Real-Time Volumetric Telepresence: A Point-Cloud Overlap System with Integrated Spatial Audio

Vitor S. Moreira[1], **Marcos F. dos Santos Soares**[1],
Hugo A. D. do Nascimento[1], **Laurita R. de Salles**[2]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)

[2]Media Lab/UFG – Universidade Federal de Goiás (UFG).

moreiravitor@discente.ufg.br, marcos.soares@discente.ufg.br,

hadn@inf.ufg.br, laurita.salles@gmail.com

Abstract. *The paper introduces a real-time volumetric telepresence system that creates a spatial overlap between a physical space and its virtual counterpart. Using an Orbbec scanner, the system captures RGB point clouds via C++ and the scanner's SDK, transmitting data through TCP to a Node.js server. Clients visualize these point clouds using A-Frame and WebSocket connections, enabling immersive 3D interactions complemented by synchronized real-time audio. Our main contribution is a low-cost, web-based pipeline enabling precise physical–virtual overlap of live point clouds with integrated spatial audio for remote collaboration.*

1. Introduction

Real-time digitization and representation of physical environments have gained significant attention due to their diverse applications in telepresence, remote collaboration, virtual reality (VR), and augmented reality (AR). Immersive systems integrating physical and virtual spaces enhance collaborative tasks, remote education, and entertainment experiences by providing users with interactive, 3D environments that foster a strong sense of presence [Zhang 2019] [Kodama et al. 2017] [Zhang et al. 2019] [Zhao et al. 2021].

A major challenge in telepresence is the real-time capture and streaming of volumetric data, such as point clouds or 3D reconstructions [Alkhalili et al. 2020]. While traditional teleconferencing is limited to 2D perspectives, volumetric representations leverage affordable depth sensors like Kinect, RealSense, and Orbbec to capture detailed RGB-D data [Kim et al. 2024]. However, bandwidth constraints, encoding and decoding complexities, synchronization of audio, and effective data transmission remain significant obstacles [Liu et al. 2021]. Although browser-based frameworks like A-Frame facilitate interactive 3D content, handling live volumetric data efficiently is still a frontier challenge [Santos and Cardoso 2019] [Hudák et al. 2020].

To address this, we propose a system that captures real-time RGB-D data, streaming it to create a robust virtual representation with synchronized spatial audio. Our solution ensures accurate spatial overlap between physical and virtual environments through precise data transformations and calibration, maintaining spatial fidelity and integrity. This paper details the system's architecture, experiments measuring latency, frame rate, and spatial accuracy, and concludes with implications, limitations, and future directions for advancing seamless physical-virtual interactions.

2. Proposed System

We adopted a distributed architecture emphasizing scalability, fault tolerance, and flexibility, structured into three core layers designed to efficiently handle data capture, processing, distribution, and visualization in real-time volumetric telepresence applications [Ostrowski and Gaczkowski 2021].

Acquisition Layer — We utilized C++14 and the Orbbec SDK to directly access raw depth and color data from the Femto Mega sensor. To enhance real-time performance while preserving essential spatial detail, we downsample frames by striding with a reduction factor $F = 5$. This is a $1/5$ reduction per dimension, so we process one pixel per 5×5 block, resulting $1/25$ of the original pixel count. Before serialization, point clouds are adjusted via rotation, translation, and scaling to align with the virtual environment's coordinate system. Each point is serialized into six floating-point values (x,y,z,r,g,b), representing spatial coordinates and color.

Server Layer — Leveraging Node.js (version 14+), we efficiently handle real-time data distribution using an event-driven, non-blocking I/O model. A TCP socket receives frames from the acquisition application, performs validation and parsing, and broadcasts the processed point clouds to multiple connected clients via WebSockets. This approach effectively separates data generation from rendering processes, ensuring efficient scalability, minimal overhead, and improved performance in multi-user scenarios.

Client Layer — On the client side, we implemented a custom A-Frame component integrated with Three.js, which decodes incoming binary frames into dynamically updated 3D geometries at approximately 15 frames per second. A static "user" entity aligned with the sensor's physical position was introduced to maintain accurate spatial audio. This allows virtual users to navigate freely, interact with the environment, and experience a cohesive, immersive spatial and audio-visual representation between physical and virtual spaces.

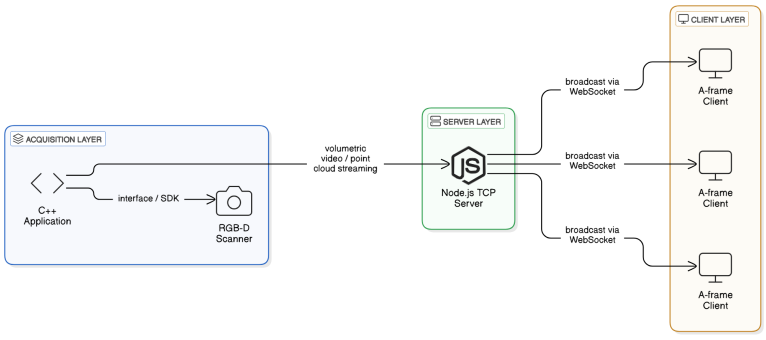

Figure 1. System architecture design

3. Experiments

In this section, we present the metrics, experimental setup, data collection procedure, and analysis of results obtained from various testing scenarios. The primary goal of these experiments was to evaluate the system's real-time performance in terms of latency, frame rate, spatial fidelity, and audio synchronization.

3.1. Evaluation Metrics

To quantitatively assess the system's performance and identify areas for improvement, four main metrics were defined:

- **End-to-End Latency**: time from sensor capture to on-screen display on the client. Measured via synchronized timestamps recorded at acquisition and client-side logging.
- **Frame Rate (Throughput)**: effective client-side rate of processed/rendered point clouds (fps).
- **Spatial Fidelity**: objective/subjective assessment of geometric accuracy and spatial relationships, via visual inspection and paired distance measurements in physical vs. virtual spaces.
- **Bidirectional Spatial Audio Synchronization**: qualitative echo tests and speech intelligibility checks to verify low-latency, directionally consistent audio.

Success criteria for the proof-of-concept were set to end-to-end latency < 500 ms and a stable frame rate of 10–15 fps.

3.2. Experimental Setup and Procedure

Experiments were conducted in the GameLab, a laboratory equipped with typical furniture and equipment, offering stable, moderate lighting conditions beneficial for consistent depth sensing. Additional tests in a domestic environment and an elongated corridor were performed to evaluate diverse scenarios regarding scene complexity and illumination.

The Orbbec Femto Mega sensor was positioned near a wall, approximately 1.2 meters above ground, slightly inclined downward to maximize the captured area. Variations in sensor height and angle were tested to study impacts on capture quality and coverage. Initial tests at maximum resolution (3840×2160) provided baseline performance data without network-induced latency.

Subsequent tests incorporated the complete distributed architecture, with one laptop handling data acquisition and another managing the Node.js server and A-Frame client. Even within a local network, latency significantly increased due to data serialization, TCP packing, and WebSocket transmission, highlighting the impracticality of maximum sensor resolution for real-time interaction.

Introducing the detailed 3D GameLab model to the client further increased rendering load, reducing frame rates and causing occasional freezes due to complex draw calls. To address these issues, sensor resolution was gradually reduced and buffers optimized. Adaptive strategies, like dynamic resolution reduction during network congestion, proved too complex for the initial proof-of-concept. Thus, resolution reduction and discarding out-of-range depth values achieved the best balance between performance and fidelity.

3.3. Results

After iterative refinement, the final configuration (final interaction test depicted in Figure 2) demonstrated reliable real-time performance under typical local network conditions, with latencies ranging from 300–500 ms. Gigabit connections consistently delivered latencies closer to 300 ms, providing fluid interactions. Conversely, standard 2.4 GHz Wi-Fi introduced sporadic instabilities and higher latencies.

Figure 2. Final test of the proposed system with multi-user interaction.

Despite latency variations, a stable frame rate of approximately 10–15 fps was maintained, effectively capturing participant movements and environmental layouts. Subjective user evaluations immediately after test sessions indicated strong feelings of virtual presence at these performance levels, enhanced by consistent audio-video synchronization, accurate spatial relationships, and absence of abrupt interruptions. Thus, the results demonstrate that moderate bandwidth, reduced frame rates, and limited latency can adequately support collaborative tasks requiring only global posture recognition and object positioning.

3.4. Discussion

These experiments validate that combining volumetric point cloud data with spatial audio significantly enriches immersive telepresence experiences. The systematic exploration of trade-offs between bandwidth, sensor resolution, and real-time rendering highlights both the feasibility and technical challenges of low-cost volumetric capture. Several technical and interaction constraints emerged during development and evaluation:

- **Network Sensitivity**: Although TCP ensures reliable transmission, network congestion or packet loss can cause noticeable frame drops or latency spikes, affecting user experience fluidity.
- **Rendering Overhead**: Browser-based 3D rendering pipelines struggle with dense point clouds, especially on CPU/GPU-constrained devices, degrading frame rates significantly.
- **Resolution Trade-off**: Achieving real-time performance necessitates reduced sampling of sensor data, sacrificing finer geometric details, impacting realism and interaction precision.
- **Single-Sensor Coverage**: Reliance on a single Orbbec sensor limits capture volume and introduces occlusions, restricting potential applications.
- **Performance on Mobile Devices**: Tests on mobile devices successfully loaded point clouds but revealed substantial processing demands, causing low frame rates and delayed rendering, emphasizing the need for further optimizations or adaptive streaming.

Depending on the application (e.g., telemedicine), maintaining higher point-cloud resolution can be critical, which in turn requires more advanced data-optimization techniques and operation over higher-bandwidth, low-latency networks; these limitations underscore future improvements, including dynamic resolution scaling, multiple-sensor integration to reduce occlusions, and advanced interpolation algorithms to enhance spatial coherence, collectively reinforcing the need for a multilayered architecture that carefully balances acquisition, transmission, and rendering to optimize performance and quality.

4. Conclusion

We presented a pipeline for the real-time capture, transmission, and rendering of RGB-D point clouds using an Orbbec sensor within a web-based virtual environment, complemented with synchronized spatial audio. The experiments demonstrated that latencies below 350 ms are achievable under typical local area network (LAN) conditions, preserving sufficient fidelity for interactive telepresence scenarios. Successful integration of volumetric data with spatial audio significantly enhances the sense of co-presence, highlighting the potential of low-cost volumetric systems in collaborative domains.

Additionally, the system distinguishes itself from related work by implementing precise spatial overlap between physical and virtual environments. This configuration enables virtual users to directly interact with the physical space, while physically present users wearing VR headsets experience corresponding alignment between both worlds. Consequently, the system fosters continuous bidirectional interaction between virtual and physical participants.

However, this study also highlights several challenges — particularly regarding occlusions resulting from single sensors, rendering overhead within browser environments, and the delicate balance between transmission efficiency and graphical detail. An ongoing enhancement plan is being developed to address key improvement areas:

- Implementing additional Orbbec sensors and developing robust calibration and point cloud registration methods to extend coverage and minimize blind spots, enabling continuous 360-degree telepresence scenarios.
- Exploring compression techniques such as octree-based methods, graph-based algorithms, or GPU-accelerated solutions to reduce network load without sacrificing essential geometric and chromatic details, and FoV-adaptive visibility-prediction methods that stream only likely-visible portions of point-cloud video [Li et al. 2025].
- Introducing real-time feedback on network conditions and client resources to dynamically adjust frame rates, resolution, and compression levels, ensuring interactivity under unstable conditions.
- Moving beyond visualization to support collaborative editing of virtual objects or real-time annotations on point clouds, benefiting fields such as education, design, and telemedicine.

Acknowledgment

This work was supported by The Advanced Knowledge Center in Immersive Technologies (AKCIT), with financial resources from the PPI IoT/Manufatura 4.0 / PPI HardwareBR of the MCTI grant number 057/2023, signed with EMBRAPII, and by the Fundação de Apoio à Pesquisa do Estado de Goiás (FAPEG), Brazil.

References

Alkhalili, Y., Meuser, T., and Steinmetz, R. (2020). A survey of volumetric content streaming approaches. In *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, pages 191–199.

Hudák, M., Korečko, , and Sobota, B. (2020). Advanced user interaction for web-based collaborative virtual reality. In *2020 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 000343–000348.

Kim, H. S., Hong, S. J., Yu, C. R., and Gil, Y. H. (2024). Rgb-d surface reconstruction using 3d gaussian splatting. In *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 2217–2218.

Kodama, R., Koge, M., Taguchi, S., and Kajimoto, H. (2017). Coms-vr: Mobile virtual reality entertainment system using electric car and head-mounted display. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 130–133.

Li, C., Zong, T., Hu, Y., Wang, Y., and Liu, Y. (2025). Spatial visibility and temporal dynamics: Rethinking field of view prediction in adaptive point cloud video streaming. In *Proceedings of the 16th ACM Multimedia Systems Conference*, MMSys '25, page 24–34, New York, NY, USA. Association for Computing Machinery.

Liu, Z., Li, Q., Chen, X., Wu, C., Ishihara, S., Li, J., and Ji, Y. (2021). Point cloud video streaming: Challenges and solutions. *IEEE Network*, 35(5):202–209.

Ostrowski, A. and Gaczkowski, P. (2021). *Software Architecture with C++: Design modern systems using effective architecture concepts, design patterns, and techniques with C++20*. Packt Publishing.

Santos, S. G. and Cardoso, J. C. S. (2019). Web-based virtual reality with a-frame. In *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–2.

Zhang, X. (2019). The college english teaching reform supported by multimedia teaching technology and immersive virtual reality technology. In *2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, pages 77–80.

Zhang, Z., Wang, C., Weng, D., Liu, Y., and Wang, Y. (2019). Symmetrical reality: Toward a unified framework for physical and virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1275–1276.

Zhao, Y., Baghaei, N., Schnack, A., and Stemmet, L. (2021). Assessing telepresence, social presence and stress response in a virtual reality store. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 52–56.