Theia: Gaze-driven and Perception-aware Volumetric Content Delivery for Mixed Reality Headsets

Nan Wu
George Mason University
nwu5@gmu.edu

Kaiyan Liu
George Mason University
kliu23@gmu.edu

Ruizhi Cheng
George Mason University
rcheng4@gmu.edu

Bo Han
George Mason University
bohan@gmu.edu

Puqi Zhou
George Mason University
pzhou@gmu.edu

Abstract

Minimizing bandwidth consumption while maintaining satisfactory visual quality becomes the holy grail of volumetric content delivery. However, due to the huge amount of 3D data to stream, the stringent latency requirement, and the high computational workload, achieving this ambitious goal could be challenging for mobile mixed reality headsets, which can naturally enable viewers' motion with six degrees of freedom but have limited computing power. Motivated by our critical observations from a user study of eye movements with 50+ participants, in this paper, we present Theia, a first-of-its-kind gaze-driven and perception-aware volumetric content delivery system that effectively incorporates the following innovations into a holistic system: (1) real-time creation of foveated volumetric content to reduce network data usage; (2) efficient augmentation of foveal content to boost user experience; and (3) adaptive omission of peripheral content for further bandwidth savings based on eye movements. We implement a prototype of Theia using Microsoft HoloLens 2 headsets and extensively evaluate its performance. Our results reveal that compared to the state-of-the-art, Theia can drastically reduce bandwidth consumption by up to 67.0% and enhance visual quality by up to 92.5%.

CCS Concepts

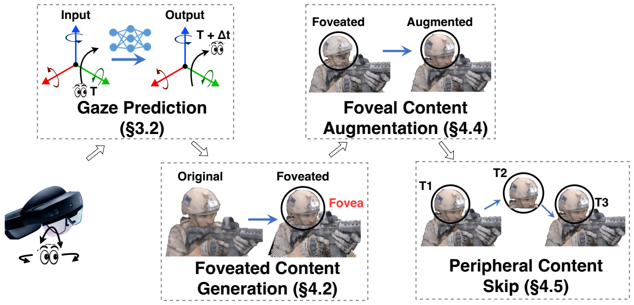
• **Information systems → Multimedia streaming**; • **Computing methodologies → Mixed / augmented reality**.

Keywords

Volumetric Video Streaming, Foveated Streaming

ACM Reference Format:
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 *The 22nd Annual International Conference on Mobile Systems, Applications and Services (MOBISYS '24), June 3–7, 2024, Minato-ku, Tokyo, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3643832.3661858

MOBISYS '24, June 3–7, 2024, Minato-ku, Tokyo, Japan
© 2024 Copyright is held by the owner/author(s).
ACM ISBN 979-8-4007-0581-6/24/06.
https://doi.org/10.1145/3643832.3661858



Input
Output
T + Δt
T
Gaze Prediction (§3.2)
Foveated
Augmented
Foveal Content Augmentation (§4.4)
Original
Foveated
Fovea
Foveated Content Generation (§4.2)
T1
T2
T3
Peripheral Content Skip (§4.5)

Figure 1: Key components of Theia**. Foveated content is created with high (reduced) quality in the fovea (periphery).**

1 Introduction

Holographic communication [24] that delivers 3D content to enable interactive and immersive applications has been envisioned as a top use case for 6G [88, 90]. A hologram, which can be approximated with volumetric content for capturing 3D scenes, is typically represented by a point cloud or mesh [11, 22]. Existing work on volumetric content delivery either directly streams compressed point clouds [38, 56] or pre-renders them into 2D content before delivery [37, 67], because point cloud is simple and flexible compared to mesh. Due to the 3D nature of volumetric content, its streaming is not only bandwidth-hungry (*e.g.*, >1 Gbps throughput [78]) but also computation-intensive for decoding, transcoding, and rendering [56]. Thus, the state-of-the-art usually deals with medium-quality volumetric content, for example, with up to 250K points per frame, on average, in ViVo [38].

Volumetric content allows viewers to adjust their viewing direction and navigate freely in 3D space, granting them the six degrees of freedom (6DoF) motion. Hence, to achieve a truly immersive and engaging experience when viewers consume point-cloud content with mixed reality (MR) headsets such as Microsoft HoloLens 2 [2], the point density that determines the visual quality should be high to avoid visual artifacts [47]. For example, it may demand >1M points per frame when users are close to the displayed content, which leads to ~3.6 Gbps bandwidth consumption (§2.3). This requirement makes volumetric content delivery challenging for MR headsets with limited computing resources. Thus, existing systems have mostly been designed for more powerful smartphones that display content on a 2D screen [38, 56, 67], leading to a barely satisfactory user experience due to the unnatural interaction with 3D volumetric content (§2.1).

Foveated rendering [36, 57] has been extensively studied for optimizing on-device computation overhead when displaying high-fidelity content on headsets [12, 83]. The high-level idea is to leverage unique features of the human visual system (HVS, §2.2) and reduce the amount of rendered content in the peripheral area outside where the user's eye gaze is located (*i.e.,* the foveal area). However, this technique *could not reduce the required bandwidth for streaming*, as the optimization is done for only rendering, and full-resolution content still needs to be delivered. To address this problem, foveated streaming has recently been explored by the computer graphics community, mainly for delivering dynamic virtual reality (VR) content and 360° videos [58, 69]. However these approaches are not fully applicable to 3D volumetric content (*e.g.,* point clouds). This is because they overlook the varying sensitivity of the HVS, and lack the consideration of occlusion optimization, which saves substantial bandwidth in 3D content streaming (§6.2). Moreover, *none of them examined gaze prediction*, which is essential to optimize the quality of experience (QoE) in a dynamic environment with fluctuating network bandwidth and delay. In these network conditions, gaze prediction is vital for pre-generating high-quality foveated content and reducing latency in content updates.

In this paper, we propose Theia[1], which is, to the best of our knowledge, the first gaze-driven and perception-aware volumetric content delivery system (Figure 1). The overarching goal of Theia is to make foveated streaming practical for volumetric content. While the concept of foveated streaming is straightforward, when designing Theia, we face the following key challenges: (1) the feasibility of accurate gaze prediction with fast eye movements; (2) the instant creation of high-fidelity foveated volumetric content with gaze data; and (3) the enhancement of QoE under demanding scenarios such as short viewing distance and low network bandwidth.

The design of Theia is motivated by the crucial observations from our IRB-approved user study with 52 participants to characterize eye movements and explore their predictability. Our key insight is that while gaze motion is inherently more flexible than head motion [31, 45], it is feasible to predict future eye movements when users consume volumetric content with MR headsets. This is because in MR scenes, virtual objects are integrated with the real world, and Theia streams them without the need to transmit background data. This differs from 360° videos or VR, where dynamic and new background leads to faster eye movements as users respond to unfamiliar visuals (§3.1). This distinction enables more precise gaze predictions. Hence, leveraging deep-learning-empowered gaze prediction, Theia generates, in real time, foveated content based on predicted gaze motion, effectively augments foveal content when viewing distance is short, and dynamically skips peripheral content during fast eye movements, benefiting from the temporal effects of HVS (§2.2). Overall, Theia incorporates the following innovations and contributions into a holistic system.

Characterizing Eye-gaze Movements (§3.1). We construct the first gaze-motion dataset with eight diverse volumetric videos. It consists of 530+ minutes of gaze trajectories from 52 participants. Compared to 360° panoramic videos, the display area of volumetric content is typically limited (*e.g.,* positioned at fixed locations), making gaze movements less random (*i.e.,* focused on the content). Thus, this nature of volumetric content facilitates more accurate gaze predictions.

Accelerating and Optimizing Foveated Content Generation (§4.2). We propose a lightweight yet efficient scheme to create foveated volumetric content in real time based on predicted gaze motion. It projects 3D points onto a 2D plane, performs a log-polar transformation [15] that leverages unique features of the HVS to drastically reduce the number of points required for maintaining satisfactory QoE, and converts the resulting 2D data back to a foveated point cloud. When doing this, Theia not only generates high-fidelity foveated volumetric content but also effectively takes viewing distance and content occlusion into consideration. Theia's design operates on individual points simultaneously and enables parallel execution on GPUs for fast processing. Thus Theia can process high-quality point clouds in real time.

Enhancing User Experience with Foveal Content Augmentation (§4.4). The visual quality of foveated content may still fall short at closer viewing distances, affecting the QoE. Existing solutions to enhance content quality, such as inpainting [29] and super-resolution [60, 99], are computation-intensive and not applicable to Theia. We design a lightweight scheme to adaptively augment the visual quality of foveal content, considering each point's distance to the fovea.

Alleviating Network Load with Peripheral Content Skip (§4.5). The above foveation-based optimizations may still generate content with substantial data size, leading to high bandwidth consumption. Leveraging the temporal effects of human perception, we propose to adaptively skip the delivery of peripheral content and reuse that of the previous frame during rapid eye movements, to further reduce bandwidth consumption while maintaining a good QoE, especially when the bandwidth is limited.

Implementing and Evaluating Theia **(§5, §6).** We build a prototype of Theia on Microsoft HoloLens 2 and thoroughly evaluate its performance via controlled experiments and another IRB-approved user study with 20+ participants. We summarize our key experimental results as follows.

- Compared to ViVo, jointly applying all optimizations in Theia reduces network data usage by 67.0% and 9.93% under unthrottled high-throughput WiFi networks and when the bandwidth is fluctuating/limited, respectively.
- Theia's server and client consistently operate at 30 FPS (frames per second) with an end-to-end latency of <100 ms across various network conditions, ensuring a smooth QoE by making gaze prediction accurate (with a short window).
- Measured by two metrics for foveated content, Theia demonstrates remarkable improvements in visual quality over ViVo, especially under fluctuating/limited bandwidth.
- Our second user study for performance evaluation indicates that various components of Theia can achieve a 45.0% – 92.5% improvement in QoE compared to ViVo.

The source code of Theia is available at https://github.com/wunan96nj/Theia_MobiSys2024. We hope they can facilitate further research on foveated streaming and gaze prediction. This work does not raise any ethical issues.

[1] Theia is the Greek goddess of sight, vision, and divine light.

2 Background and Motivation

2.1 Introduction of Volumetric Content

In contrast to 2D images with pixels, volumetric content can be represented in formats such as point cloud and mesh [70]. Point cloud, comprising 3D points with attributes such as color, has gained popularity [38, 56, 67] due to its simplicity and flexibility in representing non-manifold structures [27]. This paper focuses on point clouds, aligning with current practices in state-of-the-art streaming systems [34, 38, 56, 100, 102]. We can potentially extend the high-level concepts of Theia to 3D mesh by adjusting peripheral resolution.

Volumetric content delivery requires substantial network bandwidth and computation resources to ensure a satisfactory QoE. Recent research has focused on optimizing bandwidth consumption and computation overhead. These strategies include visibility-aware content reduction [38, 81], remote rendering to minimize client and network workload [37, 67], novel compression and decoding schemes [56], and 3D super-resolution to enhance the content quality [100]. Meanwhile, millimeter wave technology that provides multi-Gbps bandwidth has been utilized for supporting high-quality volumetric content delivery, even in multi-user scenarios [101, 102].

While volumetric content could be displayed on various devices, including personal computers (PCs), smartphones, and headsets, the key differences lie in their user interface and display methods. PCs can only emulate 6DoF motion with a mouse and keyboard. Smartphones track 3DoF rotational movements but struggle with translational movement localization [16]. Mobile headsets, such as Microsoft HoloLens 2 [2], can naturally support 6DoF motion with dedicated sensors and specialized software building blocks. Moreover, given the small display size of smartphones, their entire screen, and thus all displayed content, may fall into the foveal area, making foveated streaming less effective.

2.2 Human Visual System and Perception

Human Visual System (HVS) works with photoreceptors in the retina and neural pathways in the brain to interpret and comprehend visual stimuli from the environment [40, 94]. Human vision consists of two regions: the fovea and the periphery [36]. The fovea is a small area with a high density of cone cells and nearly half of the optic nerve fibers, enabling clear and detailed central vision. The remaining optic nerve fibers are dispersed in the peripheral retina for detecting information from the encompassing visual field [32].

Saccadic Omission. Eye movements can be broadly classified into three patterns based on speed: fixation, smooth pursuit, and saccades [62]. Saccades are rapid eye movements in a short time period, resulting in diminished visual perception, which is referred to as saccadic omission [28, 95]. The omission is a neural mechanism that prevents the transmission of visual data to the brain to avoid the negative experience caused by motion blur during rapid eye movements. It initiates ~50 ms before a saccade and persists throughout its duration. The sensitivity of HVS typically returns to its full capacity within 40–60 ms after a saccade ends [68, 95].

Temporal Effects of HVS are about how human vision works over time [12]. Specifically, these effects involve the persistence of information within the HVS for a brief period after the visual stimulus disappears. Although both the foveal and peripheral regions of HVS are susceptible to temporal effects [17], within their threshold, the HVS maintains a continuous and stable visual quality [49].

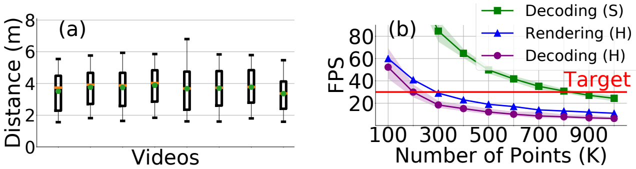


Figure 2: (a): Viewing distances to volumetric content (from our user study). (b): Decoding and rendering framerate of Samsung Galaxy S21 (S) and Microsoft HoloLens 2 (H). We omit the framerate of decoding point clouds with less than 200K points on S21, as it is >180.

Note that while perception awareness has been explored in previous work, such as Pano [35] for spatial visual quality adaptation in 360° video streaming, Theia incorporates the unique saccadic omission and temporal effects inherent to the HVS into volumetric content delivery. Theia leverages eye-gaze data, rather than head movements, to offer further mobile data reduction and user-experience improvement.

2.3 Motivation

In contrast to smartphones, delivering high-quality volumetric content for MR headsets brings forth two significant challenges: notably increased computational demands on the client side and even higher bandwidth consumption.

Goal: High-quality Volumetric Content Delivery for MR Headsets. Since volumetric content enables 6DoF motion, satisfactory visual quality depends on the viewing distance of users [38]. Among existing public datasets of volumetric content, 8i [1] offers a high fidelity with up to 1M points per frame. Those points are voxelized, with the size of each voxel ~1.75 mm. Considering normal visual acuity with 20/20 vision [87] (*i.e.,* the ability to distinguish objects/colors with an eccentric angle as low as 1 arcminute), low-resolution artifacts could be noticed even when rendering ~1M points in each frame if the viewing distance is <6 m (§4.4). Figure 2(a) plots the viewing distance to volumetric content from the traces collected during our user study (§3). The typical short distance shown in this figure necessitates the delivery of high-quality volumetric content for MR headsets.

Challenge 1: High Computation Overhead. Decoding and rendering high-fidelity volumetric content is computation-intensive for MR headsets. In Figure 2(b), we plot the frame-rate of point cloud rendering and decoding (with Draco [5]) on HoloLens 2 [2] and the decoding framerate of Samsung Galaxy S21 (a smartphone released in January 2021) for volumetric content with different point densities. In the plot, we depict the 5th, 25th, 75th, and 95th percentiles, medium, and mean (green dots). HoloLens 2 can support the rendering of volumetric content at 30 FPS with only up to 300K points. Point cloud decoding also presents a bottleneck, achieving higher than 30 FPS with only ~200K points per frame, which is significantly less than the number of points (*e.g.,* ~1M points) that may be required for rendering high-fidelity content. While parallelization could accelerate the decoding [56], it competes with

Name	Avg # of Pts/Frm	Total # of Frames	Bitrate w/o Comp.	Bitrate w/ Comp.
Soldier	1,075K	300	3,870	780
LD	834K		3,002	693
Loot	794K		2,858	600
RnB	727K		2,617	567
Lubna	402K	300	1,447	362
Matis	406K		1,461	336
V1	29K	1,938	104	29
V2	60K	2,612	216	58

Table 1: Dataset with eight diverse videos. The first four videos are from 8i [1] (LD: Long Dress; RnB: Red and Black). The next two videos are from V-SENSE [79, 80]. The last two videos were captured by ourselves, one capturing a cosplay actor, and another capturing a singer. Bitrate is in Mbps.

rendering for GPU resources. Figure 2(b) also demonstrates that the computing power of HoloLens 2 is much lower than Samsung Galaxy S21 (200K vs. 800K for 30 FPS decoding). However, considering the heat dissipation of head-mounted displays [72] such as HoloLens 2, adding more computation resources to make them comparable to smartphones remains challenging.

Challenge 2: High Bandwidth Requirements. Delivering high-quality volumetric content for MR headsets is extremely bandwidth-demanding. For example, streaming uncompressed point clouds containing >1M points at 30 FPS requires ~3.6 Gbps bandwidth, assuming each point takes 15 bytes with 4 bytes per (X, Y, Z) position coordinates and 1 byte per (R, G, B) color dimensions. Even with various optimizations proposed in previous work, such as ViVo [38], the required bandwidth could be as high as 450 Mbps when delivering point clouds with more than 1M points [56].

The above two challenges motivate us to optimize computation overhead and bandwidth requirements by taking advantage of the unique features of HVS for delivering high-quality volumetric content to MR headsets (*e.g.,* leveraging foveated streaming).

3 Gaze Movement & Prediction

3.1 Characterizing Eye-gaze Movement

Basics of Eye Tracking. Eye tracking is a vital component in foveated rendering, which monitors and records eye movements in real time and is now available on MR headsets, such as HoloLens 2 [7]. Various eye-tracking techniques have been developed, including the tracking of pupil movement via the light reflection from a VR headset's screen [61] and the study of light absorption by the pupil, facilitated by photodiodes strategically placed on AR headsets around the eyes [62].

User Study for Collecting Gaze-motion Data. Since there is no publicly available gaze-motion dataset for volumetric content, we conduct an IRB-approved user study to gain a deep understanding of gaze behavior and facilitate the design and evaluation of Theia. We recruit 52 participants (Female: 17; Male: 35) from a large university, with an average age of 24.7±2.1. Table 1 summarizes the characteristics of the eight videos in our dataset, which are diverse in terms of averaged number of points per frame (~30K to >1,000K), video length (10 s to >85 s), and bitrate (~100 Mbps to >3,800 Mbps before compression; ~30 Mbps to ~800 Mbps after compression).

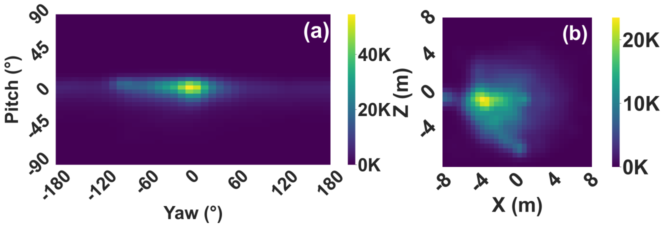


Figure 3: Heatmaps for (a) rotational and (b) translational dimensions of gaze data collected in our user study.

For data collection, we employ the built-in gaze tracking of HoloLens 2, which operates at 90 Hz [7]. Considering that the 8i and V-SENSE videos have a duration of only 10 seconds, we repeat them eight times before proceeding to the next one to gather enough data samples. Since volumetric content enables 6DoF motion, repeating the same video does not diminish the interactivity/engagement of users as they can consume the content from varying angles and distances during replay. We let users watch our captured long videos V1 and V2 only once. In total, our dataset contains 530+ minutes of gaze motion trajectories for volumetric content.

Analyzing Gaze Movement. To investigate gaze motion patterns when users consume volumetric content on MR headsets, we analyze the heatmaps of gaze data for rotational and translational dimensions and calculate the proportions of different types of gaze movements, including fixation, smooth pursuit, and saccade.

Figure 3 indicates how displayed volumetric content influences users' 5DoF [2] motion and attention. We omit the Y dimension in Figure 3 as our dataset shows that vertical motion is limited, which is consistent with the observation in ViVo [38]. Figure 3(a) presents the heatmap for the rotational dimension, showing the majority of gaze data is concentrated. Moreover, the scattering along the yaw dimension is more pronounced than the pitch dimension, reflecting the natural tendency of human gaze to cover a wider range horizontally. Figure 3(b) shows the heatmap for the translational dimension. This heatmap is highlighted around $(-4, 0)$ because the volumetric content at $(0, 0)$ is facing this direction.

We compare the proportions of different eye-movement stages when users watch volumetric content on MR headsets and 360° videos on VR headsets by contrasting our dataset with that released by Xu *et al.* [97] (referred to as **`360-Video`** thereafter). The latter is a large-scale gaze-tracking dataset for 360° videos, consisting of 45 participants watching 208 videos at 25 FPS. Based on the motion-velocity settings [62] we classify eye movements into three stages. Gaze motion with a velocity slower than 5 °/*s* is categorized as fixation, between 5 and 40 °/*s* as smooth pursuit, and exceeding 40 °/*s* as saccades. Using these criteria, we generate the cumulative distribution function (CDF) for gaze velocity in both datasets, as illustrated in Figure 4. We employ the z-test [93] and reveal significant differences in gaze patterns between the two datasets, with a

[2] Since the gaze vector only describes which direction to look in (*i.e.,* yaw and pitch), gaze information for volumetric content possesses 5DoF (X, Y, Z, yaw, pitch), where X is for left and right, Y for up and down and Z for forward and backward.

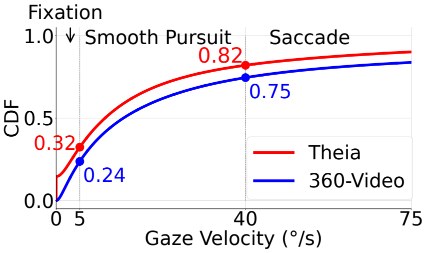


Figure 4: Comparison of gaze velocities of the `360-Video` dataset and ours.

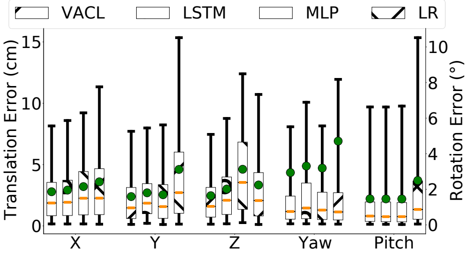


Figure 5: Gaze prediction errors of different models. Our VACL model performs the best.

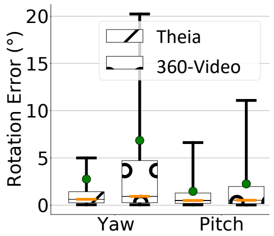


Figure 6: Rotational gaze prediction error.

p-value close to 0. Compared to the `360-Video` dataset, the fixation stage in our dataset makes up a larger percentage (33.3% higher), while the saccade stage takes up a smaller percentage (28.0% lower).

3.2 Gaze Prediction

In this section, we demonstrate our insight in §3.1. When users watch volumetric content on MR headsets, gaze prediction tends to be more accurate than when they view 360° videos on VR headsets.

VACL Model. Accurate gaze prediction is vital for enhancing the QoE in foveated streaming. By predicting where a user will look next, we can generate foveated content with optimal visual quality in advance, reducing the latency between gaze movements and the resulting updates of foveal content. Our prediction model, named VACL (Velocity-Acceleration-CNN-LSTM), predicts each dimension (X, Y, Z, yaw, and pitch) individually by combining a convolutional neural network (CNN) for feature extraction with a long short-term memory (LSTM) for learning sequential gaze data. The model also takes gaze velocity and acceleration as inputs, which improves prediction accuracy [46]. VACL's CNN component has one layer with an output channel size of 256, and the LSTM has a hidden layer size 128. For training, we utilized the Adam optimizer [53] with a learning rate of 0.001. The model is trained with a batch size of 50,000 and uses mean absolute Error (MAE) as the loss function. We set the prediction window to 100 ms, considering both the computational latency on the edge and client (§6.3) and the capabilities of 5G networks (§6.1). We also show the effectiveness of VACL with a longer prediction window in §6.4. We set the history window as 100 ms due to observed negligible differences between history windows ranging from 100 ms to 300 ms.

Evaluation Results. We compare VACL with three baselines: linear regression (LR), multilayer perceptron (MLP), and a state-of-the-art LSTM proposed by Illahi *et al.* [46]. We use different user sets for training and testing, applying it across all content. Figure 5 shows that the VACL model outperforms the others by handling the complexity of gaze dynamics with the CNN and LSTM components, exhibiting lower/comparable errors across all dimensions. The average translational errors are approximately 2.64 cm, and rotational errors are around 2.68°.

Gaze prediction for volumetric content poses a greater challenge than 360° videos, requiring accurate prediction for both rotational and translational dimensions. However,volumetric content in an MR setting, seamlessly integrates with the real world, eliminating the need for background information. This differs from 360° videos, where backgrounds affect gaze direction and cause more frequent saccades. Figure 6 compares the predictions on the `360-Video` dataset and ours in rotational dimensions, revealing that gaze prediction for volumetric content exhibits significantly lower errors.

We measure the inference time of the VACL model on an NVIDIA GeForce RTX 3060 GPU. While a larger history window adds to the VACL model more input features, this model is lightweight (with 128 units for each of the two layers) and maintains an average inference time of 0.95 ms across various history window sizes. We find this gaze prediction latency to be acceptable, compared to the 33 ms processing time budget allocated for computing on the edge in 30 FPS streaming.

4 System Design of Theia

4.1 Overview

Theia is a volumetric content delivery system designed for MR headsets, which leverages their eye-tracking capabilities and the unique features of HVS via foveated streaming. Theia focuses on 3D content streaming to mitigate content drifts (§6.1)), and demonstrates techniques designed for 2D frames are suboptimal for volumetric content, as most of them ignored gradual sensitivity drops in HVS (§6.2)) and disregarded the occlusion points. The primary objective of Theia is to make foveated streaming practical for volumetric content, by benefiting from the synergy between mobile computing, networked systems and multimedia.

Figure 7 illustrates the architecture and workflow of Theia. It focuses on optimizing the edge-to-client streaming, for which the edge consistently prefetches the coarse-grained, *viewport-adaptive* volumetric content from a remote server by adopting existing designs [38, 56]. For edge-to-client streaming, the Theia client continuously performs gaze tracking and uploads the gaze trajectory and estimated network capacity to the edge. Based on the uploaded information and the prefetched content, Theia first efficiently generates foveated, *gaze-adaptive* content in real time and determines the proper point size for rendering. Thus, different from viewport-adaptive techniques [38, 56], which typically adjust content based on users' head motion, Theia's foveated streaming leverages gaze data to further optimize bandwidth consumption and on-device computational resource usage for viewport content. Theia then intelligently augments the foveal content by considering the viewing distance and adaptively skips the current frame's peripheral content by reusing the previous one based on the eye movement speed. Theia uses Draco [5] for content encoding, a state-of-the-art

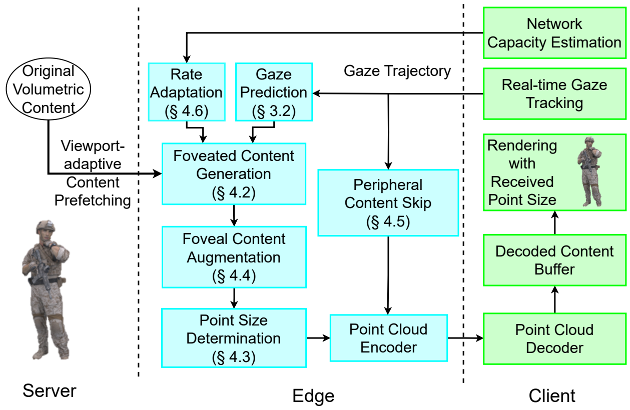

Original Volumetric Content
Viewport-adaptive Content Prefetching
Server
Rate Adaptation (§ 4.6)
Gaze Prediction (§ 3.2)
Gaze Trajectory
Foveated Content Generation (§ 4.2)
Foveal Content Augmentation (§ 4.4)
Point Size Determination (§ 4.3)
Peripheral Content Skip (§ 4.5)
Point Cloud Encoder
Edge
Network Capacity Estimation
Real-time Gaze Tracking
Rendering with Received Point Size
Decoded Content Buffer
Point Cloud Decoder
Client

Figure 7: System architecture and workflow of Theia**.**

encoding method in systems such as ViVo [38], YuZu [100], MetaStream [34], and CaV3 [66]. Once receiving the delivered volumetric content, the Theia client decodes and renders point clouds with the received point size. We emphasize that most of Theia's design choices aim to *enable parallel execution on GPUs for fast processing*, such as operating on individual points simultaneously. Thus, processing a higher number of points (*e.g.,* in multi-object scenarios) does not significantly increase the computation latency (§6.3).

4.2 Real-time Foveated Content Generation

Problem. During streaming, the foveated volumetric content is generated at the edge based on the (predicted) gaze movements of users. To optimize QoE, Theia necessitates an efficient content creation scheme that aligns well with the HVS. Such a scheme demands real-time generation of foveated content with a minimal data size that could achieve the visual quality required by the HVS.

Challenge. Foveated streaming should carefully balance the computation latency of content generation, the size of created content, and the resulting visual quality. Consider a straightforward approach proposed by Tefera *et al.* [92], which divides content in the viewport into multiple concentric cones and gradually reduces each cone's quality (*i.e.,* the further away from the center, the lower). While this method offers fast execution by merely determining the corresponding concentric cone for each point, dividing the content into concentric cones results in unsmooth changes of visual quality from the fovea to the periphery. Furthermore, the foveal angle (*i.e.,* the size of the foveal area) lacks consensus in previous studies (*e.g.,* 1.5° [12, 41], 5° [36], and 7.5° [42]). This variance makes it challenging to determine a universally accepted foveal angle for content generation. More importantly, this scheme fails to utilize depth information, which is helpful for discarding occluded points and indistinguishable ones for long viewing distances to save bandwidth [38].

Our Approach. Theia's approach is inspired by the log-polar transformation [15] for 2D videos, which has demonstrated its capabilities of generating, in real time, foveated content with a smooth transition between different levels of detail [74]. As shown in Figure 8(a), it utilizes a 2D buffer where each row represents a circular ring that is equally spaced in the log-polar coordinate system, and each column represents a sector that is equally spaced in the same system. When the row index u increases, the buffer element at (u, v) covers a larger area, similar to a human's acuity drop [36].

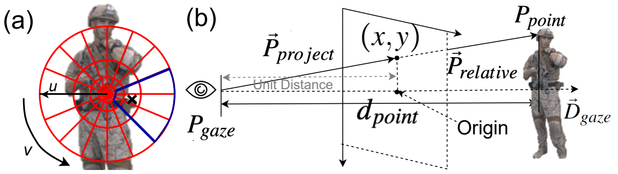


Figure 8: Illustrations of foveated content generation and foveal content augmentation: (a) log-polar transformation (X: target empty buffer element) and (b) 3D to 2D projection on a dummy plane.

We propose a novel log-polar transformation with 3D-2D-3D projection in Theia to generate foveated volumetric content, instead of directly handling 3D data, which could be computation-intensive. The high-level idea is to project 3D points onto a dummy 2D plane while recording their depth information, apply a log-polar transformation to the projected points to generate foveated 2D content, and then use the recorded depth information to convert the resulting 2D content back to a foveated point cloud. When doing this, our lightweight transformation could seamlessly incorporate occlusion- and distance-visibility optimizations into Theia, eliminating the additional overhead required for such processing. Note that similar optimizations in ViVo [38] determines the occluded content at the cell level, which is coarse-grained, to save computation overhead. On the other hand, Theia determines the occlusion at the fine-grained point level by employing the 3D-2D-3D projection.

We first project each point to a dummy 2D plane, which is perpendicular to the gaze direction and is placed at one unit distance away from the user's eyes, through 3D to 2D perspective projection [33]. As shown in Figure 8(b), we project the point P_{point} to the dummy plane at $P_{project}$, (x, y). To get this projected point, we calculate the relative position vector of P_{point} to the gaze origin (*i.e.,* the center of two eyes) as $\vec{P}_{relative} = P_{point} - P_{gaze}$. The depth along the gaze direction is $d_{point} = \vec{P}_{relative} \cdot \vec{D}_{gaze}$, where $\vec{D}_{gaze}$ is the gaze vector. The perspective projected position vector is $\vec{P}_{project} = \vec{P}_{relative}/d_{point}$. By dividing d_{point}, the depth of $\vec{P}_{project}$ along the gaze direction is set to one unit, making it end on the dummy plane. Accordingly, we can retrieve $P_{project}$, (x, y), on the dummy plane (*i.e.,* $\vec{P}_{project}$'s endpoint). After that, each point on the plane with a coordinate (x, y) is transformed to the log-polar coordinate (u, v) as follows.

$$\begin{aligned} u &= log_b(\frac{\rho}{tan(MAR_0)}),\ 0 \le u < H \\ v &= \arctan(\frac{y}{x})/2\pi \times W \end{aligned} \tag{1}$$

where $b = (W + \pi)/(W - \pi)$ follows the design in Araujo and Dias [15] and describes how fast the quality drops from the foveal to the peripheral area. $W \times H$ represents the size of the log-polar buffer. We set b to be 1.022 for high-quality transformation [36], resulting in $W = 286$. ρ is the distance from the projected point to the plane's origin. MAR_0, the smallest minimum angle of resolution (MAR), is set to be 1 arcminute, the highest foveal acuity of healthy, non-elderly adults [36]. Given that the diagonal field of view (FOV) of existing headsets is <70° [3, 8], we create a buffer with $H = log_b(tan(140°/2)/tan(MAR_0)) \approx 420$ to cover a 140° FOV, addressing the extreme case where content is rendered, for instance, at the bottom-left, while the fovea is at the top-right of FOV. To introduce occlusion and distance awareness into Theia, when multiple points are mapped to the same place, we keep only the one with the smallest d_{point}, and when closeby distant points are mapped to the same log-polar buffer, only the closest one with the smallest d_{point} is kept.

After the aforementioned processing, the content is transformed into a log-polar buffer, where (u, v) denotes the coordinate of each point in the buffer. To generate the foveated volumetric content, we finally reverse the log-polar buffer back to the original Cartesian representation. The transformation of the buffer element at (u, v) to (x', y') on the 2D plane is as follows.

$$\begin{aligned} x' &= \tan(MAR_0) \times b^u \times \cos(\theta) \\ y' &= \tan(MAR_0) \times b^u \times \sin(\theta) \\ \theta &= 2\pi \times v/W \end{aligned} \tag{2}$$

To convert the transformed (x', y') back to 3D point P'_{point}, we first retrieve the transformed position vector $\vec{P}'_{project}$ given the dummy plane and (x', y'). Then, we can get the position vector $\vec{P}'_{point} = \vec{P}'_{project} \times d_{point} + P_{gaze}$, whose endpoint is part of the foveated volumetric content.

4.3 Point Size Determination

The size of a 3D point is an important attribute that affects the visual quality of rendered volumetric content [55], but has not yet been studied in existing work on volumetric content delivery [38, 56, 67, 100, 102]. For point-cloud-based volumetric content, each point represents a voxel in 3D space. The proper point size depends on the point density (*e.g.,* the size should be small for a dense area) to avoid visual artifacts. For example, in the 8i dataset [1], points are uniformly sampled with each covering a voxel with a size s of 1.75 mm. Thus, setting the point size to 1.75 mm is essential for optimal visual quality.

The density of points generated by Theia decreases from the fovea to the periphery. Therefore, to achieve high visual quality, we should adjust each point's size based on its distance to the fovea for high-quality rendering. The key insight is that elements of log-polar buffer at different rows cover voxels with different sizes: $\tan(MAR_0) \times (b^{u+1} - b^u) \times d_{point}$, where u represents the row index of the buffer element, and d_{point} is the recorded depth for the point that is mapped to the buffer element. We set the rendering size of a point to be the same as the voxel size it represents to ensure optimal visual quality. The determined point sizes are transmitted to MR headsets for rendering.

4.4 Foveal Content Augmentation

Problem. The visual quality of high-fidelity content may still fall short at closer viewing distances, affecting the QoE. Since the foveated content generation in Theia follows the HVS, when the viewing distance is short (*e.g.,* <6 m in the 8i dataset [1][3]), many buffer elements may be empty. This is because the log-polar buffer becomes denser near the fovea, and the original content may not be dense enough to fill this region, leading to visual artifacts in the foveal area.

Challenge. We can improve the quality of foveal content by either augmenting the log-polar buffer or upsampling the foveated point cloud. Previous work in the computer vision community has explored inpainting [29] for filling missing 2D pixels or 3D super-resolution [60] for point upsampling. However, both of them are computation-intensive [18, 100] and thus cannot be directly applied to Theia. For example, on an edge that is equipped with an Intel i7-11700 CPU and an NVIDIA GeForce RTX 3060 GPU, the inpainting function implemented by OpenCV [20] takes more than 500 ms. While PU-GAN [60], one of the state-of-the-art super-resolution models, takes >200 ms to upsample the point clouds by 4×.

Our Approach. In Theia, we design a lightweight approach to enhance foveated content and improve QoE by directly augmenting the 2D log-polar buffer due to its simplicity compared to 3D point clouds. As shown in Figure 8(a), a log-polar buffer element at (u, v) may lack a mapped point if the original content is not sufficiently dense. To address this, we propose to search the non-empty neighbors around (u, v) and fill the buffers using their interpolated color and depth values. However, it is essential to confine the search distance. The excessive searching distance can result in inaccurate fills for buffer elements that should remain empty (*i.e.,* where no content is meant to be rendered). As a result, our search range is confined between $(u - u_{diff}, v - v_{diff})$ and $(u + u_{diff}, v + v_{diff})$ within the log-polar buffer.

For determining the values of u_{diff} and v_{diff}, we first consider a scenario where the maximum distance from any point within two adjacent voxels to a voxel center is the voxel size s. Thus, the minimum searching distance that guarantees to reach a voxel center from these points is s. We consider any empty log-polar buffer element that is transformed back to these points should be augmented. On the dummy plane (§4.2), s in 3D space is transformed to $\tau = s/d_{point}$. Thus, we derive v_{diff} by calculating the equivalent size of τ in the log-polar space. Specifically, $v_{diff} = \tau/(2\pi \times \tan(MAR_0) \times b^u/W)$. u_{diff} is determined by looping from 0 to $H - u$, until $\tan(MAR_0) \times (b^{u+u_{diff}} - b^u) > \tau$. Theia searches for at most 4 neighbors by following the design in bilinear interpolation [54]. If all neighbors within the searching region are vacant, the buffer element at (u, v) is left empty, indicating that there is no content available for rendering.

4.5 Peripheral Content Skip

Problem. With the log-polar buffer size defined in §4.2, the number of delivered points could be as high as 420×286 = 120,120, which happens when all buffer elements are not empty. Recall that each point takes 15 bytes (§2.3). If we use a float number (4 bytes) to represent the point size, the required bandwidth of the raw point

[3] In the 8i dataset [1], each point covers a voxel size of 1.75 mm. We can calculate the maximum distance D at which human eyes can distinguish two adjacent points with voxel sizes of s with $D = s/tan(MAR_0)$. With a voxel size s = 1.75 mm, $D \approx 6$ m. This means that visual artifacts in point clouds can be noticed by humans with healthy eyes within 6 m.

cloud could be up to 120,120 × 19 (bytes) × 8 (bits) × 30 (FPS) ≈ 547 Mbps. The reduction from high bandwidth (*e.g.,* ~3.8 Gbps for Soldier) to ~500 Mbps illustrates the potential bandwidth savings of our log-polar domain encoding, with ~500 Mbps representing the theoretical maximum buffer size. Thus, we should further reduce the amount of transmitted data for good QoE, especially under low network bandwidth.

Challenge. Given that our log-polar transformation, which follows the requirements of HVS, has reduced the number of points as much as possible without sacrificing QoE, further performing spatial compression will inevitably introduce visual artifacts. On the other hand, while temporal frame skip can potentially reduce a large amount of data, naively skipping frames may degrade visual quality, resulting in issues such as stuttering [75].

Our Approach. Although existing systems, such as ViVo [38], adopt *spatial* skipping of peripheral content out of users' predicted viewport, none of them has explored *temporal* skipping to save bandwidth. We propose to dynamically skip peripheral content in frames out of the foveal area, instead of the viewport, when eye-movement speed is high. When the current frame's peripheral content is temporally skipped, we reuse that from the previous frame for rendering.

The rationale behind this design choice is that humans are less sensitive to changes in the peripheral area during rapid eye movements (§2.1). We incorporate this design to save bandwidth for streaming higher-quality foveal content, which could enhance QoE with optimized bandwidth usage, particularly under fluctuating/limited conditions (§6.5). Theia defines the peripheral area by considering a 7.5° foveal angle, the maximum from the literature [12, 36, 41, 42]. We first estimate the speed of gaze motion based on recently received and the predicted gaze data. Then, we compare the estimated speed with a threshold T_s to determine whether to skip the peripheral content. A larger value of T_s skips less peripheral content with minimal impact on visual quality, but resulting in limited bandwidth reduction. In contrast, a smaller value of T_s leads to more frequent skipping of peripheral content during fast eye movements, saving more bandwidth but may negatively affect QoE. Theia empirically sets T_s as 10 °/s based on our performance evaluation of different values for T_s in §6.4.

4.6 Rate Adaptation

A straightforward method for adapting content quality during streaming is to reduce point density based on estimated network capacity, similar to adaptive bitrate streaming for traditional videos [48, 71]. However, directly applying this method to point clouds generated by Theia can significantly affect QoE. This is because Theia carefully calculates each point's size, and dropping points without updating their sizes can result in artifacts (*i.e.,* holes) in the rendered content.

In Theia, the point size is determined during log-polar transformation, and thus we propose to resize the log-polar buffer by adjusting its dimension before performing the transformation to accommodate network dynamics. We calculate the reduction rate r of log-polar buffer, compared to the buffer size defined in §4.5, based on an existing throughput estimation algorithm [48]. Then, Theia resizes the buffer to $H/\sqrt{r} \times W/\sqrt{r}$ before performing the log-polar transformation. Accordingly, each point's size is increased by $\sqrt{r}$ to cover a larger voxel due to the reduction of point density.

4.7 Integration of Gaze Prediction

Theia incorporates the VACL model (§3.2) for gaze prediction to enhance QoE, aligning with the high throughput and low latency envisioned in 6G. Theia employs a default prediction window of 100 ms to accommodate latency introduced by data processing and transmission, as well as addressing network jitter. This window is adjustable: a smaller window enhances gaze prediction accuracy but demands quicker content generation and delivery, while a larger window provides more time for these tasks but may result in slower response to gaze changes, potentially lowering QoE. We show how a longer prediction window affects the visual quality in §6.4.

The Theia client periodically sends gaze data, along with content requests, to the edge. Upon receiving the data, Theia first performs gaze prediction and then generates content based on the predicted gaze motion. We do not design further measures to handle inaccurate gaze prediction, as it typically occurs during saccadic eye movements [13, 14, 44, 76]. In such situations, saccadic omission (§2.2) serves to alleviate the negative impact of gaze-prediction error on QoE, as users may not perceive quality drops resulting from inaccurate prediction caused by saccades. This is verified by another user study in §6.5, which evaluates users' QoE while watching volumetric content with Theia.

Our research initially focused on single-object scenarios to motivate the development of our design, while it proves effective in multi-object scenarios. Specifically, the edge leverages viewport-visibility content prefetching (§4.1) to limit bandwidth consumption in scenarios when multiple objects are not simultaneously visible. We further show that the occlusion-aware and parallel processing designs make Theia remain effective in multi-object scenarios through evaluations (§6).

5 Implementation

We implement the Theia server in C++ on Linux and test it with Ubuntu 18.04. We develop the prototype of the Theia client on HoloLens 2 [2] using Unreal Engine 4.26 [10]. Our implementation is compatible with other headsets that support eye tracking. On the server, we implement gaze prediction using PyTorch C++ [82] and leverage the server's GPU with the CUDA toolkit [4] to create foveated content and augment the content quality. The client-server communication is based on a custom protocol over TCP. For the client on HoloLens 2, we enable 90 FPS eye tracking with the Extended Eye Tracking APIs [6]. We cross-compile the Draco library [5] to decode compressed point clouds and use Unreal Engine's Niagara particle system to render them. We save the decoded points' positions and sizes into float textures and their colors into uint8 textures, with each texture pixel representing a point to enable efficient data transfer to the GPU-accelerated system for rendering. In total, our implementation consists of 11,400+ lines of code (LoC) in C++: 3,900+ LoC for the server and 7,500+ LoC for the client.

6 Performance Evaluation

6.1 Experimental Setup

Devices. Our client device is Microsoft HoloLens 2 [2] which is equipped with a Qualcomm Snapdragon 850 chip. The edge server is a machine equipped with an Intel i7-11700 CPU, 32GB memory, and an NVIDIA GeForce RTX 3060 GPU.

Network Conditions. We connect the client and server with a Linksys WiFi router. The throughput is ~450 Mbps, and the round-trip time is ~6 ms. We increase the round-trip time to ~30 ms [86] with `tc` [9]. Moreover, we evaluate the performance of Theia over fluctuating/limited bandwidth in a reproducible manner by utilizing `tc` [9] to replay five network bandwidth traces collected at various locations on a large commercial cellular network in the U.S. The average bandwidths of these traces are 25.5±2.7 Mbps, 35.3±5.1 Mbps, 51.27.0 Mbps, 101.9±9.7 Mbps, and 151.4±39.7 Mbps. Those cellular traces with low bandwidth are from Vues [67] authors.

Streaming Systems. We compare the fully-fledged Theia with ViVo [38], a baseline system for foveated streaming, and three variations of Theia.

- ViVo: We re-implement ViVo, the state-of-the-art volumetric streaming system, on HoloLens 2.
- Baseline: A basic foveated streaming system that divides point clouds into concentric cones of varying qualities [92].
- Variations of Theia: Theia*(L)* for content generation using only log-polar transformation, Theia*(L+S)* that combines log-polar transformation and peripheral content skip, and Theia*(L+A)* that integrates log-polar transformation and foveal content augmentation.

We do not compare Theia with YuZu [100], M5 [102], and Vues [67], which are all orthogonal to our design and can be used to further enhance its bandwidth savings. The reason is that YuZu leverages a high-performance PC as the client to facilitate super-resolution, and no MR headsets currently support mmWave, which is required by M5. Vues [67] leverages an edge server to pre-transcode volumetric content into 2D streams, which may lead to content drifts of more than 30 cm (due to inaccurate viewport prediction). We re-implement Vues to compare it with direct streaming[4], illustrating that the drifts in Vues negatively affect the QoE.

Videos and Users. We select four videos for performance evaluation: Soldier, Long Dress, Loot, and Matis. We deploy volumetric content at a 1:1 scale, representing avatars true to their real-world sizes. We uniformly sample 10 user traces based on the average gaze prediction errors across the four evaluation videos. Note that Theia and Baseline generate foveated content with the highest point density available in the dataset (*e.g.,* around 1,075K points for the Soldier video) for a fair comparison, while the highest point density for ViVo is limited by the decoding capability of HoloLens 2 at 30 FPS, about 200K points per frame.

Metrics. We evaluate network throughput and end-to-end latency of Theia under both unthrottled WiFi networks and fluctuating/limited bandwidth and monitor the CPU and GPU utilization on the client. For visual quality, we employ two foveation-based metrics: FA-SSIM [84] and EWPSNR [63]. We conduct a user study to evaluate the real-world user experience of Theia.

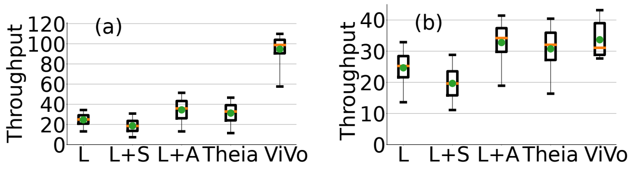


Figure 9: Comparison of network throughput (in Mbps) of ViVo and Theia **under (a) unthrottled WiFi networks and (b) fluctuating/limited bandwidth.**

6.2 Network Throughput

We compare the network throughput of Theia and ViVo under various network conditions using selected videos and user traces. The throughput of Theia is 67.0% (shown in Figure 9(a)) and 9.93% (shown in Figure 9(b)) lower than that of ViVo under unthrottled WiFi networks and fluctuating/limited bandwidth, respectively. The difference is particularly significant under unthrottled WiFi networks because ViVo transmits high-quality point clouds for content that falls into the user's viewport, while Theia delivers content for which only the foveal area has high visual quality. By reducing bandwidth usage, Theia potentially aids in saving energy consumption of mobile devices due to the reduced demands for both data transmission and processing.

We compare Theia with Baseline under unthrottled WiFi networks. Baseline's average throughput is 150±40 Mbps (not shown in Figure 9(a)), which is ~5× that of Theia. Baseline's high bandwidth consumption arises from its design of dividing content into concentric cones with the same quality within each cone which disregards the gradual sensitivity drops in HVS, and failing to remove occluded points which leads to redundant points in content. Conversely, Theia smoothly adjusts visual quality from the fovea to the periphery based on the HVS and takes occlusion optimization into consideration, thus saving substantial bandwidth.

Next, we analyze the impact of each component of Theia on bandwidth saving shown in Figure 9. Comparing Theia*(L)* with Theia*(L+S)*, we find that dynamically skipping the peripheral content based on gaze movements, Theia saves 26.3% and 33.3% bandwidth consumption under unthrottled WiFi networks and fluctuating/limited bandwidth, respectively. We then compare Theia with Theia*(L+A)* and observe a 9.3% and 6.1% reduction under these two conditions. This bandwidth saving is insignificant compared with the previous case because the augmented foveal content has a larger size compared to the skipped content. We finally evaluate the impact of foveal content augmentation by comparing Theia with Theia*(L+S)*, and observe an average of 57.9% increase under both network conditions. Nevertheless, content augmentation is the key component to improving QoE, as it improves the streaming quality of the foveal area. We show the effectiveness of foveal content augmentation in improving user experience in §6.5, by comparing Theia*(L+A)* with Theia*(L)*.

We also perform experiments for scenarios with two to four objects. Our results show that the Theia's throughput does not show significant differences when compared to the single-object scenario. This is because our algorithm is occlusion-aware, meaning

[4] https://www.youtube.com/watch?v=F1TQlYJuUws.

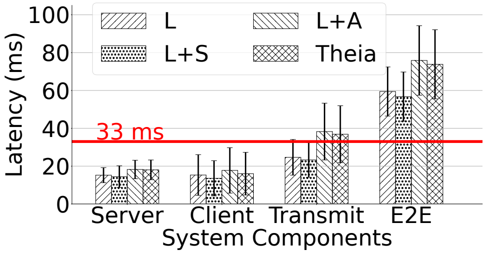


Figure 10: Breakdown of Theia**'s end-to-end latency under fluctuating/limited bandwidth.**

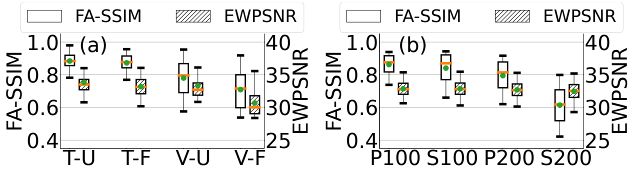


Figure 11: Comparison of FA-SSIM and EWPSNR for (a) Theia **(T) and ViVo (V) under unthrottled WiFi networks (T-U/V-U) and fluctuating/limited bandwidth (T-F/V-F) and (b)** Theia **with both predicted (P) and static (S) gaze with prediction windows of 100 ms (P100/S100) and 200 ms (P200/S200).**

it intelligently identifies and omits the transmission of occluded parts of multiple objects.

6.3 End-to-end Latency

We measure the end-to-end latency of Theia by breaking it down into three components:

- Server-side latency is largely caused by gaze prediction, foveated content generation, foveal content augmentation, peripheral content skip, and point cloud compression.
- Client-side latency encompasses mainly the time taken to decode compressed point clouds.
- The transmission latency arises from data transfer between the server and the client.

Figure 10 shows the overall latency of Theia and its three variations under fluctuating/limited bandwidth. The latency under unthrottled WiFi networks shows similar patterns and is thus omitted. The average latency of Theia's edge and client component is <33 ms, enabling foveated content streaming at 30 FPS. Moreover, the end-to-end latency of Theia is less than 100 ms, ensuring optimal QoE by enabling a small prediction window, which benefits accurate gaze prediction.

To evaluate the impact of the peripheral content skip, we compare the latency of Theia with Theia*(L+A)*, and Theia*(L+S)* with Theia*(L)*. The peripheral content skip can decrease all three components of the end-to-end latency, since it reduces the number of points that need to be compressed and transmitted. Among them, client-side processing latency has the most significant reduction, with a decrease of >10%. This is mainly due to the limited computing power of HoloLens 2, making peripheral content skipping particularly beneficial in this case. Next, we evaluate the impact of foveal content augmentation. Comparing Theia*(L+A)* with Theia*(L)*, foveal content augmentation increases server-side latency by 19.7%, client-side latency by 15.7%, and transmission latency by 58.3%, as the number of generated points grows.

We further perform experiments for scenarios with two to four objects. Our findings indicate that the end-to-end latency of Theia remains consistent compared to the single-object scenario. This is attributed to the server-side's design to utilize GPU for parallel processing. Since the volume of content transmitted in multi-object scenarios does not significantly increase (§6.2), the latency on the client side is also not significantly affected.

We also compare Theia with ViVo and Baseline under unthrottled WiFi networks. ViVo does not require edge support, while Baseline does not lead to considerable differences in server-side latency due to the parallel processing design in both systems. On the other hand, compared to Theia, ViVo and Baseline require the transmission of a larger number of data points, leading to increased transmission latency, around 3× for ViVo and 5× for Baseline. Furthermore, this higher volume of points incurs additional computational latency on the client side for decoding, amounting to around 2.5× for ViVo and 4× for Baseline.

6.4 Visual Quality

We measure the visual quality of Theia with two metrics: FA-SSIM [84] and EWPSNR [63]. Both are designed to evaluate foveated content. They allocate weights to image pixels based on the ground-truth gaze, with weights gradually decreasing from the fovea to the periphery in alignment with the HVS. We render reference images with the original high-fidelity point cloud (*e.g.*, the Soldier video with more than 1M points per frame) and compare them with content rendered by Theia and ViVo to show the metric scores. Note that the highest point density ViVo can handle is ~200K per frame (§6.1).

Visual Quality of Foveated Content. Figure 11(a) illustrates FA-SSIM and EWPSNR for content rendered by Theia and ViVo, evaluated under both unthrottled WiFi networks (T-U/V-U) and fluctuating/limited bandwidth (T-F/V-F). For both network conditions, the average FA-SSIM/EWPSNR of Theia exceeds 0.85/30, indicating good visual quality [21, 25]. Under fluctuating/limited bandwidth, Theia notably outperforms ViVo by 0.164/2.507 in FA-SSIM/EWPSNR. Even under unthrottled WiFi networks, the average FA-SSIM/EWPSNR of Theia outperforms ViVo by 0.105/0.511, and ViVo's FA-SSIM shows a poor visual quality (<0.8 on average) [25]. We also compare the visual quality at varying viewing distances. This comparison aligns with the results presented in Figure 11(a), consistently showing that Theia outperforms ViVo. This is because Theia aims to provide streaming with high-fidelity quality, while ViVo provides a maximum point density of ~200K per frame. Moreover, Theia consumes less bandwidth than ViVo, thus providing better streaming quality under fluctuating/limited bandwidth.

In addition, Theia demonstrates resilience in maintaining visual quality under fluctuating/limited bandwidth by dynamically skipping peripheral content. We vary T_s (§4.5) from 10 to 40, which corresponds to ~18% (T_s=40) to ~50% (T_s=10) gaze data (§3.1), and observe similar visual quality, with FA-SSIM around 0.88 and EWPSNR around 34.5. Consequently, we select T_s as 10, as it skips more peripheral content. Under fluctuating/limited bandwidth, Theia

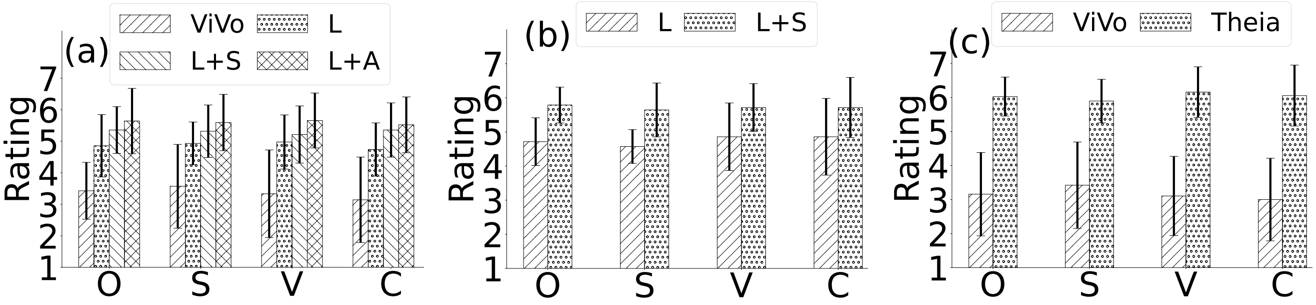


Figure 12: Comparison of ratings of (a): all users performing specific motion patterns, (b): only users with low network bandwidth, and (c): all users freely exploring volumetric content. O: overall experience; S: smoothness; V: visual quality; and C: clarity when moving close to the content. L: content generated with only log-polar transformation; L+S: content generated with log-polar transformation and peripheral content skip; and L+A: content generated with log-polar transformation and foveal content augmentation.

shows a marginal drop of 0.009/0.660 in FA-SSIM/EWPSNR compared to those under unthrottled WiFi networks. This is because by dynamically skipping periphery content, Theia can use saved bandwidth to preserve the visual quality of foveal content. For comparison, ViVo shows a drop of 0.071/2.658 in FA-SSIM/EWPSNR under fluctuating/limited bandwidth. We also expand experiments for multi-object scenarios. As expected (§4.7), the Theia’s visual quality shows similar results compared to the single-object scenario.

Significance of Gaze Prediction. A unique feature of Theia compared to other foveated streaming systems is its incorporation of gaze prediction. To demonstrate the significance of gaze prediction, we evaluate the visual quality of content generated by prediction and static gaze data from the previous frame, particularly during rapid gaze movements (*i.e.,* saccades). As evidenced in Figure 11(b), the content generated with predicted gaze improves the average FA-SSIM/EWPSNR by 0.022/0.050 and 0.180/0.423 when evaluated against the 100 ms and 200 ms prediction windows (an upper bound for the latency that Theia handles in 5G networks [96]) respectively, compared to leveraging static gaze. While static gaze provides fair quality during slow gaze movements, gaze prediction boosts quality for rapid gaze shifts, significantly raising SSIM at the 5th and 25th percentiles, proving its effectiveness in dynamic scenarios. Note that compared to EWPSNR, FA-SSIM considers gaze velocity by assigning higher weights to the foveal area at faster gaze movements. Thus, FA-SSIM is more sensitive to inaccurate gaze prediction, especially during saccades.

6.5 User Study

To evaluate the effectiveness of each component of Theia and the system as a whole from real users’ perspective, we conduct another user study involving 21 participants (Female: 9, Male: 12), with an average age of 24.1±1.7. We ask the participants to wear the HoloLens 2 and watch volumetric videos streamed with ViVo and various combinations of Theia’s components. We randomly select the network trace, video content, and experiment order for each user. Subsequently, we ask the user to perform the following movements when watching the content.

• *Specific Motion Patterns*: To enable participants to view content from diverse angles and distances, we first ask them to execute four movement patterns, as depicted in Figure 13: 1) A left-right movement spanning four meters; 2) A forward and backward movement covering four meters; 3) Circling around the volumetric content with a diameter of two meters; 4) Standing still and tracking a moving object in the video.

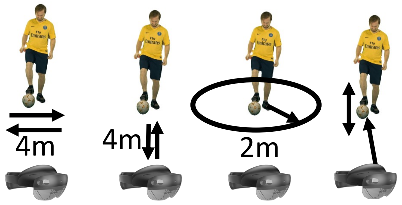


Figure 13: Four movement patterns in the user study.

• *Free-to-explore*: We allow participants to initially freely explore either Theia or ViVo for 30 seconds, based on a random assignment. Then, we ask them to replicate their trajectory for an apple-to-apple comparison between Theia and ViVo when they explore with another system.

Upon completing the tasks, we ask participants to rate their experiences with the 7-point Likert scale (from 1: very bad to 7: very good) [89] from four aspects: 1) overall experience, 2) smoothness, 3) visual quality, and 4) clarity when moving close to the content.

Specific Motion Patterns. Figure 12(a) presents ratings for different systems when performing specific movements. On average, the ratings for Theia*(L)*, Theia*(L+S)*, and Theia*(L+A)* are 45.0%± 5.29%, 58.0%±7.90%, and 66.7%±7.00% higher than those of ViVo, respectively. These results confirm the effectiveness of each component of Theia. In addition, we observe an average improvement of 18.4%±1.21% in the ratings of Theia*(L+A)* compared to Theia*(L)*. Users commented during the interview that, compared to Theia*(L)*, Theia*(L+A)* effectively concealed “black patches” (*i.e.,* visual artifacts), enhancing their experience. This result demonstrates the effectiveness of foveal content augmentation in Theia.

To evaluate the validity of peripheral content skip in Theia, we select ratings from users in low bandwidth situations (7 users, averaged bandwidth: 20.4 Mbps). Our rationale is that Theia can use the bandwidth saved by skipping peripheral content to enhance visual quality of foveal area. As illustrated in Figure 12(b), users' ratings substantiate this point, with the rating of Theia*(L+S)* improving by an average of 12.1%± 2.65% compared to Theia*(L)* for all four items.

Free-to-explore. Figure 12(c) shows ratings for ViVo and Theia when users freely explore the content. We observe that the rating for Theia is 92.5%±7.43% higher than that of ViVo. This is because ViVo can display only the basic shape of the content, particularly under fluctuating/limited bandwidth. In contrast, Theia maintains the capability to render high-fidelity content in the foveal area.

6.6 Energy and Computation Utilization

To profile the energy consumption, we continuously replay video streaming on HoloLens 2 under the unthrottled WiFi network for 1 hour. We start each experiment on the fully-charged device. After the 1-hour experiment, the battery level decreases from 100% to 64% for Theia, and to 55% for ViVo. The average CPU/GPU utilization on HoloLens 2 is 93%/68% for Theia and 96%/85% for ViVo. Compared to ViVo, Theia provides high-quality streaming with fewer points and thus reduces the resource consumption of the client.

7 Discussion

Improving Gaze-prediction Accuracy. While Figures 5 and 6 in §3 show that future gaze can be accurately predicted when consuming volumetric content on MR headsets, the prediction accuracy could be further improved in two possible directions. (1) By accurately predicting saccade landing positions [13, 14, 76], we can model the smooth transitions between fixations, resulting in more natural and realistic gaze predictions. (2) Motivated by the success of enhancing viewport-prediction accuracy for 360° video streaming by leveraging saliency maps of panoramic frames [30], Theia can potentially make gaze prediction more accurate with saliency maps created from point clouds [91, 104]. However, computing saliency maps for point clouds can not only be computationally expensive but also make the model complicated, a significant challenge when real-time prediction is required.

Privacy. Theia uploads viewers' gaze trajectories to the edge server to enable foveated streaming. This may raise privacy concerns as gaze data may reveal personal information (*e.g.,* gender, age, and ethnicity) [64]. While we can protect gaze data through differential privacy technology [43, 59], randomized encoding [19], additive noises, and temporal and spatial downsampling [26], there is a tradeoff between gaze-prediction accuracy, streaming efficiency, and privacy protection, which is part of our future work.

Quality Assessment of Volumetric Content Delivery. While subjective assessment could better reflect the user experience of video streaming than the objective counterpart, it is not only time-consuming but also costly. On the other hand, objective quality assessment of volumetric content delivery is still in its early stage, not even to mention its foveated version. We plan to develop deep-learning-based quality metrics [77] in our future work.

8 Related Work

Volumetric Content Delivery. There is plenty of work on improving the QoE of streaming volumetric content [23, 38, 56, 67, 100, 102]. Early work aimed to reduce mobile data usage by leveraging visibility-aware optimizations (*e.g.,* ViVo [38]) and to accelerate point-cloud decompression (*e.g.,* GROOT [56]). Recent efforts include Vues [67] that transcodes a point cloud into multiple 2D views, YuZu [100], enhancing the super-resolution for volumetric video streaming, and M5 [102], utilizing 6DoF motion prediction to adapt mmWave beams for multi-user streaming. Those methods ignore gaze, which Theia leverages to enable foveated streaming.

Foveated Streaming. Existing work on foveated streaming focuses on VR content [41, 58, 69]. There are only a few works on foveated streaming for point clouds [73, 92]. Tefera *et al.* [92] divide 3D space into concentric areas, resulting in unsmooth quality changes. Meng *et al.* [73] investigate the streaming of static point clouds. None of the above works considers gaze prediction, which we leverage to boost the QoE of volumetric content delivery.

Perception-aware Video Processing. There is plenty of work leveraging human perception to enhance video quality and user experience from various angles, including developing visual quality metrics [65], identifying factors impacting viewing experience [98], and proposing encoding/compression methods to reduce network overhead [51]. For example, Pano [35] takes advantage of quality perception to optimize 360° video streaming while maintaining QoE. Different from the above work, Theia benefits from eye tracking of MR headsets to design a gaze-driven content delivery system.

Gaze Prediction is a broad concept in the literature, which refers to not only the prediction of future gaze with historical information [46, 97] (§3) but also the estimation of current gaze from various sources [44, 52, 62, 85]. In contrast, we aim to build a real-time and high-precision gaze prediction model that utilizes only historical gaze motion for foveated volumetric content delivery.

Gaze-aware Applications. Besides foveated rendering and streaming, eye-gaze information is beneficial for other applications such as reduction of page load time for improving user experience [50], optimization of energy efficiency for processing holograms [103], and interaction with remote robots for wheelchair users [39].

9 Conclusion

In this paper, we presented the design, implementation, and evaluation of Theia, a novel volumetric content delivery system for MR headsets that benefits from their eye-tracking capabilities and unique features of human perception to reduce bandwidth consumption and boost QoE. By devising efficient methods for real-time foveated content generation, lightweight foveal content augmentation, and dynamic peripheral content skip, Theia significantly outperforms the state-of-the-art, demonstrated by our extensive performance evaluation. We hope our study can stimulate novel mobile MR applications that take advantage of volumetric content.

Acknowledgments

We thank the anonymous reviewers and our shepherd Chenren Xu for their insightful comments, and the user study participants for their valuable contributions. This work was partially supported by the U.S. National Science Foundation, under Grants CNS-2212296 and CNS-2235049.

Artifact Appendix

The research artifacts accompanying this paper are available via https://doi.org/10.5281/zenodo.11095706.

References

[1] 2017. 8i Voxelized Full Bodies (8iVFB v2) - Dynamic Voxelized Point Cloud Dataset. http://plenodb.jpeg.org/pc/8ilabs. [accessed on 11/30/2023].

[2] 2019. Microsoft HoloLens 2. https://www.microsoft.com/en-us/hololens. [accessed on 11/30/2023].

[3] 2023. About HoloLens 2. https://learn.microsoft.com/en-us/hololens/hololens2-hardware. [accessed on 11/30/2023].

[4] 2023. CUDA Toolkit. https://developer.nvidia.com/cuda-toolkit. [accessed on 11/30/2023].

[5] 2023. Draco 3D Data Compression. https://google.github.io/draco/. [accessed on 11/30/2023].

[6] 2023. Extended Eye Tracking in Native Engine. https://learn.microsoft.com/en-us/windows/mixed-reality/develop/native/extended-eye-tracking-native. [accessed on 11/30/2023].

[7] 2023. Eye Tracking on HoloLens 2. https://docs.microsoft.com/en-us/windows/mixed-reality/eye-tracking. [accessed on 11/30/2023].

[8] 2023. Magic Leap 2. https://www.magicleap.com/magic-leap-2. [accessed on 11/30/2023].

[9] 2023. tc(8) - Linux man page. https://linux.die.net/man/8/tc. [accessed on 11/30/2023].

[10] 2023. Unreal Engine. https://www.unrealengine.com. [accessed on 11/30/2023].

[11] 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

[12] Rachel Albert, Anjul Patney, David Luebke, and Joohwan Kim. 2017. Latency Requirements for Foveated Rendering in Virtual Reality. *ACM Transactions on Applied Perception* 14, 4 (2017), 1–13. https://doi.org/10.1145/3127589

[13] Elena Arabadzhiyska, Cara Tursun, Hans-Peter Seidel, and Piotr Didyk. 2023. Practical Saccade Prediction for Head-Mounted Displays: Towards a Comprehensive Model. *ACM Transactions on Applied Perception* 20, 1 (2023), 1–23. https://doi.org/10.1145/3568311

[14] Elena Arabadzhiyska, Okan Tarhan Tursun, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. 2017. Saccade Landing Position Prediction for Gaze-Contingent Rendering. *ACM Transactions on Graphics* 36, 4 (2017), 1–12. https://doi.org/10.1145/3072959.3073642

[15] Helder Araujo and Jorge M Dias. 1996. An Introduction to the Log-polar Mapping [Image Sampling]. In *Proceedings of II Workshop on Cybernetic Vision*. https://doi.org/10.1109/CYBVIS.1996.629454

[16] Paramvir Bahl and Venkata N Padmanabhan. 2000. RADAR: An In-Building RF-based User Location and Tracking System. In *Proceedings of IEEE INFOCOM*. https://doi.org/10.1109/INFCOM.2000.832252

[17] Behnam Bastani, Eric Turner, Carlin Vieri, Haomiao Jiang, Brian Funt, and Nikhil Balram. 2017. Foveated Pipeline for AR/VR Head-Mounted Displays. *Information Display* 33, 6 (2017), 14–35. https://doi.org/10.1002/j.2637-496X.2017.tb01040.x

[18] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. 2001. Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting . In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2001.990497

[19] Efe Bozkir, Ali Burak Ünal, Mete Akgün, Enkelejda Kasneci, and Nico Pfeifer. 2020. Privacy Preserving Gaze Estimation Using Synthetic Images via a Randomized Encoding Based Framework. In *Proceedings of ACM Symposium on Eye Tracking Research and Applications (ETRA)*. https://doi.org/10.1145/3379156.3391364

[20] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[21] David R. Bull and Fan Zhang. 2021. Chapter 4 - Digital Picture Formats and Representations. In *Intelligent Image and Video Compression (Second Edition)*. Academic Press, Oxford, 107–142. https://doi.org/10.1016/B978-0-12-820353-8.00013-X

[22] 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

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

[24] Alexander Clemm, Maria Torres Vega, Hemanth Kumar Ravuri, Tim Wauters, and Filip De Turck. 2020. Toward Truly Immersive Holographic-type Communication: Challenges and Solutions. *IEEE Communications Magazine* 58, 1 (2020), 93–99. https://doi.org/10.1109/MCOM.001.1900272

[25] 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

[26] Brendan David-John, Diane Hosfelt, Kevin Butler, and Eakta Jain. 2021. A Privacy-Preserving Approach to Streaming Eye-Tracking Data. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2555–2565. https://doi.org/10.1109/TVCG.2021.3067787

[27] Leila De Floriani, Franco Morando, and Enrico Puppo. 2003. Representation of Non-Manifold Objects. In *Proceedings of ACM Symposium on Solid Modeling and Applications*. https://doi.org/10.1145/781606.781656

[28] Mark R Diamond, John Ross, and Maria C Morrone. 2000. Extraretinal Control of Saccadic Suppression. *Journal of Neuroscience* 20, 9 (2000), 3449–3455. https://doi.org/10.1523/JNEUROSCI.20-09-03449.2000

[29] Omar Elharrouss, Noor Almaadeed, Somaya Al-Maadeed, and Younes Akbari. 2020. Image Inpainting: A Review. *Neural Processing Letters* 51 (2020), 2007–2028.

[30] Ching-Ling Fan, Jean Lee, Wen-Chih Lo, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2017. Fixation Prediction for 360° Video Streaming in Head-Mounted Virtual Reality. In *Proceedings of ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*. https://doi.org/10.1145/3083165.3083180

[31] Yu Fang, Ryoichi Nakashima, Kazumichi Matsumiya, Ichiro Kuriki, and Satoshi Shioiri. 2015. Eye-Head Coordination for Visual Cognitive Processing. *PLOS ONE* 10, 3 (2015), e0121035. https://doi.org/10.1371/journal.pone.0121035

[32] Clarence Errol Ferree, Gertrude Rand, and C Hardy. 1931. Refraction For The Peripheral Field of Vision. *Archives of Ophthalmology* 5, 5 (1931), 717–731. https://doi.org/10.1001/archopht.1931.00820050039003

[33] James D Foley. 1996. Computer Graphics: Principles and Practice. In *Intelligent Image and Video Compression (Second Edition)*. Addison-Wesley Professional.

[34] 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 the International Conference on Mobile Computing and Networking*.

[35] Yu Guan, Chengyuan Zheng, Xinggong Zhang, Zongming Guo, and Junchen Jiang. 2019. Pano: Optimizing 360 Video Streaming with a Better Understanding of Quality Perception. In *Proceedings of ACM SIGCOMM*. https://doi.org/10.1145/3341302.3342063

[36] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D Graphics. *ACM Transactions on Graphics* 31, 6 (2012), 1–10. https://doi.org/10.1145/2366145.2366183

[37] Serhan Gül, Dimitri Podborski, Jangwoo Son, Gurdeep Singh Bhullar, Thomas Buchholz, Thomas Schierl, and Cornelius Hellge. 2020. Cloud Rendering-based Volumetric Video Streaming System for Mixed Reality Services. In *Proceedings of ACM MMSys*. https://doi.org/10.1145/3339825.3393583

[38] Bo Han, Yu Liu, and Feng Qian. 2020. ViVo: Visibility-Aware Mobile Volumetric Video Streaming. In *Proceedings of ACM MobiCom*. https://doi.org/10.1145/3372224.3380888

[39] John Paulin Hansen, Alexandre Alapetite, Martin Thomsen, Zhongyu Wang, Katsumi Minakata, and Guangtao Zhang. 2018. Head and Gaze Control of a Telepresence Robot with an HMD. In *Proceedings of ACM Symposium on Eye Tracking Research & Applications (ETRA)*. https://doi.org/10.1145/3204493.3208330

[40] Joy Hirsch and Christine A Curcio. 1989. The Spatial Resolution Capacity of Human Foveal Retina. *Vision Research* 29, 9 (1989), 1095–1101. https://doi.org/10.1016/0042-6989(89)90058-8

[41] Luke Hsiao, Brooke Krajancich, Philip Levis, Gordon Wetzstein, and Keith Winstein. 2022. Towards Retina-Quality VR Video Streaming: 15ms Could Save You 80% of Your Bandwidth. *ACM SIGCOMM Computer Communication Review* 52, 1 (2022), 10–19. https://doi.org/10.1145/3523230.3523233

[42] Chih-Fan Hsu, Anthony Chen, Cheng-Hsin Hsu, Chun-Ying Huang, Chin-Laung Lei, and Kuan-Ta Chen. 2017. Is Foveated Rendering Perceivable in Virtual Reality?: Exploring the Efficiency and Consistency of Quality Assessment Methods. In *Proceedings of ACM International Conference on Multimedia*. https://doi.org/10.1145/3123266.3123434

[43] Miao Hu, Zhenxiao Luo, Yipeng Zhou, Xuezheng Liu, and Di Wu. 2022. Otus: A Gaze Model-based Privacy Control Framework for Eye Tracking Applications. In *Proceedings of IEEE INFOCOM*. https://doi.org/10.1109/INFOCOM48880.2022.9796665

[44] Zhiming Hu, Sheng Li, Congyi Zhang, Kangrui Yi, Guoping Wang, and Dinesh Manocha. 2020. DGaze: CNN-Based Gaze Prediction in Dynamic Scenes. *IEEE Transactions on Visualization and Computer Graphics* 26, 5 (2020), 1902–1911. https://doi.org/10.1109/TVCG.2020.2973473

[45] Zhiming Hu, Congyi Zhang, Sheng Li, Guoping Wang, and Dinesh Manocha. 2019. SGaze: A Data-Driven Eye-Head Coordination Model for Realtime Gaze Prediction. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (2019), 32002–2010. https://doi.org/10.1109/TVCG.2019.2899187

[46] Gazi Karam Illahi, Matti Siekkinen, Teemu Kämäräinen, and Antti Ylä-Jääski. 2022. Real-time Gaze Prediction in Virtual Reality. In *Proceedings of ACM International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE)*. https://doi.org/10.1145/3534086.3534331

[47] Alireza Javaheri, Catarina Brites, Fernando Pereira, and Joao Ascenso. 2020. Point Cloud Rendering After Coding: Impacts on Subjective and Objective Quality. *IEEE Transactions on Multimedia* 23 (2020), 4049–4064. https://doi.org/10.1109/TMM.2020.3037481

[48] Junchen Jiang, Vyas Sekar, and Hui Zhang. 2012. Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE. In *Proceedings of ACM CoNEXT*. https://doi.org/10.1145/2413176.2413189

[49] Donald H Kelly. 1979. Motion and vision. II. Stabilized spatio-temporal threshold surface. *Journal of the Optical Society of America* 69, 10 (1979), 1340–1349. https://doi.org/10.1364/JOSA.69.001340

[50] Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir Das. 2017. Improving User Perceived Page Load Time Using Gaze. In *Proceedings of USENIX NSDI*. https://dl.acm.org/doi/10.5555/3154630.3154675

[51] Sehwan Ki, Sung-Ho Bae, Munchurl Kim, and Hyunsuk Ko. 2018. Learning-Based Just-Noticeable-Quantization- Distortion Modeling for Perceptual Video Coding. *IEEE Transactions on Image Processing* 27, 7 (2018), 3178–3193. https://doi.org/10.1109/TIP.2018.2818439

[52] Joohwan Kim, Michael Stengel, Alexander Majercik, Shalini De Mello, David Dunn, Samuli Laine, Morgan McGuire, and David Luebke. 2019. NVGaze: An Anatomically-Informed Dataset for Low-Latency, Near-Eye Gaze Estimation. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*. https://doi.org/10.1145/3290605.3300780

[53] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. https://arxiv.org/abs/1412.6980. [accessed on 11/30/2023].

[54] Earl J Kirkland and Earl J Kirkland. 2010. Bilinear Interpolation. *Advanced Computing in Electron Microscopy* (2010), 261–263.

[55] Petrus EJ Kivi, Markku J Mäkitalo, Jakub Žádník, Julius Ikkala, Vinod Kumar Malamal Vadakital, and Pekka O Jääskeläinen. 2022. Real-Time Rendering of Point Clouds With Photorealistic Effects: A Survey. *IEEE Access* 10 (2022), 13151–13173. https://doi.org/10.1109/ACCESS.2022.3146768

[56] 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

[57] Marc Levoy and Ross Whitaker. 1990. Gaze-directed Volume Rendering. In *Proceedings of Symposium on Interactive 3D Graphics (I3D)*. https://doi.org/10.1145/91385.91449

[58] David Li, Ruofei Du, Adharsh Babu, Camelia D Brumar, and Amitabh Varshney. 2021. A Log-Rectilinear Transformation for Foveated 360-degree Video Streaming. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2638–2647. https://doi.org/10.1109/TVCG.2021.3067762

[59] Jingjie Li, Amrita Roy Chowdhury, Kassem Fawaz, and Younghyun Kim. 2021. Kalεido: Real-time Privacy Control for Eye-tracking Systems. In *Proceedings of USENIX Security Symposium*. https://www.usenix.org/conference/usenixsecurity21/presentation/li-jingjie

[60] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. 2019. PU-GAN: A Point Cloud Upsampling Adversarial Network. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2019.00730

[61] Tianxing Li, Qiang Liu, and Xia Zhou. 2017. Ultra-Low Power Gaze Tracking for Virtual Reality. In *Proceedings of ACM SenSys*. https://doi.org/10.1145/3131672.3131682

[62] Tianxing Li and Xia Zhou. 2018. Battery-Free Eye Tracker on Glasses. In *Proceedings of ACM MobiCom*. https://doi.org/10.1145/3241539.3241578

[63] Zhicheng Li, Shiyin Qin, and Laurent Itti. 2011. Visual Attention Guided Bit Allocation in Video Compression. *Image and Vision Computing* 29, 1 (2011), 1–14. https://doi.org/10.1016/j.imavis.2010.07.001

[64] Daniel J Liebling and Sören Preibusch. 2014. Privacy Considerations for a Pervasive Eye Tracking World. In *Proceedings of ACM UbiComp*. https://doi.org/10.1145/2638728.2641688

[65] Weisi Lin and C.-C. Jay Kuo. 2011. Perceptual visual quality metrics: A survey. *Journal of Visual Communication and Image Representation* 22, 4 (2011), 297–312. https://doi.org/10.1016/j.jvcir.2011.01.005

[66] 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)*.

[67] 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

[68] Lester C Loschky and Gary S Wolverton. 2007. How Late Can You Update Gaze-contingent Multiresolutional Displays Without Detection? *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 3, 4 (2007), 1–10. https://doi.org/10.1145/1314303.1314310

[69] Pietro Lungaro, Rickard Sjöberg, Alfredo José Fanghella Valero, Ashutosh Mittal, and Konrad Tollmar. 2018. Gaze-Aware Streaming Solutions for the Next Generation of Mobile VR Experiences. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1535–1544. https://doi.org/10.1109/TVCG.2018.2794119

[70] Adrien Maglo, Guillaume Lavoué, Florent Dupont, and Céline Hudelot. 2015. 3D Mesh Compression: Survey, Comparisons, and Emerging Trends. *ACM Computing Surveys* 47, 3 (2015). https://doi.org/10.1145/2693443

[71] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In *Proceedings of ACM SIGCOMM*. https://dl.acm.org/doi/10.1145/3098822.3098843

[72] Rachel McAfee, Cole Haxton, Matthew Harrison, and Joshua Gess. 2020. Thermal Characterization of a Virtual Reality Headset during Transient and Resting Operation. In *Proceedings of Semiconductor Thermal Measurement, Modeling & Management Symposium (SEMI-THERM)*. https://doi.org/10.23919/SEMI-THERM50369.2020.9142850

[73] Fang Meng and Hongbin Zha. 2003. Efficient Streaming of Point-based Models for Interactive 3-D Geometry Transmission. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*. https://doi.org/10.1109/CIRA.2003.1222130

[74] Xiaoxu Meng, Ruofei Du, Matthias Zwicker, and Amitabh Varshney. 2018. Kernel Foveated Rendering. In *Proceedings of ACM on Computer Graphics and Interactive Techniques*. https://doi.org/10.1145/3203199

[75] Zili Meng, Tingfeng Wang, Yixin Shen, Bo Wang, Mingwei Xu, Rui Han, Honghao Liu, Venkat Arun, Hongxin Hu, and Xue Wei. 2023. Enabling High Quality Real-Time Communications with Adaptive Frame-Rate. In *Proceedings of USENIX NSDI*. https://www.usenix.org/conference/nsdi23/presentation/meng

[76] Aythami Morales, Francisco M Costela, and Russell L Woods. 2021. Saccade Landing Point Prediction Based on Fine-Grained Learning Method. *IEEE Access* 9 (2021), 52474–52484. https://doi.org/10.1109/ACCESS.2021.3070511

[77] Yana Nehmé, Johanna Delanoy, Florent Dupont, Jean-Philippe Farrugia, Patrick Le Callet, and Guillaume Lavoué. 2023. Textured Mesh Quality Assessment: Large-Scale Dataset and Deep Learning-based Quality Metric. *ACM Transactions on Graphics* (2023). https://doi.org/10.1145/3592786

[78] 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)*. https://doi.org/10.1145/2984511.2984517

[79] Rafael Pagés, Konstantinos Amplianitis, Jan Ondrej, Emin Zerman, and Aljosa Smolic. 2022. Volograms & V-SENSE Volumetric Video Dataset. (2022). https://doi.org/10.13140/RG.2.2.24235.31529/1

[80] Rafael Pagés, Emin Zerman, Konstantinos Amplianitis, Jan Ondřej, and Aljosa Smolic. 2021. Volograms & V-SENSE Volumetric Video Dataset. *ISO/IEC JTC1/SC29/WG07 MPEG2021/m56767* (2021).

[81] Jounsup Park, Philip A. Chou, and Jenq-Neng Hwang. 2019. Rate-Utility Optimized Streaming of Volumetric Media for Augmented Reality. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9, 1 (2019), 149–162. https://doi.org/10.1109/JETCAS.2019.2898622

[82] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of NeurIPS*. https://dl.acm.org/doi/10.5555/3454287.3455008

[83] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. 2016. Towards Foveated Rendering for Gaze-tracked Virtual Reality. *ACM Transactions on Graphics* 35, 6 (2016), 179:1–179:12. https://doi.org/10.1145/2980179.2980246

[84] Snježana Rimac-Drlje, Goran Martinović, and Branka Zovko-Cihlar. 2011. Foveation-based Content Adaptive Structural Similarity Index. In *Proceedings of International Conference on Systems, Signals and Image Processing*.

[85] Jihoon Ryoo, Kiwon Yun, Dimitris Samaras, Samir R Das, and Gregory Zelinsky. 2016. Design and evaluation of a foveated video streaming service for commodity client devices. In *Proceedings of ACM MMSys*. https://doi.org/10.1145/2910017.2910592

[86] William Sentosa, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Haitham Hassanieh, and Bruce Maggs. 2023. DChannel: Accelerating Mobile Applications With Parallel High-bandwidth and Low-latency Channels. In *Proceedings of USENIX NSDI*. https://www.usenix.org/conference/nsdi23/presentation/sentosa

[87] Hans Strasburger, Ingo Rentschler, and Martin Jüttner. 2011. Peripheral Vision and Pattern Recognition: A Review. *Journal of Vision* 11, 5 (2011), 13–13. https://doi.org/10.1167/11.5.13

[88] 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

[89] 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

[90] 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. https://doi.org/10.1109/MWC.001.1900488

[91] Flora Ponjou Tasse, Jiri Kosinka, and Neil Dodgson. 2015. Cluster-Based Point Set Saliency. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2015.27

[92] Y Tefera, Dario Mazzanti, Sara Anastasi, Darwin Caldwell, Paolo Fiorini, and Nikhil Deshpande. 2022. Towards Foveated Rendering For Immersive Remote Telerobotics. In *Proceedings of the International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions at HRI*. https://hal.science/hal-03610377/file/foveated.pdf

[93] Graham Upton and Ian Cook. 2014. *A Dictionary of Statistics*. Oxford University Press, USA.

[94] Christian J Van den Branden Lambrecht and Olivier Verscheure. 1996. Perceptual Quality Measure Using a Spatiotemporal Model of the Human Visual System. In *Proceedings of Digital Video Compression: Algorithms and Technologies*. https://doi.org/10.1117/12.235440

[95] Frances C Volkmann, Lorrin A Riggs, Keith D White, and Robert K Moore. 1978. Contrast Sensitivity during Saccadic Eye Movements. *Vision Research* 18, 9 (1978), 1193–1199. https://doi.org/10.1016/0042-6989(78)90104-9

[96] Dongzhu Xu, Anfu Zhou, Xinyu Zhang, Guixian Wang, Xi Liu, Congkai An, Yiming Shi, Liang Liu, and Huadong Ma. 2020. Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption. In *Proceedings of ACM SIGCOMM*. 479–494. https://doi.org/10.1145/3387514.3405882

[97] Yanyu Xu, Yanbing Dong, Junru Wu, Zhengzhong Sun, Zhiru Shi, Jingyi Yu, and Shenghua Gao. 2018. Gaze Prediction in Dynamic 360° Immersive Videos. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR.2018.00559

[98] Jingteng Xue and Chang Wen Chen. 2014. Mobile Video Perception: New Insights and Adaptation Strategies. *IEEE Journal of Selected Topics in Signal Processing* 8, 3 (2014), 390–401. https://doi.org/10.1109/JSTSP.2014.2313456

[99] Hyunho Yeo, Chan Ju Chong, Youngmok Jung, Juncheol Ye, and Dongsu Han. 2020. NEMO: Enabling Neural-enhanced Video Streaming on Commodity Mobile Devices. In *Proceedings of ACM MobiCom*. https://doi.org/10.1145/3372224.3419185

[100] Anlan Zhang, Chendong Wang, Bo Han, and Feng Qian. 2022. YuZu: Neural-enhanced Volumetric Video Streaming. In *Proceedings of USENIX NSDI*. https://www.usenix.org/conference/nsdi22/presentation/zhang-anlan

[101] Ding Zhang, Bo Han, Parth Pathak, and Haoliang Wang. 2021. Innovating Multi-user Volumetric Video Streaming through Cross-layer Design. In *Proceedings of ACM HotNets*.

[102] 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

[103] Shulin Zhao, Haibo Zhang, Cyan Subhra Mishra, Sandeepa Bhuyan, Ziyu Ying, Mahmut Taylan Kandemir, Anand Sivasubramaniam, and Chita Das. 2021. HoloAR: On-the-Fly Optimization of 3D Holographic Processing for Augmented Reality. In *Proceedings of Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. https://doi.org/10.1145/3466752.3480056

[104] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. 2019. PointCloud Saliency Maps. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*. 1598–1606. https://doi.org/10.1109/ICCV.2019.00168