Split Rendering for Mixed Reality: Interactive Volumetric Video in Action

Jangwoo Son
Fraunhofer HHI

Serhan Gül
Fraunhofer HHI

Gurdeep Singh Bhullar
Fraunhofer HHI

Gabriel Hege
Fraunhofer HHI

Wieland Morgenstern
Fraunhofer HHI

Anna Hilsmann
Fraunhofer HHI

Thomas Ebner
Volucap GmbH

Sven Bliedung
Volucap GmbH

Peter Eisert
Fraunhofer HHI
Humboldy University of Berlin

Thomas Schierl
Fraunhofer HHI

Thomas Buchholz
Deutsche Telekom AG

Cornelius Hellge*
Fraunhofer HHI

ABSTRACT

This demo presents a mixed reality (MR) application that enables free-viewpoint rendering of interactive high-quality volumetric video (VV) content on Nreal Light MR glasses, web browsers via WebXR and Android devices via ARCore. The application uses a novel technique for animation of VV content of humans and a split rendering framework for real-time streaming of volumetric content over 5G edge-cloud servers. The presented interactive XR experience showcases photorealistic volumetric representations of two humans. As the user moves in the scene, one of the virtual humans follows the user with his head, conveying the impression of a true conversation.

CCS CONCEPTS

• **Information systems** → **Multimedia streaming**; • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools*; • **Networks** → *Cloud computing*.

KEYWORDS

volumetric video, mixed reality, cloud rendering

1 INTRODUCTION

Volumetric video (VV) reproduces 3D spaces and objects in a photo-realistic manner and is one of the enabling technologies for mixed reality (MR). During playback, viewers can freely navigate with six degrees of freedom (6DoF) and watch the volumetric content from different viewpoints [Schreer et al. 2019b]. However, VV is usually limited to playback of recorded scenes. The classical approach of using hand-crafted computer graphics (CG) models for animation lack photorealism. Our recent work has proposed to use a new hybrid geometry- and video-based animation method that combines the photorealism of VV with the flexibility of CG models [Hilsmann et al. 2020]. Our approach allows a direct animation of high-quality data itself instead of creating an animatable CG model that resembles the capture data. Thus, a photorealistic re-animation and alternation of an actor's performance captured in a volumetric studio becomes possible.

*Corresponding author: cornelius.hellge@hhi.fraunhofer.de

However, rendering VVs on mobile devices is a challenging task due to the complex nature of 3D data. Particularly, interactivity requires changing the volumetric object according to the user input or position, which is especially challenging on mobile devices with low processing power. The presence of multiple volumetric objects in the scene further increases the rendering complexity. Moreover, no efficient hardware decoders for point clouds or meshes are available yet in mobile devices.

This work demonstrates a cloud-based interactive VV streaming system that offloads heavy rendering to a 5G edge cloud and thus splits the workload between the cloud server and the MR device (*split* rendering). We have developed different client applications for Nreal Light MR glasses [Nreal 2019] and web browsers using WebXR [W3C 2019]. In the virtual demo, we showcase our system on an Android phone in an MR environment developed using ARCore [Google 2018].

2 DESIGN AND IMPLEMENTATION

2.1 Animatable volumetric video

The creation of high-quality animatable VV starts with the creation of free viewpoint video by capturing an actor's performance in a volumetric studio and computing a temporal sequence of 3D meshes using a professional capturing with sophisticated multi-view setup, e.g. [Schreer et al. 2019a]. Alternatively, lighter and cheaper systems [Alldieck et al. 2019; Robertini et al. 2016] can be applied, depending on the use case requirements. The reconstructed temporally inconsistent meshes are converted into spatio-temporally coherent mesh

Figure 1: Animating the head pose of a virtual human, to achieve a more natural conversation by following the user.

(sub-) sequences to facilitate texturing and compression. We follow a key-frame-based approach and register a number of key-frames to the captured data [Morgenstern et al. 2019]. This provides local temporal stability while preserving the captured geometry and mesh silhouette. In order to make the volumetric data animatable, we fit a parametric human model [Fechteler et al. 2016; Robertini et al. 2016] to the captured data [Fechteler et al. 2019]. Thereby, we enrich the captured data with semantic pose and animation data taken from the model, used to drive the animation of the captured VV data itself.

The kinematic animation of the individual volumetric frames is facilitated through the body model fitted to each frame. We compute correspondences between the captured mesh and the fitted template model. For each vertex of the mesh, the location relative to the closest triangle of the template model is calculated, virtually gluing the mesh vertex to the template triangle with constant distance and orientation. During playback, we modify the joints in the neck of the fitted template model in order to orient the actor's face directly towards the user's point-of-view. Artifacts in geometry and the display of unnatural poses can be avoided by restricting the offset of joint rotations to a range around the original pose. The changes in the body model are transferred to the captured mesh through the use of the computed correspondences. We thus preserve the real and fine detail of the captured content, which is lost when using traditional animation approaches. An example frame of a volumetric video capture and the same object with the modified head pose is shown in Figure 1.

2.2 Cloud-based volumetric video streaming

At the cloud server, the created interactive VV is stored as a single MP4 file with different tracks containing the compressed texture atlas and mesh data, encoded using H.264/AVC and Google Draco, respectively. We created a cross-platform cloud rendering library written in C++ that is integrated into a Unity application as a native plug-in. Our 5G edge cloud server runs the Unity application that pre-renders a 2D view from the VV depending on the user's head pose and interaction with the MR scene. For media processing, we use the Gstreamer framework with our custom plug-in for demultiplexing the MP4 file and decoding the Draco mesh stream. The obtained 2D video stream is then compressed using Nvidia NVENC [Nvidia 2020] and then streamed to the client using WebRTC. Control data e.g. user's head pose is signaled using WebSockets.

For a smooth user experience and feeling of true presence, the overall latency i.e. motion-to-photon (M2P) latency of the system should be kept as low as possible. We implemented several optimizations in our system e.g. hardware video encoding using Nvidia NVENC, edge computing, WebRTC streaming to reduce the latency caused by different components in our system. Running the server on an AWS instance in Frankfurt and with the WiFi connected client in Berlin we measured an average network latency of 13.3 ms and a M2P latency around 60 ms.

A detailed software architecture of our system is described in [Gül et al. 2020].

2.3 Mixed reality client

We have implemented different client applications for Nreal Light MR glasses, web browsers (using WebXR) and Android devices (using ARCore). Our Nreal client offers an improved feeling of presence/immersion due to the lightweight and high-quality see-through MR glasses. Moreover, the stereo viewing capability of our client offers a more realistic view of the virtual objects.

For each frame, the server captures/transmits a new view of each object based on the user pose. Also, the volumetric mesh is altered based on user input so that the head of the virtual human is turned based on the user pose. Our client applications render the 2D scene onto two different planes (one for each eye) orthogonal to the user's point of view. Since this plane is always rotated towards the user, the user perceives the different 2D views rendered onto the orthogonal plane as though a 3D object were present in the scene. A custom shader is used to remove the background of the volumetric object inside the video stream such that the object is perceived to be integrated into the real world.

3 DEMONSTRATION

For the virtual demonstration, we recorded a video to showcase our MR system on an Android phone (OnePlus 7, Android v10). In the demo, two photorealistic volumetric objects (virtual humans) are present and integrated into the real world. Users can freely walk around and view the virtual humans from different angles and distances on the display of the phone. Depending on the position and orientation of the user, one of the virtual humans rotate his head towards the user and thus convey a feeling of a true interaction.

REFERENCES

Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. 2019. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*. 2293–2303.

Philipp Fechteler, Anna Hilsmann, and Peter Eisert. 2016. Example-based Body Model Optimization and Skinning. In *Proc. Eurographics 2016*. Eurographics, Lisbon, Portugal.

Philipp Fechteler, Anna Hilsmann, and Peter Eisert. 2019. Markerless Multiview Motion Capture with 3D Shape Model Adaptation. *Computer Graphics Forum* 38, 6 (2019), 91–109.

Google. 2018. ARCore. https://developers.google.com/ar.

Serhan Gül, Dimitri Podborski, Thomas Buchholz, Thomas Schierl, and Cornelius Hellge. 2020. Low-latency Cloud-based Volumetric Video Streaming Using Head Motion Prediction. In *Proceedings of the 30th Workshop on Network and Operating Systems Support for Digital Audio and Video - NOSSDAV'20*. ACM Press.

Anna Hilsmann, Philipp Fechteler, Wieland Morgenstern, Wolfgang Paier, Ingo Feldmann, Oliver Schreer, and Peter Eisert. 2020. Going beyond Free Viewpoint: Creating Animatable Volumetric Video of Human Performances. *IET Computer Vision* (2020).

Wieland Morgenstern, Anna Hilsmann, and Peter Eisert. 2019. Progressive Non-Rigid Registration of Temporal Mesh Sequences. In *European Conference on Visual Media Production (CVMP 2019)*. ACM, London, UK, Article 9, 10 pages.

Nreal. 2019. Nreal Light MR glasses. https://www.nreal.ai/specs.

Nvidia. 2020. Nvidia Video Codec SDK. https://developer.nvidia.com/nvidia-video-codec-sdk.

Nadia Robertini, Dan Casas, Helge Rhodin, Hans-Peter Seidel, and Christian Theobalt. 2016. Model-based Outdoor Performance Capture. In *Proc. of 3DV*. IEEE, Stanford, USA.

Oliver Schreer, Ingo Feldmann, Peter Kauff, P Eisert, D Tatzelt, C Hellge, K Müller, T Ebner, and S Bliedung. 2019a. Lessons learnt during one year of commercial volumetric video production. In *Proceedings of IBC conference, Amsterdam, Netherlands, September*.

Oliver Schreer, Ingo Feldmann, Sylvain Renault, Marcus Zepp, Markus Worchel, Peter Eisert, and Peter Kauff. 2019b. Capture and 3d video processing of volumetric video. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 4310–4314.

W3C. 2019. WebXR Device API. https://www.w3.org/TR/webxr/.