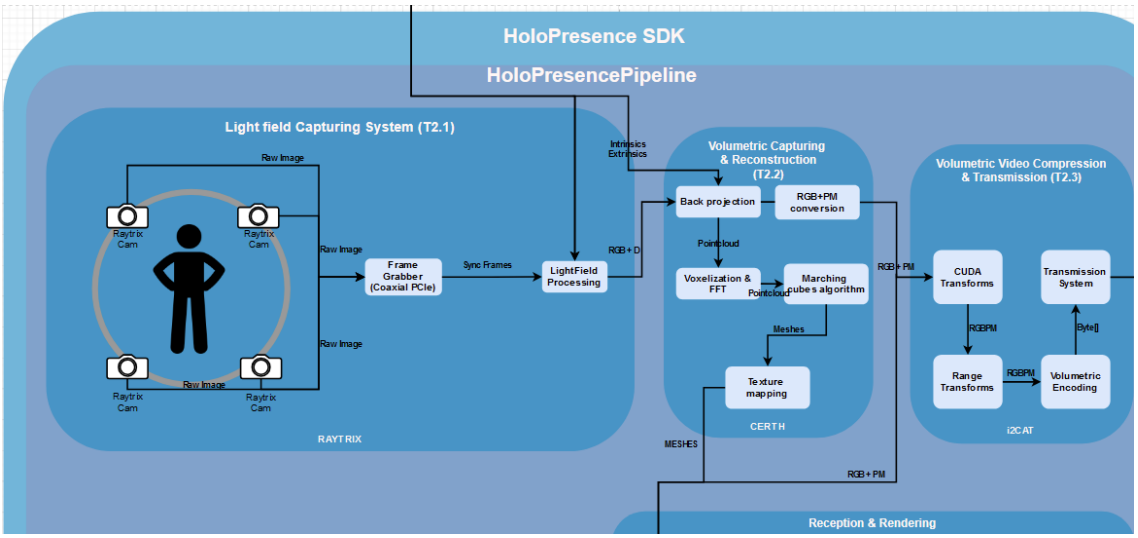


Figure 12: The HoloPresence pipeline showcasing the flow of data from capturing to the volumetric reconstruction module and further on its propagation to the compression and rendering modules.

4.2.1. Developments & key achievements

The T2.2 development process focused on two key aspects of volumetric capturing and reconstruction: multi-view camera extrinsics calibration (i.e., global pose estimation) and volumetric reconstruction techniques. The following sections will detail the efforts to develop a user-friendly extrinsics calibration method that improves accuracy, along with advancements in volumetric reconstruction algorithms.

Calibration

A preliminary step of reconstruction is performing the extrinsics calibration of the multi-view system. This ensures the alignment of the different camera views into a unified 3D reconstructed representation. In our current architecture, this procedure is not part of the HoloPresence SDK. It is a separate application which is executed once, while setting up the system. The Calibrator application then provides the intrinsics and extrinsics of each camera of the setup to the SDK.

One requirement of PRESENCE is the easy and quick procedure of calibrating the cameras. Towards this direction, we have implemented a Calibrator application via which the user of the Holoportation system can follow a very simple 2-step procedure (see Figure 13):

- Set up a 4-box structure in the middle of the scene
- Connect the cameras and let the application automatically calibrate them

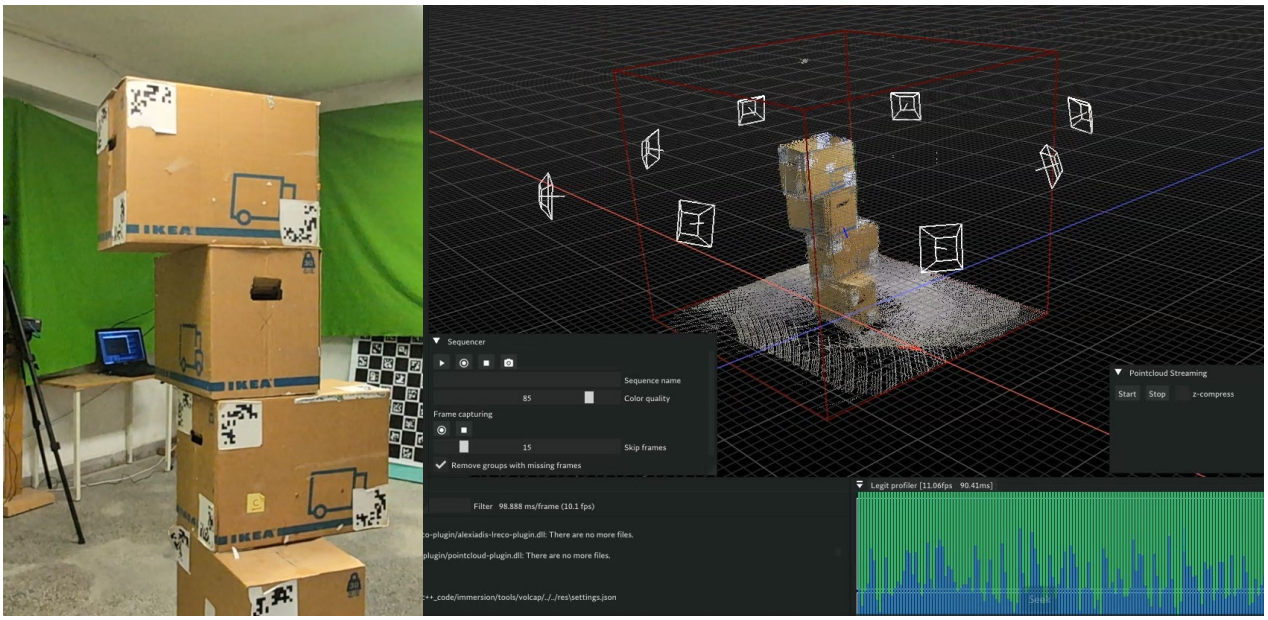


Figure 13: Real setup with the calibration boxes (left). Reconstruction software showing the fully reconstructed point cloud after the calibration step (right)

In more detail, the calibration pipeline depends on data-driven correspondence establishment (as opposed to marker-based correspondences) for the initial matching and global optimization. An enhanced semantic segmentation model with soft Procrustes analysis is used to identify the sides of the boxes. For every single-view depth capture of this structure, the 3D coordinates of the structure's keypoints are estimated (centroid of each identified side of the box) and establish 3D-point correspondences with the structure's virtual model. The virtual structure model serves as a global anchor for all views. Those keypoint correspondences are used to perform sensor pose

estimation with respect to the global coordinate system by using standard Procrustes analysis. These initial pose estimations are finally used by a dense optimization refinement algorithm (DenseCRF2D) and further on with a graph optimization algorithm (g2o) in order to achieve a precise global pose estimation. The calibration pipeline execution takes 2 min on average. In case the calibration needs some extra fine-tuning, a special GUI tool has been developed so that the user is able to correct minor calibration issues by hand.

Furthermore, the basis for a new calibration method has also been set to make the calibration process even easier for the user, by eliminating the first step and the need of any specific structure, checkerboards or other complex techniques for calibration. The goal is to obtain a good estimation of relative poses between pairs of cameras by using the 3D geometry of the scene. The proposed method is to use a learnable pipeline which is invariant to camera poses and to common office setups of a Holoportation scene. In such a scene, we expect to have low overlap regions of point clouds due to the sparse camera setup (e.g. 4 light field cameras in a 3x3 scene). To tackle this issue, we have been experimenting with geometry enhanced fully convolutional feature extractor networks (KPFCN [Ref. 23]) and transformers with self and cross-attention [Ref. 24] in order to enhance the feature extraction and optimize the point cloud registration. The first results of geometry enhanced and learned features are promising as we managed to increase at about 40% the ratio of the inliers of feature matching for the low overlap regions, which leads to superior registration.

Volumetric Reconstruction

The initial volumetric capturing and reconstruction system employed a real-time point cloud and a mesh volumetric reconstructor. Thiis reconstruction algorithm uses a volumetric Fourier Transform-based reconstruction method with GPU acceleration to achieve real-time 3D human shape reconstruction. After the depth data acquisition from multiple RGB-D sensors, the raw 3D points are extracted using backprojection, and their surface normals are estimated for accurate geometry representation. Confidence weights are assigned based on viewing angles and edge proximity to improve robustness. The 3D scene is then voxelized, and a 3D Fast Fourier Transform (FFT) is applied to construct a smooth gradient field. After filtering and integration in the frequency domain, an isosurface representing the reconstructed object is extracted using the Marching Cubes algorithm, converting the volumetric representation into a triangle mesh. To enhance visual realism, Multi-view Texture Blending is applied, combining colors from different camera angles while minimizing artifacts. The entire process is parallelized using CUDA GPU computing, ensuring near real-time high-quality 3D reconstruction [Ref. 25]. While this approach achieved reasonable performance (see Section 4.2.2 KPIs Status metrics below), it proved insufficient for dynamic computations and exhibited limitations in reconstructing certain geometrical features, particularly in lower limb regions.

To address these limitations, we have prototyped a new reconstruction method based on fast winding number calculation. This approach interprets captured point clouds as surface samples and determines whether each voxel in a grid lies inside or outside the reconstructed surface. The method follows an electrostatic field analogy, as proposed by Barill et al [Ref. 26], to compute winding numbers for each voxel.

To improve efficiency, we employ octree structures and approximate k-Nearest Neighbors (k-NN) algorithms to group winding numbers, reducing the time complexity from $O(m * n)$ to $O(m * log(n))$.

This allows for significantly faster processing while maintaining high reconstruction accuracy. Additionally, the implementation retains the use of Marching Cubes for mesh extraction and blending texture mapping techniques for improved visual quality. The overall execution time of the algorithm was benchmarked at 19ms i.e. ~52fps, using an RTX4090 NVidia GPU and capturing RGB-D data from 4 cameras at 720p color resolutions. The process is depicted in Figure 14.

The significantly improved execution speed of this algorithm has enabled the potential integration of additional enhancement techniques, including live smoothing and embedded deformable graphs, which have the potential of resulting in superior reconstruction quality. Another advantage of the new reconstruction algorithm is that it is highly parameterized and these are live-adjustable parameters which speed up the experimentation in order to achieve superior reconstruction quality.

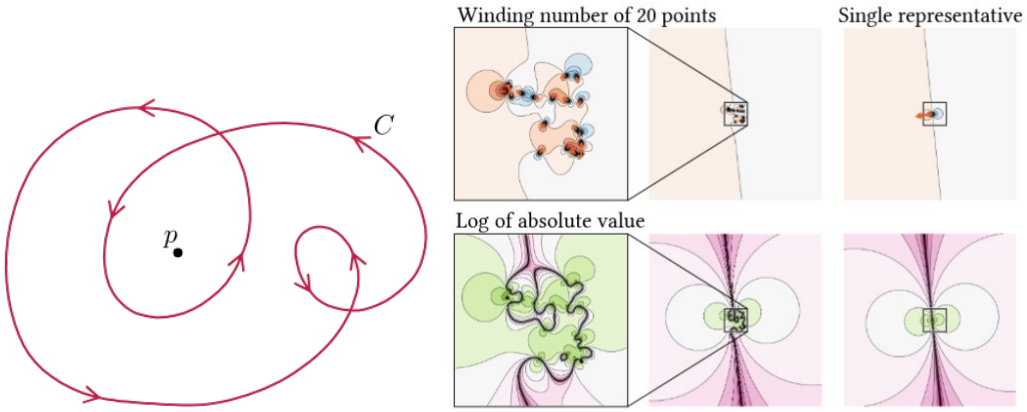


Figure 14: A simple example of the calculation of the winding number of a curve around a point p (left). The method that is followed in order to reduce the time complexity of the algorithm assumes that e.g. 20 points will have the same winding number as the single representative (right).

Figure 15: Real-time winding numbers reconstruction (left) and offline (right) using higher quality parameters.

4.2.2. KPIs Status

Task 2.2 is directly related to the following KPIs:

> KPI 2.1: *Deliver a real-time holoportation system with photo-realistic quality:* Deliver a real-time holoportation system that achieves users' acceptance levels of ≥4 on a 5-point scale in subjective tests using validated questionnaires (e.g., IPQ, Mon22 [Ref. 27]).

This KPI includes several goals within it: photorealism and real-time objectives.

The real-time aspect has been achieved for the point cloud volumetric representation as the reconstruction algorithm benchmarked an impressive 5ms process time per frame. The mean number of vertices for one human in the scene were 50-60K and the maximum at around 370K. As point clouds are easy to compress, this will benefit downstream activities of the HoloPortation pipeline.
On the other hand, point clouds lack photorealism and a mesh representation should be used to compensate for that. For the mesh representation a near real-time benchmark has been achieved at ~40ms execution time per frame (~25 fps) and it is anticipated to drop more as duplicate vertices will be properly handled in the future.

All the results along with the pros and cons of each representation are shown in Table 2.

Reconstructor DLL Metrics (4 cameras)						
Representation	**Execution time (ms)**	**fps**	**Vertices**	**Pros**	**Cons**	**Notes**
Point cloud	5	>>30	• mean=50-60K (one human) • max~370K	• Very fast • Easy to compress • Very simple Unity shader is needed	• Needs very accurate calibration • Photorealism is low	
Mesh	40	25	• expected mean after removing duplicates ~40K • max = 2^{22} (theoretical)	• Unity integration • Enhanced photorealism • Handling occlusions in geometry	Custom complex Unity shader	Goal: 33 ms execution time

Table 2: Performance and evaluation summary of the reconstructor DLL

KPI 2.1 dictates that a quantitative evaluation should take place where end-users will be provided with questionnaires to assess their experience of the Holoportation system and more specifically for T2.2, the volumetric **reconstruction quality** should score an acceptance levels ≥4 on a 5-point scale, in subjective tests using validated questionnaires (e.g. IPQ, Mon22). An evaluation protocol similar to the one used in Mon22 [Ref. 27] was finally chosen, albeit slightly modified for the purposes of the Holoportation virtual environment.

The evaluation protocol designed to assess photorealism in a real-time Holoportation System utilizes validated questionnaires to gauge user perceptions of 3D human representations across various scenarios. The participant pool will include individuals preferably with experience in virtual reality (VR) and/or CGI imagery composed out of a mix of developers, researchers, students and general

VR enthusiasts, targeting a sample size of 15-30 users to ensure robust statistical analysis. The experiment will capture different actors performing a variety of movements under diverse lighting conditions and the participants themselves in the HoloPortation scene. Both head-mounted displays (HMDs) and screen views/videos will be employed to evaluate user experiences in immersive and non-immersive contexts. Participants will also experience two visual representations separately, point clouds and meshes with textures, to determine their perceptions of photorealism in each scenario.
Participants will evaluate the visual representations based on a 5-point Likert scale, focusing on attributes such as overall visual quality, surface quality, geometry, motion consistency, depth perception and spatial presence. Key factors for assessment include the photorealism of the holoported individuals, the intricacy of their features, the accuracy of the lighting, the naturalness of their movements and the sense of 3D depth and volume. Specific statistics will also be extracted for the different scenarios (immersive vs. non-immersive) and representations (point clouds vs. meshes with textures) in order to determine the higher photorealism ratings.

The goal is to achieve an average user acceptance score of at least 4 on the scale, signifying a satisfactory level of photorealism in the tested volumetric reconstructed representations.

4.2.3. Deviations & Mitigation Plan

The integration of the capturing module of T2.1 with the reconstruction module of T2.2. experienced a slight delay, primarily due to the significant technical complexity of aligning captured data and volumetric reconstruction technologies along with their respective data representations. These technical hurdles were compounded by communication gaps between the work package teams which further complicated the integration process. By implementing more rigorous and focused technical synchronization meetings and establishing detailed interface specifications and API's between the two tasks, the team successfully navigated these challenges and set a robust course of action for concluding the integration process.

4.3. Holoportation pipeline: Volumetric compression and streaming

Volumetric media transmission requires an enormous amount of data to accurately represent 3D objects or scenes, for example point cloud videos with one million points requires up to 5 Gbps which makes them almost impossible to stream in real-time. In contrast to traditional video where 2D frames are used to display a fixed point of view, volumetric video extends the field of view allowing experiences with 6 degrees of Freedom (6DoF). However, adding an extra dimension makes the data compression and transmission a lot more complex, more resources are required and streaming geometry is more difficult than colors.

The Moving Picture Expert Group (MPEG) has been developing compression standards, such as Video based PCC (V-PCC) [Ref. 28], obtaining good efficiency in the bandwidth reduction of volumetric data, however lacking good real-time performances. The proposed solution addresses the issues previously discussed providing a high performance solution for real-time applications. In the scope of this task, we have developed a volumetric video compression system designed to significantly reduce the amount of data that is required to accurately represent 3D point clouds, this is possible by transforming geometry into colors to leverage 2D video codecs.

Combining color with depth (RGBD) is an image-based technique which is mainly used to represent height-maps and simulate terrains in real time. However, it is possible to adapt this technique to

produce volumetric content. Instead of encoding displacement in a gray-scale map, it is possible to use a combination of the intrinsic parameters of the device and the depth to project the 3D geometrical data into a 2D image which maps the xyz values to the RGB channel, as described in [Ref. 29].

A complete volumetric representation requires three key inputs. The first is the color frame, stored as a three-channel image, where each channel has an 8-bit depth (values ranging from 0 to 255). The second input is a grayscale depth map, represented as a single-channel image with a 16-bit depth (values ranging from 0 to 65,535), where the values are linearly mapped. The third input is an array of floating-point values that store the camera's intrinsic calibration and distortion parameters [Ref. 29]. This array, known as the XYTable, ensures accurate positioning of the geometry in three-dimensional space.

Each input undergoes specific transformations throughout the processing pipeline. First, the color resolution is adjusted to match the depth frame resolution provided by the device, ensuring a one-to-one correspondence between color and depth data. The adjusted color image is then stored in an auxiliary buffer and used for both local and remote point cloud representations. Meanwhile, the depth frames are saved as 16-bit grayscale images for efficient storage and processing. Lastly, the XYTable is generated from the intrinsic and distortion parameters, creating a mapping that defines the x and y positions for each depth value. This table, combined with the depth image, is used to compute the volumetric representation of the hologram.

These transformations capture both the color and geometry of the point clouds. However, the resulting values are typically represented as real numbers, which are not directly compatible with standard video codecs, as these codecs are designed to compress 8-bit integer values for color information. This incompatibility arises because video codecs, such as those used for image and video compression, are optimized to handle discrete, integer-based values, not continuous, real-valued data.

To address this limitation, an additional step is applied to the geometric data. Specifically, the 3D coordinates (x, y, z) of the point cloud are split into three separate images, each representing one coordinate (x, y, or z) as an 8-bit channel. By doing so, each geometric coordinate is stored in 8 bits per channel, resulting in a 24-bit representation for the 3D geometry (8 bits per coordinate). This allows the geometric data to be encoded in a format compatible with video codecs while still preserving the necessary precision for reconstruction.

Figure 16 illustrates the resulting images, showing how the x, y, and z components of the point cloud are stored across three separate channels, enabling efficient storage and compression for video processing while retaining critical geometric information.

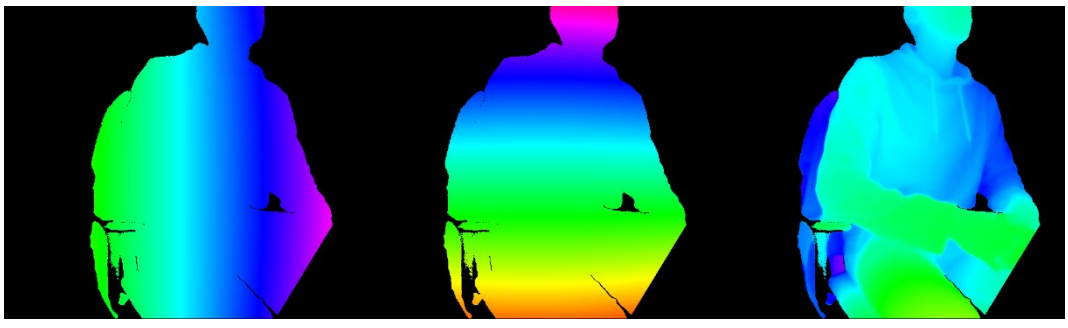


Figure 16: Image representation of the geometry, each image is used to represent the range of values of the points (x,y,z)

The compression system was tested with different parameters and configurations to determine which ones provided the highest quality and performance using different video codecs and parameters.

Figure 17 shows the results of testing different codecs as well as different parameters and they compare the following information: Constant Rate Factor (CRF), which describes the amount of compression applied to each frame, it ranges from 0-51. The lower the value the less compressed the frames are. BPV (bits per voxel) which represent the bits needed to represent a voxel for each frame, the lower the value the better. We tested the PSNR to measure the rate distortion of the resulting point cloud compared to the source. The results are also summarized in Table 3.

CRF/QP	BPV h.264	PSNR h.264	BPV h.264	PSNR h.265	BPV VPCC	PSNR VPCC
2	39.63014	32.9264	54.3388	37.9812	3.3868	62.5332
16	18.2548	28.3902	35.5809	36.5986	3.2763	62.3396
25	8.6370	26.1987	24.1586	35.2487	2.9933	61.1207
34	3.8383	26.7947	0.1013	13.6883	2.74090	56.9029
48	0.1007	32.0881	0.0889	2.3654	0	0

Table 3: Results summary of performance and quality loss compression tests

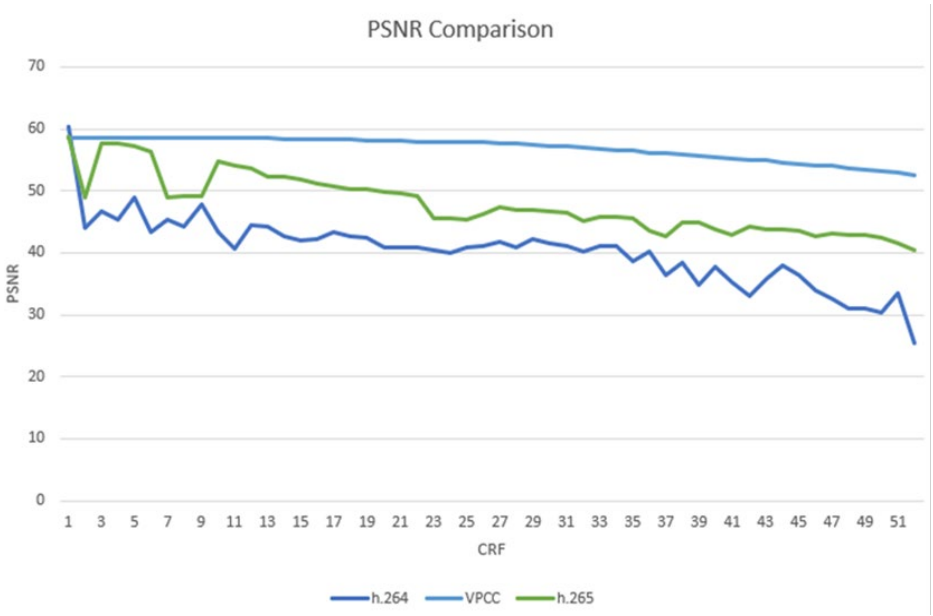


Figure 17: PSNR results for different encoding algorithms and CRF values

Figure 18 shows the achieved quality in terms of PSNR and bits per voxel. Overall, the quality is comparable to VPCC in several cases. However, the performance in terms of processing time is much better for our compression method. For instance, compressing a single frame can take VPCC several hours, whereas our method is sometimes 2 orders of magnitude faster, allowing real-time performance.

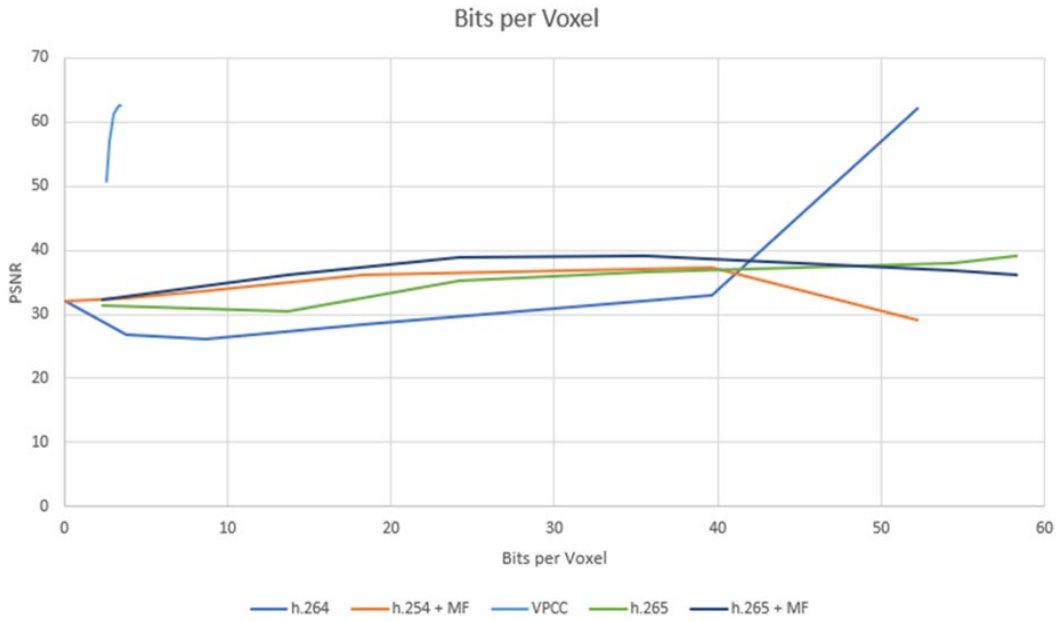


Figure 18: Comparison in terms of PSNR and Bits per Voxel of our pipeline using H.264 and H.265 versus the volumetric video compression standard (VPCC).

Figure 19 shows the results of adjusting certain compression parameters, on the left column the original input is shown, the second column shows the results of encoding a volume determined by a bounding box, the third column shows how the compression looks like when the bounding box fits the view frustum, and finally the last column shows the results of adjusting the range of values to a bounding box that fits the captured human. Next, the rows show the results of applying those parameters with different levels of CRF, top row is CRF 0, the middle row sets the CRF to 17 and the bottom row applies the highest compression value of the experiment, CRF 40. It is important to note that the quality preserves and those values can be set and adjusted depending on the network conditions. Those details are further explained in section 4.3.1.

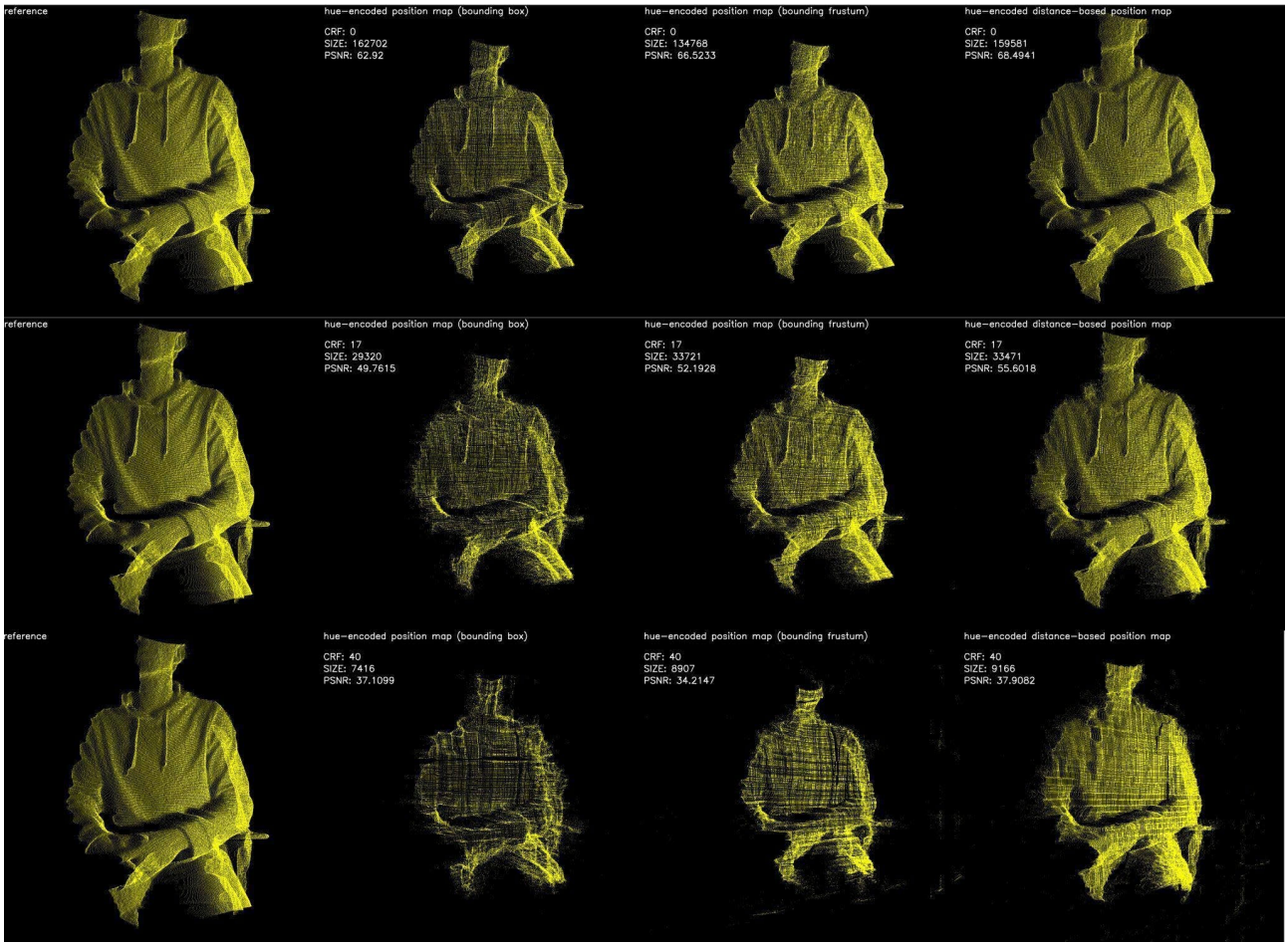


Figure 19: Columns: (first) original input, (second) results of encoding a volume determined by a bounding box, (third) how the compression looks like when the bounding box fits the view frustum, (fourth) hows the results of adjusting the range of values to a bounding box that fits the captured human. Rows: (top) CRF 0, (middle) CRF 17, (bottom) CRF 40.

4.3.1. KPIs Status

Task 2.3 contributes directly to KPI 2.1, which aims to deliver a real-time holoportation system with photorealistic quality, achieving user acceptance of at least 4 on a 5-point scale through subjective tests using validated questionnaires. The progress made in volumetric compression and streaming significantly enhances both real-time performance and visual quality of the system, which are essential for meeting this KPI.

By reducing the data rate required for volumetric data transmission, Task 2.3 ensures the holoportation system can operate in lower bandwidth or higher latency environments without sacrificing quality. This optimization enables seamless communication even under less ideal network conditions, directly improving the user experience.

The compression pipeline developed in T2.3 is crucial for preserving the photorealistic quality of 3D data while minimizing bandwidth requirements. It strikes a balance between reducing file size and maintaining texture fidelity and depth accuracy, key to achieving high visual quality. During the subjective tests for KPI 2.1, the impact of compression on the perceived quality of the 3D models will be evaluated, including sharpness and realism.

Ultimately, this work ensures that the holoportation system provides a high-quality user experience, even in challenging network environments, contributing to the achievement of user acceptance and meeting the photo-realistic quality target.

4.4. T2.4 - Multi user and scalable Holoconferencing

At project's start, a fully-functional multiuser holo-conferencing platform (i.e., HoloMIT, by i2CAT) was available [Ref. 4][Ref. 20], although with significant adaptability and scalability limitations. Within the context of T2.4, a set of improvements and extensions are being devised to overcome such limitations, thus pushing further the state-of-the-art in the field of holographic communications. The main associated developments, achievements and KPIs are described next.

4.4.1. Developments & key achievements

Holo-Orchestrator and Selective Forwarding Unit (SFU):

At the project's start, the server-side module of the holo-portation platform by i2CAT, i.e. HoloMIT, adopted a monolithic and single-package software implementation, denoted as the Orchestrator (Figure 20). By relying on *Node.js*[3] and *Socket.io*[4], the departing Orchestrator implemented different relevant features, like user management, session management, connection management, and media forwarding, this latest feature by means of a basic Selective Forwarding Unit (SFU). Although fully functional, it was clearly not the most adequate and efficient implementation, as it mixed control and data plane functionalities within the same software piece, and it lacked versatility for allowing an efficient scalability and dynamic instantiation based on the needs for/from specific holo-portation sessions.

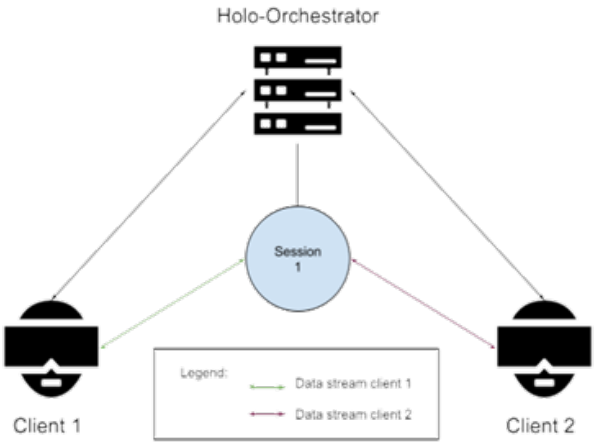


Figure 20: Departing monolithic architecture and implementation of the Holo-Orchestrator at PRESENCE's start

[3] https://nodejs.org/
[4] https://socket.io/

Therefore, such a server-side component has been significantly evolved and improved in PRESENCE, by adopting a modular and decoupled implementation (Figure 21).

The main evolution and refinement of the Holo-Orchestrator resides in the de-coupling of the Control and User Plane functions. This allows having single instances of reusable and more general services for the Control Plane, e.g. statically deployed on the Cloud, while having dedicated instances of other more stringent User Plane functions for specific sessions, on the Cloud or even on Edge servers, this serving as a first *Scalability Enabler (SE1)* for multiuser and multi-session holo-portation / holo-conferencing.

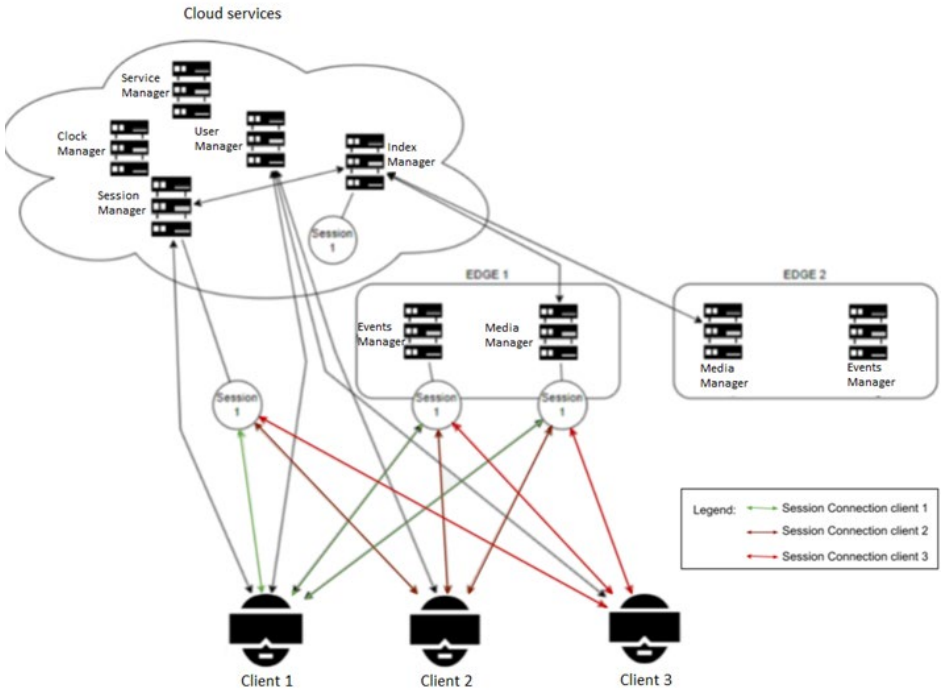


Figure 21: New modular and decoupled architecture of the Holo-Orchestrator in PRESENCE

The main Control Plane functions from the Orchestrator are briefly described next:

- User Manager: it is in charge of registering, managing and offering information / data from registered clients, scenarios and other in-cloud components, by using a MongoDB database. By using the holo-portation platform evolved in PRESENCE by i2CAT (i.e. HoloMIT), users need to be logged in the platform before creating / joining a session, and then they must select a virtual scenario on which the session will be established (e.g., a virtual meeting room, a museum).

- Session Manager: it is in charge of managing the lifecycle of multi-user sessions (i.e., creating, joining, leaving and eliminating sessions) for each involved user/client and for each selected virtual scenario, by storing the associated information on a MongoDB (it can be the same one as for the User Manager). It is also in charge of interfacing other services of the Orchestrator, like the Clock Manager and the Index/Connection Manager (both introduced next), to be able to select the most appropriate server-side function(s) to handle the communications for each session (i.e., SFU, transcoding service), the in-cloud servers where

to instantiate them, and then communicate this information to the involved clients of the target session.

- Clock Manager: it is in charge of ensuring a coherent notion of time to all involved entities in the session. It can act as a clock source against which to synchronize to, or it can just provide a reference to a Network Time Protocol (NTP) server.
- Index/Connection Manager: it is in charge of selecting the most appropriate location where to deploy in-cloud media functions for communication (i.e., SFUs) and/or transcoding services, and managing their lifecycle.

Likewise, the User Plane media functions from the original Orchestrator have been decoupled as self-contained SFUs for purely managing media forwarding functionalities between origin and destination holo-portation clients. As a *second Scalability Enabler (SE2)*, the SFU has been further decoupled to be able of independently managing the forwarding of audiovisual information (i.e. volumetric video and audio streams), by means of a Media Manager, and of control / events information (i.e., information about viewports, positions, interactions…), by means of an Events Manager. Both modules of the SFU have been built on top of Node.js and manage the exchange of streams from origin to destination clients via TCP WebSocket connections by using socket.io, as illustrated in Figure 22.

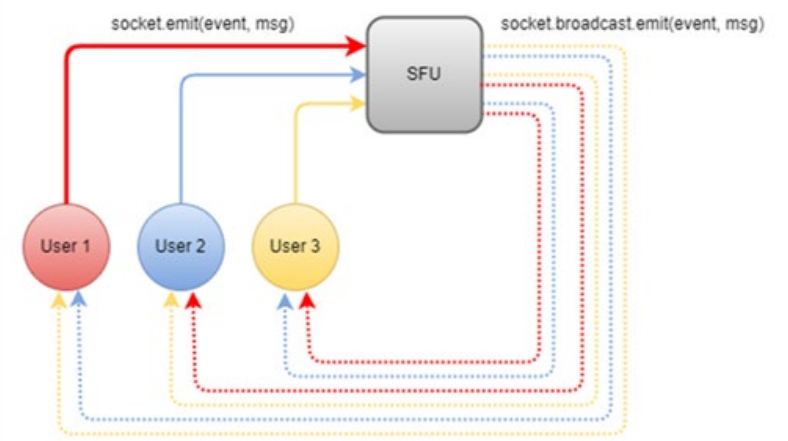


Figure 22: High-level communication architecture when adopting a Selective Forwarding Unit (SFU) for multiuser holo-conferencing

As a *third Scalablity Enabler (SE3)*, both modules of the SFU have been virtualized, provided either as a Docker and as a Helm Chart, thus having become Virtualized Network Functions (VNF) that can be dynamically instantiated over the cloud continuum (e.g., on a selected edge server), under request for specific sessions. The modules of the Holo-Orchestrator have been also virtualized, although they are not meant for dynamic multi-instantiation, but for single and static (reusable) cloud deployments.

When adopting an SFU for a holo-portation session with *N* clients, each client sends in the uplink direction 1 media (more precisely, 1 audio + 1 video) stream to the SFU and receives back *N-1* streams (i.e., the ones from all the rest clients, which are in fact *N-1* video and *N-1* audio streams) from the SFU. An analogous situation applies for the events+control information. This implies that, in total, the SFU needs to send *N*(N-1)* volumetric video streams in downlink (and the same for

audio and for control information), which becomes a clear bottleneck in sessions with a high number of clients.

For such a purpose, deployment on an Edge server can save resources and result in higher resource usage efficiency. Therefore, as a *fourth Scalability Enabler (SE4)*, novel interfaces between the (Holo-)Orchestrator and the SFUs have been devised and implemented so that specific instances of SFUs can be selected for new active sessions, based on specific criteria (e.g., deployment domains, edge resources).

Still, in such situations the SFU needs to send *N*(N-1)* volumetric video streams in downlink, so a *fifth Scalability Enabler (SE5)* has been devised in PRESENCE to support holo-conferencing sessions making use of more than 1 concurrent SFU (Figure 23). If more than 1 SFUs are adopted for a given session, then each client will still send its media streams to a unique SFU but may receive media streams from the other clients in a balanced manner from all the active SFUs, thus smoothing the bandwidth requirements for each SFU, and thus enhancing the scalability of media sessions.

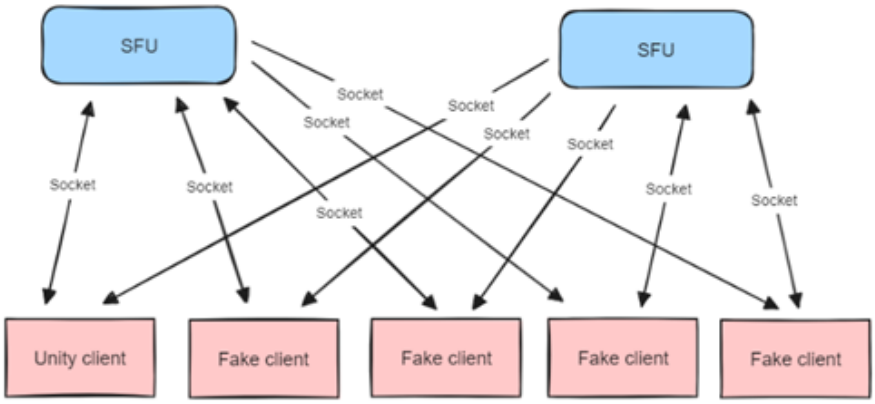


Figure 23: High-level scheme of a session with clients connecting to two different SFUs

Selective SFU with position- and Viewport-aware delivery module as Scalability Enabler 6 (SE6):

In the most basic – yet most widely adopted – operation of SFUs, the received streams from one client are forwarded to the rest of clients in the session, without performing any type of media processing task, like users' dynamics exploration, stream multiplexing, and/or transcoding.

However, in multiuser 3D environments, in which each user can have dynamic relative positions, viewpoints, and 6 Degrees of Freedom (6DoF), each of the users will be unable to see and perceive the details of the whole information at any time. For instance, users / information outside the Field of View (FoV) will not be visible, and the details for far objects / elements / holograms might not be perceived, as illustrated in Figure 24.

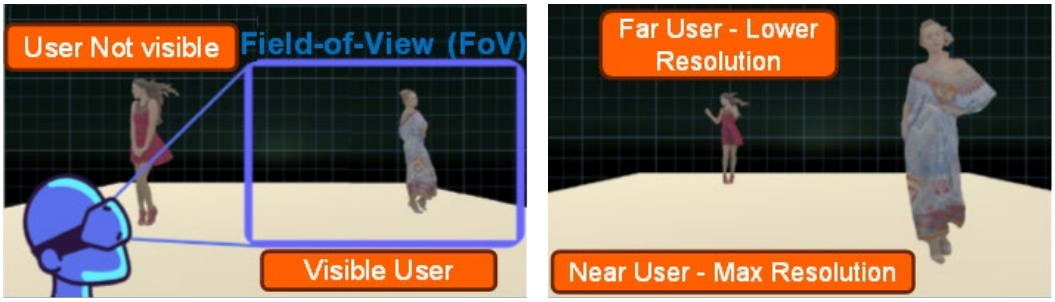


Figure 24: Impact on FoV and relative distances in 3D virtual environments

Thus, a *sixth Scalability Enabler (SE6)* has been devised to leverage the instantaneous per-client / per-stream positions and FoVs to optimize the overall communication process, adopting a selective binary forwarding strategy that consists of only delivering to each target client the audiovisual streams that are visible to him/her (Figure 25).

To achieve this, each client needs to send his/her position and viewport periodically (e.g., T=5 times / second, every 200 ms) to the Events Manager. With this information, the Media Manager builds and dynamically updates a *visibility matrix* with binary visibility variables (i.e. [0 ,1]) for each potential origin<>destination connection in the session (see Figure 25). That matrix contains a map for each client *i-th*, including an id for each remote user *j-th* and a value indicating whether he/she is visible for that reference client (i.e., $v_{ij} = 1$ = [0, 1]. Based on such information, the Media Manager can decide to which clients the incoming streams need to be forwarded. For instance, in the scenario layout shown in Figure 26, *Client_1* will not receive the streams from *Client_5* and *Client_6*, as they fall outside his/her FoV at that particular moment.

$$v = \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{pmatrix}$$

Figure 25: Visibility matrix based on the delivery strategy (SE6) devised in PRESENCE

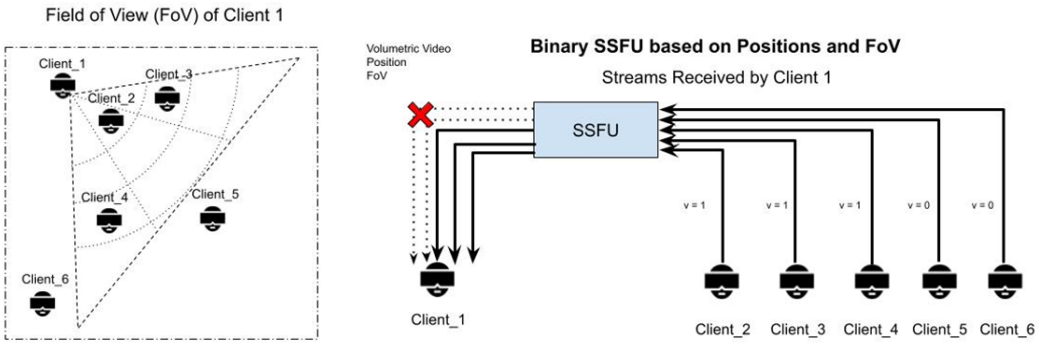


Figure 26: Basic selective (binary) position- and FoV-aware delivery strategy (SE6) devised in PRESENCE

Scalability Enabler / Transcoder:

Apart from the SFU, a separate module has been implemented: The Scalability Enabler or Transcoder (name can be used interchangeably). As shown in Figure 26, this module is designed to allow for dynamic Level of Detail according to the user's FOV and distance to other users in the scene. Additionally, it is also intended to enable general transcoding from and to different codecs and pixel formats. Thus allowing for higher compatibility between different devices and a greater number of users to be in the same session than what could be achieved with the SFU alone. To keep modularity and minimize coupling with the SFU, the Transcoder has been developed as an independent program that connects via TCP and awaits requests to process frames and change their codec or quality.

It is key to understand the differences between the SFU working in a standalone fashion and when coupled with the Transcoder module. In the former case, frames are either displayed in full detail or not at all. Whereas in the latter, a gradient of distances and FOVs is established where frames are rendered in a lower detail the farther and less centered they are to the user's point of view. Hence, we enable higher scalability by permitting a higher amount of simultaneous users to be rendered while significantly minimizing bandwidth costs for less important objects in the scene.

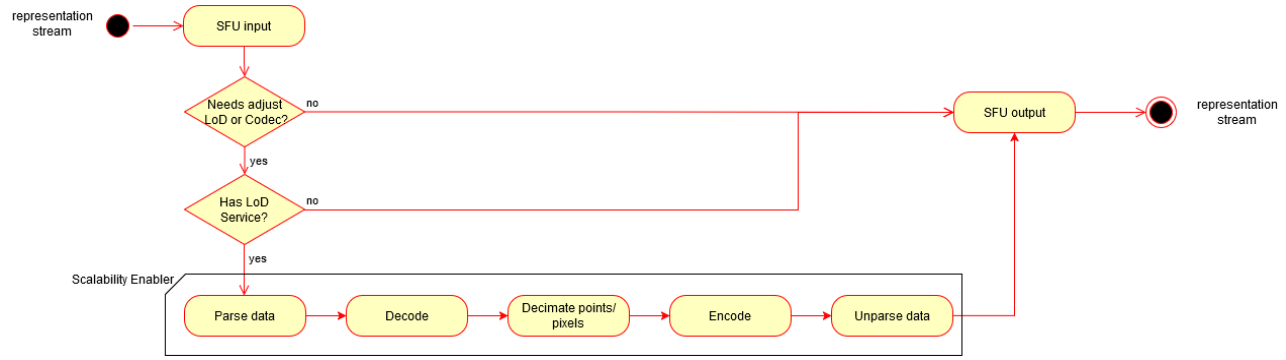


Figure 27: Design flow chart of the Scalability Enabler module, its transcoding pipeline components and how it communicates with the SFU.

In Figure 27 we display the design flow chart for this module. As shown, the SFU will only send a processing request to the Scalability Enabler in case the Level of Detail or codec needs to be adjusted and the module is available and connected to said SFU. Once a request has been received by the Transcoder, the data is parsed and then the transcoding pipeline is executed. After all data has been properly processed according to the SFU's demands, it is unparsed and sent back to the SFU so it can distribute it to the necessary users. Therefore, we retain modularity as the SFU can operate normally without the service provided by the Scalability Enabler.

As seen in both Figure 26 and Figure 27 this module has several components forming a complete transcoding pipeline. We will review the functionality and implementation of each component in detail:

Once a TCP connection is established, the Data Parser component is in charge of handling I/O for the Transcoder module. It handles the reception of incoming requests, reading the input parameters for said request and performing any necessary data preprocessing such as memory allocation, type conversion, etc. Subsequently, once the request has been handled, the Data Parser is called again

at the end of the pipeline to unparse the data and create the argument list for the output packet to be sent back to the SFU.

From the Parsed data, relevant decoding information such as the codec and pixel format used in the compressed stream can be extracted. Therefore, an appropriate decoder is created and initialized. This component outputs a raw RGB stream from the encoded data. In case the input pixel format is not 3-channel 8-bit RGB, it will be converted via ffmpeg's re-scaler API. This color space conversion is necessary to efficiently operate on the pixel-encoded data since other representations such as YUV do not allow for leveraging of spatial locality. This raw data contains the RGB-converted information of a user's Point Cloud representation, as discussed in the previous section.

Prior to re-encoding the raw data, an optional processing step can be done on the raw RGB image: Point/Pixel decimation. This step consists of directly pruning pixels in the RGB-encoded image to both reduce the required bandwidth to distribute the stream among the other clients and reduce the amount of points that need to be rendered at the endpoint. Presently, a simple algorithm that decimates points taking advantage of GPU acceleration has been implemented. However, in its current state, the algorithm is only able to prune pixels in regular patterns (i.e: Removing entire columns or rows of the image at once). Thus. resulting in some visual artifacts like noticeable gaps in the final Point Cloud. Some sort of visual degradation is to be expected in such an aggressive subsampling technique. However, we believe that it can be greatly improved by implementing random or semi-random sampling patterns and it will be explored in next steps.

Subsequently, using the arguments from the SFU's request, an encoder is created and initialized with the desired codec and output pixel format. Presently, the admitted configurations are H.264 codec with YUV 4:2:0 pixel format and H.265 codec with YUV 4:4:4 pixel format both with and without hardware acceleration by CUDA variants. Other parameters such as bit rate, resolution and GOP are either deduced from the incoming stream or configured from a metadata field in the request. The RGB data is first converted into the target pixel format if necessary. Compression parameters like CRF (Constant Rate Factor) or QP (Quantization Parameter) are set according to a "quality level" argument, which is in turn calculated and sent by the SFU depending on the user's FOV and distance to the other users. Currently, 3 discrete Levels of Detail are recognized by the transcoder: Low, medium and high. With QP (in H.265) or CRF (in H.264) values of 23, 30, and 51, respectively.

Finally, a TCP header is created, the encoded stream is unparsed into packets and sent back to the SFU with appropriate time stamps, user IDs and other relevant metadata. The SFU will utilize said information to determine to which users distribute the transcoded data and whether the packet should be discarded or not (i.e: In case the timestamp marks an obsolete packet due to transcoding problems, buffering or lost packets).

At the time of writing the deliverable, the proposed scalability enabler is stable for a 1-to-1 delivery pipeline for one transcoded stream at one fixed quality level, with a processing delay below the real-time remote communications threshold [Ref. 30]. Thorough performance tests will be carried out in the coming weeks and will be included in future versions of this deliverable.

Although the current development iteration for the Scalability Enabler already showcases significant potential in reducing bandwidth costs, the following next steps have been identified which should lead to either a greater bandwidth reduction or lesser monetary costs in deployment:

1. **Improved Point Decimation:** The main drawback of the present iteration of our point decimation algorithm is that the regular subsampling patterns create “holes” and overall visual artifacts in the resulting point cloud. By exploring vertex reduction techniques we intend to implement a better subsampling algorithm that is able to reduce the number of points for dense point clouds without impacting PSNR and visual quality as much.
2. **Multiple quality levels per request:** Presently every request sent by the SFU only allows one quality level to be set. This forces the SFU to send the same frame multiple times to the Scalability Enabler to get all the desired Levels of Detail. By reorganizing the communication protocol, we can establish a system where multiple quality levels are set and multiple encoded streams received within one request. Thus reducing transcoding time by leveraging the same parse and decoding steps for all encoded streams.
3. **Parallel request handling:** Requests are handled sequentially in CPU for this iteration. Exploring parallelism both in CPU and GPU could result in more efficient use of the hardware, which could enable for more requests being handled by a single SFU + Transcoder module. Therefore, both monetary costs of deployment and even latency (due to request buffering) could be reduced.

4.4.2. KPIs Status

So far, the efforts have been mainly focused on the development and functional testing of each one of the aforementioned Scalability Enablers (SEs). The SEs description is summarized in Table 4 The relevant KPIs for some of such SEs can be anticipated, for particular holo-conferencing setups:

- SE2 and SE3: support of 8 or 9 concurrent users per session, with a single SFU deployed on a Standard_DS1_v2 or Standard_DS2_v2 Virtual Machine on Azure, respectively, when using pre-recorded holograms with 15 fps and around 10 Mbps data rate.
- SE4: delay differences >=50ms between using a close vs far SFU for multiuser holo-conferencing
- SE5: scalability increase up to 12 and 15 concurrent users when using 2 and 3 SFUs / session.
- SE6: scalability increase up to 25 in a holo-conferencing scenario with users colocated in T shape, and with a significant number of user outside the FoV and far away from a target user
- SE7: so far the in-cloud transcoding pipeline is functional for a 1-to-1 scenario and 1 transcoded stream.

4.4.3. Deviations & Mitigation Plan (if applicable)

T2.4 is making progress according to the original plan, and achieving promising results that will advance the state-of-the-art, having devised a set of Scalability Enablers for multiuser holo-conferencing, summarized in Table 4.

Scalability Enabler (SE)	Motivation / Benefits
SE1. Decoupling of User and Control Plane functions for server-side holo-portation components	It allows reusability of Control Plane functions, which require less processing load and bandwidth than User

Scalability Enabler (SE)	Motivation / Benefits
	Plane functions, which could be duplicated and/or scale up under request
SE2. Decoupling of User Plane functions, like the SFU, to independently manage the forwarding processes for audiovisual and control + events information	It reduces the processing load and bandwidth exchange for each individual manager / SFU module
SE3. Virtualization of server-side User Plane functions, like the SFU, for dynamic instantiation under request on the cloud/edge server(s) of interest	It allows a dynamic and agile onboarding, deployment and instantiation of User Plane media functions for managing the communications in the server(s) of interest
SE4. New interfaces between the Orchestrator and SFUs so that specific instances of SFUs can be selected for new specific sessions	It allows selecting the most appropriate SFU for each new session being established
SE5. Support for modular multi-SFU communication architectures	It allows instantiating and selecting multiple SFUs per session to balance the exchanged traffic in the session, specially in the bandwidth link from SFU to clients
SE6. Position- and Viewport-aware delivery	It allows avoiding the re-transmission of not visible streams to the targets, and thus reducing significantly the amount of traffic exchanged in the session (and received by clients)
SE7. Dynamic in-cloud quality / rate adaptation (i.e., transcoding)	It allows downgrading the quality, and thus reducing the data rate, of streams belonging to clients scarcely visible to each destination client
SE8. Dynamic client-side quality / rate adaptation	It allows downgrading the quality, and thus reducing the data rate, in the uplink and downlink paths, from each client based on detected congestion or Quality of Service (QoS) drops

Table 4: Scalability Enablers devised under the umbrella of T2.4

A deviation - which is natural and was expected - can be cited though: all Scalability Enablers so far have been tested for captured streams using low-cost affordable RGB-D sensors, like Azure Kinect, but not yet Raytrix light field ones. The reason behind this is because the volumetric capture and reconstruction methods from T4.1 and T4.2 are still in-progress, which is aligned to the expected timeline evolution of the project. However, the holo-conferencing communication modules and each of the Scalability Enablers are already making use of the volumetric video compression method from T2.3, so they will be ready to seamless adopt and support the new and more heavy streams from the volumetric capture setups by PRESENCE, once available.

5. Preliminary Evaluation

This section provides an initial assessment of the developments made in WP2. The evaluation considers the performance, feasibility, and effectiveness of the implemented technologies, including

3D data acquisition, volumetric reconstruction, compression and streaming, and multi-user scalability. The assessment is based on internal tests, demonstrations, and preliminary validation against project KPIs.

5.1. Performance Assessment

3D Data Acquisition and Volumetric Capturing

The real-time 3D data acquisition system has undergone preliminary testing with the Raytrix R32 light-field camera setup. Results indicate that the system can produce high-quality volumetric data at 30 fps on an NVIDIA RTX 4090 GPU. Initial calibration challenges, such as chromatic aberration and lens distortion, have been mitigated with a new camera model and improved intrinsic calibration techniques.

Volumetric Representations and Reconstruction

The volumetric reconstruction pipeline successfully processes multi-camera RGB+D inputs, generating point cloud and mesh representations. Point cloud reconstruction operates at way above real-time speeds (5 ms per frame process time), while mesh reconstruction achieves a near real-time frame rate of ~25 fps. User feedback from internal demonstrations suggests that the point cloud representation is sufficient for motion tracking but lacks the photorealism required for fully immersive experiences. Texture-mapped mesh models are preferred for their realism, despite a higher processing overhead. The first results of a new prototype method for texture-mapped mesh reconstruction are very promising, as the mean execution time of the new algorithm is 19ms which accounts to around 52fps when using an RTX4090 NVidia GPU and capturing RGB+D data from 4 cameras at 720p color resolutions.

Holoportation Compression and Streaming

The newly implemented compression pipeline effectively reduces volumetric data transmission requirements while maintaining high visual fidelity. Testing has demonstrated that the system can achieve compression rates comparable to MPEG's VPCC standard while performing significantly faster (up to two orders of magnitude). Initial subjective assessments suggest that compression artifacts are minimal, especially at moderate bitrates. The system has been successfully tested over local and cloud-based networks, with real-time streaming performance validated under various network conditions.

Multi-User Scalability

The modular redesign of the Holo-Orchestrator and Selective Forwarding Unit (SFU) has enabled more efficient session management and multi-user support. The introduction of viewport-aware delivery mechanisms and dynamic level-of-detail adjustments has improved bandwidth efficiency while maintaining a high-quality user experience. The system is currently stable for a simple scenario but needs to be further tested for more demanding and realistic scenarios.

5.2. Quality, Feasibility and Usability

Preliminary user feedback from internal testing sessions highlights the system's potential for real-world applications. Key observations include:

- **Ease of Setup:** The updated calibration tools and user-friendly capturer.dll API have facilitated a more seamless integration process.
- **Visual Quality:** Test users found the mesh-based volumetric representations more realistic and engaging compared to point clouds.
- **Latency and Responsiveness:** The real-time compression and streaming pipeline provides a smooth holoportation experience, with minimal perceptible delay under optimal network conditions.

However, these observations, and other feasibility, usability and quality metrics need to be evaluated both quantitatively and qualitatively via user questionnaires. This is one of the key goals for the next versions of the deliverable.

6. Outlook

In this deliverable, we have outlined the progress made in WP2, focusing on the development of the real-time holoportation system. We have implemented the first iteration of the light-field camera setup and volumetric video compression pipeline, contributing significantly to the photo-realistic 3D reconstruction of human subjects. Task 2.1 has successfully deployed a multi-camera setup for capturing high-quality volumetric video, and Task 2.2 has worked towards achieving real-time volumetric data representation using innovative methods for 4D reconstruction. Additionally, Task 2.3 has explored and optimized volumetric video compression to ensure the system performs well under various network conditions while maintaining high visual fidelity. These achievements lay the foundation for future work on a real-time, scalable holoportation system.

Looking ahead, we aim to further refine the holoportation pipeline to improve the quality and real-time performance of the system. A key focus will be enhancing the compression techniques to ensure high-quality video transmission even under constrained bandwidth conditions, which is crucial for enabling remote communication in real-time. Additionally, we will continue improving the integration of light-field cameras with the real-time reconstruction system, refining algorithms to reduce latency, increase accuracy in human body modeling, and allow for more complex body movements and interactions while reducing the hardware and processing requirements. This will be vital for the realistic representation of human subjects, enabling more immersive communication experiences.

Another major objective is to extend the system's scalability to support multi-user interactions in real-time. As part of this, we will work on integrating solutions for multi-party communication, ensuring that users can interact seamlessly in virtual environments with multiple other participants. Key to this is the optimization of the pipeline for handling large volumes of volumetric data, making it possible to support simultaneous transmissions without compromising quality. The next phase will involve implementing and testing real-time collaboration scenarios, where users from diverse locations interact in virtual environments using their volumetric avatars, such as virtual conferences or professional meetings.

The user experience will be central to the continued development of WP2. In the coming months, we will continue to iterate on the system based on user feedback. This will include the integration of the system into realistic Use Cases in the field of Professional Manufacturing and Training, Medical

Applications or Cultural Heritage. Further user evaluations will help assess the system's acceptability and usability, with a particular focus on how the compression pipeline impacts the perceived video quality. The goal will be to ensure that users find the system both intuitive and immersive while addressing any challenges with system performance and comfort.

Finally, the end goal is to make WP2 developments available to the rest of the consortium and other research third party entities for seamless integration of the holoportation system into various XR applications. These APIs will allow for easier integration into different XR platforms, opening up new opportunities for a wider range of immersive experiences. In parallel, we will explore opportunities for further enhancing the real-time performance of the system by incorporating optimizations in volumetric video data compression, edge computing, and cloud-based processing. By delivering a fully integrated, real-time holoportation system, we will move closer to the vision of enabling hyper-realistic, multi-user communication in XR applications, advancing the project's goals in innovative and impactful ways.

As the project progresses, we are also planning to expand the scope of the user tests and evaluation protocols to gather deeper insights into user interactions in real-world scenarios. These will focus on evaluating the effectiveness and user engagement of the system in multi-user XR environments, where the holoportation experience will be tested under various usage conditions and settings. By refining the pipeline and incorporating the feedback from these evaluations, we will continue to improve the holoportation system to ensure it supports a wide range of immersive communication and collaborative experiences.

6.1. Planned Experiments

As part of WP2, a series of planned experiments will be conducted to evaluate the performance, quality, scalability, and usability of the system across various hardware, camera types, and operational environments. These experiments aim to provide a comprehensive understanding of the system's capabilities and limitations. The following experiments are foreseen:

1. **Raytrix Performance Evaluation on Newer Hardware (RTX 5090):** This experiment will evaluate the overall performance of the system when deployed on the recently released RTX 5090, which is foreseen to have wider memory throughput and parallel computing capabilities. One of the key goals is to understand if multiple Raytrix cameras can be handled by a single GPU. Key performance metrics such as frame rate, latency, system stability, and real-time processing capabilities will be assessed for both the camera system and associated software components.

2. **Quality Comparison Between Raytrix and Commercial RGB-D Cameras (Orbbec):** A comparative study will be conducted between the Raytrix camera system and commercial RGB-D cameras, specifically focusing on the Orbbec models. The goal is to evaluate the quality of depth and color data captured by both systems under identical conditions. The comparison will measure parameters like depth resolution, image fidelity, noise levels, and overall accuracy, helping to benchmark the Raytrix system against established commercial alternatives.

3. **Performance Test of Volumetric Reconstruction Algorithm:** This experiment will evaluate the performance of the real-time volumetric reconstruction system using four light-field cameras at a multi-view setup capturing a single full-body human in the scene. The goal

of this experiment is to measure reconstruction execution times and the resulting rendering performance in a Unity environment. Potential bottlenecks of the capturing/reconstruction/rendering pipeline will be revealed and the algorithms will be optimized in order for the whole pipeline to have real-time performance.

4. **Quality of Experience (QoE) Test for Volumetric Reconstruction:** This experiment evaluates the photo-realism of the volumetric reconstruction with validated questionnaires in a multi-view setup. Participants will assess two conditions: HMD-based immersive viewing and screen-based non-immersive viewing, comparing point clouds and mesh-based reconstructions. The goal of this experiment is to assess the perceived photo-realism of the reconstruction in terms of human appearance, motion, visual quality and depth. By comparing point clouds vs. mesh-based reconstructions and HMD vs. screen-based viewing, the study aims to determine whether the reconstruction achieves a high level of realism and user acceptance ensuring its suitability for immersive holoportation.

5. **Compression Performance Test:** This test will investigate the performance of different compression techniques in real-time scenarios, especially when increasing the resolution or the number of cameras used in the system. This experiment will highlight potential bottlenecks and guide optimization efforts for maintaining high-quality data streams in real-time environments.

6. **Scalability Enabler Evaluation with Multiple Users:** This experiment will assess the performance levels and robustness of the Scalability Enablers (i.e. multi-SFU scenarios, viewport-aware delivery, in-cloud transcoding), by means of determining the scalability gains and associated Quality of Service (QoS) - e.g., delays, fps - and resources usage - e.g., bandwidth, CPU / GPU usage - levels.

7. **Reconstruction Quality Subjective Evaluation:** This experiment will evaluate the quality of 3D reconstructions in two distinct operational setups: a local configuration (where data is processed without transmission) and a remote setup (where data is transmitted and processed across a network). The goal is to assess first the reconstruction quality provided by our pipeline. Then, we aim to understand how network latency, bandwidth limitations, and other transmission factors affect reconstruction accuracy and effective quality. Subjective user feedback will be gathered through questionnaires to understand the perceived differences in reconstruction quality between the two setups and to identify any degradation in data quality or user experience in remote scenarios.

6.2. Planned Publications

The target publication will depend to certain extent on the obtained results from the planned experiments. However, a tentative publication plan can be anticipated:

- From Task 2.1 and 2.2, the comparison among volumetric reconstructions when using Raytrix cameras and more affordable cameras, like Orbbec, can be published at a high-impact conference, like ACM NOSSDAV.

- From Task 2.2, we are planning to submit the results of the newly prototyped winding numbers based reconstruction pipeline to various computer vision/3D/4D human body scanning conferences like 3DBODY.TECH, CVPR and 3DV.

- From Tasks 2.3 and 2.4, a few dissemination objectives have been set: (i) the study on comparing architectures, performance and resources usage when adopting different volumetric video pipelines can be sent for publication to a high-impact journal, like Virtual Reality; (ii) the results for some Scalability Enablers (i.e., close vs far comm server, one vs multiple comm servers, and initial proof-of-concept for viewport-aware delivery) will be sent to IEEE Transactions on Network and Service Management; (iii) the novel pipeline with in-cloud transcoding and associated performance results will be sent for publication at a high-impact conference, like ACM Multimedia.

6.3. Planned Pilots

The following pilots are planned to demonstrate the system's capabilities in real-world scenarios, showcasing performance, scalability, and integration with industry use cases:

- **Local Performance and Reconstruction with Raytrix Cameras:** This pilot will showcase the performance and 3D reconstruction quality using at least two Raytrix cameras in a local setup. This is a first step for evaluation of the final reconstruction pipeline using Raytrix Lightfield cameras.
- **Full Pipeline Demonstration for Remote Users:** A demonstration of the complete system pipeline will be conducted, showcasing the experience of remote users with real-time volumetric capture, immersive visualization and social interaction.
- **Realistic Scalability Demonstration:** This pilot will test the system's scalability by simulating multiple remote users interacting within the same virtual environment, assessing real-time synchronization, data load, and system stability.
- **Integration with Professional Collaboration and Manufacturing Training Use Cases:** The system will be integrated into professional collaboration and manufacturing training use cases, demonstrating immersive remote collaboration and safe, realistic XR training scenarios for industrial applications.

7. Abbreviations and definitions

7.1. Abbreviations

AR	**Augmented Reality**
FoV	Field of View
GPU	Graphics Processing Unit
KPI	Key Performance Indicator
MR	Mixed Reality
NTP	Network Time Protocol
RGB-D	Red Green Blue - Depth

SFU	Selective Forwarding Unit
VPCC	Video-based Point Cloud Compression
VR	Virtual Reality
XR	Extended Reality

7.2. Definitions

Holoportation	A system that enables real-time volumetric video capture, transmission, and rendering to create immersive XR experiences.
Light-field Camera	A camera that captures both spatial and angular light information, allowing for advanced depth perception and 3D reconstruction.
Point Cloud	A collection of data points in 3D space representing the surface of an object.
Mesh Reconstruction	A process of converting point cloud data into a structured surface representation using polygons.
Compression Pipeline	A system for reducing the data size of volumetric video while preserving visual fidelity.
Viewport-aware Delivery	A streaming optimization technique that transmits only the relevant data based on the user's field of view.

8. References

Ref. 1 Petit, Benjamin *et. al, "Multi-Camera Real-Time 3D Modeling for Telepresence and Remote Collaboration"*. International Journal of Digital Multimedia Broadcasting. 2010

Ref. 2 Ignacio Reimat, Evangelos Alexiou, Jack Jansen, Irene Viola, Shishir Subramanyam, and Pablo Cesar. 2021. *"CWIPC-SXR: Point Cloud dynamic human dataset for Social XR",* 12th ACM Multimedia Systems Conference (MMSys '21). Association for Computing Machinery, New York, NY, USA, 300–306.

Ref. 3 I. Viola, J. Jansen, S. Subramanyam, I. Reimat and P. Cesar, *"VR2Gather: A Collaborative, Social Virtual Reality System for Adaptive, Multiparty Real-Time Communication",* in IEEE MultiMedia, vol. 30, no. 2, pp. 48-59, April-June 2023

Ref. 4 S. F. Langa, M. Montagud, G. Cernigliaro and D. R. Rivera, *"Multiparty Holomeetings: Toward a New Era of Low-Cost Volumetric Holographic Meetings in Virtual Reality",* in *IEEE Access*, vol. 10, pp. 81856-81876, 2022, doi: 10.1109/ACCESS.2022.3196285.

Ref. 5 Zhou, Shuyao & Zhu, Tianqian & Shi, Kanle & Li, Yazi & Zheng, Wen & Yong, Junhai. "*Review of light field technologies"*, Visual Computing for Industry, Biomedicine, and Art. 2021

Ref. 6 Y. Zhang, Z. Li, W. Yang, P. Yu, H. Lin and J. Yu, *"The light field 3D scanner," 2017 IEEE International Conference on Computational Photography (ICCP)*, Stanford, CA, USA, 2017, pp. 1-9

Ref. 7 Feng, Xiaohua & Ma, Yayao & Gao, Liang. (2022). *"Compact light field photography towards versatile three-dimensional vision."* Nature Communications. 2022.

Ref. 8 Lu Y, Yu H, Ni W, Song L. "*3D real-time human reconstruction with a single RGBD camera"*. Springer Nature, 2022.

Ref. 9 Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. "*Real-time non-rigid reconstruction using an RGB-D camera"*. ACM Trans. Graph. 33, 4, Article 156 (July 2014)

Ref. 10 Tiancheng Zhi *et. al "TexMesh: Reconstructing Detailed Human Texture and Geometry from RGB-D Video",* arXiv, 2020

Ref. 11 Marco Pesavento *et. al,* "ANIM: Accurate Neural Implicit Model for Human Reconstruction from a Single RGB-D Image". CVPR 2024.

Ref. 12 Bernhard Kerbl *et. al, "*3D Gaussian Splatting for Real-Time Radiance Field Rendering", ACM Transactions on Graphics, 2023

Ref. 13 Shunyuan Zheng, *et. al,* "GPS-Gaussian: Generalizable Pixel-wise 3D Gaussian Splatting for Real-time Human Novel View Synthesis" IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024

Ref. 14 Gumhold, Stefan & Straßer, Wolfgang. "Real Time Compression of Triangle Mesh Connectivity", Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998.

Ref. 15 B. Kuth, *et. al, "Towards Practical Meshlet Compression" ArXiv, 2024*.

Ref. 16 Walter Zimmer, *et. al "PointCompress3D: A Point Cloud Compression Framework for Roadside LiDARs in Intelligent Transportation Systems"*, arXiv, 2024

Ref. 17 R. Mekuria, K. Blom and P. Cesar, *"Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video,"* in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828-842, April 2017

Ref. 18 Petkova, R.; Poulkov, V.; Manolova, A.; Tonchev, K. "*Challenges in Implementing Low-Latency Holographic-Type Communication Systems."* Sensors, 2022.

Ref. 19 Dun Yuan, *et. al, "Realizing XR Applications Using 5G-Based 3D Holographic Communication and Mobile Edge Computing"*, arXiv, 2023.

Ref. 20 Sergi Fernandez, Mario Montagud, David Rincón, Juame Moragues, and Gianluca Cernigliaro. "*Addressing Scalability for Real-time Multiuser Holo-portation: Introducing and*

Assessing a Multipoint Control Unit (MCU) for Volumetric Video". In Proceedings of the 31st ACM International Conference on Multimedia 2023

Ref. 21 Mei, X., Sun, X., Zhou, M., Jiao, S., Wang, H., & Zhang, X. "*On building an accurate stereo matching system on graphics hardware"* In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), 2011.

Ref. 22 Christian Perwaß, Lennart Wietzke, *"Single lens 3D-camera with extended depth-of-field,"* Proc. SPIE 8291, Human Vision and Electronic Imaging XVII, 829108 (17 February 2012)

Ref. 23 Thomas, Hugues, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. *"Kpconv: Flexible and deformable convolution for point clouds."* In Proceedings of the IEEE/CVF international conference on computer vision, pp. 6411-6420. 2019.

Ref. 24 Yang Li and Tatsuya Harada, *"Lepard: Learning partial point cloud matching in rigid and deformable scenes"*, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022

Ref. 25 Alexiadis, Dimitrios S., Anargyros Chatzitofis, Nikolaos Zioulis, Olga Zoidi, Georgios Louizis, Dimitrios Zarpalas, and Petros Daras. *"An integrated platform for live 3D human reconstruction and motion capturing."* IEEE Transactions on Circuits and Systems for Video Technology 27, no. 4 (2016): 798-813.

Ref. 26 Barill, Gavin, Neil G. Dickson, Ryan Schmidt, David IW Levin, and Alec Jacobson. *"Fast winding numbers for soups and clouds.*" ACM Transactions on Graphics (TOG) 37, no. 4 (2018): 1-12.

Ref. 27 Montagud, Mario, Jie Li, Gianluca Cernigliaro, Abdallah El Ali, Sergi Fernández, and Pablo Cesar. *"Towards socialVR: evaluating a novel technology for watching videos together.*" Virtual Reality 26, no. 4 (2022): 1593-1613.

Ref. 28 Graziosi D, Nakagami O, Kuma S, Zaghetto A, Suzuki T, Tabatabai A. "*An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC)". APSIPA Transactions on Signal and Information Processing*. 2020

Ref. 29 Szeliski, Richard. *"Computer vision algorithms and applications."* 2011.

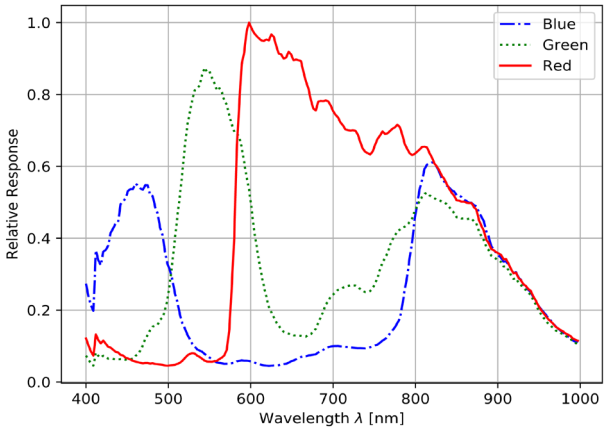
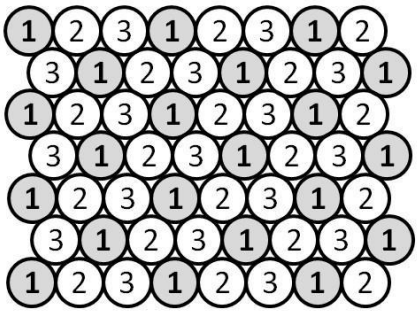
Ref. 30 Carlos Cortés, Irene Viola, Jesús Gutiérrez, Jack Jansen, Shishir Subramanyam, Evangelos Alexiou, Pablo Pérez, Narciso García, and Pablo César. "*Delay Threshold for Social Interaction in Volumetric eXtended Reality Communication."* ACM Trans. Multimedia Comput. Commun. Appl. 20, 7, Article 206 (July 2024)

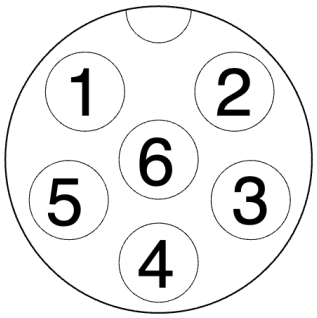
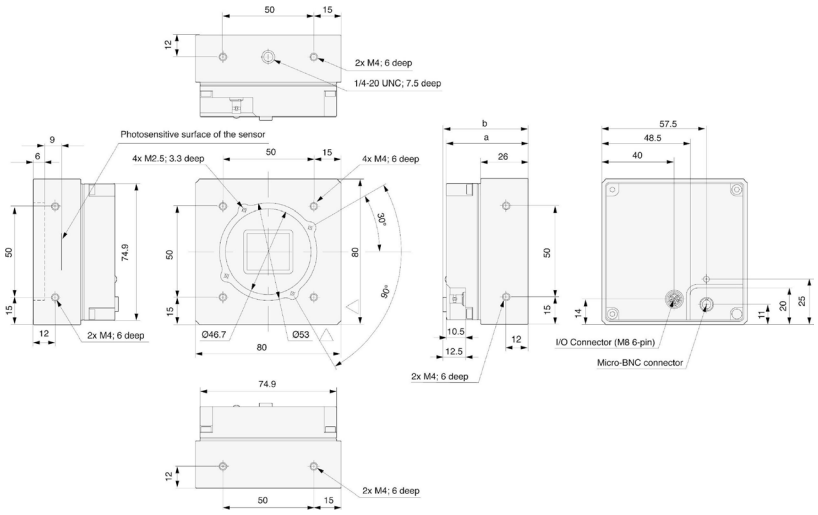
9. Annexes

9.1. Annex A

The following table show the Technical Specifications of Raytrix's R32 Light-field camera:

R32 factsheet	
	
Sensor	
Image sensor	Onsemi XGS 32000
Lateral resolution (H x V)	6560 x 4948 pixel^2
Lateral resolution (MegaPixel)	32.4 MP
Effective lat. resolution (H x V)	3280 x 2474 pixel^2
Effective lat. resolution (MegaPixel)	8.1 MP
Active area	21.0 x 15.8 mm^2
Pixel length	3.2 µm
Shutter type	Global shutter
Frame rate	36 fps
ADC resolution	12 bit
Spectrum	Color

Spectral response				
Cover glass removed?	Yes			
Sensor interface	HiSPi			
Micro lens array				
Design Iteration	V1		V2	
MLA type	L3-D125-A018-VRE-VI		L1-D500-A010-Vae-VI	
Light field mode	Galilean multi focused plenoptic 2.0			
Micro lens types	3			
Geometry	hexagonal			
Layout				
Aperture	f/1.8		f/3.8	
Lens pitch	125 µm		500 µm	
Camera interface	CoaXPress (2.5–12.5 Gbps); Micro-BNC (HD-BNC) connector			
Camera interface bandwidth	CXP Speed	Bandwidth	Cable length	fps @ full res.
	CXP-2	2.5 Gbps	180m	7
	CXP-3	3.125 Gbps	100m	9
	CXP-5	5 Gbps	60m	14
	CXP-6	6.25 Gbps	40m	18
	CXP-10	10 Gbps	40m	29

	CXP-12	12.5 Gbps	30m	36
Power	Recommended: Power over CoaXPress (PoCXP): 24 VDC supplied via the camera's Micro-BNC (HD-BNC) connector. 11 W (typical) Not Recommended: Power supply via I/O connector: operating voltage 24 VDC. Minimum 18.6 VDC. Maximum 26 VDC.			
I/O	M8 6-pin female connector (IEC 61076-2-104) Recommended mating connector: M8 6-pin male			
Pinout	Pin	Line	Function	
	1	-	24 VDC power	
	2	Line 1	Opto-coupled I/O input	
	3	-	Ground for opto-coupled I/O	
	4	Line 2	General purpose I/O (GPIO)	
	5	Line 3	General purpose I/O (GPIO)	
	6	-	Ground for camera power and General Purpose I/O (GPIO)	
Size (L x W x H)	50 x 80 x 80 mm^3			
				
Weight	550 g			
Mount	Custom, thermal decoupling of lens and camera body			
OEM	Basler Lens			
Type	Based on F-S35-2528-45M-S-SD			
Focal length	24.5 mm			
Aperture*	f/2.0			
Focus range*	0.2 - 3.5m			
Angle of View (on R32)	Horizontal: 55.7° Vertical: 39.1°			

9.2. Annex B: Capturer DLL API Specifications

The summary of the main endpoints implemented for the Capturer DLL API is given below:

Authors: Dimitrios Pattas, Theofilos Tsoris (CERTH)
Date: 8/1/2025
Version 0.2

Initialization

- `IntArray`[5] `find_cameras()`
  Find all cameras connected to the system and return their ids for future reference. We assume that each camera will have a unique id for the whole time of the capturing.
  Returns an empty array if no cameras have been found.
  Returns an array with a negative integer as the first element indicating the error that has occurred.

Cameras' parameters

Assuming a camera parameters structure:

```
struct CamParameters {
      int color_width;
      int color_height;
      int depth_width;
      int depth_height;
      int fps;
}
```

- `int set_camera_parameter(int camera_id, CamParameters  parameters)`
  Set camera parameters regarding color resolution, depth resolution of the captured image and depth respectively and frame rate of the capturing
  Returns a negative integer indicating the error that has occurred, otherwise 0
  Note: If this call is not possible before starting the capturing then we will pass the same parameters in `start_capturing` (see below).
- `int get_camera_parameter(int camera_id, CamParameters& parameters)`
  Gets the camera parameters of a specific camera.
  Returns a negative integer indicating the error that has occurred, otherwise 0.

- `float* get_color_intrinsics(int camera_id)`
  Assuming a pinhole model of camera, we expect this function to return a `float*` holding 9 values for a 3x3 matrix (row major) of the intrinsics of the color camera as described in OpenCV's documentation.

[5] `IntArray` can be a `std::vector<int>` or any other structure that contains the size of the array

Returns a NULL pointer if an error has occurred.

- `float* get_depth_intrinsics(int camera_id)`
  Assuming a pinhole model of camera, we expect this function to return a `float*` holding 9 values for a 3x3 matrix (row major) of the intrinsics of the depth camera as described in OpenCV's documentation.
  Returns a NULL pointer if an error has occurred.

- `float* get_color_distortions(int camera_id)(NOT NEEDED: undistorted images)`
  If the captured color frame is distorted, we will also need the distortion coefficients as described in OpenCV's documentation.
  Then this function returns a `float*` holding 5 values (k1, k2, p1, p2, k3)
  Returns a NULL pointer if an error has occurred.
- `float* get_depth_distortions(int camera_id): (NOT NEEDED: undistorted depth images)`
  If the captured depth frame is distorted, we will also need the distortion coefficients as described in OpenCV's documentation.
  Then this function returns a `float*` holding 5 values (k1, k2, p1, p2, k3)
  Returns a NULL pointer if an error has occurred.

Capturing

- `int start_capturing(IntArray camera_ids)`
  This function will initiate the capturing process, initialising any structures that may be needed during the whole process, connect to the cameras and start capturing from the cameras whose ids are given as an argument.
  Returns a negative integer indicating the error that has occurred, otherwise 0.

  In case the camera parameters have not be set yet, an overloaded function can be used:
  `int start_capturing(IntArray camera_ids, CamParameters parameters)`
- `int end_capturing()`
  This function will stop the capturing process, letting the DLL take care of any structures that have been allocated, disconnect the cameras etc.
  Returns a negative integer indicating the error that has occurred, otherwise 0.

For the functions below, we will assume that the Capturer DLL can provide synchronised captured frames of all (or some) connected cameras when they are called.
It is expected that with every call of the functions below the ***last*** captured frame will be provided along with its timestamp.

- `int get_sync_depth_all_cameras(CudaDepthPtrArray[6]& depth_cuda_pointers, TimeStampArray[7]& timestamps)`
  Fills the `depth_cuda_pointers` array with the CUDA pointers of captured depths of all connected cameras. Fills the `timestamps` array with their respective timestamps.
  It is expected that both arrays are sorted based on the camera's ids (ascending).
  Returns a negative integer indicating the error that has occurred, otherwise 0.

  Note: We need to know in advance if depth images are row or column major
- `int get_sync_depth(IntArray camera_ids, CudaDepthPtrArray& depth_cuda_pointers, TimeStampArray& timestamps)`
  The same as the function above but it does so only for the camera ids that are provided

- `int get_sync_color_all_cameras(CudaColorPtrArray[8] color_cuda_pointers, TimeStampArray& timestamps)`
  Fills the `color_cuda_pointers` array with the CUDA pointers of captured images in ***full focus*** of all connected cameras. Fills the `timestamps` array with their respective timestamps.
  It is expected that both arrays are sorted based on the camera's ids (ascending).
  Returns a negative integer indicating the error that has occurred, otherwise 0.
- `int get_sync_color(IntArray camera_ids, CudaColorPtrArray& color_cuda_pointers, TimeStampArray& timestamps)`
  The same as the function above but it does so only for the camera ids that are provided

Assuming there is a structure `PointCloud` in place:

```
struct PointCloud
{
	float3* vertices;
	uchar3* color;
	float* normals; // If they are not readily available, it should be NULL
	int num_of_vertices;
}
```

- `int get_sync_pointclouds_all_cameras(PointCloudArray[9]& pointclouds, TimeStampArray& timestamps)`

[6] `CudaDepthPtrArray` can be a `std::vector<float*>` or any other structure that contains the size of the array
[7] `TimeStampArray` can be a `std::vector<long long>` or any other structure that contains the size of the array
[8] `CudaColorPtrArray` can be a `std::vector<uchar*>` or any other structure that contains the size of the array
[9] `PointCloudArray` can be a `std::vector<PointCloud>` or any other structure that contains the size of the array

Fills the `pointclouds` array with the calculated pointclouds of all connected cameras. Fills the `timestamps` array with their respective timestamps.
It is expected that both arrays are sorted based on the camera's ids (ascending).
Returns a negative integer indicating the error that has occurred, otherwise 0.

- `int get_sync_pointclouds(IntArray camera_ids, PointCloudArray& pointclouds, TimeStampArray& timestamps)`
  The same as the function above but it does so only for the camera ids that are provided
- `void point_to_image_coordinates(int camera_id, double* point_3d, double& img_xy[2])`
  It takes a point in 3D space coordinates and a camera id as input and returns the back-projected `x,y` pixel coordinates of the image acquired by the camera. (0,0) is assumed to be the upper left corner of the image.

CudaImage data structure

Documentation of CudaImage: https://raytrix.de/LFR%208.0/class_rx_1_1_l_f_r_1_1_cuda_image.html

To access the image data the method GetData() can be called. This returns a pointer to a one-dimensional array of the row-wise ordered pixel data. The number of array entries varies by image size and pixel type. The data type is also not fixed.
All information necessary for interpretation can be accessed from the CRxImageFormat class.
https://raytrix.de/LFR%208.0/class_rx_1_1_c_rx_image_format.html

An instance of this class describing a given CudaImage can be accessed with the GetFormat() method.
CRxImageFormat has public members Width, Height, PixelType and DataType. (This changed since the latest online documented version. Refer also to the locally installed documentation.)
Width and Height give the number of pixels while PixelType encodes the number of values per pixel. E.g. RGB uses three values per pixel whil RGBA uses four.
Please refer to pixel type:
https://raytrix.de/LFR%208.0/namespace_rx_1ac37864f9a600d573902409fbd6fe9e97.html
and data type:
https://raytrix.de/LFR%208.0/namespace_rx_1add4d321bb9cc51030786d53d76b8b0bd.html
to interpret the image data array.

Future developments

- Implementation of an intrinsic pinhole camera model simulation, ensuring that color and depth share the same parameters, adapted for different resolutions.
- Development of a second producer/consumer mechanism to generate total focus images and Depth3D data, similar to the existing point cloud production system.
- Multi-GPU support for image processing.
- Computation of surface normals.