# MeshReduce: Split Rendering of Live 3D Scene for Virtual Teleportation


Tao Jin*
Carnegie Mellon University

Edward Lu†
Carnegie Mellon University

Mallesham Dasari‡
Northeastern University

Kittipat Apicharttrisorn§
Nokia Bell Labs

Srinivasan Seshan¶
Carnegie Mellon University

Anthony Rowe‖
Carnegie Mellon University
Bosch Research



Figure 1: An example scene showing MeshReduce delivering live captured 3D video to devices with WebXR-enabled browsers. MeshReduce captures the live environment as textured meshes, streams the captured scene to a split rendering framework, and composites high-resolution remote rendered people with a locally rendered low-polygon photogrammetry background.


## Abstract

The pursuit of immersive telepresence has always aimed to capture and stream 3D environments, enabling remote viewers to observe scenes from any view angle. However, realizing this vision remains demanding, especially with current mobile AR/VR devices, due to intricate scene details, network latency, and bandwidth constraints. This demo introduces MeshReduce, an innovative approach that integrates a novel distributed 3D scene capture technique with a split rendering framework. Our demo shows a prototype of a cross-platform, live 3D telepresence system that can be viewed on standard web browsers. The capture setup consists of multiple depth sensors, capturing users and the background scene in real time. MeshReduce uniquely allows for real-time rendering of remotely captured 3D scenes, seamlessly merging them with content on the user's device.



**Index Terms:** Computing methodologies—Computer graphics—Graphics systems and interfaces—Mixed / augmented reality; Information systems—Information systems applications—Multimedia information systems—Multimedia content creation


## 1 Introduction

Immersive telepresence, crucial in logistics, healthcare, and hospitality, relies on real-time 3D scene capture and streaming, merging high-rate data from various sensors to comprehensively visualize an environment. Systems like Holoportation [5], Volograms [7], and Project Starline [3] have been pioneering this domain. However, operating these systems effectively over standard networks, with the presence of background traffic, remains a significant challenge.

Capturing 3D scenes traditionally involves either generating RGB-D, point cloud [1], or creating textured meshes [8]. However, the former requires clients to convert large-scale scenes into geometries for rendering, increasing the graphics load. In contrast, textured meshes can be decimated to achieve significantly less scene complexity without sacrificing texture resolution and perceptual quality, making it a more efficient representation for 3D spaces.

Traditional 3D scene capture methods struggle with high computational loads and memory usage, particularly for detailed, large-scale scenes. Systems like Holoportation require substantial GPU memory, and streaming 3D content, especially with interactive AR/VR elements, often leads to significant latency. Although existing research [9] explores real-time compression and streaming of 3D human performance, it fails to composite interactive VR content with real-time captured environments effectively. This necessitates improved methods to blend interactive virtual elements with real-time, high-resolution 3D captures, balancing low-latency interaction with superior rendering quality.

MeshReduce demonstrates an end-to-end telepresence system that addresses these challenges. First, it uses a network of depth sensors to capture spaces in real-time at scale. Through deploying compute nodes directly at the sensor side, MeshReduce efficiently converts the sensor data into the meshes without suffering from compute and memory bounds [2]. In addition, a merging server takes in these intermediate meshes from sensors and incrementally merges them into a full scene description. Secondly, MeshReduce uses RenderFusion [4], a split rendering framework, to effectively balance latency and visual quality, allowing an optimized distribution of rendering tasks: contents that are complex and exceed local compute capacity are offloaded to remote servers, while interactive virtual content that requires low latency response is rendered locally on the device. MeshReduce combines these two approaches to produce a seamless and immersive telepresence experience, effectively merging remotely streamed spaces with interactive virtual content.


*e-mail: taojin@andrew.cmu.edu
†e-mail: elu2@andrew.cmu.edu
‡e-mail: m.dasari@northeastern.edu, work done while at CMU
§e-mail: kittipat.apicharttrisorn@nokia-bell-labs.com, work done while at CMU
¶e-mail: srini@cs.cmu.edu
‖e-mail: agr@andrew.cmu.edu

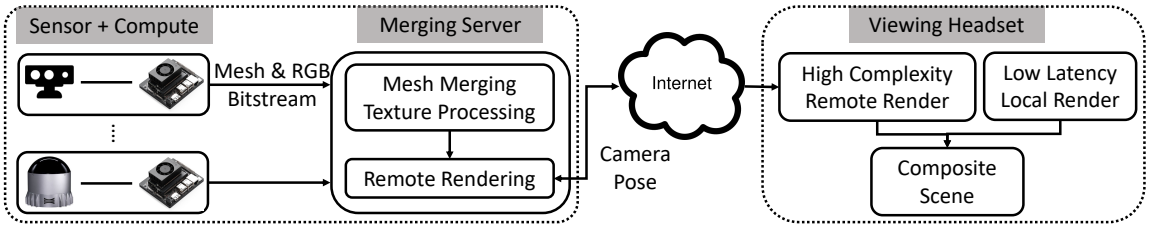


Figure 2: MeshReduce's end-to-end scene capture, merge, and rendering pipeline.

## 2 System Overview

MeshReduce leverages sensor side compute nodes to perform partial scene reconstruction. It supports heterogeneous sensors such as RGB-D cameras and LiDARs. A merging server pulls the partial reconstructions together and merges them into a single scene. On the client side, a split rendering approach is used to composite remotely rendered high-complexity content with locally rendered latency-sensitive content. Fig. 2 shows a system block diagram.

### 2.1 Distributed 3D Scene Capture

To address the challenges around GPU memory limitations and compute latency, MeshReduce employs a distributed capture pipeline. This setup involves multiple compute nodes, each dedicated to a single camera. On these nodes, per-camera scene reconstruction is performed. Specifically, we use the Truncated Signed Distance Function to model an implicit surface representation, followed by Marching Cubes for mesh extraction. For mesh decimation, we implement a parallel Quadric-Error Metric surface decimation. The system efficiently combines these individual camera feeds into a unified 3D model from separate compute nodes. Each node, equipped with adequate GPU resources, is capable of handling the scene reconstruction and decimation for its respective camera, ensuring minimal latency in the process.

### 2.2 Mesh Merging

After gathering individual sensor reconstructions, MeshReduce integrates them into a single 3D model at the merging server. Theoretically, merging mesh models involves combining data and removing duplicate surfaces. However, inaccuracies in camera calibration and depth noise often lead to misalignments. To address this, MeshReduce identifies and retains only the visible mesh layers through raycasting, discarding obscured parts. It then merges mesh edges by connecting the nearest neighboring vertices. This approach streamlines the integration process, producing a unified final model.

### 2.3 Split Rendering

Our viewing application is built using ARENA [6] and can run on web browsers on standalone WebXR-enabled headsets for cross-platform compatibility. Rendering live 3D meshes at full resolution on thin WebXR clients would be nearly impossible due to the massive data rates and rendering power required. Instead of decimating the meshes even further, which would reduce quality, we offload rendering of the live meshes to a nearby, powerful server. Additionally, to keep user-perceived latency low, we employ split rendering. Our solution divides the scene into high-resolution elements rendered by a remote server (live 3D meshes) and low-latency objects rendered on the headset (controller models and UI elements).

The server streams a virtual camera frame synchronized to a user's head pose to the client as video (we stream a stitched side-by-side color + depth frame). The viewing application then reprojects the color frames to mask motion-to-photon latencies for head movements. The locally rendered objects are rendered with a standard GPU rendering pipeline, and the result is composited with the reprojected color frame, using the depth frame to account for local-remote object occlusions. With this, we are able to render a complex, live volumetric video on a standard web browser without sacrificing interaction latency.

During the demo, we present several portions of our system:

- We show a live 3D capture of the demo area, reconstructed in real-time, using several Azure Kinect cameras.
- We let participants try an immersive viewing application run in a web browser on AR/VR headsets (i.e., Magic Leap 2, Quest Pro), tablets, and laptops. Headset users are able to view the live 3D textured mesh in full stereoscopic 3D.
- We also have a small interactive application to show that user-perceived latency for inputs is low.


## Acknowledgments

This work was supported by NSF award CNS-1956095, Bosch Corporate Research, NSF Graduate Research Fellowship DGE-2140739, and CMU Manufacturing Futures Institute. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. We thank Haoming Jing for helping with data collection.


## References


[1] B. Han, Y. Liu, and F. Qian. Vivo: Visibility-aware mobile volumetric video streaming. In *Proceedings of the 26th annual international conference on mobile computing and networking*, pp. 1–13, 2020. 1

[2] T. Jin, M. Dasari, C. Smith, K. Apicharttrisorn, S. Seshan, and A. Rowe. Meshreduce: Scalable and bandwidth efficient 3d scene capture. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2024. 1

[3] J. Lawrence, D. B. Goldman, S. Achar, G. M. Blascovich, J. G. Desloge, T. Fortes, E. M. Gomez, S. Häberling, H. Hoppe, A. Huibers, et al. Project starline: A high-fidelity telepresence system. 2021. 1

[4] E. Lu, S. Bharadwaj, M. Dasari, C. Smith, S. Seshan, and A. Rowe. Renderfusion: Balancing local and remote rendering for interactive 3d scenes. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 312–321. IEEE, 2023. 1

[5] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, et al. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th annual symposium on user interface software and technology*, pp. 741–754, 2016. 1

[6] N. Pereira, A. Rowe, M. W. Farb, I. Liang, E. Lu, and E. Riebling. Arena: The augmented reality edge networking architecture. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 479–488. IEEE, 2021. 2

[7] Volograms. Volograms. https://www.volograms.com/. Online. Accessed: Sep 2022. 1

[8] P. Stotko, S. Krumpen, M. B. Hullin, M. Weinmann, and R. Klein. Slamcast: Large-scale, real-time 3d reconstruction and streaming for immersive multi-client live telepresence. *IEEE transactions on visualization and computer graphics*, 25(5):2102–2112, 2019. 1

[9] D. Tang, M. Dou, P. Lincoln, P. Davidson, K. Guo, J. Taylor, S. Fanello, C. Keskin, A. Kowdle, S. Bouaziz, et al. Real-time compression and streaming of 4d performances. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 1