MagicStream: Bandwidth-conserving Immersive Telepresence via Semantic Communication

Ruizhi Cheng
George Mason University
Fairfax, VA, USA
rcheng4@gmu.edu

Nan Wu
George Mason University
Fairfax, VA, USA
nwu5@gmu.edu

Vu Le
University of Massachusetts Amherst
Amherst, MA, USA
vdle@umass.edu

Eugene Chai
Nokia Bell Labs
Murray Hill, NJ, USA
eugene.chai@nokia-bell-labs.com

Matteo Varvello
Nokia Bell Labs
Murray Hill, NJ, USA
matteo.varvello@nokia.com

Bo Han
George Mason University
Fairfax, VA, USA
bohan@gmu.edu

ABSTRACT

Immersive telepresence has the potential to revolutionize remote communication by offering a highly interactive and engaging user experience. However, state-of-the-art exchanges large volumes of 3D content to achieve satisfactory visual quality, resulting in substantial Internet bandwidth consumption. To tackle this challenge, we introduce MagicStream, a first-of-its-kind *semantic-driven* immersive telepresence system that effectively extracts and delivers compact semantic details of captured 3D representation of users, instead of traditional bit-by-bit communication of raw content. To minimize bandwidth consumption while maintaining low end-to-end latency and high visual quality, MagicStream incorporates the following key innovations: (1) efficient extraction of user's skin/cloth color and motion semantics based on lighting characteristics and body keypoints, respectively; (2) novel, real-time human body reconstruction from motion semantics; and (3) on-the-fly neural rendering of users' immersive representation with color semantics. We implement a prototype of MagicStream and extensively evaluate its performance through both controlled experiments and user trials. Our results show that, compared to existing schemes, MagicStream can drastically reduce Internet bandwidth usage by up to 1195× while maintaining good visual quality.

CCS CONCEPTS

• **Information systems → Multimedia streaming**; • **Computing methodologies → Mixed / augmented reality**.

KEYWORDS

Immersive Telepresence, Semantic Communication, Neural Rendering, User Experience

ACM Reference Format:
Ruizhi Cheng, Nan Wu, Vu Le, Eugene Chai, Matteo Varvello, and Bo Han. 2024. MagicStream: Bandwidth-conserving Immersive Telepresence via Semantic Communication. In *The 22nd ACM Conference on Embedded Networked Sensor Systems (SENSYS '24), November 4–7, 2024, Hangzhou, China.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3666025.3699344

1 INTRODUCTION

Immersive telepresence, a primary use case in the envisioned 6G [31, 85, 88], holds the promise of revolutionizing remote communication by providing deeply engaging and interactive user experiences. Immersive content that depicts 3D objects/scenes is typically represented by point clouds or meshes [14, 19], allowing users to not only change view directions but also move freely in 3D space, known as six degrees of freedom (6DoF) motion. This capability has propelled the adoption of immersive content across various domains, such as healthcare, education, professional training, scientific data visualization, and entertainment [95]. As a result, recent years have seen intensifying efforts to advance immersive content delivery and enhance its quality of experience (QoE) [40, 49, 60, 103, 115, 117].

While existing solutions primarily optimize *video-on-demand* services that distribute pre-recorded content, *live immersive content delivery* offers a broader spectrum of compelling applications for telepresence, such as telesurgery [25] and remote collaborations [92]. However, achieving a truly immersive and highly interactive user experience for telepresence poses the following challenges.

- Due to its 3D nature, streaming high-fidelity immersive content demands substantial network bandwidth, for example, >1 Gbps throughput in Holoportation [71].
- The interactivity of immersive telepresence necessitates ultra-low end-to-end latency for live content delivery, typically under 100 ms for one-way communication [16, 18, 69].
- Real-time delivery of immersive content requires maintaining at least 30 frames per second (FPS) streaming rate (*i.e.,* <33 ms processing time per frame), to ensure fluid motion and continuity in user interactions [39, 40, 48, 60, 115, 117].

Meanwhile, today's Internet may not adequately support bit-by-bit communication of raw data for immersive telepresence with representations such as point cloud and mesh. Existing point-cloud-based systems for immersive telepresence, even at medium quality, can demand over 70 Mbps of bandwidth per participant [39], which is close to what is offered by standard broadband services in the U.S. (*i.e.,* 100 Mbps [28]). However, sudden bandwidth drops are common and can significantly affect QoE [83]. Also, network links

SENSYS '24, November 4–7, 2024, Hangzhou, China

ACM ISBN 979-8-4007-0697-4/24/11
https://doi.org/10.1145/3666025.3699344

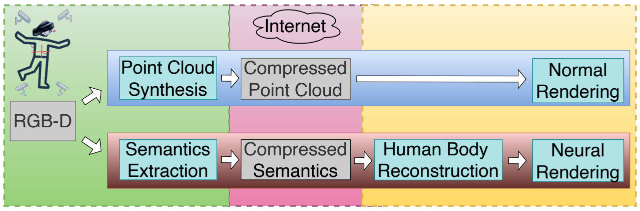

Figure 1: Comparison of a representative existing scheme MetaStream [39] (top) and our MagicStream (bottom).

are typically shared among multiple applications, making it undesirable for a single application to consume the majority of the available bandwidth [64]. Furthermore, certain use cases for immersive applications, such as video conferencing, often involve multiple concurrent users, which exacerbates bandwidth demands [64]. Therefore, it is imperative to optimize the bandwidth consumption of immersive telepresence systems.

In response to these challenges, we present MagicStream, which is, to the best of our knowledge, the first semantic-driven immersive telepresence system. Our key insight is that delivering immersive content with exact bit-by-bit precision, which leads to high-bandwidth demands, is often unnecessary in certain use cases such as teleconferencing. Instead, the focus should be on conveying pivotal interactions or notable events, such as a speaker's key gestures and facial expressions. Thus, for telepresence that involves mainly users, we can transmit only their *meaningful semantic details*, which are used to reconstruct their immersive representation. Figure 1 compares the common bit-by-bit communication employed by the state-of-the-art, MetaStream [39], and semantic communication adopted by MagicStream. We further compare MagicStream with other existing systems [47, 48, 71] in Table 1.

The overarching goal of MagicStream is to leverage semantic communication to significantly reduce bandwidth usage, while preserving low end-to-end latency and ensuring satisfactory visual quality simultaneously for users wearing a resource-constrained mobile headset such as Microsoft HoloLens 2 [2]. Realizing semantic communication for immersive telepresence typically entails the extraction of semantics at the sender, transmission over the Internet, and reconstruction of immersive content from derived semantics at the receiver. Hence, fundamentally, the principal challenges for designing MagicStream are: 1) *properly identifying and accurately extracting semantics from captured 3D data of users*, and 2) *reconstructing users' immersive representation from received semantics with high visual quality, both in real time*. To address these challenges, MagicStream incorporates the following innovations into a holistic system.

Efficient Extraction of Color & Motion Semantics (§4.1). Extracting semantics is the first key step in semantic communication for MagicStream. Typically, a user's representation can be decomposed into skin/clothing color and body movement. For color semantics, our insight is that the user's perceived color is a combination of the base color of the skin/cloth, which refers to the original color before applying any lighting effects in computer graphics, and the lighting characteristics of a scene. Under a reasonable assumption that the user's base color does not change during telepresence, deriving color semantics is equivalent to extracting the lighting characteristics of the scene. Utilizing a lightweight lighting estimation model [121], MagicStream achieves real-time extraction of color semantics. For motion semantics, our insight is that tracking the positions of specific important points (known as keypoints) in the human body, such as joints, provides a sufficient basis to model user movements. Thus, MagicStream capitalizes on an efficient keypoint detection framework [62] to extract motion semantics.

System	Tput	Full Body	Vis. Qual.	Headset
Holoportation [71]	H	✓	H	✓
Project Starline [47]	H	✗	H	✗
FarfetchFusion [48]	M	✗	M	✗
MetaStream [39]	H	✓	M	✓
MagicStream	L	✓	H	✓

Table 1: Comparison of MagicStream with existing immersive telepresence systems. Tput: throughput and Vis. Qual.: Visual Quality. L: low; M: medium; and H: high.

Real-time Reconstruction from Motion Semantics (§4.2). Upon extracting motion semantics, MagicStream reconstructs the human body by utilizing keypoints to drive SMPL-X [73], a parametric human model that can precisely represent body movements over time. Despite the vital importance of SMPL-X in depicting human form and motion, accurately estimating its parameters in real time is challenging. Indeed, existing approaches sustain only <2 FPS (§2.2), inadequate to ensure fluid motion and interactivity. Thus, we propose a novel regression-based method for estimating in real time SMPL-X parameters with high accuracy from a sparse set of keypoints. Given that this task is still compute-intensive, MagicStream conducts it at the sender side and delivers SMPL-X parameters, instead of keypoints, to balance the computation overhead of the sender and receiver.

On-the-fly Neural Rendering with Color Semantics (§4.3). On the receiver side, upon obtaining color semantics and SMPL-X parameters, which are used to create an SMPL-X mesh, MagicStream employs an optimized neural-rendering pipeline to display high-quality content of the remote user in real time. This process involves a two-stage rendering approach: initial rasterization [35] for rendering a basic 2D image based on the SMPL-X mesh and color semantics, followed by refinement with an image-based neural network for rendering a high-quality representation of the remote user. As neural rendering is time-consuming, we design a patch-based acceleration strategy, which updates only specific portions of the human body with noticeable changes, balancing the need for real-time performance and high visual quality for immersive telepresence. By doing this, we can also enable parallel rendering of multiple patches.

Implementation and Evaluation of MagicStream (§5, §6). We build a prototype of MagicStream and thoroughly evaluate its performance with controlled experiments on a dataset collected via an IRB-approved user study to make our results reproducible and separate user trials to assess the QoE. We summarize our key experimental results as follows.

- When offering similar visual quality as the state-of-the-art, MetaStream [39], MagicStream drastically reduces bandwidth consumption by 1195×, operating at only 0.2 Mbps.

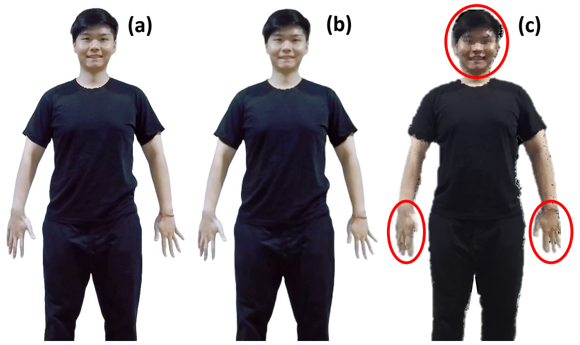

Figure 2: Qualitative comparison of (a) ground truth, (b) MagicStream**, and (c) MetaStream [39]. The network has a symmetric bandwidth of ∼150 Mbps.**

- Under a round-trip time of 40 ms [30], MagicStream achieves a low one-way end-to-end latency of ∼95 ms, satisfying the stringent requirement of interactive applications (*i.e.,* <100 ms) [18], and marks a 44.1% reduction compared to MetaStream [39].
- Under various network conditions, MagicStream consistently operates at 30+ FPS, surpassing MetaStream [39], whose FPS is <10 for bandwidth-constrained scenarios.
- MagicStream achieves an SSIM (structural similarity) [101] index of 0.92, which is calculated with screenshots collected on the Microsoft HoloLens 2 headset [2] at the receiver and the ground-truth images captured at the sender, indicating a good visual quality [29]. In contrast, MetaStream leads to an SSIM index of ∼0.8.
- Our second user study indicates that MagicStream results in a better QoE compared to MetaStream [39] under different network conditions, with up to 86.7% improvement.

Beyond the above key results, we demonstrate how MagicStream can quickly adapt to various changes in user appearance by fine-tuning its neural rendering model in §6.2. Figure 2 qualitatively compares MagicStream and MetaStream [39] over a network with ∼150 Mbps symmetric bandwidth. We present more qualitative comparisons of them in §6.3 and record a video to visually demonstrate their rendering quality[1]. Note that compared to 3D reconstruction with wireless sensing [100, 106, 108, 119], which requires additional hardware and struggles to obtain colors, the vision-based approach in MagicStream is more suitable for telepresence by directly capturing colors with cameras, an essential component of such applications. This work does not raise any ethical issues.

2 BACKGROUND AND MOTIVATION

2.1 Background

Immersive Telepresence involves real-time capture, creation, delivery, and rendering of immersive content, typically with multiple RGB-D (D for depth) cameras to cover different viewing angles [39, 48, 71]. The synthesis of RGB-D images from these cameras, achieved through synchronization, calibration, and filtering, enables the generation of free-view 3D models, which are commonly represented as textured meshes or point clouds [114]. Mesh, a network of interconnected vertices forming a cohesive structure, incorporates both geometry (defining the shape and structure) and texture (adding surface detail such as color). In contrast, point clouds are a set of discrete points with colors in 3D space. In this paper, we primarily focus on users as the main subject of immersive content, since they represent the key component in telepresence [39, 47, 48, 71].

To ensure a satisfactory QoE, delivering immersive content in point clouds or meshes necessitates significantly higher network bandwidth than 2D video streaming. Relying exclusively on the compression of immersive content is not sufficient due to its low compression ratio [40]. Consequently, recent research has shifted focus towards strategies that optimize both communication and computation overhead [39, 48]. Despite these advancements, the bandwidth requirement for transmitting medium-quality, full-body immersive content is still high (*e.g.,* ∼70 Mbps, as shown in §2.2).

Although volumetric content can be displayed across a variety of devices, including PCs, smartphones, tablets, and headsets, its level of interactivity is fundamentally different across devices. PCs can only emulate 6DoF motion through mouse and keyboard input. While smartphones and tablets can track 3DoF rotational movements, localization along translational dimensions remains a challenging issue for them [15]. Current systems predominantly use on-screen finger operations to manipulate content across translational dimensions [40], resulting in a sub-optimal user experience. Conversely, headsets such as Microsoft HoloLens 2 [2] can naturally support 6DoF motion thanks to their dedicated sensors and specialized software framework.

Human Keypoints. Keypoints are specific and unique features identifiable on an object. For human beings, keypoints are predominantly situated in areas including the body, hands, and face [17]. Human keypoint detection, also known as human pose estimation, involves detecting the positions (*i.e.,* coordinates) of body parts or joints from images or videos. Recent developments have demonstrated that keypoint detection models can achieve high accuracy and real-time performance [17, 84]. However, reconstructing visually satisfactory 3D content from keypoints is challenging. Existing solutions that directly create a mesh from keypoints operate on a single-frame basis [24, 58, 68]. Thus, they do not effectively capture the temporal dynamics inherent in human motion, potentially resulting in unsatisfactory visual quality due to temporal discontinuity and visual artifacts [74]. Moreover, as keypoints do not encode texture information, the meshes reconstructed from them lack texture, resulting in a geometry that portrays a non-clothed body structure [24, 58, 67, 68].

Parametric Human Model. To enable accurate modeling of human movements over time, recent efforts [67, 79] resort to parametric human models such as SMPL-X [73]. These models are extensively pre-trained on vast video datasets to capture a wide array of human movement patterns, enabling them to accurately and smoothly model human motion in various poses. SMPL-X is a state-of-the-art parametric model with fully articulated hands and an expressive face, which are essential in immersive telepresence. It can be formulated as $V_w = W(\phi_w, \theta_w, \beta_w, \psi_f)$, where W is a linear blend skinning (LBS) function [45], $\phi_w \in \mathbb{R}^3$ represents the global orientation of the whole body, $\theta_w \in \mathbb{R}^{(21+15+15)\times 3}$ consists

[1] https://youtu.be/u7U9iPND26E

OpenPose Keypoints	SMPL-X Parameter	SMPL-X Mesh
0.35	0.16	10.1

Table 2: Comparison of required bandwidth (Mbps) at 30 FPS for OpenPose keypoints [17], SMPL-X parameters [73], and SMPL-X Mesh after data compression.

of whole-body pose parameters accounting for pose-dependent deformation, $\beta_w \in \mathbb{R}^{10}$ denotes the shape of the face, hands, and body, and $\psi_f \in \mathbb{R}^{10}$ encompasses facial expression parameters. Specifically, θ_w is subdivided into body pose parameters ($\theta_b \in \mathbb{R}^{21\times3}$), left-hand pose parameters ($\theta_{lh} \in \mathbb{R}^{15\times3}$), and right-hand pose parameters ($\theta_{rh} \in \mathbb{R}^{15\times3}$). All pose parameters are defined in the axis-angle representation [4], which denotes the relative rotation to the parent joints as defined in the kinematic map [97]. The output of SMPL-X, $V_w \in \mathbb{R}^{10,475\times3}$, is a 3D mesh comprising 10,475 vertices. The spatial position of each vertex is determined by SMPL-X parameters and its LBS weight.

2.2 Motivational Study

Traditional Bit-by-bit Communication. To benchmark the requirements of immersive telepresence employing bit-by-bit communication, we re-implement MetaStream [39], a state-of-the-art point-cloud-based telepresence system. Our implementation has a similar performance to that reported in the original paper. Over five sessions of five minutes each, the average bandwidth consumption after applying compression and communication optimization is 72.3 Mbps (SD: 8.65). Note that MetaStream transmitted only ~200K points for each frame. However, high-quality streaming may require the delivery of >1M points per frame at 30 FPS [49], potentially demanding a network bandwidth as high as 450 Mbps [49], even after considering various visibility-aware optimizations [40]. Moreover, our experiments indicate that decoding 1M points on HoloLens 2 [2] achieves only <7 FPS, much lower than the required real-time frame rate (*i.e.,* at least 30 FPS).

Semantic Communication. To drastically reduce the bandwidth demand of immersive telepresence, a potential solution is to leverage *semantic communication* where, instead of delivering raw data, we transmit "instructions" (*e.g.,* semantics for body motion and skin/cloth color) to reconstruct 3D content. We next investigate the potential benefits of semantic communication for immersive telepresence, along with its proper setup and open challenges.

We use a ZED 2i depth camera [11] to capture RGB-D data in 2K resolution and leverage OpenPose [17] to identify up to 135 2D keypoints across the human body. We then estimate SMPL-X parameters with SMPLify-X [73], a state-of-the-art optimization-based solution widely recognized as a benchmark in various studies [26, 33]. SMPLify-X uses the RGB image and the keypoints detected by OpenPose [17] to estimate SMPL-X parameters, which can be used to generate the SMPL-X mesh. We capture five sessions with ~600 frames each where users perform arbitrary poses. We use a desktop machine with an NVIDIA RTX 4090 GPU, an AMD Ryzen 9 7900X CPU, and 32GB RAM at both the sender and receiver sides of a test-case immersive telepresence.

(1) *Data Size.* We first compare the data size of OpenPose keypoints, SMPL-X parameters, and SMPL-X mesh. To compress SMPL-X mesh, we utilize Draco [5], a 3D content compression framework that is commonly used in existing systems [39, 40]. To compress OpenPose keypoints and SMPL-X parameters, we test four popular compression schemes: Zstandard [13], LZMA [6], zlib [12], and LZ4 [8]. We find that LZMA offers the highest compression ratio.

Keypoint Detection	SMPL-X Parameter Estimation	SMPL-X Mesh Reconstruction
32.2/4.23	0.018/0.007	331/1.32

Table 3: The averaged FPS and its standard deviation of OpenPose keypoint detection, SMPL-X parameter estimation from SMPLify-X [73], and SMPL-X mesh reconstruction.

Table 2 shows the required bandwidth (after compression) for these three data types at 30 FPS. Transmitting keypoints or SMPL-X parameters has the potential to dramatically reduce bandwidth usage (*i.e.,* 0.16-0.35 Mbps *vs.* 10.1 Mbps required by SMPL-X mesh). Note that although streaming the SMPL-X mesh takes only ~10 Mbps, the visual quality is low if we directly display it.

(2) *Runtime.* We next examine the FPS achieved by keypoint detection, SMPL-X parameter estimation, and SMPL-X mesh reconstruction, as shown in Table 3. Both keypoint detection and SMPL-X mesh reconstruction can operate in real time (*i.e.,* >30 FPS). However, SMPL-X parameter estimation achieves only <0.02 FPS. This is because SMPLify-X [73], while accurate, requires iterative processing during its optimization phase. In contrast, since an SMPL-X mesh has a fixed topology, generating it from parameters involves only fast vertex deformation based on pre-defined LBS weights.

(3) *Texture Transmission and Rendering.* An SMPL-X mesh contains only geometry without texture (*e.g.,* color information) [73], which is essential for photo-realistic content rendering. A straightforward solution is to transmit high-resolution, multi-view RGB-D images for mapping texture on the SMPL-X mesh at the receiver [36]. To understand the bandwidth requirement and the visual quality of this approach, we utilize three ZED 2i cameras [11] for capturing a user from different views. Following MetaStream [39], we first use H.264 [3] to encode RGB-D images. The consumed bandwidth at 30 FPS is 32.1 Mbps (SD: 5.37). We then conduct texture mapping and calculate the SSIM [101] with camera-captured images. SSIM is a metric that integrates luminance, contrast, and structural comparisons between two images to assess their similarity. Its value ranges from 0 to 1, with a larger value indicating higher similarity and, consequently, better visual quality, and vice versa. The SSIM is only 0.63 (SD: 0.02), indicating poor visual quality [29]. This primarily stems from the low-quality geometry of SMPL-X mesh, which negatively impacts rendering quality [35].

Takeaways. Our study highlights the potential of semantic communication to significantly reduce the bandwidth usage of immersive telepresence. However, semantic communication introduces the following major challenges that we aim to address in MagicStream. First, estimating SMPL-X parameters requires significant acceleration (*i.e.,* from <0.02 FPS to >30 FPS) without comprising reconstruction quality. Second, directly transmitting texture-related data still requires high bandwidth. Therefore, we need to properly identify color semantics. Third, due to potential information loss during the extraction of semantics, rendering high-quality, photo-realistic content of a user is non-trivial.

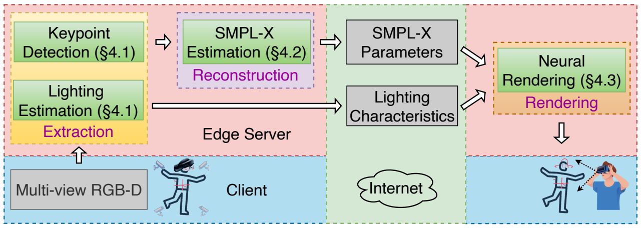

Figure 3: System architecture and workflow of MagicStream**. For simplicity, we depict only one-way communication for telepresence. The opposite direction mirrors this figure. "Reconstruction" is executed at the sender side for load balancing between the sender and receiver. The reason is that the computation-intensive neural rendering should be performed by the receiver.**

3 OVERVIEW OF MAGICSTREAM

Figure 3 depicts the system architecture of MagicStream and its workflow of semantic communication for immersive telepresence. Due to the resource restrictions of mobile headsets, users are served by an edge server that performs computation-intensive tasks (*i.e.,* SMPL-X parameter estimation and neural rendering). MagicStream delivers SMPL-X parameters, instead of keypoints, since estimating SMPL-X parameters at the sender can effectually balance the computation overhead of the sender and the receiver, which needs to conduct resource-demanding neural rendering. Also, as shown in Table 2, the data size of SMPL-X parameters is essentially smaller than that of keypoints.

Prior to telepresence, MagicStream requires a user profiling phase to fine-tune/train the following models. During this phase, participants are asked to spin a circle (~10–20s) in front of multiple recording RGB-D cameras. The data gathered from these recordings is used to fine-tune the lighting estimation model (§4.1), and train the models for SMPL-X parameter estimation (§4.2) and neural rendering (§4.3). Meanwhile, we obtain the user's base color with an existing approach [87] and transmit it along with the neural rendering model to the receiver ahead of telepresence sessions.

4 MAGICSTREAM DESIGN

4.1 Efficient Extraction of Color and Motion Semantics from RGB-D Data

Insights. Successfully extracting semantics requires precise identification of key features that represent the human body during immersive telepresence. Our design choice is motivated by the fact that 3D models such as meshes can be essentially decomposed into surface details such as colors and the geometry that defines shape and structure (§2.1). Thus, we propose to derive semantics that represent color and geometry separately in MagicStream.

To extract color semantics, consider a scenario where the user waves the hand. As the hand moves towards a light source, its color perceptibly shifts towards that of the light. Conversely, when the hand moves away from the light, it may appear darker. Given that an individual's skin color should be constant during telepresence, the observed changes in color are attributed to alterations in how light interacts with the skin. These alterations are governed by the characteristics of the light source(s) within the scene. Drawing from the above analysis, the factors influencing color representation are: (1) the inherent base color, (2) human pose, and (3) light characteristics. We can safely assume that the base color of the skin/cloth does not change during a telepresence session, and thus, it can be obtained during user profiling. Further, the human pose can be described as keypoints, which we will discuss next. Thus, our first key insight is that *lighting characteristics are the essential color semantics.*

The next step is to extract semantics to represent the geometry of the human body. One caveat is that this should not be done on a single-frame basis, which does not capture the temporal dynamics inherent in video frames, leading to unsatisfactory visual quality due to temporal discontinuity and artifacts [74]. Thus, we propose to extract motion semantics to represent the geometry, by capturing key body movements across frames. These movements are essentially driven by the articulation of bones and joints [41], which are inherently complex. Our observation is that focusing on certain critical points in the body is sufficient to approximate human motion. For example, by extracting the position of each joint in the user's hand and tracking these positional changes, we can accurately capture hand movements. These critical points are typically identified as keypoints. Therefore, our second key insight is that *keypoints could serve as motion semantics*, as tracking their changes can effectively model human body movements over time.

Solution. To extract color semantics, we employ Xihe [121], a lightweight lighting estimation framework. Xihe outputs a 27-dimensional coefficient vector to encapsulate the comprehensive lighting characteristics of the scene. However, one issue of Xihe is that its initial design is not tailored to human-centric scenarios. To address this problem, we utilize the pre-trained Xihe model as the foundation and fine-tune it with user-profiling data, which is supervised by an accurate, diffusion-based [110], and face-centered lighting estimation model [75]. We choose facial features for light estimation, instead of those of other body parts, because the facial region offers a strong geometric prior that can improve the accuracy [50]. Additionally, using facial features allows us to efficiently prevent potential distortions caused by clothing materials, such as reflections and varying textures.

The frequency of estimating and delivering light characteristics is contingent upon the specific lighting conditions of the telepresence environment. For instance, in settings exposed to sunlight, more frequent updates are needed due to the dynamic nature of natural light. Conversely, in indoor environments with stable artificial light and limited changes in the intensity and number of light sources, less frequent updates are sufficient. Nevertheless, the representation of lighting characteristics as a compact 27-dimensional coefficient vector allows for efficient data communication. Even when MagicStream transmits these coefficients for each frame, the bandwidth consumption remains remarkably low, with 53.76 Kbps at 30 FPS after applying the LZMA compression [6].

To extract motion semantics, MagicStream utilizes MediaPipe [62], a lightweight keypoint detection model, which provides more keypoints (524 *vs.* 135), faster execution (~10 ms), and higher detection accuracy [27] than OpenPose [17].

4.2 Real-time Human Body Reconstruction from Motion Semantics

Challenges. As directly reconstructing a 3D model from keypoints will lead to poor visual quality, MagicStream first reconstructs a non-colored human body by mapping keypoints to the output mesh of SMPL-X [73], a parametric human model that can represent body movements over time (§2.1). This mesh will be further refined by our proposed neural rendering model in §4.3. However, precisely estimating SMPL-X parameters that control the SMPL-X mesh in real time is non-trivial because they integrate complex representations for hand postures and facial expressions. As shown in §2.2, optimization-based schemes [73] are notoriously time-consuming. While recent studies have shown that image-centric methods [26, 33] execute faster than optimization-based ones, their processing time (*e.g.,* ∼200 ms per frame [26]) is still too high to meet the real-time requirements of interactive telepresence.

This poor performance of image-centric methods [26, 33] is due to two main drawbacks. First, they estimate SMPL-X parameters based on dense features extracted from images, incurring a high computation overhead. Second, to accurately extract dense features, they typically utilize complex deep-learning models such as attention [94], further prolonging the delay in parameter estimation.

Solution. To achieve real-time SMPL-X parameter estimation, we design a lightweight regression-based method that relies on keypoints as input, instead of images. As keypoints contain less information than images, this enables faster execution than image-centric methods [26, 33].

The goal of our regression-based method is to train a deep-learning model which establishes a relationship between the input keypoints and the output SMPL-X mesh through its SMPL-X parameters, controlling the generation of the mesh. However, directly establishing an accurate mapping between keypoints and SMPL-X mesh is non-trivial. An SMPL-X mesh has 10,475 vertices. Nonetheless, only 524 MediaPipe keypoints are available. To address this problem, MagicStream introduces an innovative step before training: for each keypoint, we add a corresponding vertex to the first ground-truth SMPL-X mesh. For example, given a keypoint on the left wrist, we add a vertex to the SMPL-X mesh at the same position. In essence, this alignment becomes an indicator of accuracy in our training objectives. Our insight is that if the reconstructed SMPL-X mesh accurately reflects the human form, any detected keypoints should naturally align with their corresponding vertices on this mesh, when they move independently over time along with body motion. Note that to enable such correspondence, we introduce additional vertices to the SMPL-X mesh, rather than seeking correspondence in it, as certain keypoints' locations/coordinates may not align with any existing vertices of the mesh.

These newly added vertices, termed *keypoint-anchored vertices* (KAVs), act as conduits. As shown in Figure 4, they directly connect keypoints with the output SMPL-X mesh, and thus aid in the precise estimation of SMPL-X parameters. We add KAVs to only the first ground truth SMPL-X mesh. For each keypoint, we add a vertex to the mesh based on the keypoint's 3D coordinates[2]. For each added vertex v_k, we find its nearest vertex v_g on the SMPL-X mesh and assign the LBS weight of v_g to v_k. By doing this, we ensure that KAVs can be controlled by estimated SMPL-X parameters and realistically deformed in harmony with the surrounding vertices (§2.1). For meshes of other frames, we reuse KAVs from the previous one while extracting keypoints from newly captured images to align with those KAVs.

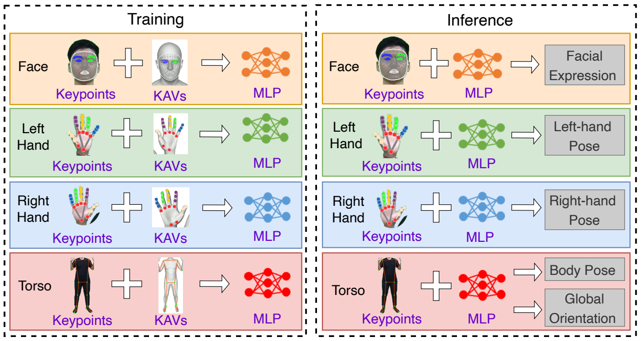

Figure 4: Workflow of our proposed SMPL-X parameter estimation (KAVs: keypoint-anchored vertices). MagicStream **trains separate models for different body parts with keypoints and KAVs. During inference, given a set of input keypoints,** MagicStream **concurrently executes trained models to estimate SMPL-X parameters for different body parts.**

By building the connection between the keypoints and the SMPL-X mesh, we can train a lightweight model to estimate SMPL-X parameters from these keypoints without compromising accuracy. Specifically, we design a five-layer multilayer perception (MLP) model and deploy four parallel instances of it for the left hand, right hand, face, and torso. To ensure that the output SMPL-X mesh faithfully replicates the actual body pose, facial expressions, and hand gestures captured by the keypoints over time, we minimize both the mean squared error (MSE) between SMPL-X parameters and their ground truth, as well as the mean per-joint position error (MPJPE) [41] between each 3D keypoint and its corresponding KAV. To supervise the training of the regression-based method, we extend SMPLify-X [73] by incorporating keypoints observed from multi-view cameras to generate SMPL-X parameters and meshes offline as the ground truth due to its high accuracy, a common approach used in previous studies [118, 122].

During inference, MagicStream processes incoming 3D keypoints from MediaPipe and applies the trained MLP models to estimate SMPL-X parameters.

4.3 On-the-fly Neural Rendering with Color Semantics

Challenges. On the receiver side, upon obtaining SMPL-X parameters (which are then used to generate an SMPL-X mesh) and color semantics, rendering a photo-realistic representation of the remote user in real time is essential for achieving a high QoE. However, it is non-trivial, given the sparse nature of the SMPL-X mesh (*i.e.,* having only 10,475 vertices). Existing methods for rendering meshes primarily involve conventional computer graphics pipelines, such as explicit mesh rasterization [35]. This approach, while efficient, often falls short of achieving high-quality rendering [89].

[2] We can lift detected 2D keypoints to 3D with depth information.

Solution. To obtain photo-realistic visual quality, we leverage neural rendering [89], an emerging rendering paradigm that utilizes deep-learning models. It combines generative deep learning [78] with physical knowledge from computer graphics, such as integrating differentiable rendering [61] into model training, aiming for high-quality rendering from sparse input data (*e.g.,* 3D meshes or 2D images).

Two-stage Rendering. While directly feeding an SMPL-X mesh into the neural renderer is feasible [44], it will lead to significant inference overhead due to the 3D nature of input mesh data. Therefore, we propose a two-stage rendering: initially employing rasterization with lighting characteristics (*i.e.,* color semantics) to generate low-quality 2D images [38], followed by an image-based neural rendering model for photo-realistic results.

To ensure high-quality outcomes, the ground-truth images for training the neural rendering model should encompass various viewports. However, cameras are usually positioned horizontally, making it difficult to capture images for challenging scenarios (*e.g.,* when users view content with their heads tilted). Rotating cameras during training data acquisition is not only cumbersome but also inefficient for covering all angles. To address this problem, we implement image augmentation techniques [81], including translation and rotation, to enhance the diversity of training data and generate images for different head-tilt viewing angles.

Patch-accelerated Inference. To improve neural-rendering efficiency for immersive telepresence with stringent real-time requirements, we propose an optimization strategy. Our rationale is that in certain use cases, such as teleconferencing, only a few regions of the human body, such as the hands and head, exhibit significant movements. In light of this, instead of rendering the entire frame from scratch, we focus on updating specific patches[3] showing noticeable changes, reducing rendering time. To identify patches that need updates, the receiver's edge server calculates the movement distance of each vertex in the reconstructed SMPL-X mesh. When the distance exceeds a threshold, we interpret it as a sign of *noticeable* change. We set the thresholds for different body parts based on prior research [77]. While a similar patch-aware rendering technique has been previously proposed by FarfetchFusion [48], it focuses on only the head. Accurately identifying patches that do not require updates during full-body motion without compromising rendering quality is more challenging. Additionally, different from FarfetchFusion [48] that designates pre-defined invariant regions without updates across all frames, our solution adopts a more fine-grained approach.

While updating only patches with noticeable changes can reduce rendering overhead, the challenge lies in patch design. We find that both a large number of small patches and a small number of large patches extend inference time. When the number of patches is too small, it requires processing large areas, as human motion may cause changes in all patches. Conversely, having too many patches leads to high GPU memory consumption [57] and thus may lead to congestion during inference, resulting in patches not being updated simultaneously. Moreover, applying non-overlapping patches can introduce artifacts (*e.g.,* discontinuities) at patch boundaries. We further observe that vertical splitting of patches adversely affects visual quality, particularly in facial regions, again by causing discontinuities in rendering. To balance these factors, our design features eight equally sized and vertically stacked patches with a 10% overlap across adjacent patches. This approach guarantees spatial consistency and mitigates artifacts, providing an effective compromise for efficient neural rendering in MagicStream. We extensively evaluate the effects of various patch designs in §6.2.

5 IMPLEMENTATION

Hardware. We employ three ZED 2i [11] RGB-D cameras for capturing telepresence users. Similar to MetaStream [39], each camera is connected to an NVIDIA Jetson Xavier NX [9] embedded system, which processes RGB-D images. Each edge server is equipped with an NVIDIA RTX 4090 GPU and an AMD Ryzen 9 7900X CPU. We use the Microsoft HoloLens 2 mixed reality (MR) headset [2] as the client device.

Software. We employ Unreal Engine [10] to render and display content on HoloLens 2, MediaPipe [37] to detect keypoints, and LZMA [6] to compress SMPL-X parameters and lighting characteristics. We utilize PyTorch [72] to build all deep-learning models, including SMPL-X parameter estimation, lighting estimation, and neural rendering.

Neural Rendering. We employ U-Net [76] as the foundational neural rendering model. For training, we use the coarse-grained rendering results from rasterization as the input (§4.3), while camera-captured images serve as the ground truth for supervised learning. The training process minimizes the perceptual VGG loss [43] with a learning rate of 1e-4. We set the rendering resolution at 1280×720, which has been shown to provide a satisfactory QoE [60].

In total, MagicStream consists of over 6,800 lines of code (LoC): 1,200+ LoC in C++ on the client side and 5,600+ LoC in Python for the remaining components.

6 PERFORMANCE EVALUATION

6.1 Experimental Setup

Datasets. Due to the absence of publicly available telepresence datasets, we conduct an IRB-approved user study and collect a comprehensive dataset with 20 participants. This cohort consists of 5 females and 15 males with an average age of 22.1±1.4 years. Participants engage in two 10-minute sessions: teleconferencing and dancing. While teleconferencing is the primary use case of telepresence [21, 48], dancing is intriguing for immersive telepresence [46, 98], as it effectively embodies activities that entail extensive body movements, similar to those in remote collaboration [111]. The dataset includes varied poses with over 90% of frames differing from the previous one, demonstrating the diverse motion of participants. The dataset also presents varied lighting conditions at different times of the day.

Baselines. We compare MagicStream with MetaStream [39] in terms of bandwidth consumption, end-to-end latency, visual quality, frame rate, and user experience. We further compare the SMPL-X parameter estimation model of MagicStream with ExPose [26] and PIXIE [33], and the neural rendering model of MagicStream with X-Avatar [79], PoseVocab [55], and HumanNeRF [102].

[3] A patch refers to a small, rectangular/square region in an image [112].

We do not compare MagicStream with other immersive telepresence systems such as FarfetchFusion [48], Project Starline [47], and Holoportation [71] for the following reasons. (1) FarfetchFusion and Project Starline are not designed for full-body telepresence; instead, they focus on a small part of the human body, such as the face. In contrast, MagicStream supports full-body streaming, presenting more challenges than face-only systems. (2) Holoportation and Project Starline are industrial prototypes requiring special setups that we cannot replicate. For instance, Project Starline necessitates custom-built hardware for display, which is not compatible with mobile headsets. Finally, we do not compare MagicStream with existing model-based streaming systems such as YuZu [115] that benefit from 3D super-resolution to upsample delivered low-density point clouds to high-density ones. The reason is that their bandwidth savings are limited and depend on the super-resolution ratio, which is typically small (*e.g.,* 4 or 8).

Network Setup. Mobile headsets of the sender and receiver are connected to different networks with separate Linksys WiFi routers. The edge servers are connected to their corresponding router via Ethernet. The two WiFi routers communicate over Ethernet, with an average throughput of ~150 Mbps and a one-way latency of ~3 ms. To realistically represent Internet delays, we increase the round-trip latency between WiFi routers to 40 ms (*i.e.,* a typical latency within the U.S. [30]), using Linux `tc` [7]. We collect four traces of network bandwidth from different locations on a large commercial cellular network in the U.S. and replay them using `tc`. Their average bandwidths are 11.2±1.2 Mbps, 22.5±2.8 Mbps, 31.8±4.1 Mbps, and 50.4±6.1 Mbps. In the following, these conditions are referred to as scenarios with available bandwidths of 10, 20, 30, and 50 Mbps.

6.2 Component-wise Evaluation

We first evaluate the performance of key MagicStream components: lighting & base color estimation, SMPL-X parameter estimation, and neural rendering.

Estimation of Lighting Characteristics & Base Color. We use the SMPL-X mesh as the underlying geometry to evaluate the effectiveness of our fine-tuning of the Xihe model [121] to estimate lighting characteristics (§4.1). We compare the SSIM between the renderings with and without the fine-tuning. We consider texture mapping [36] that obtains colors from RGB images as the baseline. Fine-tuning yields an SSIM of 0.91±0.03, indicating good quality [29]. Without fine-tuning, the SSIM of the rendered content drops to 0.82±0.05. This result demonstrates the effectiveness of our fine-tuning strategy. As we will show later, the visual quality can be further improved by neural rendering. We also evaluate the overhead of obtaining the base color, which takes only <1 minute.

Estimation of SMPL-X Parameters. We compare the model of SMPL-X parameter estimation in MagicStream (§4.2) with ExPose [26] and PIXIE [33], two existing regression-based models. To assess the effectiveness of our proposed design of keypoint-anchor vertices, we implement another model that minimizes only the L1 loss between the output parameters and the ground truth during training. We calculate the vertex-to-vertex (V2V) distance for the face, hands, and full body of the mesh generated by each method with the ground truth (§4.2), with lower values indicating better performance. Additionally, we evaluate the frame rate achieved by each model. The results are represented in Figure 5, which shows the 95th, 75th, 25th, and 5th percentiles, median, and mean (blue dots). We observe that MagicStream achieves the same or even superior accuracy over others. In terms of frame rate, MagicStream can perform in real time (>30 FPS) with ~5× accelerations compared to ExPose and PIXIE.

# of Patches	Overlap	FPS	SSIM
1	0%	6.37/1.2	0.95/0.03
4	0%	18.2/2.3	0.91/0.03
6	0%	31.5/2.6	0.86/0.08
6	10%	32.8/4.7	0.90/0.03
8	0%	40.9/3.3	0.81/0.16
8	10%	37.2/4.3	0.92/0.02
8	30%	28.4/6.8	0.92/0.03
10	10%	24.4/6.3	0.90/0.04

Table 4: FPS and SSIM for neural rendering with varying numbers of patches and overlap percentages.

Neural Rendering. We next evaluate the neural rendering model of MagicStream, examining its patch design, comparing it with other models [55, 79, 102], and assessing its generalizability.

Impact of Patch Design. We delve into the impact of different patch configurations on visual quality and rendering efficiency. We find that the vertical split of patches severely degrades visual quality. For example, dividing the content into two vertical patches results in an SSIM of only 0.80±0.04. This is because users typically move horizontally, which means that vertical patch split can cause discontinuities in the representation of the human form. Thus, we focus on horizontal patch split.

Table 4 presents the impact of varying the number of patches and the percentage of patch overlap on visual quality and frame rate. Introducing a moderate number of patches gradually improves the frame rate. Nonetheless, an excessive number of patches (*e.g.,* 10) can paradoxically lead to a decline in frame rate. This is because a small number of patches necessitates processing extensive areas, whereas an excessive number of patches increases the amount of patches to update (§4.3). Furthermore, incorporating a certain degree of overlap between patches contributes to maintaining a satisfactory visual quality by facilitating smoother transitions at patch boundaries. Our results suggest that a 10% overlap strikes a balance between visual fidelity and computational overhead, ensuring high-quality rendering without significantly impacting the frame rate. Excessively large overlap areas can lead to diminishing returns, potentially processing redundant content and increasing computational load without proportional gains in visual continuity (*e.g.,* a 44% drop in frame rate when increasing the overlap from 10 to 30% for the eight-patch setup). Based on our extensive evaluation, we select a configuration of eight horizontally stacked patches with a 10% overlap.

Comparison with Other Models. We next compare the neural rendering model of MagicStream with three existing approaches, X-Avatar [79], PoseVocab [55], and HumanNeRF [102]. As shown in Figure 6, MagicStream achieves real-time performance with >40 FPS on average, whereas other models perform at <1 FPS.

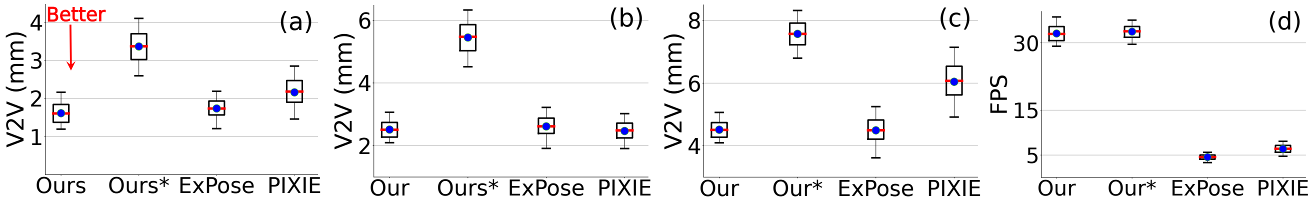

Figure 5: Comparison of vertex-to-vertex (V2V) distance for (a)–(c): face, hands, and full body, as well as (d) FPS of SMPL-X parameter estimation in MagicStream, MagicStream without keypoint-anchored vertices (Ours*), ExPose [26] and PIXIE [33].

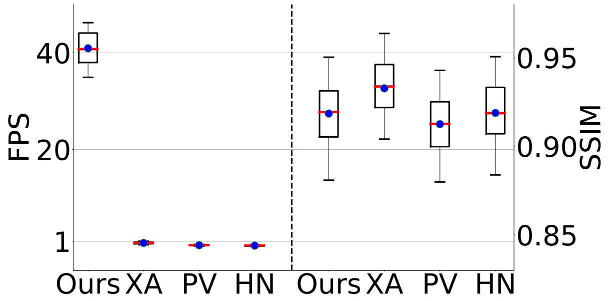

Figure 6: Comparison of neural rendering in MagicStream **(Ours) with X-Avatar (XA) [79], PoseVocab (PV) [55], and Human-NeRF (HN) [102].**

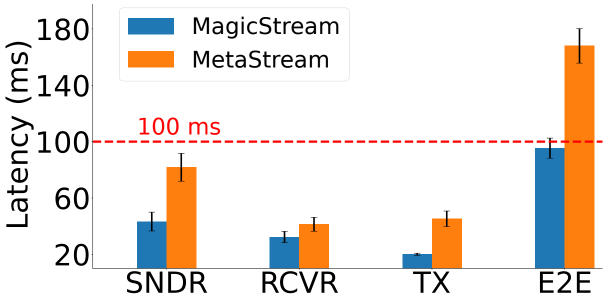

Figure 7: Breakdown of end-to-end (E2E) latency of MagicStream **and MetaStream [39], including sender (SNDR), receiver (RCVR), and transmission (TX).**

Additionally, MagicStream maintains comparable or even better SSIM than other models (*e.g.,* on average 0.92 for MagicStream *vs.* 0.91 for PoseVocab [55]). The heavy computational overhead of existing models mainly arises from their reliance on complex rendering techniques to achieve high visual quality, such as neural radiance fields (NeRF) [66] employed by PoseVocab [55] and HumanNeRF [102], or the use of inefficient root-finding loops for 3D model deformation, such as SNARF [20] utilized by X-Avatar [79].

In contrast, MagicStream employs several strategies to accelerate neural rendering while preserving visual quality. First, it initially generates coarse-grained 2D images based on color semantics, reducing the complexity of the input to the neural rendering model from 3D data to 2D images (§4.3). Second, MagicStream meticulously designs a patch-based acceleration approach to improve computational efficiency without compromising rendering quality (§4.3). Furthermore, the underlying neural rendering model used by MagicStream is based on U-Net [76], which is more lightweight than NeRF [66]. This reduction in complexity is made possible by MagicStream’s effective model for SMPL-X parameter estimation (§4.2), which accurately reconstructs the user’s body, and its use of data augmentation techniques when training the neural rendering model (§4.3).

Generalizability. We then verify the generalizability of the neural rendering model of MagicStream, which should be well-trained before each telepresence session. Ideally, it should be generalizable to a novel appearance, instead of training from scratch every time, which causes significant computation overhead. This generalizability is plausible because both motion (*i.e.,* keypoints) and color semantics (*i.e.,* lighting characteristics) are inherently user-independent. To verify this, we design an experiment in which the same user participates in six separate trials, each time wearing a different colored outfit. We train the neural rendering model from scratch for the first trial and then fine-tune the trained model for the subsequent ones. We stop the training/fine-tuning when the SSIM of the content rendered by the model reaches 0.9, the indicator of good visual quality [101]. Each experiment is repeated five times with randomly selected trial orders.

Our results show that although training from scratch takes an average of 84 minutes (SD: 13), fine-tuning for the last trial takes only 2.7 minutes on average (SD: 0.4), which could be further reduced with more data for fine-tuning. This significant reduction in training time highlights the model’s potential generalizability. These findings align with recent efforts [79, 89, 90], which demonstrate that neural rendering models, when trained on large and diverse datasets, can adapt to novel appearance during inference without requiring training from scratch or extensive fine-tuning.

6.3 End-to-end Evaluation

In this subsection, we present results in one-way telepresence. Two-way communication mainly affects the resource utilization of edge servers, which we will evaluate in §6.4.

Bandwidth Consumption. We compare the bandwidth consumed by MagicStream, which delivers semantic information, and MetaStream, which requires the transmission of point clouds. We observe that by transmitting semantics instead of 3D content, MagicStream consumes a bandwidth of only 0.2 Mbps, on average. In stark contrast, MetaStream results in 72.3 Mbps throughput, on average, 360× higher than that of MagicStream. Note that MetaStream can stream only ∼200K points per frame in real time, resulting in poor visual quality. As we will show later, to achieve a similar visual quality as MagicStream, MetaStream demands 1195× higher bandwidth.

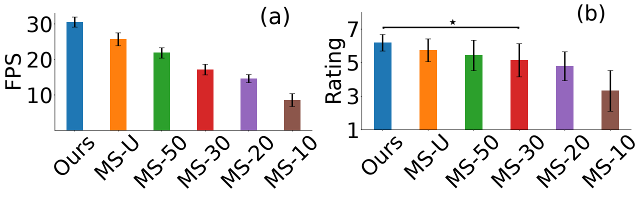

Figure 8: Comparison of (a) FPS and (b) user ratings between our MagicStream and MetaStream (MS) with varying bandwidth settings (in Mbps). "U" means unthrottled networks. The FPS of MagicStream is not affected by available bandwidth due to the small size of its transmitted data. ⋆: $p \leq 0.05$.

End-to-end Latency. We next compare the end-to-end latency of MagicStream and MetaStream, by breaking it down into the computation latency of the sender and the receiver and the transmission delay, as shown in Figure 7. MagicStream achieves an average one-way end-to-end latency of 95 ms, which satisfies the latency requirement of interactive applications (*i.e.,* <100 ms) [16, 18, 69]. Specifically, the sender-side processes involve keypoint detection (∼10 ms), lighting estimation (∼10 ms), and SMPL-X parameter estimation (∼30 ms). Since keypoint detection and lighting estimation are conducted in parallel, the combined latency totals ∼40 ms. On the receiver side, coarse-grained rendering is executed in ∼3 ms, and neural rendering takes ∼24 ms.

In comparison, MetaStrem achieves an average latency of ∼170ms, almost 2× higher than MagicStream. The most substantial reduction in the latency of MagicStream can be primarily attributed to the following factors. First, it relies on keypoint detection to identify the human body from camera-captured images, which takes ∼10 ms, as opposed to MetaStream's approach of executing a segmentation model for the same purpose, taking ∼30 ms. Second, MagicStream efficiently executes two lightweight models in parallel to generate semantics on the sender. In contrast, MetaStream needs to create and filter dense point clouds. Third, MagicStream reduces the transmission delay to ∼20 ms (from ∼45 ms by MetaStream), thanks to the considerably smaller bandwidth requirements of extracted semantics.

Frame Rate. Figure 8(a) shows the FPS of MagicStream and MetaStream under varying network conditions. MagicStream consistently reaches 30+ FPS, fulfilling the requirement for real-time telepresence. The high frame rate is achieved because each computational module, including keypoint extraction and lighting estimation (∼10 ms, executed in parallel for a single frame), SMPL-X parameter estimation (∼30 ms), and neural rendering (∼27 ms), operates <33 ms. Additionally, MagicStream processes multiple frames in parallel, ensuring that the overall system can achieve >30 FPS. In contrast, MetaStream is unable to reach 30 FPS, even in unthrottled network environments. This limitation is primarily due to the computation overhead of point cloud synthesis/processing. As the available bandwidth decreases, MetaStream experiences a significant decline in frame rate, dropping to <10 FPS when the available bandwidth is around 10 Mbps, which is largely attributed to increased transmission latency.

User Experience. To compare the experience perceived by real users for MagicStream and MetaStream, we conduct another user study involving 17 participants: 4 females and 13 males, with an average age of 21.1±1.7. They are asked to wear a HoloLens 2 headset and freely explore the content randomly selected from our data collection (§6.1), similar to the setup adopted in prior work, FarfetchFusion [48]. The content is streamed with MagicStream and MetaStream under different network conditions for two minutes each. To eliminate bias, we randomize the order of streaming systems used for each viewing session, and participants do not know which system is being used to generate the content. Upon completing the tasks, we ask participants to rate their experiences with the 7-point Likert scale (1: very bad; 7: very good) [86].

Figure 8(b) shows the ratings for MagicStream and MetaStream under different network conditions. We utilize the Shapiro-Wilk test [1] and find that all these ratings are not normally distributed. Thus, we apply the Wilcoxon signed-rank test [1] to conduct a significance test between MagicStream and MetaStream under each network condition. When the available bandwidth drops below 30 Mbps, the rating of MagicStream is significantly higher than that of MetaStream (*e.g.,* when the available bandwidth is ∼10 Mbps, $p < 0.01$[4]; Rank-biserial correlation $r = 0.64$[5], with an improvement of 86.7%). This is because when the bandwidth is limited, both the FPS and visual quality of MetaStream significantly drop. In contrast, MagicStream is not affected by the limited bandwidth, resulting in consistently higher user satisfaction.

Visual Quality. To evaluate the visual quality of MagicStream, we first quantitatively compare its SSIM [101] with MetaStream, which is calculated between screenshots collected on HoloLens 2 and the camera-captured ground-truth images. MagicStream achieves, on average, an SSIM of 0.92, indicating good visual quality [29]. In contrast, the average SSIM of MetaStream is ∼0.8. We then qualitatively compare the rendering results of MagicStream and MetaStream. Specifically, we focus on rendering results for novel views (*i.e.,* not from camera perspectives), which is essential in telepresence as users can freely observe their peers from different angles. This is challenging for MagicStream as the ground-truth images for these views may not exist in the training data for neural rendering. Figure 9 reveals MagicStream's ability to render the human body with high fidelity, unlike MetaStream, which struggles, especially in rendering faces and hands clearly. We also observe some black spots/areas in the rendered results of MetaStream, which is also visible in Figure 2(c). This is because RGB-D cameras may fail to calculate depth under certain conditions such as strong ambient light [48]. In contrast, MagicStream is less affected by such artifacts, thanks to its accurate lighting estimation and optimized neural rendering models.

MagicStream *vs.* MetaStream with More Points per Frame. We next compare MagicStream and MetaStream that generates more points per frame in order to achieve better visual quality. As shown in Figure 10, while this approach can improve visual quality for MetaStream, it will further increase bandwidth consumption and transmission latency, as well as decrease frame rate. The significant drop in frame rate is primarily due to the substantial computational

[4] p represents the result of the statistically significant test. $p < 0.05$ is considered statistically significant [91].

[5] Rank-biserial correlation r is the effect size of the Wilcoxon signed-rank. $r > 0.1$ indicates the claim is valid [65].

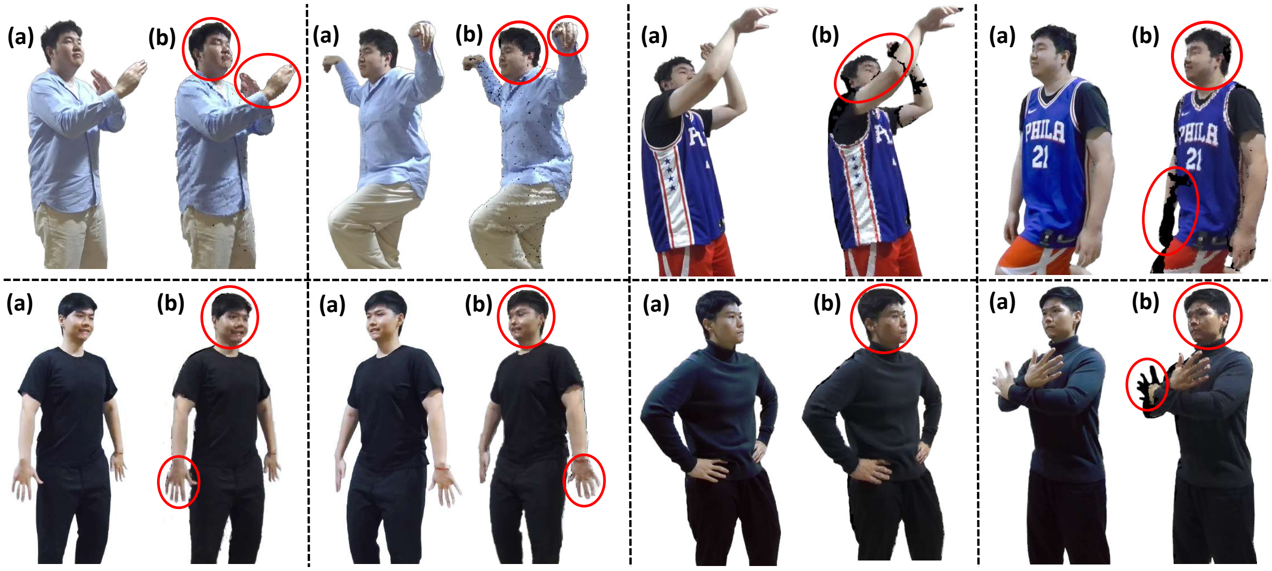

Figure 9: Qualitative comparison of (a) MagicStream and (b) MetaStream for novel views.

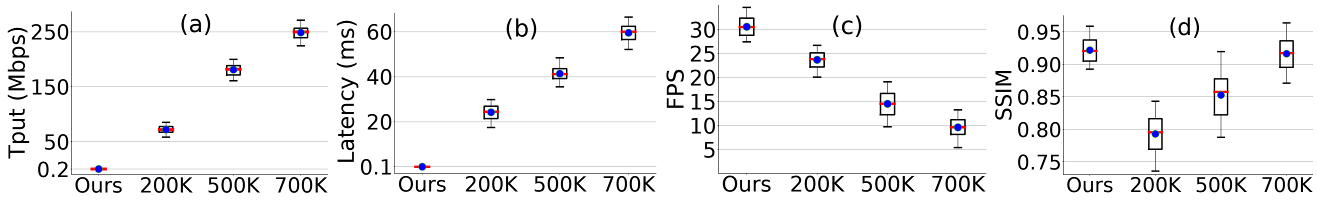

Figure 10: Comparison of (a) throughput at 30 FPS, (b) transmission latency (excluding the propagation delay between edge servers), (c) FPS, and (d) SSIM for MagicStream (Ours) and MetaStream with different number of points per frame.

overhead for processing dense point clouds. When streaming 700K points per frame, MetaStream can achieve similar visual quality as MagicStream, but demands 1195× higher bandwidth consumption and increases the transmission latency from 0.1 to ∼60 ms.

6.4 Resource Utilization

Finally, we evaluate the resource utilization of MagicStream on the edge server (*i.e.,* commercially available machines introduced in §5) and mobile headset (*i.e.,* Microsoft HoloLens 2). MagicStream utilizes 7.6 GB GPU memory, <5% CPU resources, and <20% host memory on the edge server for two-way communication between two users. To further investigate MagicStream's scalability on resource utilization, we simultaneously execute its models for multiple users, each under a dedicated capturing environment, while sharing the same edge server for computation. Our experiments reveal that the edge server can support three concurrent users on each side of telepresence without affecting MagicStream's performance. To measure on-device resource utilization, we fully charge a HoloLens 2 headset and then deploy it in a telepresence session for 1 hour under an unthrottled network. After the 1-hour experiment, the battery level decreases to 75%, and the average CPU/GPU utilization is 30%/38%. Overall, we believe the resource utilization and energy consumption of MagicStream are acceptable.

7 DISCUSSION

Use Cases. In this paper, we benefit from the task-driven nature of immersive telepresence that facilitates semantic communication. This approach is particularly effective in scenarios where conveying key elements of human interactions, rather than duplicating complete 3D content, is vital, especially when the network bandwidth is limited. For example, in teleconferencing, the pivotal elements are often the speaker's distinct gestures and facial expressions. Our recent measurement study reveals that Apple FaceTime also utilizes semantic communication to reduce bandwidth consumption for its spatial personas on the Vision Pro headset [22]. However, semantic communication is not a one-size-fits-all solution. In high-precision applications such as immersive scientific data visualization [32], traditional bit-by-bit communication might be preferable. Therefore, although MagicStream offers substantial advantages in reducing Internet bandwidth usage, its suitability largely depends on the specific goals and requirements of telepresence use cases.

Scalability. Since MagicStream relies on deep-learning models for semantics extraction and content reconstruction, the resource consumption of the edge server is largely attributed to GPU memory usage (§6.4). To improve MagicStream's scalability, one simple solution is to deploy more powerful GPUs or additional edge servers. We can also adapt the resolution of neural rendering based on the viewer's distance to the displayed content, further optimizing resource utilization. For example, when the viewing distance is long, high-resolution rendering may not be necessary [40]. For multi-site setups, it mainly increases the computation overhead of users' edge servers that will perform neural rendering for multiple parties. To mitigate the computational demands on edge servers, one possible solution is to deploy multipoint control units [34] that fuse/process the data from users before distributing it to edge servers.

Impact of Lighting Conditions. In telepresence, lighting conditions significantly influence the quality of captured content. Thus, existing systems such as Starline [47] typically require controlled lighting environments. For extreme scenarios, such as in dark environments, RGB-D cameras may not function well [107], impacting both the quality of generated point clouds and the accuracy of keypoint extraction. In addition to employing controlled lighting, another potential solution is to utilize LiDAR sensors to improve the quality of captured data in challenging environments [53]. These solutions are orthogonal to semantic communication, as they relate to mainly data capturing.

Generalizability to Other Objects. MagicStream extracts and transmits motion and color semantics from the captured data of the human body to enable semantic communication. Such a design can be adapted to other objects. For motion semantics, as MagicStream is designed for the human body with both rigid and non-rigid movements, it can be generalized to other entities with similar movements (*e.g.,* animals or small children). For objects with mainly rigid motions, for example, vehicles, we can initially transmit their entire 3D models and subsequently only their position/orientation as motion semantics. For color information, since it involves only the interaction between the base color and lighting characteristics (§4.1), this process is not specific to the human body and can potentially be applied to other objects [120, 121].

Lower-layer Network Optimization. While MagicStream predominantly works on the application layer to reduce bandwidth requirements over the Internet, ensuring reliable streaming necessitates consideration of optimizations across other layers in the protocol stack. For instance, packet loss, a common issue in Internet content delivery, demands the design of a system resilient to such losses. Implementing loss-resilient mechanisms [23] is crucial for the seamless transmission of semantic information. Regarding network protocols in the transport layer, the adoption of emerging protocols such as QUIC [93] presents a more advantageous alternative for live video streaming compared to traditional protocols such as TCP. However, effectively integrating QUIC into semantic communication systems requires careful consideration of how its features can be leveraged to prioritize and transmit semantically relevant data efficiently, ensuring both speed and data integrity in high-fidelity telepresence.

8 RELATED WORK

Immersive Video Streaming. There is a plethora of work on improving the QoE for immersive video streaming [39, 40, 47–49, 56, 59, 60, 71, 103, 115, 117]. Among these, video-on-demand services have been extensively studied. For instance, M5 [117] utilizes 6DoF motion prediction to adapt mmWave beams for multiuser video streaming, and Theia [103] leverages foveated streaming to reduce data usage. Additionally, recent efforts focus on live video streaming, offering a wide range of applications for telepresence [39, 47, 48, 71]. However, prior studies rely on bit-by-bit transmission, leading to notable bandwidth demands. In contrast, MagicStream introduces an innovative approach, utilizing semantic communication to drastically reduce bandwidth usage.

3D Reconstruction via Wireless Sensing. Existing efforts demonstrate that wireless sensing, for example, via WiFi [42, 51, 52, 100] and mmWave [54, 106, 108, 109], can be utilized for 3D reconstruction. However, as wireless sensing cannot capture color information, these studies either focus on keypoint detection [42, 54, 109], similar to the extraction of motion semantics in MagicStream (§4.1), or non-textured mesh reconstruction [100, 106, 108], similar to the reconstruction from motion semantics in MagicStream (§4.2).

Semantic Communication has garnered substantial interest for its potential to reduce transmission overhead [21, 63, 80, 104, 105, 116, 123]. Initial work primarily concentrates on its direct interpretation, targeting the delivery of text data [104]. Subsequent advancements in the field have extended this concept to include a wider range of modalities, such as images [105], broadening the applications of semantic communication into various new domains, for example, the emerging Metaverse [116]. A recent work [123] loosely uses the term "semantics" as it is similar to traditional point cloud compression techniques, achieving only limited bandwidth reduction [21]. In contrast, MagicStream significantly reduces bandwidth consumption while maintaining high-quality rendering.

Keypoint-driven 2D Videos. Recent studies have begun to explore the use of keypoints in 2D video streaming due to their informativeness [70, 82, 96, 99, 113], such as generating neural models for head avatars [99, 113]. However, these studies are confined to 2D videos. MagicStream diverges from this path by leveraging keypoints for streaming 3D content, which is more complex but enables more dynamic and engaging use cases than conventional 2D videos.

9 CONCLUSION

In this paper, we presented the design, implementation, and comprehensive evaluation of MagicStream, a novel semantic-driven immersive telepresence system. MagicStream excels in precisely extracting and efficiently delivering semantics of body motion and skin/cloth color, as well as in effectively reconstructing and rendering immersive content based on received semantic details. This approach substantially reduces Internet bandwidth usage while maintaining low end-to-end latency and a satisfactory visual quality. Our thorough performance evaluations demonstrate that MagicStream significantly outperforms state-of-the-art solutions. We hope our study can pave the way for future advancements in semantic-based immersive telepresence, promising to expand the reach of holographic communication.

ACKNOWLEDGMENT

We thank the anonymous reviewers and our shepherd for their insightful feedback, which has greatly strengthened this work. Additionally, we sincerely thank the participants of our user studies and experiments for their invaluable time and contributions, enabling us to gather meaningful data and validate our findings. This work was partially supported by the National Science Foundation under Grants CNS-2212296 and CNS-2235049.

REFERENCES

[1] 2014. Statistical Methods for HCI Research. https://yatani.jp/teaching/doku.php?id=hcistats:start. [accessed on 10/07/2024].

[2] 2019. Microsoft HoloLens 2. https://www.microsoft.com/en-us/hololens. [accessed on 10/07/2024].

[3] 2021. H.264 : Advanced video coding for generic audiovisual services. https://www.itu.int/rec/T-REC-H.264. [accessed on 10/07/2024].

[4] 2024. Axis–angle Representation. https://en.wikipedia.org/wiki/Axis-angle_representation.

[5] 2024. Draco 3D Data Compression. https://google.github.io/draco/. [accessed on 10/07/2024].

[6] 2024. Lempel–Ziv–Markov Chain Algorithm. https://en.wikipedia.org/wiki/Lempel-Ziv-Markov_chain_algorithm. [accessed on 10/07/2024].

[7] 2024. Linux TC Man Page. https://linux.die.net/man/8/tc.

[8] 2024. LZ4 (compression algorithm). https://en.wikipedia.org/wiki/LZ4_(compression_algorithm). [accessed on 10/07/2024].

[9] 2024. Nvidia Jetson Technical Specifications. https://developer.nvidia.com/embedded/jetson-modules.

[10] 2024. Unreal Engine. https://www.unrealengine.com. [accessed on 10/07/2024].

[11] 2024. ZED 2i. https://www.stereolabs.com/zed-2i/l. [accessed on 10/07/2024].

[12] 2024. zlib. https://en.wikipedia.org/wiki/Zlib. [accessed on 10/07/2024].

[13] 2024. Zstandard. https://facebook.github.io/zstd/. [accessed on 10/07/2024].

[14] Lukas Ahrenberg, Philip Benzie, Marcus Magnor, and John Watson. 2008. Computer Generated Holograms from Three Dimensional Meshes using an Analytic Light Transport Modell. *Applied Optics* 47, 10 (2008), 1567–1574. https://doi.org/10.1364/AO.47.001567

[15] Paramvir Bahl and Venkata N Padmanabhan. 2000. RADAR: an In-Building RF-based User Location and Tracking System . In *Proceedings of IEEE INFOCOM*.

[16] Mario Baldi and Yoram Ofek. 2000. End-to-end Delay Analysis of Videoconferencing Over Packet-switched Networks. *IEEE/ACM Transactions On Networking* 8, 4 (2000), 479–492.

[17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *Proceedings of IEEE/CVF CVPR*.

[18] Kaifei Chen, Tong Li, Hyung-Sin Kim, David E Culler, and Randy H Katz. 2018. MARVEL: Enabling Mobile Augmented Reality with Low Energy and Low Latency. In *Proceedings of ACM SenSys*.

[19] Rick H-Y Chen and Timothy D Wilkinson. 2009. Computer Generated Hologram from Point Cloud using Graphics Processor. *Applied Optics* 48, 6 (2009), 6841–6850. https://doi.org/10.1364/AO.48.006841

[20] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. 2021. SNARF: Differentiable Forward Skinning for Animating Non-rigid Neural Implicit Shapes. In *Proceedings of IEEE/CVF CVPR*.

[21] Ruizhi Cheng, Kaiyan Liu, Nan Wu, and Bo Han. 2023. Enriching Telepresence with Semantic-driven Holographic Communication. In *Proceddings of ACM Workshop on Hot Topics in Networks (HotNets)*.

[22] Ruizhi Cheng, Nan Wu, Matteo Varvello, Eugene Chai, Songqing Chen, and Bo Han. 2024. A First Look at Immersive Telepresence on Apple Vision Pro. In *Proceedings of ACM Internet Measurement Conference (IMC)*.

[23] Yihua Cheng, Ziyi Zhang, Hanchen Li, Anton Arapin, Yue Zhang, Qizheng Zhang, Yuhan Liu, Kuntai Du, Xu Zhang, Francis Y Yan, et al. 2024. GRACE:Loss-Resilient Real-Time Video through Neural Codecs. In *Proceedings of NSDI 24*.

[24] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. 2020. Pose2Mesh: Graph Convolutional Network for 3D Human Pose and Mesh Recovery from a 2D Human Pose. In *Proceedings of ECCV*.

[25] Paul J Choi, Rod J Oskouian, and R. Shane Tubbs. 2018. Telesurgery: Past, Present, and Future. *Cureus* 10, 5 (2018), e2716.

[26] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. 2020. Monocular Expressive Body Regression Through Body-Driven Attention. In *Proceedings of ECCV*.

[27] Jen-Li Chung, Lee-Yeng Ong, and Meng-Chew Leow. 2022. Comparative Analysis of Skeleton-based Human Pose Estimation. *Future Internet* 14, 12 (2022), 380.

[28] Federal Communications Commission. 2024. FCC Increases Broadband Speed Benchmark. https://docs.fcc.gov/public/attachments/DOC-401205A1.pdf.

[29] Eduardo Cuervo, Alec Wolman, Landon P. Cox, Kiron Lebeck, Ali Razeen, Stefan Saroiu, and Madanlal Musuvathi. 2015. Kahawai: High-Quality Mobile Gaming Using GPU Offload. In *Proceedings of ACM MobiSys*. https://doi.org/10.1145/2742647.2742657

[30] The Khang Dang, Nitinder Mohan, Lorenzo Corneo, Aleksandr Zavodovski, Jörg Ott, and Jussi Kangasharju. 2021. Cloudy with a Chance of Short RTTs: Analyzing Cloud Connectivity in the Internet. In *Proceedings of ACM Internet Measurement Conference (IMC)*.

[31] Chamitha De Alwis, Anshuman Kalla, Quoc-Viet Pham, Pardeep Kumar, Kapal Dev, Won-Joo Hwang, and Madhusanka Liyanage. 2021. Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research. *IEEE Open Journal of the Communications Society* 2 (2021), 836–886.

[32] Ciro Donalek, S George Djorgovski, Alex Cioc, Anwell Wang, Jerry Zhang, Elizabeth Lawler, Stacy Yeh, Ashish Mahabal, Matthew Graham, Andrew Drake, et al. 2014. Immersive and Collaborative Data Visualization Using Virtual Reality Platforms. In *Proceddings of IEEE International Conference on Big Data (Big Data)*.

[33] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. 2021. Collaborative Regression of Expressive Bodies Using Moderation. In *Proceedings of International Conference on 3D Vision (3DV)*.

[34] Sergi Fernandez, Mario Montagud, David Rincón, Juame Moragues, and Gianluca Cernigliaro. 2023. Addressing Scalability for Real-time Multiuser Holo-portation: Introducing and Assessing a Multipoint Control Unit (MCU) for Volumetric Video. In *Proceedings of ACM International Conference on Multimedia (MM)*.

[35] James D Foley. 1996. *Computer Graphics: Principles and Practice*. Vol. 12110. Addison-Wesley Professional.

[36] Yanping Fu, Qingan Yan, Long Yang, Jie Liao, and Chunxia Xiao. 2018. Texture Mapping for 3D Reconstruction with Rgb-d Sensor. In *Proceedings of IEEE/CVF CVPR*.

[37] Google. 2024. Face landmark detection guide. https://developers.google.com/mediapipe/solutions/vision/face_landmarker. [accessed on 10/07/2024].

[38] Robin Green. 2003. Spherical harmonic lighting: The gritty Details. In *Archives of the Game Developers Conference*.

[39] Yongjie Guan, Xueyu Hou, Nan Wu, Bo Han, and Tao Han. 2023. MetaStream: Live Volumetric Content Capture, Creation, Delivery, and Rendering in Real Time. In *Proceedings of ACM MobiCom*.

[40] Bo Han, Yu Liu, and Feng Qian. 2020. ViVo: Visibility-Aware Mobile Volumetric Video Streaming. In *Proceedings of ACM MobiCom*.

[41] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 7 (2013), 1325–1339.

[42] Sijie Ji, Xuanye Zhang, Yuanqing Zheng, and Mo Li. 2023. Construct 3D Hand Skeleton with Commercial WiFi. In *Proceedings of ACM SenSys*.

[43] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-time Style Transfer and Super-resolution. In *Proceedings of ECCV*.

[44] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3D Mesh Renderer. In *Proceedings of IEEE/CVF CVPR*.

[45] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. 2007. Skinning with Dual Quaternions. In *Proceedings of Symposium on Interactive 3D Graphics and Games*.

[46] Gregorij Kurillo, Ruzena Bajcsy, Klara Nahrsted, and Oliver Kreylos. 2008. Immersive 3D Environment for Remote Collaboration and Training of Physical Activities. In *Proceedings of IEEE Conference Virtual Reality and 3D User Interfaces (VR)*.

[47] Jason Lawrence, Danb Goldman, Supreeth Achar, Gregory Major Blascovich, Joseph G Desloge, Tommy Fortes, Eric M Gomez, Sascha Häberling, Hugues Hoppe, Andy Huibers, et al. 2021. Project Starline: a High-fidelity Telepresence System. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–16.

[48] Kyungjin Lee, Juheon Yi, and Youngki Lee. 2023. FarfetchFusion: Towards Fully Mobile Live 3D Telepresence Platform. In *Proceedings of ACM MobiCom*.

[49] Kyungjin Lee, Juheon Yi, Youngki Lee, Sunghyun Choi, and Young Min Kim. 2020. GROOT: a Real-time Streaming System of High-fidelity Volumetric Videos. In *Proceedings of ACM MobiCom*. https://doi.org/10.1145/3372224.3419214

[50] Chloe LeGendre, Wan-Chun Ma, Rohit Pandey, Sean Fanello, Christoph Rhemann, Jason Dourgarian, Jay Busch, and Paul Debevec. 2020. Learning Illumination from Diverse Portraits. In *SIGGRAPH Asia Technical Communications*.

[51] Chenning Li, Li Liu, Zhichao Cao, and Mi Zhang. 2022. WiVelo: Fine-grained Walking Velocity Estimation for Wi-Fi Passive Tracking. In *Proceedings of IEEE SECON*.

[52] Chenning Li, Manni Liu, and Zhichao Cao. 2020. WiHF: Enable User Identified Gesture Recognition with WiFi. In *Proceedings of IEEE INFOCOM*.

[53] Jialian Li, Jingyi Zhang, Zhiyong Wang, Siqi Shen, Chenglu Wen, Yuexin Ma, Lan Xu, Jingyi Yu, and Cheng Wang. 2022. LiDARCap: Long-range Marker-less 3D Human Motion Capture with LiDAR Point Clouds. In *Proceedings of IEEE/CVF CVPR*.

[54] Wenwei Li, Ruofeng Liu, Shuai Wang, Dongjiang Cao, and Wenchao Jiang. 2023. Egocentric Human Pose Estimation using Head-mounted mmWave Radar. In *Proceedings of ACM SenSys*.

[55] Zhe Li, Zerong Zheng, Yuxiao Liu, Boyao Zhou, and Yebin Liu. 2023. PoseVocab: Learning Joint-structured Pose Embeddings for Human Avatar Modeling. In *Proceedings of ACM SIGGRAPH*.

[56] Zhicheng Liang, Junhua Liu, Mallesham Dasari, and Fangxin Wang. 2024. Fumos: Neural Compression and Progressive Refinement for Continuous Point Cloud Video Streaming. In *Proceedings of IEEE Conference Virtual Reality and 3D User Interfaces (VR)*.

[57] Ji Lin, Wei-Ming Chen, Han Cai, Chuang Gan, and Song Han. 2021. MCUNetV2: Memory-Efficient Patch-based Inference for Tiny Deep Learning. In *Proceedings of Annual Conference on Neural Information Processing Systems (NeurIPS)*.

[58] Kevin Lin, Lijuan Wang, and Zicheng Liu. 2021. End-to-end Human Pose and Mesh Reconstruction with Transformers. In *Proceedings of IEEE/CVF CVPR*.

[59] Junhua Liu, Boxiang Zhu, Fangxin Wang, Yili Jin, Wenyi Zhang, Zihan Xu, and Shuguang Cui. 2023. CaV3: Cache-assisted Viewport Adaptive Volumetric Video Streaming. In *Proceedings of IEEE Conference Virtual Reality and 3D User Interfaces (VR)*.

[60] Yu Liu, Bo Han, Feng Qian, Arvind Narayanan, and Zhi-Li Zhang. 2022. Vues: Practical Mobile Volumetric Video Streaming Through Multiview Transcoding. In *Proceedings of ACM MobiCom*. https://doi.org/10.1145/3495243.3517027

[61] Matthew M Loper and Michael J Black. 2014. OpenDR: An Approximate Differentiable Renderer. In *Proceedings of ECCV*.

[62] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. Mediapipe: A Framework for Building Perception Pipelines. https://arxiv.org/pdf/1906.08172.pdf. [accessed on 10/07/2024].

[63] Xuewen Luo, Hsiao-Hwa Chen, and Qing Guo. 2022. Semantic Communications: Overview, Open Issues, and Future Research Directions. *IEEE Wireless Communications* 29, 1 (2022), 210–219.

[64] Kyle MacMillan, Tarun Mangla, James Saxon, and Nick Feamster. 2021. Measuring the Performance and Network Utilization of Popular Video Conferencing Applications. In *Proceedings of ACM IMC*.

[65] Esther Lopez Martin and Diego Ardura Martinez. 2023. The Effect Size in Scientific Publication. *Educacion XX1* 26, 1 (2023), 09–17.

[66] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[67] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. 2022. Accurate 3D Hand Pose Estimation for Whole-Body 3D Human Mesh Estimation. In *Proceedings of IEEE/CVF CVPR*.

[68] Gyeongsik Moon and Kyoung Mu Lee. 2020. I2L-MeshNet: Image-to-Lixel Prediction Network for Accurate 3D Human Pose and Mesh Estimation from a Single RGB Image. In *Proceedings of ECCV*.

[69] Jakob Nielsen. 1994. *Usability Engineering*. Morgan Kaufmann.

[70] Maxime Oquab, Pierre Stock, Daniel Haziza, Tao Xu, Peizhao Zhang, Onur Celebi, Yana Hasson, Patrick Labatut, Bobo Bose-Kolanu, Thibault Peyronel, et al. 2021. Low bandwidth video-chat compression using deep generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2388–2397.

[71] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. 2016. Holoportation: Virtual 3D Teleportation in Real-time. In *Proceedings of ACM Symposium on User Interface Software and Technology (UIST)*.

[72] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An Imperative Style, High-performance Deep Learning Library. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[73] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proceedings of IEEE/CVF CVPR*.

[74] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 3D Human Pose Estimation in Video with Temporal Convolutions and Semi-supervised Training. In *Proceedings of IEEE/CVF CVPR*.

[75] Puntawat Ponglertnapakorn, Nontawat Tritrong, and Supasorn Suwajanakorn. 2023. DiFaReli: Diffusion Face Relighting. In *Proceedings of IEEE/CVF ICCV*.

[76] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention*.

[77] Matteo Ruggero Ronchi and Pietro Perona. 2017. Benchmarking and Error Diagnosis in Multi-instance Pose Estimation. In *Proceedings of IEEE/CVF ICCV*.

[78] Ruslan Salakhutdinov. 2015. Learning Deep Generative Models. *Annual Review of Statistics and Its Application* 2 (2015), 361–385.

[79] Kaiyue Shen, Chen Guo, Manuel Kaufmann, Juan Jose Zarate, Julien Valentin, Jie Song, and Otmar Hilliges. 2023. X-avatar: Expressive Human Avatars. In *Proceedings of IEEE/CVF CVPR*.

[80] Guangming Shi, Yong Xiao, Yingyu Li, and Xuemei Xie. 2021. From Semantic Communication to Semantic-Aware Networking: Model, Architecture, and Open Problems. *IEEE Communications Magazine* 59, 8 (2021), 44–50.

[81] Connor Shorten and Taghi M Khoshgoftaar. 2019. A Survey on Image Data Augmentation for Deep Learning. *Journal of big data* 6, 1 (2019), 1–48.

[82] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First Order Motion Model for Image Animation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

[83] Vibhaalakshmi Sivaraman, Pantea Karimi, Vedantha Venkatapathy, Mehrdad Khani, Sadjad Fouladi, Mohammad Alizadeh, Frédo Durand, and Vivienne Sze. 2024. Gemino: Practical and Robust Neural Compression for Video Conferencing. In *Proceedings of USENIX NSDI*.

[84] Liangchen Song, Gang Yu, Junsong Yuan, and Zicheng Liu. 2021. Human Pose Estimation and Its Application to Action Recognition: A Survey. *Journal of Visual Communication and Image Representation* 76 (2021), 103055.

[85] Emilio Calvanese Strinati, Sergio Barbarossa, Jose Luis Gonzalez-Jimenez, Dimitri Ktenas, Nicolas Cassiau, Luc Maret, and Cedric Dehos. 2019. 6G: The Next Frontier: From Holographic Messaging to Artificial Intelligence Using Subterahertz and Visible Light Communication. *IEEE Vehicular Technology Magazine* 14, 3 (2019), 42–50. https://doi.org/10.1109/MVT.2019.2921162

[86] Gail M Sullivan and Anthony R Artino Jr. 2013. Analyzing and Interpreting Data From Likert-Type Scales. *Journal of Graduate Medical Education* 5, 4 (2013), 541–542. https://doi.org/10.4300/JGME-5-4-18

[87] Daichi Tajima, Yoshihiro Kanamori, and Yuki Endo. 2021. Relighting Humans in the Wild: Monocular Full-Body Human Relighting with Domain Adaptation. In *Computer Graphics Forum*, Vol. 40. 205–216.

[88] Faisal Tariq, Muhammad RA Khandaker, Kai-Kit Wong, Muhammad A Imran, Mehdi Bennis, and Merouane Debbah. 2020. A Speculative Study on 6G. *IEEE Wireless Communications* 27, 4 (2020), 118–125.

[89] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. 2020. State of the Art on Neural Rendering. In *Computer Graphics Forum*, Vol. 39. 701–727.

[90] Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred Neural Rendering: Image Synthesis Using Neural Textures. *Acm Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

[91] Matthew S Thiese, Brenden Ronna, and Ulrike Ott. 2016. P value interpretations and considerations. *Journal of Thoracic Disease* 8, 9 (2016), E928.

[92] Balasaravanan Thoravi Kumaravel, Fraser Anderson, George Fitzmaurice, Bjoern Hartmann, and Tovi Grossman. 2019. Loki: Facilitating Remote Instruction of Physical Tasks Using Bi-Directional Mixed-Reality Telepresence . In *Proceedings of ACM Symposium on User Interface Software and Technology (UIST)*.

[93] Jean-Marc Valin, Koen Vos, and Tim Terriberry. 2021. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000. https://datatracker.ietf.org/doc/html/rfc9000 [accessed on 10/07/2024].

[94] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[95] Irene Viola and Pablo Cesar. 2023. Volumetric Video Streaming: Current Approaches and Implementations. *Immersive Video Technologies* (2023).

[96] Anna Volokitin, Stefan Brugger, Ali Benlalah, Sebastian Martin, Brian Amberg, and Michael Tschannen. 2022. Neural Face Video Compression using Multiple Views. In *Proceedings of the IEEE/CVF CVPR*.

[97] Kenneth J Waldron and James Schmiedeler. 2016. Kinematics. In *Springer Handbook of Robotics*. 11–36.

[98] Liao Wang, Qiang Hu, Qihan He, Ziyu Wang, Jingyi Yu, Tinne Tuytelaars, Lan Xu, and Minye Wu. 2023. Neural Residual Radiance Fields for Streamably Free-Viewpoint Videos. In *Proceedings of IEEE/CVF CVPR*.

[99] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. 2021. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10039–10049.

[100] Yichao Wang, Yili Ren, Yingying Chen, and Jie Yang. 2022. Wi-Mesh: A WiFi Vision-based Approach for 3D Human Mesh Construction. In *Proceedings of ACM SenSys*.

[101] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image Quality Assessment: from Error Visibility to Structural Similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[102] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. 2022. HumanNeRF: Free-viewpoint Rendering of Moving People from Monocular Video. In *Proceedings of IEEE/CVF CVPR*.

[103] Nan Wu, Kaiyan Liu, Ruizhi Cheng, Bo Han, and Puqi Zhou. 2024. Theia: Gaze-driven and Perception-aware Volumetric Content Delivery for Mixed Reality Headsets. In *Proceedings of ACM MobiSys*.

[104] Huiqiang Xie, Zhijin Qin, Geoffrey Ye Li, and Biing-Hwang Juang. 2021. Deep Learning Enabled Semantic Communication Systems. *IEEE Transactions on Signal Processing* 69 (2021), 2663–2675.

[105] Huiqiang Xie, Zhijin Qin, Xiaoming Tao, and Khaled B Letaief. 2022. Task-Oriented Multi-user Semantic Communications. *IEEE Journal on Selected Areas in Communications* 40, 9 (2022), 2584–2597.

[106] Jiahong Xie, Hao Kong, Jiadi Yu, Yingying Chen, Linghe Kong, Yanmin Zhu, and Feilong Tang. 2023. mm3DFace: Nonintrusive 3D Facial Reconstruction Leveraging mmWave Signals. In *Proceedings of ACM MobiSys*.

[107] Zhiyuan Xie, Xiaomin Ouyang, Li Pan, Wenrui Lu, Guoliang Xing, and Xiaoming Liu. 2023. Mozart: A Mobile ToF System for Sensing in the Dark through Phase Manipulation. In *Proceedings ACM MobiSys*.

[108] Hongfei Xue, Qiming Cao, Yan Ju, Haochen Hu, Haoyu Wang, Aidong Zhang, and Lu Su. 2022. M^{4}esh: mmWave-based 3D Human Mesh Construction for

Multiple Subjects. In *Proceedings of ACM SenSys*.

[109] Hongfei Xue, Qiming Cao, Chenglin Miao, Yan Ju, Haochen Hu, Aidong Zhang, and Lu Su. 2023. Towards Generalized mmWave-based Human Pose Estimation through Signal Augmentation. In *Proceedings of ACM MobiCom*.

[110] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion Models: A Comprehensive Survey of Methods and Applications. *Comput. Surveys* 56, 4 (2023), 1–39.

[111] Zhenyu Yang, Bin Yu, Wanmin Wu, Ross Diankov, and Ruzena Bajscy. 2006. Collaborative Dancing in Tele-immersive Environment. In *Proceedings of ACM International Conference on Multimedia*.

[112] Sergey Zagoruyko and Nikos Komodakis. 2015. Learning to Compare Image Patches via Convolutional Neural Networks. In *Proceedings of the IEEE/CVF CVPR*.

[113] Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. 2020. Fast Bi-layer Neural Synthesis of One-shot Realistic Head Avatars. In *Proceedings of ECCV*.

[114] Emin Zerman, Cagri Ozcinar, Pan Gao, and Aljosa Smolic. 2020. Textured Mesh vs Coloured Point Cloud: A Subjective Study for Volumetric Video Compression. In *Proceedings of International Conference on Quality of Multimedia Experience (QoMEX)*.

[115] Anlan Zhang, Chendong Wang, Bo Han, and Feng Qian. 2022. YuZu: Neural-enhanced Volumetric Video Streaming. In *Proceedings of USENIX NSDI*.

[116] Bowen Zhang, Zhijin Qin, Yiyu Guo, and Geoffrey Ye Li. 2022. Semantic Sensing and Communications for Ultimate Extended Reality. https://arxiv.org/abs/2212.08533. [accessed on 10/07/2024].

[117] Ding Zhang, Puqi Zhou, Bo Han, and Parth Pathak. 2022. M5: Facilitating Multi-User Volumetric Content Delivery with Multi-Lobe Multicast over mmWave. In *Proceedings of ACM SenSys*. https://doi.org/10.1145/3560905.3568540

[118] Tianshu Zhang, Buzhen Huang, and Yangang Wang. 2020. Object-occluded Human Shape and Pose Estimation from a Single Color Image. In *Proceedings of IEEE/CVF CVPR*.

[119] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. 2018. RF-based 3D Skeletons. In *Proceedings of ACM SIGCOMM*.

[120] Yiqin Zhao and Tian Guo. 2020. PointAR: Efficient Lighting Estimation for Mobile Augmented Reality. In *Proceedings of ECCV*.

[121] Yiqin Zhao and Tian Guo. 2021. Xihe: a 3D Vision-based Lighting Estimation Framework for Mobile Augmented Reality. In *Proceedings of MobiSys*.

[122] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. 2021. PaMIR: Parametric Model-Conditioned Implicit Representation for Image-based Human Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 6 (2021), 3170–3184.

[123] Yuanwei Zhu, Yakun Huang, Xiuquan Qiao, Zhijie Tan, Boyuan Bai, Huadong Ma, and Schahram Dustdar. 2022. A Semantic-aware Transmission with Adaptive Control Scheme for Volumetric Video Service. *IEEE Transactions on Multimedia* 25 (2022), 7160–7172.