Virtual Reality Telepresence: 360-Degree Video Streaming with Edge-Compute Assisted Static Foveated Compression

Xincheng Huang, James Riddell, Robert Xiao

(a) (b) (c)

Fig. 1: Example setup of virtual reality telepresence using our system: a) The local user uses a ∼6K 360° camera to stream their environment. b) The remote guest wears a VR headset to receive the real-time video from the local user streamed by our system c) The remote user is telepresent in the local environment with a first-person perspective.

Abstract—Real-time communication with immersive 360° video can enable users to be telepresent within a remotely streamed environment. Increasingly, users are shifting to mobile devices and connecting to the Internet via mobile-cellular networks. As the ideal media for 360° videos, some VR headsets now also come with cellular capacity, giving them potential for mobile applications. However, streaming high-quality 360° live video poses challenges for network bandwidth, particularly on cellular connections. To reduce bandwidth requirements, videos can be compressed using viewport-adaptive streaming or foveated rendering techniques. Such approaches require very low latency in order to be effective, which has previously limited their applications on traditional cellular networks. In this work, we demonstrate an end-to-end virtual reality telepresence system that streams ∼6K 360° video over 5G millimeter-wave (mmW) radio. Our use of 5G technologies, in conjunction with mobile edge compute nodes, substantially reduces latency when compared with existing 4G networks, enabling high-efficiency foveated compression over modern cellular networks on par with WiFi. We performed a technical evaluation of our system's visual quality post-compression with peak signal-to-noise ratio (PSNR) and FOVVideoVDP. We also conducted a user study to evaluate users' sensitivity to compressed video. Our findings demonstrate that our system achieves visually indistinguishable video streams while using up to 80% less data when compared with un-foveated video. We demonstrate our video compression system in the context of an immersive, telepresent video calling application.

Index Terms—360-Degree Video, Virtual Reality, Telepresence

1 Introduction

360° images and videos allow users to immersively explore a remote environment with omni-directional freedom. As such, they are natural applications for the rising virtual reality (VR) technologies. Streaming on-demand 360° videos in VR headsets have enabled rich multi-media applications in gaming and immersive entertainment [5, 9]. We expect the next frontier in 360° videos and virtual reality is to "cut the cords" and enable live and mobile applications. The ability to provide high-quality, live 360° videos in VR anytime and anywhere can bring about more immediate telepresence experiences, enabling efficient collaboration and education over long distances [28, 38, 46, 47]. Towards this direction, there have been industrial products combining 5G technology with VR devices [3], extending their network bandwidth and making them more suitable for mobile applications.

360° videos require a much higher resolution than regular videos to achieve a satisfactory perceived visual quality in VR headsets, which poses a challenge in video processing and network bandwidth. Before the 5G era, prior research has investigated foveated compression techniques [12, 13, 18, 24] including tile-based viewport adaptation [29,39,50] and foveated rendering [41,42,49] to save network bandwidth. While tile-based viewport adaptation prioritizes streaming video tiles within a user's field of view (FOV), foveated rendering streams high-resolution video in the focal area while heavily compressing the remaining pixels. However, both methods require additional techniques to mitigate delayed video tile delivery and video artifacts [19, 27] caused by latency in conventional 4G/LTE networks [4]. Recent research has explored 360° video streaming with 5G networks [23, 25], utilizing its high bandwidth and low network latency. However, they are evaluated in simulated network environments without a user-centered perspective.

In this work, we implemented an end-to-end VR telepresence system that streams live 6K H.264 360° videos with foveated compression using 5G millimeter wave (mmW). Our approach is a static foveation technique [1, 31] which uses head tracking. While it is straightforward to use eye-tracking with our system, head-tracking is more widely compatible with most deployed VR systems [43].

We implemented video processing and foveation on a 5G multi-access edge computing (MEC) server to reduce the local processing overhead and latency. With our system, a user can stream their environment, via a commodity 6K 360° video camera, to another user wearing a VR headset. The VR headset runs a client application that remaps the dual fisheye 360° videos to the skybox of the scene. The MEC server receives live updates of the user's head position and re-encodes the video stream with foveated compression.

- *Xincheng Huang is with the University of British Columbia. E-mail: xchuang@cs.ubc.ca*
- *James Riddell is with the University of British Columbia. E-mail: riddell6@student.ubc.ca*
- *Robert Xiao is with the University of British Columbia. E-mail: brx@cs.ubc.ca*

We evaluated how using MEC-assisted 5G mmW and static foveation affects the system performance with measurements of latency and visual quality. Specifically, we measured the latency between a head movement and a corresponding foveated frame, and tested the visual quality of the video with varying levels of foveated compression. Our results show that 5G mmW reduces the network latency to that of wired Ethernet, providing a substantial reduction compared to conventional 4G/LTE networks. Additionally, we found that foveated rendering provides higher visual quality given the same bitrates. Our work demonstrates that foveated compression of high-resolution 360° video over 5G networks is imminently practical, enabling cellular networks to carry immersive, live, and telepresent virtual reality video.

To identify how these improvements in system performance are reflected in actual user experiences, we conducted two tests of just noticeable differences (JND) on the size of focal areas and visual quality (i.e., as a result of different bitrates), finding that our system's low latency and rapid response to head motion make it feasible to set a focal area that is close to the VR headset's FOV. Meanwhile, the results of just noticeable visual degradation show that our system can reduce bitrate by 80% while achieving the same perceived visual quality. In addition to the JND tests, we instructed the participants to conduct an end-to-end video conference with the system, and gathered qualitative results on potential interactive techniques and applications.

We contribute 1) a real-life implementation of an end-to-end 360° Video VR telepresence system with MEC-assisted 5G millimeter-wave, and 2) a set of system measurements and user experiments that evaluate such a system from a user-centric perspective.

2 Related Work

360° videos have found applications in telepresence [5] and remote collaboration [46]. Here we review the state of the art of 360° telepresence, as well as the technologies (e.g., foveated compression and rendering [27, 33, 39], MEC-Assisted millimeter-wave 5G [21, 30]) that made streaming high-quality 360° videos more feasible.

2.1 360-Degree Video Telepresence and Collaboration

360° images and videos provide users with an immersive viewing experience by allowing them to explore the captured space omnidirectionally. This feature makes 360° videos ideal for telepresence [5]. Prior research has extended 360° videos to remote instruction and collaboration [46]. Such collaboration systems often feature an asymmetric setup, where one user captures a live video with a 360° camera for another user to view in a mixed-reality device [20,28,38,47]. For example, Piumsomboon et al. [38] used 360° video streaming to create a miniature virtual presence for a remote user to facilitate multi-scale remote instruction. OmniGlobeVR [28] demonstrated that 360° videos can also be streamed to spherical globes to be simultaneously viewed by multiple collaborators from a third-person perspective.

However, streaming high-quality 360° videos is challenging. When watching 360° videos in VR headsets, users only view a portion that corresponds to the headset's FoV. Therefore, to achieve a satisfactory perceived visual quality, the 360° videos need to be streamed in a resolution that is much higher than the VR headset's resolution [27, 39, 40], which poses a challenge to network bandwidth and video processing.

2.2 Foveated Compression and Rendering

Foveated compression [12, 13, 18, 24] saves network bandwidth and processing overhead for high-quality visual content by dynamically adapting to a user's viewport or focal area. One common approach is tile-based foveated compression [16, 17, 29, 34, 37, 48, 50]. Such approaches separate a 360° video into multiple tiles and prioritize streaming the tiles that are within a user's focal area or field of view. However, in conventional 4G LTE networks, such an approach can suffer from delayed delivery of requested tiles caused by network latency [4]. To mitigate this, prior research developed viewport prediction algorithms [11, 34, 39, 40] to pre-fetch video tiles that a user is likely to watch. Most viewport prediction algorithms are trained based on the history of user viewing trajectories, making them unsuitable for live streaming [11]. Prior research [11, 39] has also introduced viewport prediction algorithms based on the past few seconds of a user's viewing trajectory, but they are subject to prediction errors.

Instead of breaking a video into tiles, foveated rendering [12, 13, 22, 33, 41, 42, 49] dynamically controls the compression rate of the entire video. For such techniques, the foveated region can be dynamic or static [1, 31]. In dynamic foveated compression, the foveated region corresponds to the user's gaze position and thus requires a low-latency eye-tracker [43]. In contrast, static foveation has a fixed foveated region at the center of the display and thus only relies on head tracking. Foveated compression saves network bandwidth by heavily compressing content that is outside of the viewer's focal area.

However, foveated compression also suffers from both processing and network latency [4], which leads to video artifacts showing up in the peripheral area of a user's field of view. Research has proposed methods such as Log-Rectilinear transformation [27] and neural reconstruction methods [19] to deal with latency-inflicted video artifacts. However, such methods introduce additional computational overhead to the video processing pipeline. The rise of 5G networks and MEC-assisted computing bring extended mobile network bandwidth and inherently lower latency, potentially alleviating latency-inflicted video artifacts in foveated rendering. Therefore, we believe it is worth re-evaluating the problem space of foveated 360° video streaming in the 5G context.

2.3 MEC-Assisted 5G Millimeter Wave Video Streaming

Millimeter wave (mmW) 5G is able to bring mobile network bandwidth to 1Gbps with a low latency [14]. Prior research has explored the potential of providing multimedia services with 5G networks [8, 10, 36] for applications such as ultra-realistic VR experiences [7] and gaming arenas [9]. Millimeter wave 5G is often coupled with Multi-Access Edge Computing (MEC) due to its highly distributed nature. The ability to offload computation to MEC servers can further reduce the computational overhead at end clients [21, 51]. Meanwhile, as MEC servers are typically situated near the downstream link in the carrier's network, MEC-assisted network pipelines can minimize the latency to the client. Researchers have explored panoramic video streaming systems with MEC assistance [15, 23, 25, 26, 30, 45]. However, they only evaluated their system performance in simulated network environments. Therefore, it is necessary to implement a real-life mmW 5G-based system and evaluate it from a user-centric perspective.

3 System Implementation

We implemented an end-to-end system for low-latency 360° video VR telepresence via MEC-Assisted millimeter-wave (mmW) 5G. Our system is comprised of 1) a sender application that streams ∼6K 360° video from a commodity camera, 2) a Multi-access Edge Computing (MEC) server to which we offload foveated compression, and 3) a VR client that renders the statically foveated video received while updating the MEC with its most recent head positions.

3.1 Hardware and Software Apparatus

Our system facilitates sending 360° live video from a camera to a VR user through 5G wireless connections, while using head data (headset view direction) to provide foveated compression. To stream and render 360° videos, we use 1) an *Oculus Quest* VR headset[1] with a per-eye resolution of 1440x1600 and about 90° FOV along the horizontal axis, and 2) an *Insta360 Evo* 360° camera[2] with a

[1] https://en.wikipedia.org/wiki/Oculus_Quest

[2] https://www.insta360.com/product/insta360-evo/

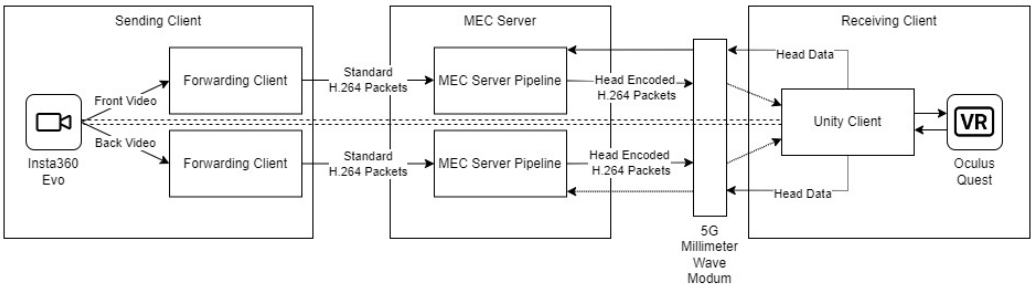


Fig. 2: Communication from the 360° camera to the VR headset utilizing our system. The end-to-end system is comprised of three entities, including a Sending Client, MEC Server, and Receiving Client. Two independent pipelines between the end components transmit the front and back hemispheres of the 360° video. Each video stream is relayed to the MEC Server where it enters an instance of the MEC Server Pipeline. Each MEC Server Pipeline instance accepts the video stream of standard H.264 packets from the Sending Client, along with the head data from the Receiving Client. The MEC Server Pipelines then produce the accompanying head encoded H.264 packet streams carried by 5G communication to the 5G Millimeter Wave Modem. The Unity Client component in the Receiving Client collects the two streams and produces a single rendering of the sender's camera feed on the Oculus display. Simultaneously, the Receiving Client streams updated head data (view direction) back to the MEC Server over a 5G connection.

maximum resolution of 5760x2880 (∼6K). Note that while viewing a ∼6K 360° video, the visible portion of the video in the Oculus Quest's FOV is nearly at maximum resolution (1440x1440 pixels). Our system connects to a millimeter wave (mmW) 5G indoor station on the FR2 band with an InseeGo 5G MiFi M1100 modem [3]. We offloaded video processing and foveated compression to a MEC server, which is a part of the 5G core network. Our MEC runs Red Hat Enterprise Linux (v7.9) with 32 GB RAM and an 8-core Intel Xeon 8268 at 2.9 GHz. We installed all software dependencies with yum.

We customized the x264 library[4] for our foveated compression procedure to output video in the standard H.264 codec, and handled video encoding and decoding with the open-source FFmpeg 4.4[5] multimedia library. We implemented the VR client in Unity and created a customized video-sending application for the *Insta360 Evo* 360° camera.

3.2 System Architecture

The system we developed consists of three software components: the *Sending Client*, *MEC Server*, and the *Receiving Client*. Together these entities connect a 360° camera and virtual headset, which are attached to the sending and receiving user's machines respectively (see Fig. 2). During a streaming session, the 360° video is sent from the *Sending Client* as a stream of H.264 compressed packets to the MEC Server.

Upon reception at the MEC Server, we apply static-foveated compression to the 360° video stream to reduce its size and more effectively transmit the information. To accomplish this we decode the video frames through the MEC Server Pipeline and re-encode them by dynamically reallocating the video quality to optimize for the user's view direction. The re-encoded video is then transmitted to the *Receiving Client* via 5G communication.

At its max resolution (∼6K) the *Insta360 Evo* camera outputs two separate video streams corresponding to the front and back 180° fisheye cameras. These streams are processed separately on the MEC, and then combined and rendered into a "skybox" texture on the *Receiving Client*.

3.3 Sending Client

The *Sending Client* is composed of the sender's 360° camera and a *Forwarding Client* application hosted on their desktop. Together the camera and application capture the scene the sender wishes to immerse the other user in and transmit the video stream to the server. Once a video stream begins, the *Forwarding Client* establishes a connection with the *MEC Server Pipeline* and starts to forward the video to the server node over TCP. The sender uploads two compressed video streams (the front and back hemispheres) as produced by the video camera's onboard encoder. Each stream is H.264 Main Profile video at 30fps, 50Mbps per camera, for a total of 100Mbps. However, here we focus on evaluating the downstream link with MEC-assisted mmW 5G.

3.4 MEC Server Pipeline

The *MEC Server* creates and maintains instances of the *MEC Server Pipeline* which connects the *Sending* and *Receiving Clients*. After establishing connections with both clients, an instance of the pipeline focally encodes a video stream using the receiver's head data to optimize bandwidth usage. Each *MEC Server Pipeline* is designed as a load-balanced pipeline composed of four task threads that receive the *Sending Client*'s stream, decode the stream, re-encode the stream using head data, and transmit the re-encoded video to the *Receiving Client* respectively. The encoder and decoder wrap FFmpeg's libavcodec, taking advantage of hardware acceleration and fine-grained encoder controls. We use a customized build of the x264 encoder and the "ultrafast, zero-latency" encoder preset to minimize encoder delay.

Our customized x264 encoder receives the stream of head data orientation vectors from the *Receiving Client* over UDP. The encoder uses the head data to dynamically adjust the output quality of the video, boosting quality in the areas visible to the user while decreasing quality in the areas outside the field of view, thereby optimizing the use of the stream bandwidth. This is explained in more detail in section 3.6. Together, our implementation of the *MEC Server Pipeline* achieves real-time frame re-encoding with sub 25 ms of latency.

It is worth pointing out that foveation (e.g., decoding and re-encoding) at the sender is also possible. However, we choose to foveate on the MEC server to optimize the user's perceived video quality. As the foveation is dependent on the head data streamed from the client to the MEC server, a high latency at the downstream link causes video artifacts to be noticeable in a user's field of view. While the sender can be located anywhere with arbitrary latency, the MEC server is typically situated near the receiver in the carrier's network. Therefore, foveating at the MEC server not only offloads computation from clients, but also minimizes the network latency between the client and the server.

3.5 Receiving Client Pipeline

The *Receiving Client* runs on a receiving user's VR headset and desktop and communicates with the *MEC server* via a 5G mmWave mo-

[3] https://static.inseego.com/us/download/userguide-m1100-global.pdf
[4] https://code.videolan.org/videolan/x264/
[5] https://ffmpeg.org/

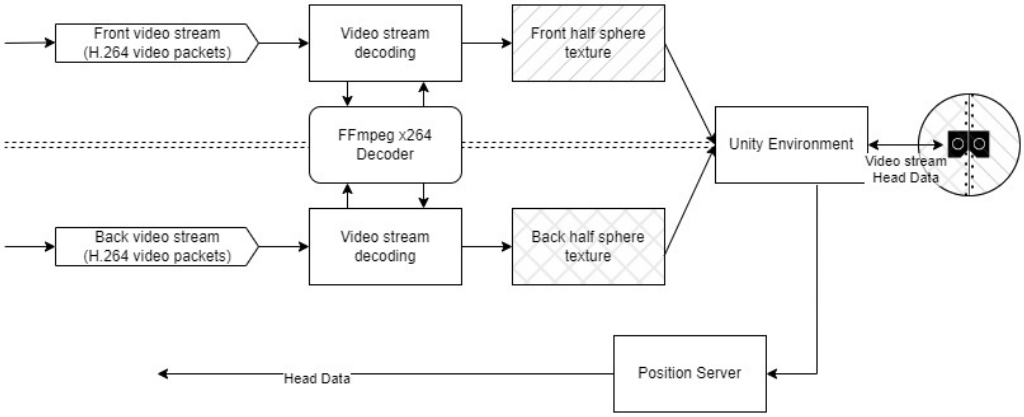


Fig. 3: The Receiving Client. The Unity Client receives two H.264 video streams, one for each hemisphere. These are ingested in parallel into the client pipeline and handled by two instances of the video stream decoding plugin. The frames produced from decoding are then stored in textures and rendered onto the surface of a spherical mesh object surrounding the user. Simultaneously, head orientation data is continually collected and sent to the MEC server.

dem. The client, implemented in Unity, receives the video streams and decodes them using FFmpeg and x264 (Fig. 3). The decoded video frames are copied into textures and rendered onto a sphere mesh object surrounding the user, using a custom shader to reverse the distortion of the Insta360's fisheye lenses. Simultaneously, the user's head orientation is continually measured to determine the user's viewing direction and sent to the MEC server as UDP packets. Here, UDP is used to minimize latency, as head data easily fits into individual packets, and loss of this data is not critical. Timestamps are used to protect against reordered reception of packets.

The *Receiving Client* principally consists of a decoding plugin, video renderer, and head position sender subcomponents. The decoding plugin, implemented in C++ and linked to Unity as a native-code plugin, wraps FFmpeg and x264 and decodes incoming video data using the "fast decode, zero-latency" preset. Each incoming frame is decoded into a 2880x2880 image and mapped to a Unity texture.

The video renderer is implemented as a custom shader, which binds the two separate decoded textures and uses texture lookups to dynamically unwrap the fisheye lens videos into full 360° video surrounding the user. Finally, the position sender component obtains the head pose of the VR camera, encodes the rotational vector as a binary packet, and transmits it at 30Hz to the MEC server as a stream of UDP packets. Here, we designed our system to produce frames at the same rate as the input, to minimize bandwidth usage and latency of video delivery.

3.6 Static-Foveated Compression Based on Head Tracking

While we want to stream a full 360° video to improve the feeling of immersion and presence, the receiving user is only capable of perceiving a small window of this video due to the limited field-of-view of human vision and the VR headset. Thus, we can save significant amounts of bandwidth by transmitting only video that is visible to the user.

However, if there is high latency between the measurement of the head pose and the transmission of the corresponding video, the user may see partial frames that do not cover their entire field of view as they rotate their head. Past attempts to solve this issue have included expanding the transmitted field-of-view to establish a buffer, or predicting how the user will move their head based on kinematics or analysis of video contents. By contrast, we take advantage of the low latency of 5G networks to enable rapid reporting of the head pose, enabling us to use a smaller transmitted field-of-view and avoid complex and error-prone gaze prediction heuristics.

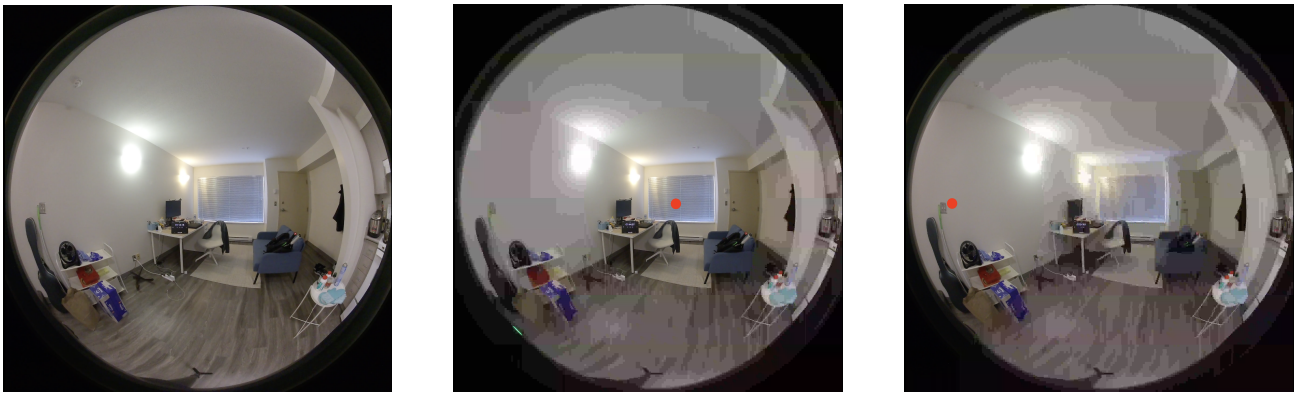
Our foveated compression system is implemented within our custom build of x264. It receives head data from the *Receiving Client* at 30Hz. We implement our foveation through modification of an x264 library which is called by FFmpeg during the encoding process. Libx264, which we leverage in our design, includes functions for applying quantization to the signal produced by an image during standard H.264 video encoding. Quantization is applied to the frame through a quantization parameter (qp) which controls the degree to which details are removed from the macroblocks which make up a frame. A higher qp corresponds to a more restricted bitrate and thus a reduced level of detail in the transmitted image.

We overwrite x264's rate control function, which determines the qp for each 16x16 macroblock, to apply static-foveated compression to the frame. Our approach to static-foveated compression uses a fixed transmission field of view. Each macroblock has its proximity measured to the center of a user's FOV in polar coordinates to determine if it is within the view of the client. As in the case of mapping for our shader, the effects of distortion caused by the fisheye lenses also need to be removed via geometric mapping. If the distance between the macroblock and the real-time position vector for the user's gaze is within the fixed field-of-view then we apply standard H.264 quantization. However, should the distance exceed the threshold, we subject it to a constant qp value of 51 which heavily reduces the bitrate of the signal for that block. Since the heavily quantized regions of the frame occur outside of the user's FOV, it does not impact the user's experience with the streamed VR media yet allows large amounts of the picture to be rendered at a lower resolution. We demonstrate example images of raw 360° camera frames pre- and post-compression in Fig. 4.

3.7 Fisheye Camera Mapping

The Insta360 Evo employs two circular fisheye lenses, each with a field-of-view of around 185 degrees, which are combined to produce a 360° video. At the highest resolution (5760x2880), the camera produces two separate square video streams, one for each camera, as shown (concatenated) in Figure 5. The Insta360's smartphone companion app automatically unwraps these videos into VR videos, but as we are capturing the raw video from the camera, we need to implement unwrapping and fisheye undistortion manually. As there are multiple different fisheye projections [2], we determined the camera's projection mapping (equidistant fisheye) and relevant distortion parameters through reverse engineering, allowing us to mathematically relate spherical angles onto corresponding points in the camera images.

For a given spherical angle (the view direction from the VR focal center to the VR video sphere), we convert the angle into a



(a) (b) (c)

Fig. 4: Example pre- and post-foveation raw 360° (front) camera frames with the correspondent head direction marked with a red dot: a) the original camera video without any foveation, b) the foveated video with the head direction at roughly the center, the video maintains high quality in the focal area, whereas the rest of the video is compressed with the maximum qp, c) as the user turn their head to the left, the focal area follows accordingly. Note that in the examples shown here, the foveated angle is 90°.

normalized view vector (x, y, z). Based on the sign of z, we select the appropriate camera image (either front- or rear-facing). The magnitude of z is translated into the distance from the center of the lens image, while the x and y values are converted into the polar angle around the circular lens image. During rendering in VR, we apply this forward transformation to obtain the texture coordinate for each rendered pixel (based on the view angle of that pixel from the camera). During macroblock quality adjustment, we apply this transformation in reverse to identify the view angle for the center of each macroblock, establish the angular distance to the user's viewing direction, and determine whether the macroblock falls within the transmission field of view.

4 System Evaluation

Our system evaluation sets out to answer key questions regarding our system's performance: 1) How much static-foveated area (90° to 180°) is suitable? and 2) How much visual quality improvement can we achieve with static-foveated compression given the same video bitrates? To answer these questions, we conducted two sets of system measurements profiling our system's *static-foveation turnaround time* (latency) and *within-FOV visual quality*. To make our experiment results as realistic as possible, we conducted the measurements and user studies (Sec. 5) in a typical office environment without paying any special attention to mitigating obstacles.

4.1 Foveation Turnaround Time

A small foveated area is ideal for our system as it can save the bandwidth of video delivery and produce high video quality in the foveated area given the same overall video bitrates. However, an over-constrained static-foveated area risks causing video artifacts to appear in the VR user's field of view as they turn their heads. A suitable foveated area thus balances system performance and user perception. A key contributing factor is the *static-foveation turnaround time*, which stands for the latency between the VR client's delivery of a head position and the reception of the correspondent foveated frame. Ideally, with 5G millimeter wave, our system should have a *foveation turnaround time* that is close to what is achievable with a hardwired connection.

4.1.1 Measurement Configuration

Recall that in our system pipeline 2, our system has the VR client report its head orientations to the MEC server as UDP packets. To measure the *foveation turnaround time*, we attach a timestamp to every head position packet transmitted by the VR client. The MEC server then sends back the most recent head-data timestamp

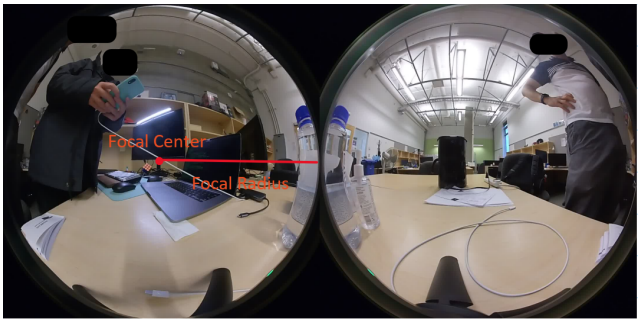


Fig. 5: 2D texture containing the hemispheres captured by the 360° camera with two circular fisheye lenses. The figure shows the focal point of one of the lenses along with the radius which defines the picture.

it receives with the static-foveated video frames it generates. The timestamps were attached as extra non-video metadata in the video packets and were not affected by the compression process. At the arrival of each video frame, the VR client calculates the *foveation turnaround time* by subtracting the timestamp on the video frame from its system time. We benchmarked the average turnaround time for static foveation during 1 minute of video streaming with our system. We conducted the same measurement with the same video under different network configurations including, 1) a wired Ethernet connection, 2) WiFi broadcasted from a router connected to the same wired ethernet, 3) 5G mmW, and 4) 4G LTE.

4.1.2 Results

We illustrated the results of our system's *foveation turnaround time* in Fig. 6. As expected, the wired Ethernet connection has the lowest network latency and leads to an average turnaround time of 46ms (std=12ms). In comparison, 5G mmW has a similarly low foveation turnaround time of 56ms (std=20ms). Both Ethernet and 5G mmW have a lower and stabler foveation turnaround time than using 4G LTE (mean=151ms, std=187ms). It is surprising that using WiFi (mean=47ms, std=12ms) achieves about the same turnaround time as using Ethernet. We determined that this is because the WiFi we used is broadcasted from a router that connects to the same wired Ethernet connection, making the difference in network latency negligible. We think the low latency provided by mmW 5G makes it feasible to set a static-foveation area close to a VR headset's FOV. We further explored the feasibility of various focal areas with user

experiments in Sec. 5.1.

4.2 Within-FOV Visual Quality

Visual quality is a key feature for any video streaming and conferencing system. Theoretically, the use of foveated rendering can allow a higher bitrate to be streamed in the selected focal area, thus providing the same perceived visual quality within a user's FOV when streamed at a lower bitrate.

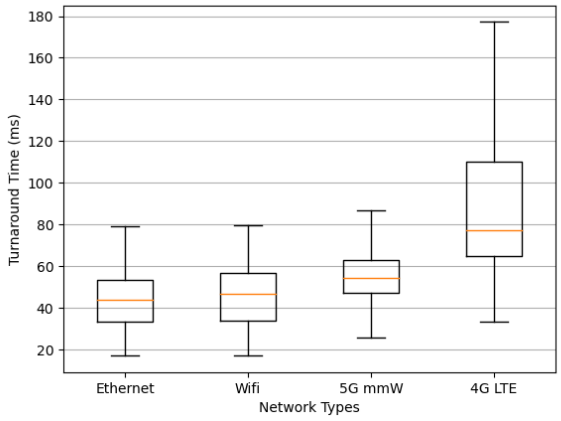


Fig. 6: Measurements for static foveation turnaround time for different network conditions. The standard deviation for Ethernet, WiFi, 5G mmW, and 4G LTE measurements are respectively 12ms, 12ms, 20ms, and 187ms. We can see that the measurement for 5G mmW is close to the ones for Ethernet, and is much lower and stabler than the measurements for 4G LTE. Since we used WiFi broadcast from a router wired to the same Ethernet connection we used for measurements here, the measurement results for it is similar to that of wired Ethernet.

4.2.1 Measurement Configuration

Similar visual quality measurements in previous work [27] usually benchmark the video quality of the entire video. Here, to better capture the visual quality a user actually perceives with our system during a VR telepresence experience, we only benchmark the average visual quality with Peak Signal-to-Noise Ratio (PSNR) and FOVVideoVDP [32] *within* the VR headset's FOV. The video we used here is a 20-second clip of an indoor scene directly recorded from our ∼6K 360° camera at 30fps. For this measurement, we adapted our MEC application to enable video streaming with pre-recorded videos saved on the server. For each measurement, we screen-recorded the VR headset's dual-eye view mirrored on our desktop. We repeated the same measurements using different combinations of bitrates (i.e., 32Mbps, 16Mbps, 8Mbps, 4Mbps, 2Mbps, and 1Mbps) and sizes of the focal area (i.e., horizontal and vertical 90°, 120°, 150°, 180°, and no foveated compression) to obtain 30 video clips. We then calculated the PSNR (dB) and FOVVideoVDP (measured in Just-Objectionable-Difference (JOD) and with the *–foveated* flag set to true) [32] between the screen recording produced with the original and the 30 statically foveated compressed video clips. We chose FOVVideoVDP as a second metric because it accounts for video flickering (temporal aliasing), and is thus more suitable for foveated videos. Note that before all the measurements, we also calculated the PSNR and JOD between two screen recordings produced with the same original videos. The results were 43.43 dB and 9.66 JOD, which establishes the highest possible PSNR and FOVVideoVDP achievable with this setup.

4.2.2 Results

We illustrated the visual quality measurement in Fig. 7. In general, the PSNR (Fig. 7a and 7b) and JOD (Fig. 7c and 7d) showed the same pattern. Note that JOD is a relevantly large measure as an image that is 1 JOD higher than another means it is preferable for 75% of people [32]. Therefore, we plotted Fig. 7c and 7d with the y-axis range from 8-10 JODs. As expected, higher video bitrates lead to higher video quality. In Fig. 7a and Fig. 7c, given the same bitrate, we can observe that the within-FOV visual quality with foveated rendering is always higher than the videos without foveation (i.e., the purple line in Fig. 7a and Fig. 7c), except for the measurements with a 90° FOV (i.e., the blue line in 7a and 7c). This is because 90° is close to our headset's FOV and is susceptible to visual artifacts at the periphery of the VR field of view. In Fig. 7b and 7d, given the same bitrates, foveated rendering leads to an increase in visual quality. This video quality improvement plateaus and decreases with a larger foveated area. Notably, the visual quality of a video streamed at a 120° focal area and 4Mbps is comparable to the same of a video streamed at 32Mbps without foveated rendering.

5 User Experiments

We conducted user experiments to further evaluate our system's foveated rendering and user-perceived visual quality. Our user experiment includes two Just Noticeable Difference (JND) [44] tests and one session of end-to-end 360° Video Conferencing. We conducted all of our user experiments with our system configured to use 5G. We recruited 15 participants (9 male, and 6 female) with an average age of 24 (min=19, max=33), all of whom had normal or corrected-to-normal vision. To gather meaningful feedback regarding our system, we recruited participants with prior experience related to virtual reality and 360° video streaming. Specifically, P1, P3, P4, and P8-12 have routine developmental/research experience in immersive/VR applications, P6 and P7 have experience in computer vision research, and P2, P5, and P13-15 have experience playing with mixed-reality applications. Each user experiment took around 45 minutes and we compensated each participant with 16 dollars for their time. All participants agreed to and signed the consent form approved by the institution's ethics board prior to the study.

5.1 Just Noticeable Static Foveation

Ideally, the users should experience 360° video conferencing without noticing the implementation of foveation. Therefore, our first JND test sets out to determine the smallest focal area that is enough for a user to notice video artifacts as they turn their heads. For this JND test, we adapted the MEC application to linearly decrease the focal area from horizontal/vertical 180° to 75° over the course of a 1-minute video. All users watched the same 360° videos of an outdoor scene streamed in the Oculus VR headset. We instructed the participants to casually move their heads and explore the scene. We prompted users to follow certain targets within the scene and timed these such that different participants had similar head movement traces (i.e., attention) during the experiment. We designed our prompts to encourage participants to move their heads faster, mitigating the potential bias toward under-reporting the foveation angle. We refrained from using a virtual target for the users to follow, as we found that such virtual objects can distract users from noticing the change in the video quality in our pilot studies. We instructed the participants to notify the study facilitator whenever they noticed video artifacts in the video, and we then recorded the corresponding focal area (in degrees of angle). This procedure was repeated 5 times for every participant to account for system and user variability. We observed the participants' behavior during the study and instructed them to think aloud.

5.1.1 Results

15 participants yielded 75 data points, reported in Table 1. From the results, we calculated the 0.5 just noticeable static-foveation

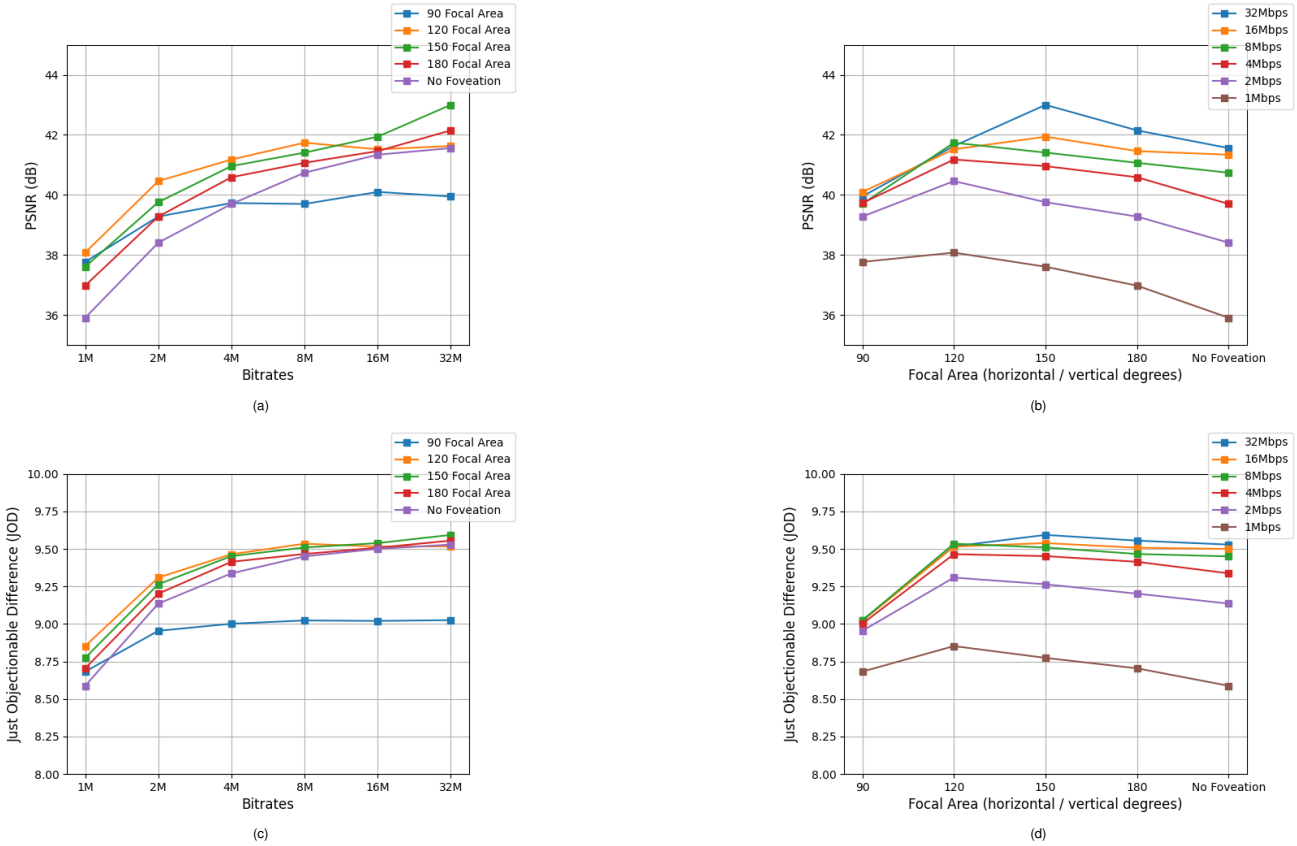


Fig. 7: Visual quality measurements in PSNR (dB) and FOVVideoVDP (JOD). In (a) and (c) we plotted the trend of the visual quality of different choices of focal areas. We can see that most within-FOV visual quality with foveated rendering has higher visual quality than the videos without (the purple line), showing that foveation can provide higher visual quality in the focal area given the same bitrate. In (b) and (d) we plotted the trend of the visual quality of different choices of bitrates. For a focal area larger than the headset's FOV (90°), given the same bitrates, the visual quality gain achieved with foveation tops at a 120° or 150° focal area and decreases with a larger foveated area. Notably, the visual quality of a video streamed at a 120° focal area and 4Mbps is already comparable to the same of a video streamed at 32Mbps without foveated rendering.

(50% percentile) is horizontal/vertical 92.78°, which is only 2.78° larger than the FOV of the VR headset we use. This result is consistent with the low foveation turnaround time we measured in Section 4.1. This shows that the low network latency provided by 5G mmW makes it feasible to set the focal area to closely match the VR headset field-of-view in foveated rendering. Note that it is normal for the results to be less than the VR headset's FOV because the video artifacts only show up in the participants' peripheral vision. We only observed 4 cases where the participants noticed the foveated video artifacts at a focal area over 120°. We attribute those edge cases to network instability, as in all these cases, the participants responded that the video artifacts showed only as a sudden blink that immediately vanishes.

5.2 Just Noticeable Visual Quality Degradation

In Section 4.2, we have shown that our foveated rendering provides a better visual quality given the same bitrates. However, mere measurements in PNSR do not always reflect the user-perceived visual quality. Here, our second JND test determines how much bitrate we can save without a perceivable visual quality degradation. A potential option is to adapt our system to gradually reduce the bitrate of the video it streams, similar to our previous JND study in Section 5.1. However, we noticed that, unlike video artifacts, noticing subtle visual quality degradation requires a person to carefully observe and compare videos. Therefore, we adapted our system to stream "mirrored" videos on the left and right hemispheres of the 360° videos (Fig. 8). For one hemisphere, we streamed the video at 32Mbps without foveated compression. For the other hemisphere, we streamed the mirrored video with foveated compression while setting the focal area to horizontal/vertical 120°. At the beginning of the study, we set the bitrate of the hemisphere with foveated compression to 16Mbps. We then instructed the participant to tell whether they notice a difference in the visual quality and if they did, to point out which hemisphere they think has the lower visual quality. If the participant indicates that there is no visual difference or incorrectly identifies which hemisphere has the lower-quality video, we decreased the bitrate of the lower-quality hemisphere by half. We repeated this process until the participants correctly determined the hemisphere of the 360° video with the lower visual quality. We then increased the bitrate of the lower-quality side by 1Mbps and repeated the process until the participants failed to correctly point out which hemisphere has the lower visual quality. Note that before each round we run a binary random number generator to decide which hemisphere would stream the video with foveated compression.

5.2.1 Results

We illustrate the results of just noticeable visual quality degradation in Fig. 9. Our result shows that, compared with a 32Mbps video, half of the users did not notice any visual quality degradation in their focal area until the bitrates dropped to less than 6Mbps, marking an up-to 80% bitrate save. This result indicates that our foveated

Table 1: Just Noticeable Static Foveation Results. The 0.5 just noticeable static-foveation (50% percentile) is horizontal/vertical 92.78°, which is only 2.78° larger than the FOV of the VR headset we use. This shows that 5G mmW provides a low network latency which makes it feasible to set the focal area to closely match the VR headset field-of-view in foveated rendering. Note that it is normal for the results to be less than the VR headset's FOV because the video artifacts only show up in the participants' peripheral vision. We only observed 4 cases where the participants noticed the foveated video artifacts at a focal area over 120°. We attribute those edge cases to network instability, as in all these cases, the participants responded that the video artifacts showed only as a sudden blink that immediately vanishes.

Participant id	Round 1	Round 2	Round 3	Round 4	Round 5
P1	90.75°	82°	96°	115.25°	97.75°
P2	94.25°	125.75°	99.5°	108.25°	99.5°
P3	76.75°	89°	104.75°	136.25°	145°
P4	90.75°	99.5°	83.75°	97.75°	136.25°
P5	96°	111.75°	99.5°	96°	89°
P6	96°	80.25°	89°	90.75°	110°
P7	97.75°	83.75°	89°	90.75°	103°
P8	76.75°	82°	87.5°	89°	83.75°
P9	76.75°	97.75°	75°	90.75°	101.25°
P10	87.25°	110°	113.5 °	101.25°	118.75°
P11	106.5°	80.25°	87.25°	76.75°	94.25°
P12	78.5°	75°	76.75°	75°	76.75°
P13	75°	78.5°	80.25°	76.75°	75°
P14	80.25°	82°	87.25°	92.5°	97.75°
P15	76.75°	80.25°	92.5°	90.75°	83.75°

Fig. 8: Demonstration of what participants see in the JND test for visual quality degradation. When they turn their heads to the stitching line of the hemispheres, they see a mirrored image. We instructed the participants to tell which side has the higher visual quality. In this example, the left side is streamed at 32Mbps and the right side is streamed at 4Mbps.

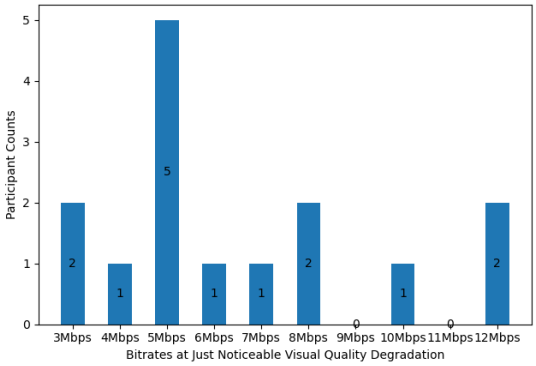


Fig. 9: The results for "Just Noticeable Visual Quality Degradation". We found that most users do not perceive a visual quality degradation from 32Mbps until the bitrate drops to less than 6Mbps.

rendering significantly reduces bitrate while preserving perceived visual quality, which is consistent with the system measurement results in Sec. 4.2.

5.3 End-to-End Video Conferencing

The last part of the user experiment set out to evaluate the user experience in an end-to-end video conference application enabled by our system, and inform future work on potential interactive techniques and applications. Specifically, the study coordinator streamed a remote environment with the 360° camera while the participant viewed the 360° video streamed in an Oculus Quest VR headset. After the video conference, the participants answered a questionnaire on: 1) the experience of 360° VR Telepresence compared to regular video conference, 2) potential interactive techniques and applications. Note that our experiment focused on evaluating the visual and interactive aspects of VR telepresence, as it is well-established in the literature that stereo audio improves immersion in VR. Thus, in this experiment, we provided audio via a separate channel (i.e. phone call). We conclude the participants' responses and our observations during the study as follows.

5.3.1 User Experience in 360° VR Telepresence

All participants mentioned that 360° VR Telepresence is more immersive and realistic. P1 especially pointed out that when the study coordinator moves the 360° camera, it feels like the coordinator is directly interacting with them. Participants also praised the ability to omnidirectionally explore the space from a first-person perspective: P1, P2, P5, P10, and P11 commented that it gave them more freedom; P8 responded that it provides them more embodiment and makes them feel like they are actually in the streamed environment. Participants also expressed concerns in terms of motion sickness, which can attribute to the fact that the VR user always remains still while the other user moves the camera. This shows that our system is more suitable for use cases with a stationary camera optimally positioned to capture most of the space without requiring repositioning.

5.3.2 Potential Interactive Techniques and Applications

Since we recruited users with experience in playing VR games or developing VR applications, we use this part of the user experiment to inform future interactive techniques and applications via their responses. We group the main suggestions and ideas that emerged

as follows:

- **The ability to interact and navigate in the remote scene**: Multiple participants (P1, P2, P7, P8, P9, P10) asked for the ability of the VR user to move and interact with the remote scene, which is essential for potential applications in remote gathering and collaboration. One potential approach for the VR user to navigate in the remote environment is to install motors on the 360° camera to make it remotely controllable.
- **More vivid 3D effects**: P3, P12 pointed out that the current implementation of the system is lacking vivid 3D effects as the 360° camera remains stationary no matter how the VR user moves their body. Besides mobilizing the 360° camera itself, a potential approach is to remap the video adapting to the user's head position (e.g., a user feels an object is closer if they lean toward it).
- **More bidirectional interactions**: P5 pointed out that the current interaction has the user with the 360° camera unidirectionally stream the video to the other user, thus there should be techniques facilitating more bi-directional interactions. In the future, it would be beneficial to anchor a 3D avatar representing the VR user to the 360° camera. Conversely, we also observed that it would be necessary to provide additional cues for the VR user to locate where the other user is, especially at the beginning of the session and when the other user moves away from their field of view.
- **Virtual artifacts and avatars**: P3, P8, and P13 suggested the potential of adding virtual assets and avatars. Specifically, P3 responded that there is the potential for our system to be extended to multi-user teleconferencing, and incorporating virtual avatars and assets can make the application more entertaining. Similarly, P8 imagined a collaborative scenario and suggested that using virtual assets can make collaboration richer and more efficient.

6 Discussion

We have shown that our system enables an immersive and realistic real-time 360° VR Telepresence experience with MEC-Assisted mmW 5G. Our system utilizes mmW 5G and edge computing, which minimizes the server-client latency to ∼50ms. With such minimal latency, our pipeline delivers a perceived visual quality less susceptible to video artifacts showing at the periphery of the FOV. With objective and subjective user experiments we have shown that our system saves up to 80% bandwidth saving while retaining the same visual quality.

6.1 Comparison with Prior Work

Here we discuss how our MEC-Assisted pipeline benefits the foveated streaming quality by comparing it with prior similar systems.

The performance of foveated video streaming is highly dependent on latency. For tile-based viewport adaptation, Nguyen et al. [35] have shown that an excessive delay in the network can lead to a stall in tile delivery and dramatically reduce the visual quality. Similarly, for foveated compression, to minimize the impact on visual quality by video artifacts, Albert et. al [4] concluded that the total system latency needs to be lower than 50-70ms. The highest bandwidth saving reported in prior work in foveated video streaming has achieved a high bandwidth savings of 80% [40], which is similar to our system. However, they evaluated their work in an ideal network setting by running the head-data processor (i.e., the server) and the headset in the same local network, minimizing the latency. Therefore, it is reasonable to expect a decreased performance with a real-life server without edge computing. In comparison, our system more practically reduces overall system latency by utilizing MEC-Assisted 5G mmW as in Sec. 4.1.

To mitigate the latency issue, prior systems have proposed head-movement prediction [19, 39]. However, such predictive models rely on training data and may be inaccurate given an unseen video. Therefore, they often trade-off bandwidth saving by streaming additional high-quality content outside of the user's FOV [39]. In comparison, our system measurements on static foveation turnaround time show that it is feasible to set the focal area of foveated rendering close to the headset's field of view when using mmW 5G. Combining our low-latency approach with head-movement prediction could further improve performance in the future.

6.2 Limitations and Future Work

One notable discovery of the just noticeable static foveation is that some users do not notice the foveation video artifacts until the focal area decreased to be less than the headset's field of view. This can partially be attributed to the low network latency of mmW 5G, as well as the observation that most participants do not tend to turn their heads rapidly while using VR applications. There is potential to save bandwidth by further increasing compression ratios by gradually reducing the qp within the headset field of view according to the precise gaze point. Prior research has investigated this [18,27]. In this work, we refrained from such an implementation as most customer-level VR headsets do not come with built-in eye-tracking capability, and eye movements tend to be even faster than head motions. In the future, it is also possible to further improve our system's foveation by dynamically controlling the parameters according to the user's head motion and latency. Specifically, the system can enlarge the focal area when the user is moving their head quickly or when the latency of the network is higher, and shrink the focal area if the opposite.

The qualitative response we obtained for the end-to-end VR teleconferencing points to future work for this system regarding user experience. One notable future direction is to make the streamed remote environment more accessible for the VR user. Mobilizing the 360° camera to create a remote agent would allow users to freely navigate the remote environment. Furthermore, techniques such as creating annotations, virtual assets, or robotic engineering may provide a richer interactive experience for the VR user. Of course, it would also be interesting to investigate whether controlling a remote agent from virtual reality headsets leads to significant motion sickness.

The current implementation is a unidirectional space-sharing system with telepresence. Future efforts should also be spent on making the interaction more bi-directional. From the side of the user streaming with the 360° camera, they should be able to see the other user's remote presence in a 3D reconstructed figure or avatar. On the other hand, we found that while the VR user is immersed in the streamed environment, they might need additional cues on the other user's presence.

Another future direction is to enhance the vividness of the 360° video itself. It might not always be feasible to control a remote agent with the 360° camera installed. Therefore, when VR user is learning or moving their body, they should be able to perceive the corresponding change in terms of motion parallax. This may be achievable by streaming depth information alongside the 3D video and enabling asynchronous time warping [6], and remapping the 3D video according to a user's head position.

7 Conclusion

In this work, we implemented and evaluated an end-to-end system for VR Telepresence by live streaming 360° videos via MEC-Assisted mmW 5G. We implemented static-foveated rendering to save bandwidth. In our system evaluation, we measured foveation turnaround time in mmW 5G, and compared the results with different setups of network conditions (i.e., wired Ethernet, WiFi, and 4G LTE). We further measured visual quality within the headset's FOV and benchmarked the visual qualities by setting different focal areas. Combined with JND tests on focal areas and visual qualities,

we found that mmW 5G makes high-quality 360° streaming with foveated rendering more feasible. Finally, we obtained qualitative feedback on a telepresence video conference enabled by our system to inform future directions.

Acknowledgments

This work was supported in part by the Natural Science and Engineering Research Council of Canada (NSERC) under Discovery Grant RGPIN-2019-05624 and by Rogers Communications Inc. under the Rogers-UBC Collaborative Research Grant: Augmented and Virtual Reality. We thank Ailin Saggau-Lyons for invaluable help in the initial development of the system.

References

[1] Essential concepts. https://developer.tobii.com/xr/learn/foveation/rendering/essential-concepts/. Accessed: 2023-6-14. 1, 2

[2] Fisheye projection. https://wiki.panotools.org/Fisheye_Projection. Accessed: 2023-6-14. 4

[3] Qualcomm snapdragon XR2 5G platform. https://www.qualcomm.com/products/mobile/snapdragon/xr-vr-ar/snapdragon-xr2-5g-platform. Accessed: 2023-3-24. 1

[4] R. Albert, A. Patney, D. Luebke, and J. Kim. Latency requirements for foveated rendering in virtual reality. *ACM Trans. Appl. Percept.*, 14(4):1–13, Sept. 2017. 1, 2, 9

[5] D. V. G. Alcabaza, Polytechnic University of the Philippines - Santa Rosa Campus, M. E. Legaspi, T. L. Muyot, K. L. C. Ofren, J. A. D. Panganiban, and R. E. Tolentino. Real-time realistic telepresence using a 360 degree camera and a virtual reality box. *Int. j. inf. technol. comput. sci.*, 11(3):46–52, Mar. 2019. 1, 2

[6] M. Antonov. Asynchronous timewarp examined. https://developer.oculus.com/blog/asynchronous-timewarp-examined/. Accessed: 2023-6-17. 9

[7] J. Chakareski, M. Khan, T. Ropitault, and S. Blandino. 6DOF virtual reality dataset and performance evaluation of millimeter wave vs. Free-Space-Optical indoor communications systems for lifelike mobile VR streaming. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*, pp. 1051–1058. ieeexplore.ieee.org, Nov. 2020. 2

[8] K. Doppler, E. Torkildson, and J. Bouwen. On wireless networks for the era of mixed reality. In *2017 European Conference on Networks and Communications (EuCNC)*, pp. 1–5, June 2017. 2

[9] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler. Edge computing meets millimeter-wave enabled VR: Paving the way to cutting the cord, 2018. 1, 2

[10] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler. Toward Low-Latency and Ultra-Reliable virtual reality. *IEEE Netw.*, 32(2):78–84, Mar. 2018. 2

[11] X. Feng, V. Swaminathan, and S. Wei. Viewport prediction for live 360-degree mobile video streaming using user-content hybrid motion tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(2), jun 2019. doi: 10.1145/3328914 2

[12] W. S. Geisler and J. S. Perry. Real-time foveated multiresolution system for low-bandwidth video communication. In B. E. Rogowitz and T. N. Pappas, eds., *Human Vision and Electronic Imaging III*. SPIE, July 1998. 1, 2

[13] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. Foveated 3D graphics. *ACM Trans. Graph.*, 31(6):1–10, Nov. 2012. 1, 2

[14] A. Gupta and R. K. Jha. A survey of 5g network: Architecture and emerging technologies. *IEEE Access*, 3:1206–1232, 2015. doi: 10.1109/ACCESS.2015.2461602 2

[15] S. Gupta, J. Chakareski, and P. Popovski. Millimeter wave meets edge computing for mobile vr with high-fidelity 8k scalable 360 video. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2019. 2

[16] M. Hosseini. View-aware tile-based adaptations in 360 virtual reality video streaming. In *2017 IEEE Virtual Reality (VR)*, pp. 423–424, Mar. 2017. 2

[17] M. Hosseini and V. Swaminathan. Adaptive 360 VR video streaming: Divide and conquer. In *2016 IEEE International Symposium on Multimedia (ISM)*, pp. 107–110, Dec. 2016. 2

[18] G. Illahi, T. Van Gemert, M. Siekkinen, E. Masala, A. Oulasvirta, and A. Ylä-Jääski. Cloud gaming with foveated graphics, 2018. doi: 10.48550/ARXIV.1809.05823 1, 2, 9

[19] A. S. Kaplanyan, A. Sochenov, T. Leimkühler, M. Okunev, T. Goodall, and G. Rufo. DeepFovea: neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. *ACM Trans. Graph.*, 38(6):1–13, Nov. 2019. 1, 2, 9

[20] S. Kasahara and J. Rekimoto. JackIn head: immersive visual telepresence system with omnidirectional wearable camera for remote collaboration. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*, VRST '15, pp. 217–225. Association for Computing Machinery, New York, NY, USA, Nov. 2015. 2

[21] M. A. Khan, E. Baccour, Z. Chkirbene, A. Erbad, R. Hamila, M. Hamdi, and M. Gabbouj. A survey on mobile edge computing for video streaming: Opportunities and challenges. *IEEE Access*, 10:120514–120550, 2022. doi: 10.1109/ACCESS.2022.3220694 2

[22] H. Kim, J. Yang, J. Lee, S. Yoon, Y. Kim, M. Choi, J. Yang, E.-S. Ryu, and W. Park. Eye Tracking-Based 360 vr Foveated/Tiled video rendering. In *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–1, July 2018. 2

[23] H.-W. Kim, T. T. Le, and E.-S. Ryu. 360-degree video offloading using millimeter-wave communication for cyberphysical system. *Trans. emerg. telecommun. technol.*, 30(4):e3506, Apr. 2019. 1, 2

[24] J. Kim, Y. Jeong, M. Stengel, K. Akşit, R. Albert, B. Boudaoud, T. Greer, J. Kim, W. Lopes, Z. Majercik, P. Shirley, J. Spjut, M. McGuire, and D. Luebke. Foveated ar: Dynamically-foveated augmented reality display. *ACM Trans. Graph.*, 38(4), jul 2019. doi: 10.1145/3306346.3322987 1, 2

[25] T.-T. Le, D. N. Van, and E.-S. Ryu. Real-time 360-degree video streaming over millimeter wave communication. In *2018 International Conference on Information Networking (ICOIN)*, pp. 857–862, Jan. 2018. 1, 2

[26] T. T. Le, D. Van Nguyen, and E.-S. Ryu. Computing offloading over mmwave for mobile VR: Make 360 video streaming alive. *IEEE Access*, 6:66576–66589, 2018. 2

[27] D. Li, R. Du, A. Babu, C. D. Brumar, and A. Varshney. A Log-Rectilinear transformation for foveated 360-degree video streaming. *IEEE Trans. Vis. Comput. Graph.*, 27(5):2638–2647, May 2021. 1, 2, 6, 9

[28] Z. Li, T. Teo, L. Chan, G. Lee, M. Adcock, M. Billinghurst, and H. Koike. OmniGlobeVR: A collaborative 360-degree communication system for VR. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, DIS '20, pp. 615–625. Association for Computing Machinery, New York, NY, USA, July 2020. 1, 2

[29] X. Liu, Q. Xiao, V. Gopalakrishnan, B. Han, F. Qian, and M. Varvello. 360° innovations for panoramic video streaming. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, HotNets-XVI, pp. 50–56. Association for Computing Machinery, New York, NY, USA, Nov. 2017. 1, 2

[30] Y. Liu, J. Liu, A. Argyriou, and S. Ci. MEC-Assisted panoramic VR video streaming over millimeter wave mobile networks, 2019. 2

[31] P. Lyu and H. Hua. Design of a statically foveated display based on a perceptual-driven approach. *Opt. Express*, 31(2):2088–2101, Jan 2023. doi: 10.1364/OE.480900 1, 2

[32] R. K. Mantiuk, G. Denes, A. Chapiro, A. Kaplanyan, G. Rufo, R. Bachy, T. Lian, and A. Patney. Fovvideovdp: A visible difference predictor for wide field-of-view video. *ACM Trans. Graph.*, 40(4), jul 2021. doi: 10.1145/3450626.3459831 6

[33] X. Meng, R. Du, M. Zwicker, and A. Varshney. Kernel foveated rendering. *Proc. ACM Comput. Graph. Interact. Tech.*, 1(1):1–20, July 2018. 2

[34] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash. Adaptive 360-degree video streaming using layered video coding. In *2017 IEEE Virtual Reality (VR)*, pp. 347–348, Mar. 2017. 2

[35] D. V. Nguyen, H. T. T. Tran, and T. C. Thang. An evaluation of tile selection methods for Viewport-Adaptive streaming of 360-degree video. *ACM Trans. Multimedia Comput. Commun. Appl.*, 16(1):1–24, Mar. 2020. doi: 10.1145/3373359 9

[36] J. Orlosky, K. Kiyokawa, and H. Takemura. Virtual and augmented reality on the 5G highway. *Journal of Information Processing*, 25:133–141, 2017. 2

[37] J. Park and K. Nahrstedt. Navigation graph for tiled media streaming.

In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pp. 447–455. Association for Computing Machinery, New York, NY, USA, Oct. 2019. 2

[38] T. Piumsomboon, G. A. Lee, A. Irlitti, B. Ens, B. H. Thomas, and M. Billinghurst. On the shoulder of the giant: A Multi-Scale mixed reality collaboration with 360 video sharing and tangible interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–17. Association for Computing Machinery, New York, NY, USA, May 2019. 1, 2

[39] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan. Flare: Practical Viewport-Adaptive 360-degree video streaming for mobile devices. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, pp. 99–114. Association for Computing Machinery, New York, NY, USA, Oct. 2018. 1, 2, 9

[40] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan. Optimizing 360 video delivery over cellular networks. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, ATC '16, pp. 1–6. Association for Computing Machinery, New York, NY, USA, Oct. 2016. 2, 9

[41] J. Ryoo, K. Yun, D. Samaras, S. R. Das, and G. Zelinsky. Design and evaluation of a foveated video streaming service for commodity client devices. In *Proceedings of the 7th International Conference on Multimedia Systems*, number Article 6 in MMSys '16, pp. 1–11. Association for Computing Machinery, New York, NY, USA, May 2016. 1, 2

[42] M. Y. Saraiji, K. Minamizawa, and S. Tachi. Foveated streaming: Optimizing video streaming for telexistence systems using eye-gaze based foveation. *Virtual Reality Society of Japan*, Sept. 2017. 1, 2

[43] N. Stein, D. C. Niehorster, T. Watson, F. Steinicke, K. Rifai, S. Wahl, and M. Lappe. A comparison of eye tracking latencies among several commercial head-mounted displays. *i-Perception*, 12(1):2041669520983338, 2021. PMID: 33628410. doi: 10.1177/2041669520983338 1, 2

[44] M. K. Stern and J. H. Johnson. *Just Noticeable Difference*, pp. 1–2. John Wiley & Sons, Ltd, 2010. doi: 10.1002/9780470479216.corpsy0481 6

[45] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai. Multi-path multi-tier 360-degree video streaming in 5G networks. In *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, pp. 162–173. Association for Computing Machinery, New York, NY, USA, June 2018. 2

[46] A. Tang, O. Fakourfar, C. Neustaedter, and S. Bateman. Collaboration with 360 videochat: Challenges and opportunities. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, DIS '17, pp. 1327–1339. Association for Computing Machinery, New York, NY, USA, June 2017. 1, 2

[47] T. Teo, L. Lawrence, G. A. Lee, M. Billinghurst, and M. Adcock. Mixed reality remote collaboration combining 360 video and 3D reconstruction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–14. Association for Computing Machinery, New York, NY, USA, May 2019. 1, 2

[48] E. Turner, H. Jiang, D. Saint-Macary, and B. Bastani. Phase-Aligned foveated rendering for virtual reality headsets. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1–2, Mar. 2018. 2

[49] O. Wiedemann, V. Hosu, H. Lin, and D. Saupe. Foveated video coding for Real-Time streaming applications. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, May 2020. 1, 2

[50] S.-C. Yen, C.-L. Fan, and C.-H. Hsu. Streaming 360° videos to head-mounted virtual reality using DASH over QUIC transport protocol. In *Proceedings of the 24th ACM Workshop on Packet Video*, PV '19, pp. 7–12. Association for Computing Machinery, New York, NY, USA, June 2019. 1, 2

[51] X. Yu, F. Xu, J. Cai, X.-Y. Dang, and K. Wang. Computation efficiency optimization for Millimeter-Wave mobile edge computing networks with NOMA, 2022. 2