Towards Retina-Quality VR Video Streaming: 15 ms Could Save You 80% of Your Bandwidth

Luke Hsiao, Brooke Krajancich, Philip Levis, Gordon Wetzstein, and Keith Winstein
Stanford University
Stanford, California, USA
{lwhsiao@cs.,brookek@,pal@cs.,gordon.wetzstein@,keithw@cs.}stanford.edu

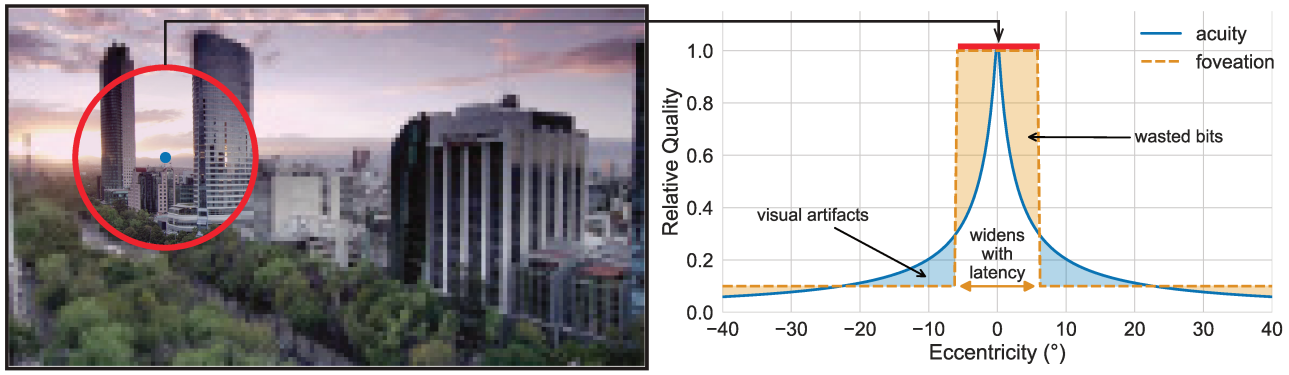

Figure 1: One reason virtual reality systems today cannot yet deliver retina-quality video experiences is due to bandwidth limitations. To reduce data rates, recent work uses the decay of visual acuity in human perception for foveated video compression, keeping a small region of high resolution while decaying quality in the periphery (left)[1]. We show that decreasing motion-to-photon latency benefits foveated video compression and enables minimally-sized regions of high resolution (right).

ABSTRACT

Virtual reality systems today cannot yet stream immersive, retina-quality virtual reality video over a network. One of the greatest challenges to this goal is the sheer data rates required to transmit retina-quality video frames at high resolutions and frame rates. Recent work has leveraged the decay of visual acuity in human perception in novel gaze-contingent video compression techniques. In this paper, we show that reducing the motion-to-photon latency of a system itself is a key method for improving the compression ratio of gaze-contingent compression. Our key finding is that a client and streaming server system with sub-15 ms latency can achieve 5× better compression than traditional techniques while also using simpler software algorithms than previous work.

CCS CONCEPTS

• **Computing methodologies** → **Image compression**; *Virtual reality*; • **Hardware** → *Displays and imagers.*

KEYWORDS

video compression, latency, virtual reality, gaze-contingent, foveated

1 INTRODUCTION

Virtual reality (VR) video strives to offer immersive experiences through high fidelity, 360° display of recorded content. Doing so requires streaming video at both high resolutions and frame rates across large fields of view with constrained computational power and bandwidth. Today's VR systems, unable to achieve this, stream videos below retina resolution, at low frame rates, or both.

Several practical challenges stand in the way of achieving immersive, retina-quality VR video. First, only powerful GPUs are capable of decoding high resolution video frames at, or greater than, the 90 Hz or higher refresh rates that are essential for VR [8]. Second, modern VR display hardware does not yet support retina-quality pixel densities. Consumer headsets today only reach about 20 % of that goal[2]. Third, the poor performance of existing systems is due in part to the sheer amount of data our retinal acuity requires. In this paper, we focus on this third challenge.

Consider an *uncompressed* 5.7K (5760 × 2880 px), 360 × 180°, VR video—the highest resolution supported by 360° cameras today. Setting aside the immense bandwidth requirements of streaming uncompressed 5.7K video (~7 Gbit/s), this would still only achieve 16 samples/°, just 27 % of the 60 samples/° standard for retina quality. Since streaming services like YouTube encode 5.7K video at 15 to 30 Mbit/s (>230× smaller than the uncompressed bitrate), the resolution after compression is even worse. Achieving retina-quality VR video with traditional techniques would require a huge increase in bitrate (higher resolutions and less compression); the bandwidth requirements alone are a barrier.

This challenge has inspired perceptually-motivated graphics; a complementary field of work that exploits the limitations of human perception to reduce bandwidth or computation. In particular, these techniques use the fact that our visual acuity (or ability to resolve spatial detail) is highest in the region of the retina called the fovea

[1]Note that the quality reduction has been exaggerated for illustration purposes.

[2]For example, one pixel of the HTC VIVE Pro is approximately 4′35″ of visual angle, or ~5× larger than the minimum angle of resolution in the foveola

and drops quickly with eccentricity (or distance from the fovea). Combined with eye tracking, this knowledge is used to degrade rendering quality [10, 13, 30], level-of-detail [26, 28, 29], or display resolution [19, 42] in regions that fall on a user's periphery, thus reducing bandwidth without perceivable quality degradation. For streaming 360° video, related work also utilizes techniques such as adapting encoding parameters [12], predicting a user's field of view [41] and upscaling highly compressed video using super-resolution [7] (Figure 1 shows an example). We focus on foveated video compression, which seeks to compress a sequence of frames while modeling visual acuity decay to concentrate the allocation of bits in the encoded video to the foveal region [16, 18, 33].

While the compression benefits of foveated techniques are significant, they are fundamentally limited by the *motion-to-photon latency* of the system. This latency is the time between a change in the viewer's gaze and the resulting change in the display's pixels. Larger latencies introduce larger uncertainty about the viewer's gaze position and consequently require a larger foveal region to avoid perception of the degradation applied in the periphery.

We present a study on the relationship between a system's motion-to-photon latency and the bitrate required to display a gaze-contingent video without degrading its perceived quality. We use a desktop setup as a proxy for future high-frame-rate, low-latency, retina-resolution VR systems. Our key finding is that with ~15 ms latency, we improve on the bitrate of traditional compression techniques by 5 × while using simpler software techniques than previous work. We also test more modest reductions in latency (e.g., to 45 ms), but did not find latency at this level to be helpful in reducing bitrate. We believe using gaze-contingent compression with low-latency systems is a key step towards realizing truly immersive VR experiences.

Contributions. This paper makes these contributions:

- We build a video streaming system using foveated video compression. The display reacts to gaze changes within 15 ms, over 3 × lower than previously demonstrated in VR HMDs.
- Through a user study using our low-latency prototype, we derive perceptual insights about the relationship between system latency and the bitrate required to display a foveated video without noticeable quality degradation.
- We find that low latency can reduce the required bitrate of a video transmission system by 5 ×, but only when end-to-end latency is well below the ~50 ms budget thought sufficient by previous work.

We focus on the impact of eye-motion-to-photon latency. As a result, our design has a few important limitations. First, our system excludes the latency introduced by separating the client and server with a realistic network. The need for low server-to-client latencies means that a video encoder would need to be near the client at the network edge (a potential use case for edge computing). Second, our prototype uses an encoder-in-the-loop approach to perform video compression and streaming in real-time. This approach has a higher computational cost than those that pre-encode video since it instead requires the server to encode video for each viewer. Last, we evaluate our system using an eye tracker and display that are among the fastest available today; comparable performance is unavailable on the consumer market or in current head-mounted displays.

2 BACKGROUND AND RELATED WORK

2.1 Human Perception

The human visual system has a field of view of approximately 220° horizontally by 135° vertically [20]. Yet only a small region (~1.5°), called the fovea, is capable of resolving spatial detail as fine as 60 cycles/° [9]. Outside the fovea, the distribution of retinal components and refractive lens effects change rapidly, resulting in decreased visual acuity [44], less sensitivity to color [2, 14], and limited stereoscopic depth discrimination [37], as well as increased sensitivity to flicker [15, 23] in our peripheral visual field.

The eyes make short, rapid movements called saccades to scan visual scenes with the high-resolution fovea. While these ballistic-like movements can occur at speeds of up to ~900 °/s [6], the temporary suspension in perception (referred to as saccadic suppression) that occurs a short period before, during, and after the eye movement (totaling 50 to 200 ms [35]) reduces the challenge they pose to gaze-contingent systems. However, even during fixation the eyes involuntarily move, albeit slower (~50 ′/s [36]), exploring fine detail with a random-walk-like pattern referred to as ocular drift and correcting the fixation position with microsaccades. During fixation, there is also a high frequency component referred to as ocular tremor (see [22] for a detailed review).

2.2 Foveated Video Compression

This knowledge of the human visual system, coupled with real-time eye tracking, has given rise to foveated graphics techniques that imperceptibly degrade the peripheral image to improve efficiency (i.e., reducing bandwidth or computation). For example, foveated graphics improves efficiency by reducing the number of vertices or fragments a GPU has to sample, ray trace, shade, or transmit to the display [21]. The most prominent approach is perhaps foveated rendering [10, 13, 30] and display [19, 42], where images and videos are rendered, transmitted, or displayed with spatially varying resolutions without affecting the perceived image quality. Related approaches also use gaze location to vary bit-depth [27], shading or level-of-detail [26, 28, 29], or reconstruct content from sparse samples [18] outside of the foveal region.

These ideas have also been applied to video compression. Traditional video compression removes temporal and spatial redundancy in a sequence of video frames. Foveated video compression builds on these techniques by using real-time gaze information to concentrate data allocation in an encoded video to the foveal region, achieving better compression in the periphery.

There are many approaches for foveated video compression. Lee et al. [24] use a nonuniform filtering scheme to increase compression. Specifically, their algorithm maximizes a foveated signal-to-noise ratio (FSNR) using a Lagrange multiplier along curvilinear coordinates. Illahi et al. [16] use a similar but simpler approach of varying quantization parameters, compressing peripheral regions more than foveal regions. Instead of compressing a single video stream, Romero et al. [33] store a video in two resolutions, low and high. A client first fetches the low-resolution stream, and then streams only the cropped, high-resolution segments based on a viewer's current gaze. Similarly, Jeppsson et al. [17] divide a video into many small blocks and pre-encodes each block in many different resolutions. Then, when streaming, the resolutions are chosen

Figure 2: A system must compensate for latency by enlarging the foveal region to avoid a viewer's gaze escaping the region before the system can react.

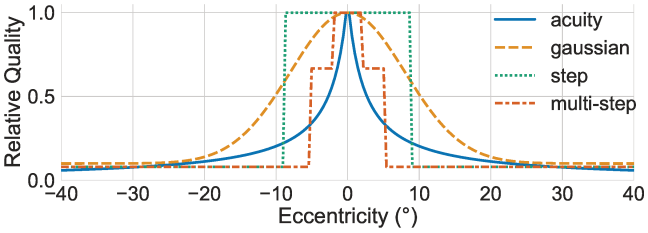

Figure 3: Related work uses a variety of functions to approximate relative quality (i.e., the allocation of bits) with the decay in visual acuity.

on the server based on gaze data and stitched together at the client into three levels of resolution. Foveated video compression can achieve bitrates that are 25 to 60 % of the bitrates of traditional compression algorithms with similar visual quality.

2.3 Latency

Being gaze-contingent, foveated compressions systems are very sensitive to motion-to-photon latency—the time between the eyes moving and the pixels of the display updating with the frame corresponding to the new gaze location. Yet none of the foveated video compression works described in Section 2.2 discuss the impact of latency on their results.

The importance of system latency has been given more attention in foveated rendering, with a number of works measuring the maximal tolerable system latency to be between 42 to 91 ms, depending on the size of the full resolution foveal image that follows the gaze, the degree of degradation applied to the image, and the type of degradation method used [1, 13, 39, 45]. Similarly, Loschky et al. [25] also observed that detection of image artifacts due to foveation in gaze-contingent, multiresolution displays did not change if latency was kept under 60 ms. However, to the best of our knowledge we are the first to show the significant compression benefits of squeezing system latency below these budgets in reducing the bitrate needed to produce the same visual quality.

3 LATENCY VS. COMPRESSION

Foveated video compression relies on accurate, real-time gaze information to allocate a larger portion of the bitrate to where a viewer is looking while decaying the quality in the periphery. Assuming accurate and instantaneous gaze information, these algorithms can compress frames to have minimally-sized regions of high-resolution without viewer detection. In practice, however, latency introduces uncertainty in a viewer's gaze position, requiring larger regions of high-resolution video[3]. Figure 2 illustrates this challenge. On the left, the gaze position used by the system matches the actual gaze position perfectly, and the periphery can be highly compressed. However, a system must also keep the foveal region large enough such that when the gaze moves, it does not escape the region before the system can react (shown on the right). This occurs if the system latency, t_L, is longer than the time it takes for the gaze to move. Consequently, there is tension between minimally sizing the foveal region for better compression and sizing it large enough to ensure a viewer does not see video artifacts.

While we understand the decay in visual acuity of the human visual system well [11, 32], our understanding of the nuances of peripheral vision (e.g., change blindness, crowding, object recognition, etc.) is still actively developing [34, 40]. Because of these nuances, there is no well-understood mapping function that a foveated compression algorithm can use to transmit the minimal number of bits while maintaining high visual quality for all types of videos.

As a result, foveation is usually achieved by empirically choosing an approximation function to model the decay in visual acuity and applying transformations that appear visually acceptable. For example, Illahi et al. [16] and Wiedemann et al. [46] choose a Gaussian function, Romero et al. [33] choose a step function, and Guenter et al. [13] choose a step function with multiple steps. Figure 3 shows examples of these approximations functions.

Figures 1 and 3 also plot the acuity model of Geisler et al. [11], fit with parameters from Robson et al. [32], in blue. This gives visual acuity, A, as a function of eccentricity, e, as follows.

$$A(e) = \ln(64)\frac{2.3}{0.106 * (e + 2.3)} \quad (1)$$

The goal of these approximations is to minimize the gap between the transmitted quality and the perceived quality. As annotated in Figure 1, transmitting too high of a quality in the periphery wastes bits while transmitting too low of a quality results in visual artifacts. Further, as system latency increases, so does uncertainty about the viewer's gaze and, consequently, the size of the foveal region. The approximation functions must be widened to accommodate this uncertainty, resulting in more wasted bandwidth.

In practice, we find that the choice of approximation function also influences implementation choices, which can in and of itself cause additional latency. For example, a common foveated compression implementation of a Gaussian approximation is to vary the degree of compression of individual subregions of a video frame according to the Gaussian function [16, 46]. This requires processing the full video resolution to produce a single video stream of smoothly varying quality. In contrast, a simple step function can be implemented using two traditionally compressed video streams—one for the cropped high-resolution foveal region and one for the low resolution background. This approach results in far less processing. For example, rather than processing a full 4K (3840 × 2160 px) video, a two-stream approach might process a small 480 × 480 px

[3]Inaccuracy in an eye tracking device also contributes to this uncertainty but is out of scope of this work.

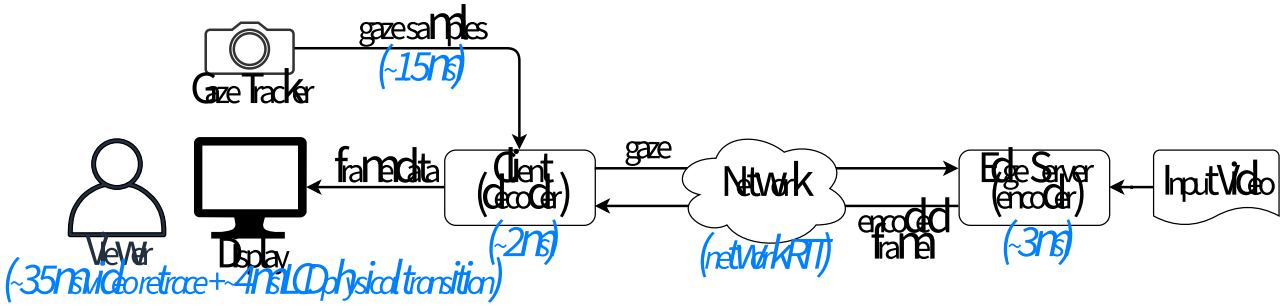

Figure 4: Overview of our low-latency, desktop-based prototype system. This system allows us to focus on the effects of latency on foveated video compression by avoiding the limitations complexities of current VR HMDs.

foveal region and a downscaled 768 × 432 px background, which combines to be <7 % of the original 4K pixels.

To focus on the impact of reducing latency, we chose a simple two-stream approach (Section 4). Our experience suggests that achieving low latencies will be key to realistically achieving retina-quality VR video over a network. We cannot have long latencies and achieve great compression; we need great latencies as well.

4 A LOW-LATENCY PROTOTYPE SYSTEM

Understanding the real-world impact of latency on foveated video compression requires a system with very low latencies. However, commercial head-mounted displays (HMDs) used for VR today have system latencies >45 ms [38]. In addition, these HMDs do not have sufficiently high resolutions (i.e., less than 4K) to be an ideal test bed for studying the impact of latency on compression of retina-quality video[4]. Consequently, we build a desktop-based system as a proxy for future VR HMDs. Doing so allows us to focus on the impact of latency without the limitations of current HMDs.

4.1 Architecture

We design our system based on a typical video-streaming architecture with a client and server model. However, rather than the client only receiving encoded video frames from the server to decode and display, the client also sends the viewer's current gaze position each time a frame is received (Figure 4). This gaze sample allows the server to encode the next video frame foveated on the viewer's gaze position. To minimize system latency, the server and client run as separate processes on the same machine and communicate using message passing, implemented with shared memory.

4.2 Two-Stream Compression

To reduce the latency spent on encoding and decoding, our system uses a simple two-stream approach. The server sequentially reads uncompressed frames at the frame rate of the input video. Then, for each gaze sample it receives from the client, it compresses up to two versions of the current frame[5]. First, it downscales the video frame to a significantly lower resolution. Second, it crops the video frame to a small area around the viewer's gaze location. The resolution of both the downscale and the crop are configurable. It then encodes these two frames to send to the client. At the client, the reverse process occurs. First, the client decodes and upscales the background frame to the size of its display. Next, it decodes the foreground frame and positions it at the corresponding gaze position with a blend[6]. Finally, it displays this composed frame.

As is typical with compression techniques, this approach trades off increased computation (real-time encoding per client) for reduced bitrate. While the server can pre-encode the background, the foreground must be encoded in real-time using the viewer's gaze.

4.3 System Details

We implement our system in Rust, using SDL2, FFmpeg, and x264. Our workstation runs Pop!_OS 20.04 and contains an AMD Ryzen 7 3700X CPU, 16 GB of 3200 MHz memory, and an NVIDIA GeForce RTX 2070 SUPER GPU. Our display is an LG 27GN95B-B (4K at 144 Hz, 7.6 ms input latency). An Eyelink 1000 provides low-latency eye tracking. The software for this system available at https://github.com/lukehsiao/fvideo.

5 EXPERIMENTS

Using our low-latency prototype system as a proxy for future VR HMDs, we seek to answer the following questions.

(1) What is the lower bound for latency of modern hardware?
(2) What is the latency of our foveated compression system, and where is the time spent?
(3) What is the relationship between system latency and achievable video compression?

5.1 Lower Bound for System Latency

The first experiment finds a lower bound for the achievable system latency using commercially available hardware. We use an eye tracker and display that are among the lowest latency available today and minimize video processing by only toggling portions of the display between black and white (i.e., omitting video encoding/decoding). We use an Eyelink 1000 to minimize the latency between a viewer's eyes moving and receiving the data in software. Although lower-latency eye trackers are continually being developed [3], the Eyelink 1000 provides a good trade-off between accuracy, latency[7], and sampling rate among those that are commercially available[8].

[4]Recent HMDs, such as the Vive Pro 2, do include 4k or higher resolution displays.
[5]We also skip both background frames when the current video frame has not changed and foreground frames if the gaze has not not changed.
[6]We set the alpha channel (opacity) to a 2D Gaussian in order to fade out the hard, square edges of the foreground. The parameters of the Gaussian are chosen empirically.
[7]We disable the built-in filters to further minimize latency
[8]Based on their advertised specifications and prior comparison by others [38].

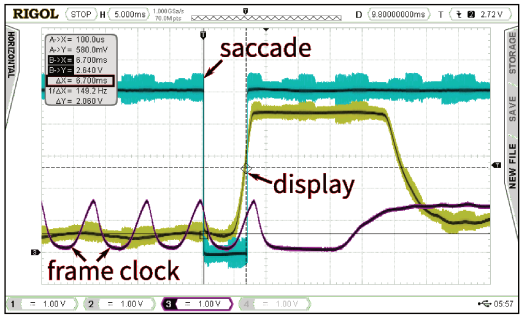

Figure 5: We use an oscilloscope to measure a lower bound for system latency—the time between an artificial saccade occurring (the falling edge in green) and the pixels of the display reacting (the rising edge in yellow).

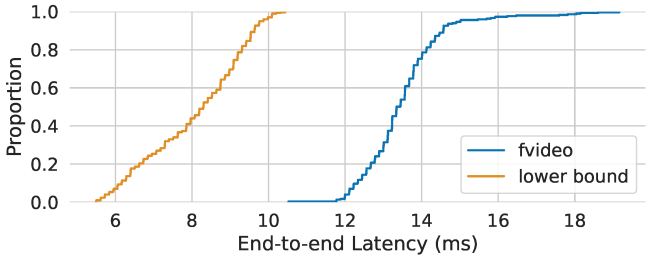

Figure 6: ECDF of end-to-end system latencies. A simple two-stream approach for compression (fvideo) only adds ~5 ms over the lower bound.

To minimize the latency between a frame being sent to the display and the pixels changing, we select a ZisWorks x28 R2 monitor (1080p resolution at 240 Hz), which advertises an input latency of ~30 µs, significantly lower than the 1.5 to 16 ms of most consumer monitors. We also opt for a simple graphics stack for this experiment by using Xubuntu 18.04 with compositing disabled.

To ensure measurements are precise, automated, and repeatable, we design our own Arduino-based artificial saccade generator (ASG)[9]. Most eye trackers (head-mounted or desktop) either track the infrared (IR) reflection of the retina or directly process a video stream of the eye to detect and track the pupil [31]. The Eyelink 1000 uses IR reflection, so we build an ASG that can be triggered using software and toggles between two IR LEDs to mimic a saccade.

Finally, we implement a minimal system that polls for changes in gaze position using the eye tracker and then uses OpenGL to change a small portion of the display from black to white. This pixel change is then detected using a photodiode circuit. The approach of using an ASG and photodiode circuit to measure latency is commonly used [4, 5, 31]. System latency is measured as the time between triggering the ASG and the mid-point of seeing the pixel change on the photodiode. Figure 5 shows an oscilloscope trace of this process with a system latency of 6.7 ms. In some cases, it is possible for the saccade to be triggered and the display pixels to change within the one refresh cycle of the monitor. However, if the saccade does not line up with the frame clock, then it may take up to an additional refresh cycle to update.

[9]See https://github.com/lukehsiao/eyelink-latency. Unable to find a suitable commercial ASG, we follow the precedent of related work by building our own.

We run this measurement for 300 repetitions and plot the empirical cumulative distribution function (ECDF) in Figure 6 (lower bound). The minimum observed latency is under 6 ms, with the majority of samples falling under 9 ms. Of this latency, an average of 1.65 ms is waiting for the updated gaze sample, and the remaining is dominated by the time it takes for the display to update (>4 ms).

5.2 Foveated Compression Latency

Next, we measure the latency of our gaze-contingent, foveated compression prototype. There are two important differences in this experiment compared to the previous lower bound baseline. First, this experiment includes the computational cost of scaling, cropping, encoding, and decoding 4K video frames. Rather than directly changing a portion of a frame from black to white, we use a synthetic video. This video is black until a saccade is detected, after which it toggles a portion of the frame to white. Second, this experiment uses the LG 27GN95B-B monitor, which supports 4K resolution at 144 Hz and has an average of 7.6 ms input latency.

We also run this measurement 300 times and plot the ECDF in Figure 6 (fvideo). On average, our system has ~5 ms longer latency than our lower bound baseline. Of this additional latency, ~2.8 ms comes from the slower display, and the remaining comes from the computational costs of encoding and decoding the two video streams. Importantly, our two-stream approach ensures that the latency of foveated video compression itself is not significantly longer than the latency of our hardware. An approximate breakdown of where time is spent is annotated in Figure 4. A gaze sample is taken and sent to the edge server, where a new frame is encoded. This frame is then sent to the client for decoding and display.

5.3 User Study: Latency vs. Bitrate

We conduct a user study to better understand the relationship between system latency and how much compression can be achieved while maintaining similar visual quality. We set up a controlled laboratory experiment to gather data on perceived video quality using our low-latency prototype (Section 4).

Because this work is motivated by the challenge of streaming retina-quality VR video (Section 1), we use a 4K video encoded at 28 Mbit/s as a proxy for the video quality of streaming platforms like YouTube[10]. In this study, we measure the compressed bitrate of a video as a function of system latency at the point of equipoise perceived video quality compared to the baseline.

As stimuli, we use two 4K videos from Derf's collection [43]. These two videos are selected due to their diverse content, and each consists of two sub-scenes. The first, `barscene`, shows one sub-scene with strong bokeh and another with dialogue between two individuals that naturally guides a viewer's gaze. The second, `square_timelapse`, shows one sub-scene of a busy crowd of people where viewers' gaze typically jumps around the scene, and another of a city skyline with many hard edges and natural scenery. We test these videos at three latency conditions. First, we evaluate our system at its unmodified latency (~14 ms). Then, we select the minimum and maximum latencies of commercially available HMDs as measured by Stein et al. [38]: 45 and 81 ms. We add artificial delay to our system to match these latencies.

[10]Specifically, we use `x264 --preset veryfast --bitrate 28000`.

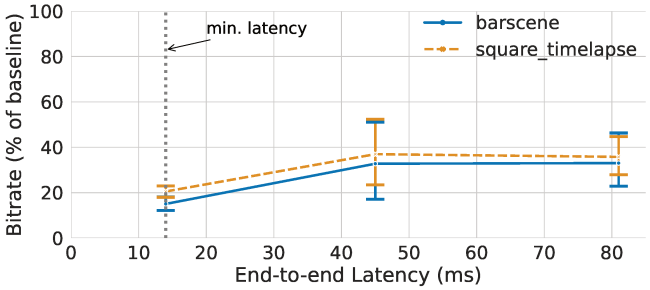

Figure 7: Latency vs. compression, with 95 %-confidence intervals. We improve compression by 5 × using a simple two-stream approach. However, the full benefit comes only at latencies lower than demonstrated by current VR HMDs.

For each video and latency combination, we prepare a set of compression configurations starting at lower resolutions with higher compression, and moving to higher resolutions with lower compression. We only evaluate 3 latency points and 2 videos, to keep the study to a reasonable duration.

5.3.1 Experimental Setup. We use the system detailed in Section 4.3. The LG display is set to 3840 × 2160 px and 144 Hz. Physically, the display is 59.67 × 33.56 cm and placed at a distance that gives ∼55° horizontal field of view, achieving retina-quality resolution. The participant's head is stabilized using a chin and forehead rest, and the eye tracker is placed between the monitor and participant.

5.3.2 Procedure. Each participant was asked to view two videos and perform the same task on each. The order in which the videos were presented was equally divided among participants. We first calibrated the eye tracker and validated the tracking accuracy for each participant. Then, participants were asked to perform four trials of a matching task. Three of the trials correspond to each latency points (14, 45, and 81 ms), and we randomly repeated one trial to check for consistency. The order of the trials was also randomized. Participants were shown a reference video and then asked to select which of ten comparison videos the reference video is most similar to in quality for each trial. The ten videos were ordered by increasing video quality. Participants could also choose to respond that none of the ten were similar in quality. Participants were allowed to take as long as they wished to make their selection and could freely navigate and re-watch any of the videos. Each participant did 8 trials, resulting in a study duration of ∼45 min (see Appendix A).

5.3.3 Participants. We recruited 13 participants[11]. All participants provided written consent before taking part in the study, and the methods were approved by Stanford's institutional review board (IRB). Before each experiment, the participants were briefed about the purpose of the study and their task. Of these 13 participants, we excluded 2 participants' data from the results because we were unable to achieve a maximum calibration accuracy error <10° (unacceptably large compared to the size of the foveal region).

5.3.4 Results. Figure 7 shows the results of our user study. We plot the mean compressed bitrate as a percentage of the 28 Mbit/s baseline along with the 95 %-confidence interval for each latency. At low latency, a simple two-stream approach can compress these videos to ∼20 % of the baseline while maintaining a similar visual quality. Despite using a simple algorithm, our results are competitive with the numbers reported in related work (Section 2.2).

To understand the statistical significance of these differences, we compute a t-test between both the 14 and 45 ms latencies and 45 and 81 ms latencies. We find that the difference between the means of 14 and 45 ms is statistically significant ($t = 2.76$, $p = 0.008$), while the difference between the means of 45 and 81 ms is not ($t = 0.10$, $p = 0.92$). This validates the trend shown in Figure 7.

The latency gap between the fastest and slowest consumer HMDs is a significant 36 ms. However, we find that reducing the latency from 81 ms to 45 ms does not significantly improve the required video bitrate. It is not until we push the system's latency to below that of commercially available HMDs that we see an additional ∼2 × compression benefit. This finding also suggests this relationship is not simply a question of making the foveal region larger as the delay increases—we suspect there is a distinct phenomenon (and a compression opportunity) at low latencies.

Related works that mention system latency often do so primarily to show that the latency is below the ∼50 ms budget proposed by prior work (Section 2). However, our finding not only suggests that driving down system latency can result in significant compression gains without changing the compression algorithm itself, but also that these gains might only be realized with system latencies much lower than previously proposed budgets.

6 CONCLUSION

We present latency reduction as a method for improving foveated video compression and validate its potential by implementing a prototype, ultra-low-latency video streaming system. Our findings indicate that reducing system latency is greatly helpful to achieving the levels of compression needed for retina-quality VR content. The latency budget we describe is tight, but in a model where an edge server can be located within a few milliseconds RTT of the client, we believe server-side video rendering at retina quality may become feasible at practical network throughputs. In concert with future advancements in VR HMDs and improvements in foveated video compression, reducing latency may play a critical role in making retina-quality VR video streaming practical over realistic networks.

It would be useful to know exactly what latency target is required to realize a significant reduction in transmitted bitrate through gaze-contingent encoding; that is to say: if reducing latency from current levels (80 ms) to 45 ms is not helpful, but reducing to 15 ms is very helpful, then what does the curve look like between 15 and 45 ms? Because of the limitations of our study (which was conducted during the COVID-19 pandemic), we cannot answer these questions today, but encourage the community to further investigate the tradeoffs between lowering latency in gaze-contingent video transmission and resulting bitrate reductions.

ACKNOWLEDGMENTS

This work was supported by Facebook Reality Labs, by NSF grants 2045714, 1909212, 2039070, and 1839974, by Google, VMware, Dropbox, and Amazon, and by a Stanford Knight-Hennessy Fellowship, an Okawa Research Grant and a Sloan Research Fellowship.

[11] The COVID-19 pandemic limited the number of participants available for this study.

REFERENCES

[1] 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 (Sept. 2017), 25:1–25:13. https://doi.org/10.1145/3127589

[2] Stephen J. Anderson, Kathy T. Mullen, and Robert F. Hess. 1991. Human Peripheral Spatial Resolution for Achromatic and Chromatic Stimuli: Limits Imposed By Optical and Retinal Factors. *The Journal of Physiology* 442, 1 (1991), 47–64. https://doi.org/10.1113/jphysiol.1991.sp018781

[3] Anastasios N. Angelopoulos, Julien N.P. Martel, Amit P. Kohli, Jörg Conradt, and Gordon Wetzstein. 2021. Event-Based Near-Eye Gaze Tracking Beyond 10,000 Hz. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2577–2586. https://doi.org/10.1109/TVCG.2021.3067784

[4] Jean-Baptiste Bernard, Scherlen Anne-Catherine, and Castet Eric. 2007. Page Mode Reading With Simulated Scotomas: A Modest Effect of Interline Spacing on Reading Speed. *Vision research* 47, 28 (2007), 3447–3459. https://doi.org/10.1016/j.visres.2007.10.005

[5] Christopher J. Bockisch and Joel M. Miller. 1999. Different Motor Systems Use Similar Damped Extraretinal Eye Position Information. *Vision research* 39, 5 (1999), 1025–1038. https://doi.org/10.1016/S0042-6989(98)00205-3

[6] Roger H.S. Carpenter. 1988. *Movements of the Eyes, 2nd Rev.* Pion Limited.

[7] Jiawen Chen, Miao Hu, Zhenxiao Luo, Zelong Wang, and Di Wu. 2020. SR360: Boosting 360-Degree Video Streaming with Super-Resolution. In *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video* (Istanbul, Turkey) *(NOSSDAV '20)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3386290.3396929

[8] Eduardo Cuervo, Krishna Chintalapudi, and Manikanta Kotaru. 2018. Creating the Perfect Illusion: What Will It Take to Create Life-Like Virtual Reality Headsets?. In *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications* (Tempe, Arizona, USA) *(HotMobile '18)*. Association for Computing Machinery, New York, NY, USA, 7–12. https://doi.org/10.1145/3177102.3177115

[9] Michael F. Deering. 1998. The Limits of Human Vision. In *2nd International Immersive Projection Technology Workshop*, Vol. 2. 1.

[10] Sebastian Friston, Tobias Ritschel, and Anthony Steed. 2019. Perceptual Rasterization for Head-Mounted Display Image Synthesis. *ACM Transactions on Graphics* 38, 4, Article 97 (July 2019), 14 pages. https://doi.org/10.1145/3306346.3323033

[11] Wilson S. Geisler and Jeffrey S. Perry. 1998. Real-Time Foveated Multiresolution System for Low-Bandwidth Video Communication. In *Human Vision and Electronic Imaging III*, Bernice E. Rogowitz and Thrasyvoulos N. Pappas (Eds.), Vol. 3299. International Society for Optics and Photonics, SPIE, 294–305. https://doi.org/10.1117/12.320120

[12] 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 the ACM Special Interest Group on Data Communication* (Beijing, China) *(SIGCOMM '19)*. Association for Computing Machinery, New York, NY, USA, 394–407. https://doi.org/10.1145/3341302.3342063

[13] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D graphics. *ACM Transactions on Graphics* 31, 6 (Nov. 2012), 164:1–164:10. https://doi.org/10.1145/2366145.2366183

[14] Thorsten Hansen, Lars Pracejus, and Karl R. Gegenfurtner. 2009. Color Perception in the Intermediate Periphery of the Visual Field. *Journal of Vision* 9, 4 (04 2009), 26–26. https://doi.org/10.1167/9.4.26

[15] E. Hartmann, B. Lachenmayr, and H. Brettel. 1979. The Peripheral Critical Flicker Frequency. *Vision Research* 19, 9 (1979), 1019–1023. https://doi.org/10.1016/0042-6989(79)90227-X

[16] Gazi Karam Illahi, Thomas Van Gemert, Matti Siekkinen, Enrico Masala, Antti Oulasvirta, and Antti Ylä-Jääski. 2020. Cloud Gaming with Foveated Video Encoding. *ACM Transactions on Multimedia Compututing, Communications, and Applications* 16, 1, Article 7 (Feb. 2020), 24 pages. https://doi.org/10.1145/3369110

[17] Mattis Jeppsson, Håvard Espeland, Tomas Kupka, Ragnar Langseth, Andreas Petlund, Peng Qiaoqiao, Chuansong Xue, Konstantin Pogorelov, Micheal Riegler, Dag Johansen, Carsten Griwodz, and Pål Halvorsen. 2018. Efficient Live and On-Demand Tiled HEVC 360 VR Video Streaming. In *2018 IEEE International Symposium on Multimedia (ISM)*. 81–88. https://doi.org/10.1109/ISM.2018.00022

[18] Anton S. Kaplanyan, Anton Sochenov, Thomas Leimkühler, Mikhail Okunev, Todd Goodall, and Gizem Rufo. 2019. DeepFovea: Neural Reconstruction for Foveated Rendering and Video Compression Using Learned Statistics of Natural Videos. *ACM Transactions on Graphics* 38, 6, Article 212 (Nov. 2019), 13 pages. https://doi.org/10.1145/3355089.3356557

[19] Jonghyun Kim, Youngmo Jeong, Michael Stengel, Kaan Akşit, Rachel Albert, Ben Boudaoud, Trey Greer, Joohwan Kim, Ward Lopes, Zander Majercik, Peter Shirley, Josef Spjut, Morgan McGuire, and David Luebke. 2019. Foveated AR: Dynamically-Foveated Augmented Reality Display. *ACM Transactions on Graphics* 38, 4, Article 99 (July 2019), 15 pages. https://doi.org/10.1145/3306346.3322987

[20] Arnold Knapp. 1938. An Introduction to Clinical Perimetry. *Archives of Ophthalmology* 20, 6 (1938), 1116–1117.

[21] G. A. Koulieris, K. Akşit, M. Stengel, R. K. Mantiuk, K. Mania, and C. Richardt. 2019. Near-Eye Display and Tracking Technologies for Virtual and Augmented Reality. *Computer Graphics Forum* 38, 2 (2019), 493–519. https://doi.org/10.1111/cgf.13654

[22] Eileen Kowler. 2011. Eye Movements: The Past 25 Years. *Vision Research* 51, 13 (2011), 1457–1483. https://doi.org/10.1016/j.visres.2010.12.014 Vision Research 50th Anniversary Issue: Part 2.

[23] Brooke Krajancich, Petr Kellnhofer, and Gordon Wetzstein. 2021. A Perceptual Model for Eccentricity-dependent Spatio-temporal Flicker Fusion and its Applications to Foveated Graphics. *arXiv preprint arXiv:2104.13514* (2021).

[24] Sanghoon Lee, M.S. Pattichis, and A.C. Bovik. 2001. Foveated Video Compression With Optimal Rate Control. *IEEE Transactions on Image Processing* 10, 7 (July 2001), 977–992. https://doi.org/10.1109/83.931092

[25] 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* 3, 4, Article 7 (Dec. 2007), 10 pages. https://doi.org/10.1145/1314303.1314310

[26] David Luebke and Benjamin Hallen. 2001. Perceptually Driven Simplification for Interactive Rendering. In *Eurographics Workshop on Rendering Techniques*. Springer, 223–234. https://doi.org/10.1007/978-3-7091-6242-2_21

[27] John D. McCarthy, M. Angela Sasse, and Dimitrios Miras. 2004. Sharp or Smooth? Comparing the Effects of Quantization vs. Frame Rate for Streamed Video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 535–542. https://doi.org/10.1145/985692.985760

[28] Hunter Murphy and Andrew T Duchowski. 2001. Gaze-Contingent Level of Detail Rendering. *EuroGraphics* (2001).

[29] T. Ohshima, H. Yamamoto, and H. Tamura. 1996. Gaze-Directed Adaptive Rendering for Interacting With Virtual Space. In *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*. 103–110. https://doi.org/10.1109/VRAIS.1996.490517

[30] 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, Article 179 (Nov. 2016), 12 pages. https://doi.org/10.1145/2980179.2980246

[31] Eyal M. Reingold. 2014. Eye Tracking Research and Technology: Towards Objective Measurement of Data Quality. *Visual Cognition* 22, 3–4 (2014), 635–652. https://doi.org/10.1080/13506285.2013.876481

[32] J.G. Robson and Norma Graham. 1981. Probability Summation and Regional Variation in Contrast Sensitivity Across the Visual Field. *Vision Research* 21, 3 (1981), 409–418. https://doi.org/10.1016/0042-6989(81)90169-3

[33] Miguel Fabian Romero-Rondón, Lucile Sassatelli, Frédéric Precioso, and Ramon Aparicio-Pardo. 2018. Foveated Streaming of Virtual Reality Videos. In *Proceedings of the 9th ACM Multimedia Systems Conference* (Amsterdam, Netherlands) *(MMSys '18)*. Association for Computing Machinery, New York, NY, USA, 494–497. https://doi.org/10.1145/3204949.3208114

[34] Ruth Rosenholtz. 2016. Capabilities and Limitations of Peripheral Vision. *Annual Review of Vision Science* 2, 1 (2016), 437–457. https://doi.org/10.1146/annurev-vision-082114-035733

[35] John Ross, M. Concetta Morrone, Michael E. Goldberg, and David C. Burr. 2001. Changes in Visual Perception at the Time of Saccades. *Trends in Neurosciences* 24, 2 (2001), 113–121. https://doi.org/10.1016/S0166-2236(00)01685-4

[36] Michele Rucci and Martina Poletti. 2015. Control and Functions of Fixational Eye Movements. *Annual Review of Vision Science* 1, 1 (2015), 499–518. https://doi.org/10.1146/annurev-vision-082114-035742

[37] John Siderov and Ronald S. Harwerth. 1995. Stereopsis, Spatial Frequency and Retinal Eccentricity. *Vision Research* 35, 16 (1995), 2329–2337. https://doi.org/10.1016/0042-6989(94)00307-8

[38] Niklas Stein, Diederick C Niehorster, Tamara Watson, Frank Steinicke, Katharina Rifai, Siegfried Wahl, and Markus Lappe. 2021. A Comparison of Eye Tracking Latencies Among Several Commercial Head-Mounted Displays. *i-Perception* 12, 1 (2021), 1–16. https://doi.org/10.1177/2041669520983338

[39] Michael Stengel, Steve Grogorick, Martin Eisemann, and Marcus Magnor. 2016. Adaptive Image-Space Sampling for Gaze-Contingent Real-time Rendering. *Computer Graphics Forum* 35, 4 (July 2016), 129–139. https://doi.org/10.1111/cgf.12956

[40] Hans Strasburger, Ingo Rentschler, and Martin Jüttner. 2011. Peripheral Vision and Pattern Recognition: A Review. *Journal of Vision* 11, 5 (12 2011), 1–82. https://doi.org/10.1167/11.5.13

[41] Liyang Sun, Yixiang Mao, Tongyu Zong, Yong Liu, and Yao Wang. 2020. Flocking-Based Live Streaming of 360-Degree Video. In *Proceedings of the 11th ACM Multimedia Systems Conference* (Istanbul, Turkey) *(MMSys '20)*. Association for Computing Machinery, New York, NY, USA, 26–37. https://doi.org/10.1145/3339825.3391856

[42] Guanjun Tan, Yun-Han Lee, Tao Zhan, Jilin Yang, Sheng Liu, Dongfeng Zhao, and Shin-Tson Wu. 2018. Foveated Imaging for Near-Eye Displays. *Optics Express* 26, 19 (Sept. 2018), 25076–25085. https://doi.org/10.1364/OE.26.025076

[43] Timothy Terriberry. [n. d.]. *Derf's Test Media Collection.* Retrieved March, 2021 from https://media.xiph.org/video/derf/

[44] L. N. Thibos, F. E. Cheney, and D. J. Walsh. 1987. Retinal Limits to the Detection and Resolution of Gratings. *Journal of the Optical Society of America A* 4, 8 (Aug. 1987), 1524–1529. https://doi.org/10.1364/JOSAA.4.001524

[45] Robin Thunström. 2014. *Passive Gaze-Contingent Techniques Relation to System Latency.* Master's thesis. Blekinge Institute of Technology.
[46] Oliver Wiedemann, Vlad Hosu, Hanhe Lin, and Dietmar Saupe. 2020. Foveated Video Coding for Real-Time Streaming Applications. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX).* 1–6. https://doi.org/10.1109/QoMEX48832.2020.9123080

A EXPERIMENT PROCEDURE DETAILS

The procedure of the user study experiment is outlined in Section 5.3.2. This section provides additional details relevant to conducting the user study.

The user study was conducted between April 10th, 2021, and April 17th, 2021. Because of building restrictions associated with the COVID-19 pandemic, study participants were drawn from members of the Stanford university community who had completed COVID protocol training and were approved for access to the building. As described in Section 5.3.3, this resulted in a study population of 13 participants, from which usable data were obtained from 11.

A.1 User Study Instructions

When users arrived to the study, the following instructions were read to them before they entered the room used for the study.

> We are doing a study comparing video compression techniques. During our study, we will be asking you to view different versions of two short video clips on a screen set up with eye-tracking.
> Since we are socially distanced, you will enter the study room by yourself while I sit in the hallway, but there will be a phone/laptop with a video call active so that we can communicate throughout the study. When you enter the room, you will see a workstation with a headrest mounted to the table. Please be careful not to trip on any cables that may be in your path as you enter. Sit down in front of the headrest and rest your chin on the headrest such that your forehead is gently touching the forehead mount. Feel free to use the various knobs to adjust it to a comfortable position.
> We will start off by calibrating the eye-tracker. A grey screen with a single dot will appear. Let me know once you are looking at the dot and I'll start the process, which will then cycle through a series of dots and positions. Just look at each one. After the calibration, we will run it again to validate the accuracy of the calibration and then you'll be ready to start the study.

At this point, the person conducting the study paused to allow the user to ask clarification questions about the instructions given thus far. After resolving any concerns or questions, they provided the next instructions.

> In the study, we will be asking you to do a matching task. You will be shown one version of the video (the comparison video) and then asked to select which of another set of videos best matches in quality. The set of videos you will have to choose from are numbered from 1 to 10, where 1 is rendered with the lowest quality and 10 the best. The comparison may not be completely straightforward, since the video artifacts that you may see could look completely different, but we would like you to select the lowest numbered video at which you do not have a preference over that video setting and the comparison video. I'll start by showing you video 10, the highest quality option, and you can let me know which video number you would like to see next (for example, we can binary search). I can also re-show you the comparison video, or any other video any time you'd like. Once you've decided on the quality setting, verbally let me know which video number you would like to select, and we'll move on to the next configuration and repeat the process. You are also welcome to indicate that none of the videos fit the criteria.
> Some quick notes:
> - Occasionally, you might see a flash of black at the beginning of a video for both the reference video and the different quality videos. Please ignore this particular bug when comparing the overall qualities.
> - Try to keep your head still in the headrest. If you move significantly (e.g., lift your head from the position), we will have to re-run a calibration. Since the position can be a little uncomfortable, if you need a break at any time, just let me know. I'll let you know when we're switching videos, since we'll redo the calibration then anyway so you can stretch.
>
> Once all of the comparison tasks are complete, I will take a few moments to verify that your data was recorded correctly. Then, I'll let you know when you can exit the room.

After the instructions were provided, the person conducting the study once again paused to give the user an opportunity to ask questions about the task and procedure, and reaffirmed that they would be able to communicate with us via video throughout the duration of the study.

A.2 Conducting the Study

Once a user entered the user study room, they sat at a desk with the eye tracker, display, and chin rest arranged as described in Section 5.3.1. In addition, there was a mobile phone with an active video call open facing the user so that the user and the person conducting the study could communicate. After adjusting the chin and forehead rest, the user was positioned as shown in Figure 8.

The person conducting the study would then walk the user through the calibration procedure, operating the Eyelink's calibration system from outside the room. If minor adjustments were needed to the position of the eye tracker, the user was instructed on the adjustments to make via the video call. In some cases, we were unable to obtain a sufficiently accurate calibration ($<10^{\circ}$) compared to the size of the foveal region. These users were still taken through the entirety of the study, but their results were filtered from the data.

During the video selection process, the person conducting the study would prompt the user which video was being shown (e.g,

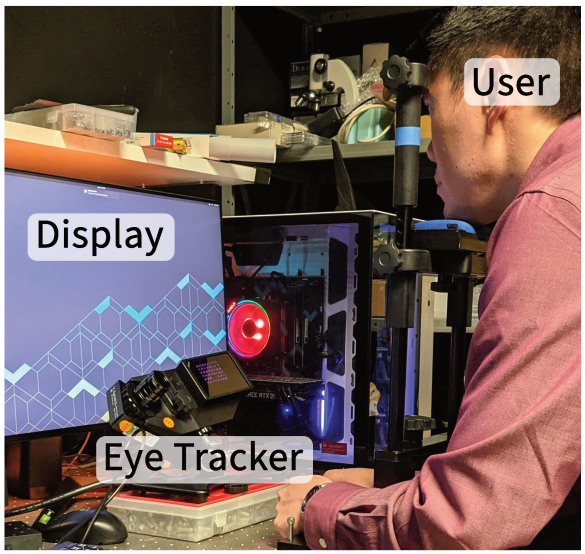

Figure 8: The physical setup of the user study.

"Here is video number 1."), and control video switching remotely. The user would vocally indicate which videos they wanted to see, and what selection they ultimately would like to make. Note that the additional complexity of operating the study from outside the user study room was required by COVID protocols.