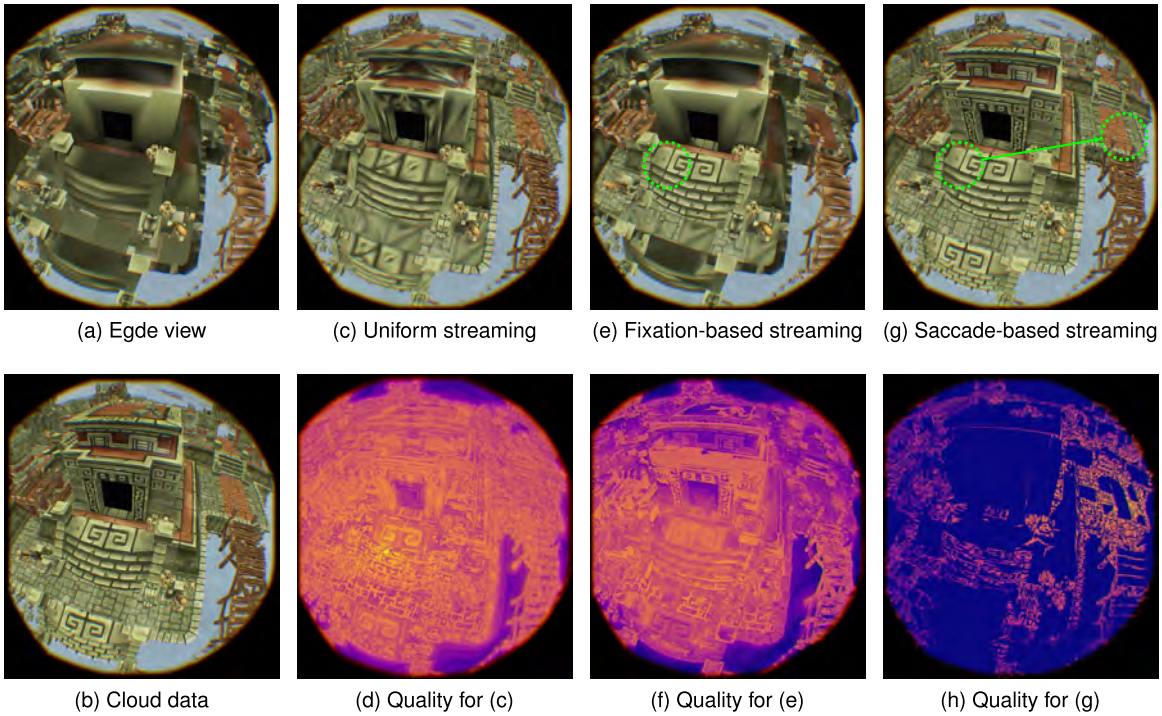
Instant Reality: Gaze-Contingent Perceptual Optimization for 3D Virtual Reality Streaming

Shaoyu Chen, Budmonde Duinkharjav, Xin Sun, Li-Yi Wei, Stefano Petrangeli,
Jose Echevarria, Claudio Silva, *Fellow, IEEE,* Qi Sun



(a) Egde view (c) Uniform streaming (e) Fixation-based streaming (g) Saccade-based streaming

(b) Cloud data (d) Quality for (c) (f) Quality for (e) (h) Quality for (g)

Fig. 1. *Our gaze-contingent immersive 3D assets streaming interface.* Starting from the partially-streamed 3D assets on the edge-side rendered at a given time (a), our method streams additional updates from the cloud to the edge for perceptually closer rendering to the full assets stored on the cloud (b). Standard uniform streaming (c) evenly updates all visible assets in the scene, causing suboptimal perceptual quality in (d), which visualizes both the temporal (popping with respect to (a)) and the spatial (quality with respect to (b)) perceptual errors. Brighter colors indicate worse artifacts. Our method, in contrast, optimizes the subset of the assets to be streamed to the edge for better spatio-temporal quality under the same network bandwidth. Our perceptual model considers both eccentricity-based acuity during fixation (e) and temporal masking during eye movement (g). If the user fixes the gaze (e), our model prioritizes regions near the gaze point (green circle) while reducing potential popping artifacts. If a saccade is detected (g), our model can safely ignore popping to stream more aggressive updates for further quality improvement (h).

Abstract— Media streaming, with an edge-cloud setting, has been adopted for a variety of applications such as entertainment, visualization, and design. Unlike video/audio streaming where the content is usually consumed passively, virtual reality applications require 3D assets stored on the edge to facilitate frequent edge-side interactions such as object manipulation and viewpoint movement. Compared to audio and video streaming, 3D asset streaming often requires larger data sizes and yet lower latency to ensure sufficient rendering quality, resolution, and latency for perceptual comfort. Thus, streaming 3D assets faces remarkably additional than streaming audios/videos, and existing solutions often suffer from long loading time or limited quality.
To address this challenge, we propose a perceptually-optimized progressive 3D streaming method for spatial quality and temporal consistency in immersive interactions. On the cloud-side, our main idea is to estimate perceptual importance in 2D image space based on user gaze behaviors, including where they are looking and how their eyes move. The estimated importance is then mapped to 3D object space for scheduling the streaming priorities for edge-side rendering. Since this computational pipeline could be heavy, we also develop a simple neural network to accelerate the cloud-side scheduling process. We evaluate our method via subjective studies and objective analysis under varying network conditions (from 3G to 5G) and edge devices (HMD and traditional displays), and demonstrate better visual quality and temporal consistency than alternative solutions.

Index Terms—Streaming, Level of detail, Virtual reality, Head-mounted display, Human perception

- *S. Chen, B. Duinkharjav, C. Silva, Q. Sun are with New York University E-mail: {sc6439, budmonde, csilva, qisun}@nyu.edu.*
- *X. Sun, L. Wei, S. Petrangeli, J. Echevarria are with Adobe Research. E-mail:{xinsun, lwei, petrange, echevarr}@adobe.com.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

1 Introduction

Thanks to the rapid growth of Internet technologies such as 5G, streaming services have unprecedentedly revolutionized how we access and consume high-quality multimedia, from listening to audios, watching videos, to scientific discovery (e.g., NASA's Eyes[1]). The edge-cloud streaming setting significantly reduces edge-side computation and redundant offline storage, allowing for portable and wearable consumer devices.

When it comes down to VR/AR, storing all the assets offline is impractical for dynamic and location-based scenarios. Thus, streaming assets from the cloud and perform the rendering on the edge-side become an effective solution. However, VR streaming poses unique challenges due to the required high resolution, high frame rate, low latency, and stereo rendering. Frame-based 2D streaming falls unrealistic given the exceptionally high demands of data volume and responsiveness. Still, current approaches fail in streaming large high-quality 3D assets, introducing unacceptably long loading times or undesirable visual and interactive artifacts (please refer to our video for examples). Comparisons of the recent 250% FLOPS gains in the last two consecutive GPU generations [2] versus a smaller growth of only 26% in global internet bandwidth during the same time [3] also suggest the main roadblock for immersive 3D experiences lies in the delayed access to 3D assets, instead of computational resource.

Perceptual mechanisms, such as foveation, have been harnessed in recent efforts to accelerate interactive rendering [21, 29]. However, the acceleration only occurs after all assets are stored at the user-end, thus improving the computation from, but not the transmission of, scene assets (especially complex geometries). Gaze-contingent effects have been used to optimize the limited network bandwidth for 2D frame-based streaming [18, 37] but not 3D assets. Existing solutions for 3D content streaming often have globally uniform granularity, such as game levels or camera distance. Finer-grained controls can be achieved based on visibility [15] or viewpoint [39] to optimize level-of-details (LoD). However, without comprehensive optimization for human perceived quality, the adaptive streaming may cause strong visual artifacts such as temporal popping or low quality with limited network bandwidth.

To this end, we propose a 3D LoD-based streaming framework that automatically schedules the data transmission priority towards optimal perceptual quality. Our method is based on modeling human spatio-temporal perception and adapts to varying network conditions. Our key idea is to estimate perceptual sensitivity based on spatio–temporal user gazes, including foveation, saccade, and popping, so that we can more aggressively optimize static image quality and dynamic frame change, such as image regions outside the visual fovea or quickly glanced over during saccade. The perceptual importance is then transferred to transmission priorities from the cloud to the edge. To further enable real-time responses without extra latency caused by the complex frequency domain computations, we also implemented a simple neural-network-based accelerator for fast importance computation. Our method is general enough to support various data formats, such as meshes, volumes, height fields, and point clouds. It targets high field-of-view and eye-tracked displays that require interactive streaming, such as cloud-based gaming. The framework is also compatible with transmitting geometric transformations in dynamic scenes, as our proof-of-concept shows in the supplementary video.

We conducted both subjective studies (via VR and traditional displays) and objective analysis based on the recorded data, which indicated higher quality and lower artifacts of our method than alternative solutions under identical bandwidth. We also tested our method under varying latencies by simulating real-world network transmission rates (from 3G to 5G).

In summary, the main contributions of this paper are:

- a perceptually optimized high-quality and low-latency 3D immersive streaming framework, supporting various 3D computer graphics data formats;
- a gaze-contingent perceptual model based on foveation, saccade, and popping to depict spatio-temporal visual behaviors during procedural streaming, including static quality acuity and dynamic change suppression;
- a neural-network-based accelerator for real-time computation of the complex perceptual mechanisms and large data volumes;
- a series of subjective studies and objective analysis validating our method under varying network conditions.

Details on the implementation can be found in the InstantReality Github repository[4].

2 Related Work

2.1 Perception-Aware Graphics

With the rapid deployment of eye-tracking and personal electroencephalography (EEG) devices, perceptually-aware computer graphics approaches have drawn extensive attention from various applications. Gaze-contingent rendering [56], as a representative example, involves understanding the visual optics [12, 30], anatomical retina [48], peripheral perceptual encoding [4] and higher-order neural process [51]. Perceptual models can also been deployed in depth perception [20, 22], saliency [46], color perception [6], visual masking [10] , visual attention [5], navigation in VR [16, 49], light/luminance perception [27, 50, 52], and spatio-temporal media [35, 40, 42]. However, previous perceptually-based graphics systems focus on optimizing local content than transmission. In cloud-based 3D scenarios, solutions on perceived low-latency and temporal consistency have been under-explored. These factors are critical for a smooth and realistic experience, comparing with the less demanding local computation on the user end.

2.2 Low-Latency Streaming in Cloud-Edge Systems

Low-latency streaming has been widely practiced in cloud-based 2D content services, such as film/video platforms and live broadcasting, based on factors such as network conditions [13, 45], foveated vision [18, 37], viewport and visible area [44, 57]. With the current frameworks, the rendering occurs on the cloud side, requiring the network to stream each individual frame. The frequent transmission may harm the experience for highly interactive and latency-sensitive VR/AR scenarios. For instance, e-sports, high quality 3D model design, or physics-based simulation.

2.3 3D Data Transmission

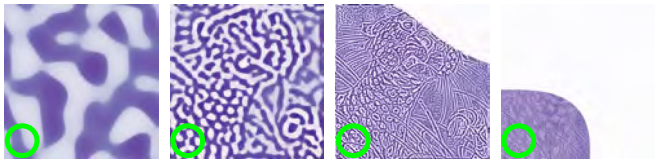
How to prioritize 3D primitives is critical to 3D streaming, which has been studied with little consideration of human perception. To incorporate the unique structures of 3D meshes, Hladky et al. [15] prioritize the triangle components in the streaming pipeline based on their individual visibility. In a real cloud-based system, object-wise LoDs shall be determined with respect to the virtual camera views and the available resources (bandwidth) and constraints (reaction time) [32]. It is also possible to perform progressive rendering without building hierarchical structures [39]. However, these systems rely on particular data types such as meshes or point clouds. A recent advancement to enable a decoupled and efficient rendering-transmission is through irradiance volume data [47]. In highly interactive scenarios, however, the virtual content may comprise various components, such as particles for physics-based simulation, meshes for characters, and point clouds from sensors. Moreover, under near-eye high field-of-view and high frame rate immersive displays, the systems may introduce the well-known popping artifacts [26]. Our system, by analytically modeling human perception, preserves both visual quality and temporal stability.

[1] https://eyes.nasa.gov/

[2] https://www.digitaltrends.com/computing/nvidia-rtx-3090-vs-rtx-2080-ti-most-powerful-gaming-gpus-duke-it-out/

[3] https://blog.telegeography.com/466-tbps-the-global-internet-continues-to-expand

[4] https://github.com/chenshaoyu1995/InstantReality



(a) 4 cycle / im (b) 16 cycle / im (c) 64 cycle / im (d) 256 cycle / im

Fig. 2. *Decomposition visualization of bandpass filtered contrast multiplied by contrast sensitivity function of corresponding frequency band ($c(\mathbf{x},\mathbf{f}_i,I)$ from Equation (2) and Equation (17) in Appendix C).* The green circle indicates gaze position. When $\mathbf{f}_i$ is low ((a) and (b)), the bandpass filtered contrast is computed over the entire visual field based on visual content. With high $\mathbf{f}_i$ ((c) and (d)), the periphery sensitivity was clamped by B_r, thus the empty content.

3 Method

Given a limited network bandwidth and a set of scene data, we optimize and determine the streaming priority with regard to individual user's spatio-temporal perception. The priority is updated according to users' static and dynamic gaze behaviors. Our goal is to maximize perceived quality (compared with a fully local rendering) and ensure dynamic smoothness (minimize popping artifacts).

We first describe the factors in our spatio-temporal perception model (Section 3.1), followed by our method in 2D image space (Section 3.2) and 3D object space (Section 3.3). Equation (3) summarizes our key idea. Finally, to enable the cloud's real-time responses to users' dynamic head/gaze motions, we accelerate the system via a deep neural network (Section 3.4). At each moment, the cloud (for computation and storage) receives the gaze and existing content from the edge (for rendering and interaction). Then, guided by our gaze-contingent and perception/content-aware model, the cloud automatically determines the streaming priority of each asset component based on their added perceptual quality and data size (with regard to a given network bandwidth). Figure 6 shows an overview of our system.

3.1 Modeling Spatio-temporal Vision

Spatial visual acuity. The human visual acuity is foveated, with a deterioration along age [9] and visual field eccentricity [28, 30]. The importance of a pixel $E(\mathbf{x})$ is determined by its distance to the retinal eccentricity $r = \sqrt{x^2+y^2}$. As the retinal eccentricity r increases, the importance function $E(\mathbf{x})$ decreases. Please refer to Appendix A for the details of $E(\mathbf{x})$.

Static stimuli. As shown in Figure 2, we model the perception of a static image by separating it into a number of frequency bands. The perception for each frequency band $\mathbf{f}_i$ is the product of its bandpass-filtered contrast and its contrast sensitivity function. The foveated retinal band $B_r(\mathbf{g},\mathbf{x}) = E(\mathbf{g}-\mathbf{x})$ will also affect the perception, results in clamped peripheral part for high frequency content. Please refer to Appendix C for details.

Dynamic stimuli. In runtime, the level of content LoD constantly changes and the quality is improved gradually, because more detailed data is streamed continually with the available network bandwidth. However, abrupt changes within a small time period may cause popping artifacts. Similar to [40], we define the perceived *change* as a temporally adapted Weber's contrast considering short-term memory.

Change blindness. The gaze fixations and the dynamic transitioning motions among them also significantly alter the perceptual quality. Besides motion parallax [41] and smooth pursuit movements to keep tracking on objects, people perform very fast eye movements (a.k.a. saccades) to change among fixations. Due to the fast movement, the visual contrast sensitivity (c) is suppressed as studied in [19]. We ran a pilot study and validated the suppressed perception of the popping artifacts along with the sensitivity drop. Therefore, we adaptively model visual sensitivities w.r.t dynamic gaze behaviors including fixations and saccades, as detailed next in Section 3.2. Albert et al. [1] suggested for foveated rendering, an overall system latency of 50-70ms is acceptable when users are specifically tasked with finding artifacts, and the requirements shall be further relaxed for regular tasks. Although reading-introduced saccades can be as short as 20ms, both short- and long- peri-saccades are combined with post-saccade suppression that stretches the applicable duration to mitigate popping artifacts. Previous literature has shown that saccadic suppression may typically last for at least 100ms even with a 50ms saccade [17]. Thus, the allowable saccadic suppression durations contain modern mobile network latencies ($\sim$ 100ms for 3G) [43].

3.2 Model in Screen Space

Spatial acuity and quality. Due to foveated vision, we prioritize quality and details in the fovea over the periphery. Thus, the importance of a given pixel $\mathbf{x}$ under gaze position $\mathbf{g}$ is computed as:

$$\hat{P}_{ec}(\mathbf{g},\mathbf{x}) = E(\mathbf{g}-\mathbf{x}), \tag{1}$$

where E is the importance function defined in Equation (11) in Appendix A.

Temporal consistency. A major problem from traditional LoD-based procedural rendering is the visual popping effect [26]. That is, when the LoD level of an area receives an update, the abrupt visual changes may easily be noticed and distractive to the experience. The human visual system perceives LoD-introduced popping artifacts in spatial frequency and retinal velocity [40].

By extending the perceived *change* as a temporally adapted Weber's contrast considering short-term memory to individual frequency band, we obtain the approximated popping (i.e., perceived temporal intensities) between two varied frames (I and I' in the screen space):

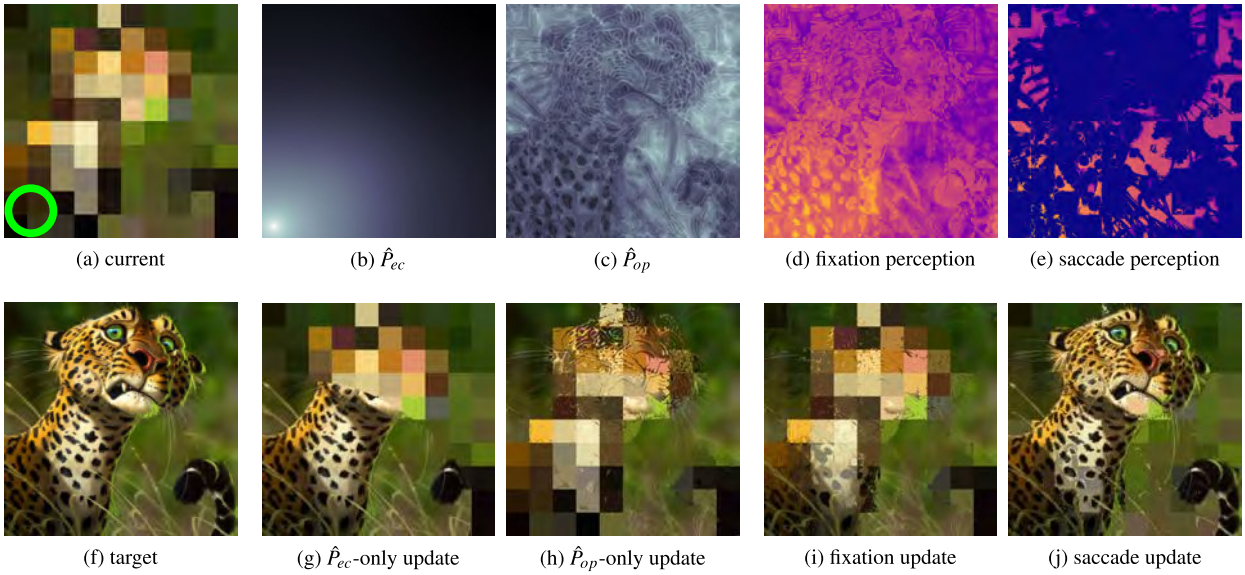
$$\hat{P}_{op}(\mathbf{g},I,I',\mathbf{x}) = \sum_{i=0}^{b-1} s(\mathbf{f}_i,L) \times \frac{|c(\mathbf{x},\mathbf{f}_i,I) - c(\mathbf{x},\mathbf{f}_i,I')|}{|c(\mathbf{x},\mathbf{f}_i,I)| + \omega}, \tag{2}$$

where $\mathbf{g}$ is the tracked gaze positions in I and I', respectively. ω is the controlling parameter that balances for low-intensity stimuli where Weber's law may fail. s is the contrast sensitivity function given the frequency band $\mathbf{f}_i$ and luminance L, as detailed in Equation (12) under Appendix B. The transition from I to I' can be from the changes of scene or camera. The corresponding coefficients are omitted here for the brevity of presentation. Although the spatial contrast sensitivity function is not reflecting the sensitivity of the human visual system to temporal changes, we have modeled the sensitivity temporal changes by introducing Weber's law on two temporally adjacent frames I and I'. c is the bandpass filtered point contrast given the frequency band $\mathbf{f}_i$, as detailed in Appendix C. It assumes slow gaze/head motion with fast (90FPS) frame update. Thus, $\mathbf{g}$ in the two frames are approximately identical. The next paragraph discusses cases when the gaze moves fast (saccade).

Adapting to dynamic gaze behaviors. As studied by [19], our visual sensitivity gets significantly suppressed during saccades. Due to the change blindness, we only perceive weak popping artifacts in this period. Thus, when saccades are detected, we can instead stream the most noticeable popping elements to reduce the popping intensity after it lands. This motivates our gaze-behavior-adaptive per-pixel sensitivity by combining both spatial acuity and temporal consistency models:

$$\hat{P}(\mathbf{g},I,I',\mathbf{x}) = \begin{cases} \hat{P}_{ec}(\mathbf{g},\mathbf{x}) - \lambda \hat{P}_{op}(\mathbf{g},I,I',\mathbf{x}) & \text{during fixation} \\ \int_{\mathbf{g}'\in I'} \hat{P}_{op}(\mathbf{g}',I,I',\mathbf{x})\mathrm{d}\mathbf{g}' & \text{during saccade} \end{cases}, \tag{3}$$

where λ is the balance between maximizing foveated perceptual quality and minimizing popping artifacts during fixation. A saccade is considered to happen if the gaze speed is greater than 180 deg/sec. Because of the open challenge in real-time saccade landing prediction [2], we integrate over the entire visual field while computing the popping for saccade instead of assuming the landing position. This ensures global robustness to any user's attentional changes. Figure 3 visualizes individual importance and the resulting image-space update.



(a) current (b) $\hat{P}_{ec}$ (c) $\hat{P}_{op}$ (d) fixation perception (e) saccade perception

(f) target (g) $\hat{P}_{ec}$-only update (h) $\hat{P}_{op}$-only update (i) fixation update (j) saccade update

Fig. 3. *Visualization of our perceptual model in 2D screen space.* Here each pixel is treated as a unit with its own LOD. (a) is a semi-transmitted low LoD image on the edge and the green circle indicates the gaze. (f) is the full LoD target on the cloud. (b) and (c) visualize the importance of the eccentricity in Equation (1) and the temporal consistency in Equation (2). Our model exploits both eccentricity and temporal consistency in Equation (3). (d) and (e) visualize the corresponding perceptual quality measures considering both static visual quality and dynamic consistency. The second row shows the corresponding screen space update.

We can adapt Equation (3) for progressive LoD update from level $i+j$ to i as follows:

$$\hat{P}(\mathbf{g}, I_{i+j}, I_i, \mathbf{x}) = \sum_{l=i}^{i+j-1} \hat{P}(\mathbf{g}, I_{l+1}, I_l, \mathbf{x}), \tag{4}$$

where I_i represents the image at the i-th LoD.

3.3 Mapping from 2D Screen to 3D Streaming

In this section, we present how to extend the screen-space model from Section 3.2 to 3D assets for a real-world cloud-based 3D streaming system. The 3D assets can have various representations, including triangle meshes, volumes, terrains, or large crowded objects, as shown in Figure 5.

Map to 3D assets. So far, our perceptual model depicting static quality and dynamic artifacts in Section 3.2 applies to individual pixels. The 3D assets, however, comprise non-uniformly distributed content such as depths and connectivities. We apply a deferred shading algorithm to convert various types of 3D primitives to 2D perception evaluations.

We divide the 3D content based on the coarsest level of LoD. We denote an individual computational unit as U_i, where i is its index among all units. It can be a coarsest triangle in a 3D mesh, a largest super-voxel in a volume, a texel in the coarsest mipmap level of a height/displacement texture, or a separate object in a swarm scene. We denote the LoD level of U_i at time frame t as $L_{U_i,t}$.

At time frame $t-1$ when the LoD levels of all units are determined and transmitted to edge already, we render a framebuffer at cloud side for the whole scene without anti-aliasing to retrieve the unit indices of every pixel, i.e., a mapping $M_{t-1} : \{\mathbf{x}\} \longmapsto \{U_i\}$ from the set of pixels $\{\mathbf{x}\}$ to the set of units $\{U_i\}$.

If the LoD level of U_i is updated to $L_{U_i,t}$ at time frame t, we approximate its sensitivity by accumulating all pixels $M_{t-1}(\mathbf{x}) = U_i$ of the unit at time frame $t-1$,

$$\begin{aligned}&\hat{P}_{U_i,t}\left(L_{U_i,t}, \mathbf{g}_{t-1}, M_{t-1}\right)\\ &\approx \sum_{\mathbf{x}\in M_{t-1}^{-1}(U_i)} \hat{P}\left(\mathbf{g}_{t-1}, I_{t-1}, \hat{I}_t\left|L_{U_i,t}\right., \mathbf{x}\right),\end{aligned} \tag{5}$$

where $\mathbf{g}_{t-1}$, M_{t-1}, and I_{t-1} are the gaze position, unit mapping, and render image at time frame $t-1$. Figure 4 visualizes the mapping. The mapping M implicitly represents the camera of each time frame. It also varies according to the LoD of all units. The approximation in Equation (5) simplified the evaluation by assuming $M_{t-1} = M_t$. $\hat{I}_t\left|L_{U_i,t}\right.$ is the render image at time frame t if the LoD level of U_i is $L_{U_i,t}$. We denote it with a hat because it is an approximation by assuming the LoD level of other units is not changed between time frames of $t-1$ and t. $\hat{P}_{U_i,t}$ is also dependent on the LoD levels of other units $\left\{L_{U_j,t-1}\,|\,j \neq i\right\}$, which we omit in Equation (5) for brevity. The evaluation of $\hat{P}_{U_i,t}$ in Equation (5) is computationally expensive, so we propose a neural evaluation, as will be detailed in Section 3.4.

(a) 2D heatmap (b) 3D heatmap (c) before (d) update

Fig. 4. *Visualization of mapping importance from 2D pixels to 3D triangles.* (a) shows the gaze-aware importance $\hat{P}$ in 2D. (b) shows mapped importance in 3D. (c) and (d) show the object before and after streaming, guided by (b). The green circle indicates the user's gaze fixation position.

Cloud-based streaming. Finally, we evaluate the perceptual quality *per bit* by updating the LoD level of a unit,

$$W_{U_i,t}(L_{U_i,t}) = \frac{\hat{P}_{U_i,t}\left(L_{U_i,t}, \mathbf{g}_{t-1}, M_{t-1}\right)}{D_{U_i}(L_{U_i,t-1}, L_{U_i,t})}, \tag{6}$$

where D is the data volume difference by updating U_i's LoD level from $L_{U_i,t-1}$ at time $t-1$ to $L_{U_i,t}$ at time t.

The edge side is a device with limited network bandwidth and storage. Therefore, the 3D scene is fully stored on cloud side, including all LoD levels of all units. In runtime at time frame $t-1$, the cloud side

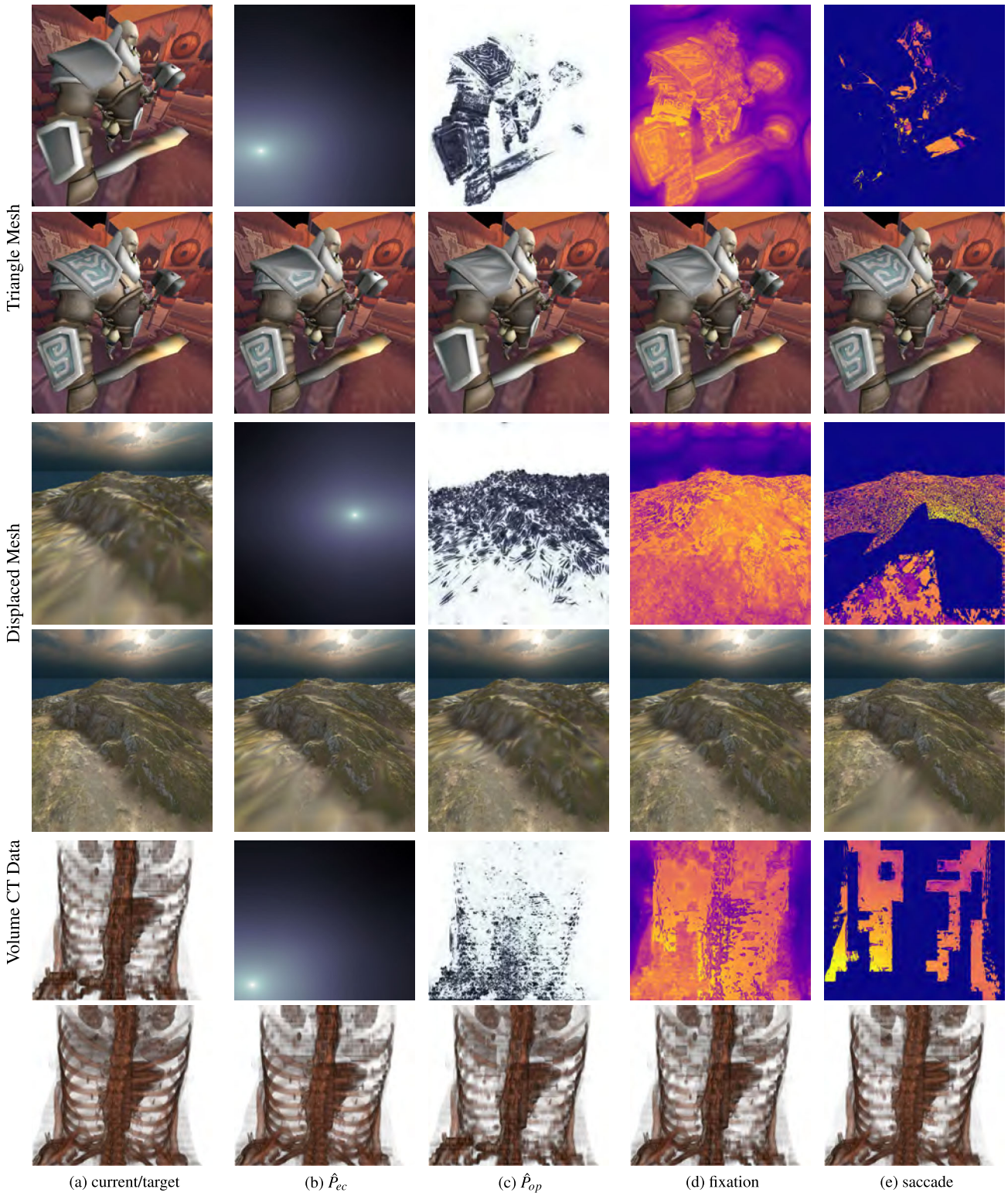


Fig. 5. *Results for various 3D data types.* From top to bottom, the 3D assets are triangle mesh, displacement-based terrain, and CT scanned volume. In each group of images, (a) shows the semi-transmitted scenes at edge on top and full scenes at cloud on bottom. (b)/(c) is the importance and corresponding updated rendering by considering only eccentricity $\hat{P}_{ec}$/temporal consistency $\hat{P}_{op}$. Brighter means higher importance. (d)/(e) is the update if the gaze fixes/saccades considering both (b)$\hat{P}_{ec}$ and (c)$\hat{P}_{op}$. Similar to Figure 1, their first rows indicate estimated perceptual quality (brighter means stronger spatio-temporal artifacts or worse quality).

sends the gaze position $\mathbf{g}_{t-1}$. The cloud side knows the LoD levels of all units on the edge side as the edge side acknowledges the received network packages containing the 3D data. According to that, the cloud will compute the perceptual quality per bits for updating LoD levels of every unit accordingly, with the updates holding highest Ws to achieve the best improvement of perceptual quality, given the constraint on the available update package size S.

$$\begin{aligned} &\underset{\{L_{U_i,t}\}}{\operatorname{argmax}} \sum_i W_{U_i,t}(L_{U_i,t}) \\ &s.t. \sum_i D_{U_i}(L_{U_i,t-1}, L_{U_i,t}) \leq S. \end{aligned} \tag{7}$$

3.4 Neural Acceleration

In a practical implementation, the cloud-side needs to compute Equation (5) for each update. Although the cloud can afford more computation than the edge, the heavy frequency domain decomposition for individual LoD and frames in Equation (16) may cause intolerable latencies on the end users. Thus, for each scene, we train a multilayer perceptron (MLP) neural network for the fast prediction of $\hat{P}_{U_i}$.

Specifically, with a camera view $\mathbf{c}_{t-1}$ and gaze position $\mathbf{g}_{t-1}$ as input, the network learns to compute

$$\mathscr{N}(\mathbf{c}_{t-1}, \mathbf{g}_{t-1}) = \left\{\hat{P}_{U_i}\left(L_{U_i,t}, \mathbf{g}_{t-1}, M_{t-1}\right)\right\} \tag{8}$$

for all U_i and possible $L_{U_i,t}$. The camera view $\mathbf{c}_{t-1}$ contains 3 vectors, which are coordinates of camera position, camera direction and up direction. In the input, there is also a flag indicating whether a saccade is detected. The current LOD state is not in the input since the network always predicts for all LODs, which will be used by the cloud to make the streaming decision. Although we do not provide the mapping M_{t-1} explicitly to the network, the network also infers it from $\mathbf{c}_{t-1}$ once trained on a specific scene. The diversity of projected areas of various units U_i cause $\hat{P}_{U_i}$ to have no upper bound, making it difficult to optimize the network. To combat this, we normalize $\hat{P}_{U_i}$ by the pixel counts of individual $\hat{P}_{U_i}$ in the projected screen space. After normalization, the value is bounded by the maximum value of Equation (2). The cloud stores the network and retrieves the fast-inferred $\hat{P}_{U_i}$ for streaming decision (Equations (6) and (7)), so the cloud doesn't need to render the actual image for the decision.

We report specific implementation details such as network architecture, loss function, and dataset generation in Section 4 and the performance/precision analysis in Section 6.2. Please also refer to our code repository for the actual implementation of the network structure. We evaluate both performance gains and prediction precision of our neural network in Section 6.2. Change the super parameters like the ω will require retrain the neural network. To avoid retraining the network when changing the λ, one solution is to train 2 networks to predict the $\hat{P}_{ec}$ and $\hat{P}_{op}$ separately.

4 Implementation

3D Assets. Our current method assumes we can set the LoD of each U independently from one another. While this works for asset types with more independent units such as point clouds [38, 39], volumes, and crowd agents, it might introduce artifacts for others, such as cracks or T-junctions for triangle meshes. To ensure quality without introducing complex implementation, we currently use vertex colors instead of textures, and build the mesh hierarchies by sub-sampling existing vertices without changing their positions. The displaced mesh (terrain scene from Figure 5) is handled in a similar way, except that additionally a height map is also sampled for vertices. For volume data, the LOD is created by sub-sampling the voxels.

Solving Equation (7) is equivalent to the knapsack problem, which is NP-Complete. For this reason, we use a greedy approximation, which always selects the unit with the largest $W_{U_i,t}(L_{U_i,t})$ in Equation (6) if the bandwidth allows so.

System. We simulate the streaming network via ZeroMQ library (`https://zeromq.org/`). The rendering system was implemented with OpenGL. The system runs on a PC with Intel i9-9880H 8-core CPU and with 32GB of RAM, and an NVIDIA RTX 2080 graphics card. For the VR headset, we use an HTC Vive Pro Eye with an integrated eye tracker. It has a 110 degree FOV and 1440×1600 spatial resolution (14 pixels per degree, i.e., 7 cycles per degree for B_d). Our system runs at 90-FPS with neural acceleration.

Neural acceleration. We prepare the dataset for the neural network in Section 3.4 by first sampling short sequences of camera and/or gaze movements inside a 3D scene at 90-FPS. The views are freely controlled by the HMD users. Then the offline computed (via Equation (5)) $\{(\mathbf{c}_{t-1}, \mathbf{g}_{t-1}), \hat{P}_{U_i}\left(L_{U_i,t}, \mathbf{g}_{t-1}, M_{t-1}\right)\}$ pairs are used to train the network. Since the adjacent frames are likely to have similar data, we sample 4 frames per second uniformly and compute the ground truth $\hat{P}_{U_i}$ for all U_i and LOD levels offline. There are 16 sequences, and each sequence lasts for about 30 seconds and contains ~ 120 samples. We use 15 sequences as the training set and the remaining one as the test set.

The neural network is fully-connected and consists of 3 inner-layers with $(100, 1000, 1000)$ neurons, each with *ReLU* activations, and an output layer with a *Sigmoid* activation. The network was trained with *L1* loss for 100 epochs. In the training, we set the learning rate to 0.001 and batch size to 128. we used an SGD optimizer with momentum = 0.9. For the neural network in the user study (Section 5.1), since we have ~ 2000 samples and each sample contains 722 (triangles) $\times 4$ (LOD) ≈ 3000 scores. The sample size, higher than the parameter numbers, avoids biased overfitting. We refer the readers to our open source implementation for reproduction details.

Parameters. We determine the optimal spatio-temporal balancing λ via a pilot user study. Users reported 3.0 as the most plausible experience. In our experiment, the Weber's law adjustment for low intensities ω was set to 10. With low network bandwidths, the peripheral content may not receive the priority of transmission. Thus, we included a pedestal constant 2.0 to the $\hat{P}_{ec}$ for each pixel.

5 Evaluation: User Study

We compare different progressive streaming methods with different usages of perceptual mechanisms under the same network bandwidth - uniformly increasing the LoD of each U based on visibility (**UNI**), "foveated" streaming without considering dynamic consistency (**ECC**, a 3D version of [37]), and our method (**OURS**). We experimented with two display environments, active HMD-based 6 DoF (degrees of freedom) navigation (Section 5.1) for temporal consistency evaluation only and passive screen-based observation (Section 5.2) for evaluation of both visual temporal consistency and static visual quality.

5.1 Eye-Tracked Study

We first conducted a user study with an eye-tracked VR HMD to assess our method's effectiveness on perceived temporal consistency.

Stimulus and setup. We used a terrain displacement scene from Figure 5 in this experiment. Based on users' real-time head and gaze motions, the stimuli is a sequence of simulated progressive streaming updates (from a coarse to adaptively finer level). During the study, users wore an HTC Vive Pro Eye Headset and remained seated. They were instructed to freely observe the virtual scene and keep their fingers on the keyboard to make selections after each trial, as shown in Figure 7a. Eight users (age $23-31$, 1 female) participated in the study. To accommodate the scene data size and study duration, we experimented with a simulated 5G network bandwidth locally.

Task. Due to the limited access to participants during COVID-19, we aim to maximize trial randomness and numbers to ensure objective measurements. We designed a two-alternative-forced-choice (2AFC) experiment. Each trial consists of a pair of conditions among the three (**UNI/ECC/OURS**). In each condition, the stimulus was initiated with the coarsest LoD. The participants were then instructed to freely observe the environment that was streamed and updated with the corresponding method. Each condition lasted for 10 seconds. At the end of

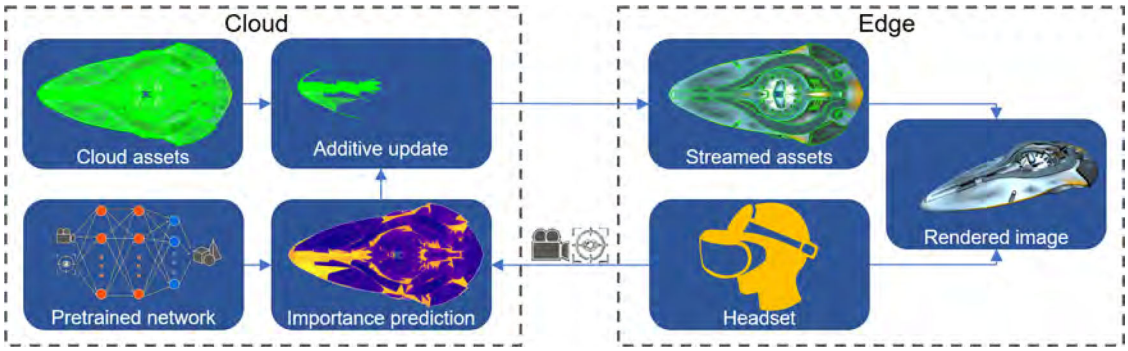


Fig. 6. *Overview of the system.* The parameters of the camera and gaze are sent to cloud, and they are used as the input of the pretrained neural network to predict the perceptual importance of primitives. The image is rendered on edge with the streamed assets.

(a) VR setting (b) Video setting

Fig. 7. *The setting of our user study.* Both (a) and (b) were captured from parts of our user studies.

each trial, the participants used the keyboard to select which one of the two conditions appeared more smoothly and comfortably updated with fewer artifacts over the entire duration.

Each experiment began with a warm-up session to familiarize the participants with the interface and task, followed by 24 counter-balanced and randomly ordered trials. Thus, each pair of conditions was evaluated 8 times per participant. To minimize fatigue, a 2-second break was enforced between trials.

Results. Figure 8a plots the results of the pair-wise percentages that participants indicated as the preferred method. Specifically, **OURS** was voted as significantly more preferred than both **ECC** ($90.6\% \pm 8.3\%$) and **UNI** ($92.2\% \pm 12.4\%$). With a significance level established at $p = 0.05$ and power $> .999$, binomial tests indicate the difference is significant for both comparisons ($p < .001$). **ECC** showed marginally higher voting rate than **UNI** ($56.3\% \pm 28.6\%$). We did not observe significance with a binomial test ($p = .38$).

Discussion. The results above showed our method's significant temporal quality preference over the alternative solutions: under the same network conditions, our method could stream 3D assets more smoothly by reducing popping artifacts. However, this shall be achieved without compromising spatial-visual quality. Our initial study design considered several evaluation metrics, including visual fidelity to the reference, temporal smoothness, and perceptual comfort. We planned to evaluate them both individually and collectively: ask participants to focus on individual metric (such as visual fidelity or temporal smoothness only), and additional post-study comments (such as why they choose one condition over another).

However, during our pilot runs, we noticed several problems with this initial design. Some participants commented that it was difficult for them to focus on more than one aspect of the evaluation criteria, especially for those with less VR or gaming experiences. Meanwhile, visual quality depends on the temporal integration of multiple instead of individual frames. The integration also depends on the exact view path, which can be freely moved by the participants and thus may differ across pairs of conditions for a valid comparison. To address these issues, we decided to focus on only temporal smoothness, which tends to be less view-path dependent than spatial quality, for this free navigation HMD experiment, and leave per-frame visual quality evaluation by fixing the gazes, as described in the following experiment.

5.2 Screen-Based Study

Due to the aforementioned design challenges and extra difficulties of physically distributing the eye-tracked VR display, we further simulate the study procedure in Section 5.1 to a remote setting with recorded stimuli. This ensures a thorough evaluation of both temporal smoothness and spatial quality.

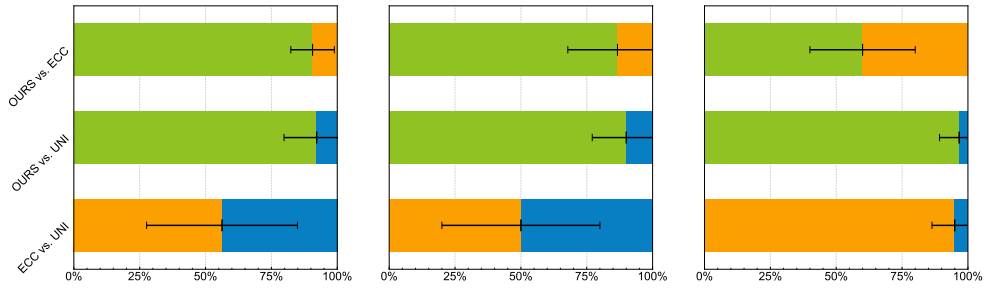
Specifically, we recorded videos by randomly pre-defined gaze paths. During the study, users were remotely monitored and instructed to keep one eye open, fix their head positions, and gaze on the green cross and follow the same procedure as Section 5.1.

Stimulus and setup. To evaluate a different data type from the eye-tracked study, we chose the triangle mesh scene in Figure 5 for this study. The visual stimuli were rendered with 1920×1080 resolution and 60 degree of vertical field of view, so that the B_d is the same for every participant. In the monitored remote study, participants first input their computer resolution and screen size. Our study protocol would automatically compute and inform participants of the correct eye-display distance. The user then fixed their head at the reported distance cross accordingly. Twelve users (age $24-41$, 5 females) participated in this study. All users had normal or corrected-to-normal vision.

Tasks. The experiment was conducted in two phases: video-based stimuli for temporal smoothness and static frame-based stimuli for visual quality. The task of the *first* phase was similar to Section 5.1. The users observed two sequentially played stimuli from the three conditions. After each trial, they selected the one with preferred temporal consistency (i.e., fewer artifacts). In the *second* phase, each trial consisted of two static images randomly sampled from the sequences in the first phase at the same timestamp. In each trial, a full-quality rendering (second row of Figure 5a) was first shown for 3 seconds as the quality reference. Then, 2 images of the 3 conditions were sequentially displayed for 3 seconds. After each trial, the participants selected the one that appeared to be closer to the reference in visual quality. Both tasks contain 15 counter-balanced and randomly ordered trials. Thus, each pair of conditions was evaluated 5 times per participant.

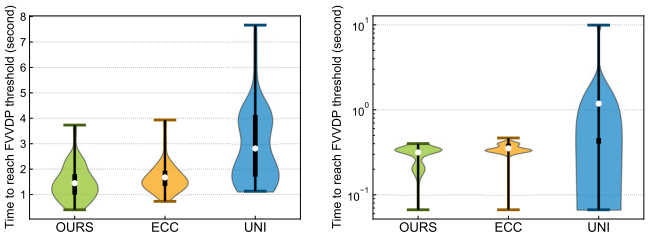
Task Design Rationale. Fixing gaze in 2AFC experiment has been practiced in assessing retinal acuity and perceived visual quality in VR [30, 48, 50]. In fact, the human visual perception is limited during natural, active viewing conditions [6]. Further, because of the suppressed vision during saccades, the visual quality is primarily determined by the frames at gaze fixations. For this reason, the current experiment where the gaze is fixed may generally represent free viewing conditions where fixations are connected by saccades.

Results. Figures 8b and 8c plot the results. For temporal consistency (phase-1), a consistent trend to Section 5.1 was observed: **OURS** had a significant higher ration of voting over the alternatives ($86.7\% \pm 18.8\%$ against **ECC** and $90.0\% \pm 12.9\%$ against **UNI**). With a significance level established at $p = 0.05$ and power $> .999$, binomial tests indicate the difference is significant for both comparisons



(a) VR eye-tracked temporal consistency (b) video-based temporal consistency (c) video-based quality

Fig. 8. *User study results.* We compare pair-wise 2AFC user vote percentage in both HMD and traditional display studies from Section 5. The green/orange/blue bar indicates the percentage voting for **OURS**/**ECC**/**UNI** and the errorbars indicates the standard deviations.



(a) static scene (b) dynamic scene

Fig. 9. *The statistics of the time FOVVideoVDP first reaches threshold per sequence among all users (Section 6.1).* The sub-figure (a)/(b) shows the results with static/dynamic scenes. The violin plot shows mean values, 1st/3rd quartiles, and whiskers min/max. Lower values indicate better streaming efficiency in additively improving the quality.

($p < .001$). **ECC**, however, did not show major preference gain over **UNI** ($50.0\% \pm 30\%, p = 1.0$).

On the extended visual quality (phase-2), marginally higher voting rate on **OURS** was observed over **ECC** ($60.0\% \pm 20\%, p = .16$). On the other hand, with a significance level established at $p = 0.05$ and power $> .999$, both **OURS** and **ECC** were voted to have significantly improved quality than **UNI** ($96.7\% \pm 7.4\%$ and $95.0\% \pm 8.7\%$ respectively, $p < .001$ for both).

Discussion. This study simulates and extends Section 5.1 to a remote screen-based setting. Beyond the agreeing observation of temporal consistency, the phase-2 study also indicated that **OURS** preserves the high visual fidelity as **ECC**. The experiments with both eye-tracked VR and traditional monitors show that our method significantly improves the perceived temporal consistency (i.e., minimizing popping artifacts) without compromising the visual quality.

6 Evaluation: Objective Analysis

6.1 Visual Quality

Section 5 shows our significant benefits of subjective quality judgement in terms of temporal consistency and artifact reduction. For further objective evaluation considering dynamic gaze adaptation stimuli (Equation (3)), we conducted a series of analytical experiments.

Static scene. Recently, a thoroughly considered metric, FovVideoVDP [28], was proposed to measure video quality considering both spatio-temporal and foveated effects. We compute the FovVideoVDP with rendering on the fully transmitted asset as a reference. The analysis comprises full 10-second sequences from each user (one sample from each tri-condition group, 8×8 in total) of the eye-tracked study. Since edge-cloud streaming is a temporally procedural quality enhancement, we measure the timing when FovVideoVDP reaches a shared threshold. Intuitively, this measures the efficiency of the streaming achieving high quality. In the experiment, we chose the threshold as 6.5 to ensure all conditions, even the slowest, can reach it with the limited trial durations. As shown in Figure 9a, the average time of **OURS** (1.446 ± 0.669) and **ECC** (1.674 ± 0.589) are significantly shorter than **UNI** (2.813 ± 1.429). Pairwise t-tests show the difference between all pairs are significant ($p < .001$ for **OURS** vs. **UNI** and **ECC** vs. **UNI**, $p = .032$ for **OURS** vs. **ECC**).

Dynamic scene. We further preformed an experiment on a dynamic scene which consists of an animation of a pile of falling soft balls as shown in Figure 10. We refer the complementary video for full animation. The gaze is from the eye-tracked study with the same setting as the static scene study, except that the dynamic assets and threshold as 6.7. As shown in Figure 9b, the average time of **OURS** (0.316 ± 0.069) and **ECC** (0.352 ± 0.049) are significantly shorter than **UNI** (1.184 ± 2.548). Pairwise t-tests show the difference between all pairs are significant ($p = .008$ for **OURS** vs. **UNI**, $p = .011$ for **ECC** vs. **UNI**, and $p < .001$ for **OURS** vs. **ECC**).

The above analysis on the eye-tracked gaze data revealed a consistent observation to the subjective studies in Section 5. That is, **OURS** and **ECC** deliver high visual quality more efficiently than **UNI**. Meanwhile, comparing with **ECC**, **OURS** does not compromise the visual quality with the benefits of improved temporal consistency.

6.2 System Performance

Performance gain from neural acceleration. We conducted a performance analysis of the neural acceleration effectiveness. With all the scenes and gaze trajectories, it takes an average of 389.28 ± 14.44 ms without the neural acceleration. In comparison, the neural network accelerates the computation to 20.30 ± 0.89 ms.

Cloud/edge latency. Large-scale (e.g., 3D) data transmission inevitably introduces latency between the cloud and the edge due to the underlying network. That is, the gaze analyzed in the cloud may be outdated when the edge receives the additive asset. To validate the impact under real-world network conditions, we conduct a simulated experiment with varied transmission speeds ranging from 3G to 5G+. With a recorded gaze motion trajectory, we compute the

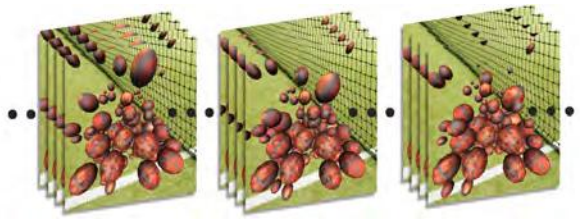
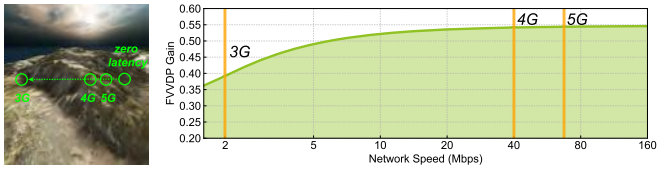


Fig. 10. *Example frames of the dynamic scene.* The animation is composed of free-falling soft balls. Each image represents a frame along time. Please refer to the supplementary video for the full visualization.



(a) illustration (b) quality gain w.r.t. bandwidth

Fig. 11. *Extendability test with varied network bandwidths.* (a) shows a gaze trajectory from right to left, starting at the position circled by *zero latency*. The 3G/4G/5G circles indicate the gaze positions for which the required data packets arrive. (b) shows the FoVVideoVDP gain of **OURS** from **UNI** along with varied simulated network conditions. The significant quality elevation in **OURS** is only subtly affected by gaze transmission latency when the speed is lower than 10Mbps.

FoVVideoVDP gain between **OURS** and **UNI**. By assuming 100KB packets, the approximated network speeds for 3G (∼2Mbps) / 4G (∼40Mbps) / 5G (∼67Mbps) are referred from [34, 36]. As seen in Figure 11, **OURS** shows significantly elevated spatio-temporal quality (i.e., higher FVVDP) compared with **UNI** under all network conditions. The elevation slightly decreases when network speed is slower than 10Mbps. The analysis demonstrates our approach's consistent outperform than **UNI** under both modern (4G and 5G) and legacy (3G - 4G) network conditions.Using the same data, we further perform a pressure experiment with artificially created latencies (from e.g., pings, database retrieval, etc.) with 3G/4G/5G network speeds. Figure 12 shows the result. Our method is only subtly affected by latencies across all conditions. Under 4G and 5G network, **OURS** can reach 0.5 in JODs compared with **UNI** when the latency is lower than 100ms. The 4G and 5G results are almost identical because their speeds are close.

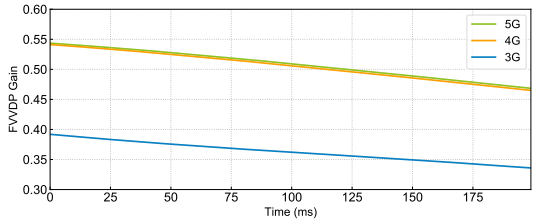


Fig. 12. *Pressure test with artificially introduced network latencies.* X-/Y-axis indicates the introduced latency/corresponding average FVVDP values. Across all network conditions, the low FVVDP changes demonstrate our method's robust benefits under real-world scenarios.

Approximation errors. Unlike 2D frame streaming, our system faces challenges of the complex 3D structures including connectivity and sparsity. Although our method analytically transforms screen-space perceptual models to 3D (Section 3.3) with neural acceleration (Section 3.4), each of them may introduce numerical loss. In this section, we also analyzed the potential error. Specifically, we compute the MSE loss among the image-based per-pixel update, per-triangle update with/without neural acceleration. Figure 13 visualizes an example sequence. The experiments showed that the loss is relatively minor and worth the performance gain enabled by the neural network.

7 Conclusion

We present a method for high-quality 3D immersive streaming with gaze-contingent perceptual optimization. Compared with 2D frame-based streaming systems, our 3D streaming method enables low-latency interaction, low cloud overload, and consistent viewing. Our evaluation demonstrates that our system delivers a statistically significant reduction of temporal artifacts without compromising the visual quality.

Limitations and Future Work. The two main parts of our method, modeling human perception for visual importance in 2D, and mapping

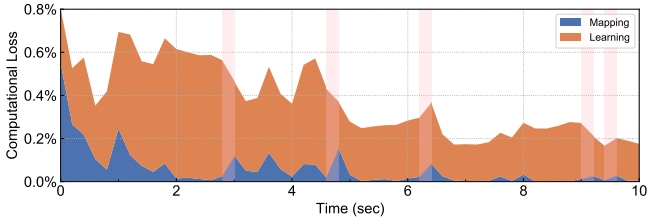


Fig. 13. *Approximation error visualization.* Areas highlighted by pink indicates that there is a saccade. Due to the 2D to 3D mapping (Section 3.3, blue area) and neural acceleration (Section 3.4, orange area), the final importance prediction may contain approximation errors. However, their overall error is below 1%.

the 2D importance to 3D for rendering and streaming, are largely orthogonal, as various perceptual factors can be considered for the 2D importance, before mapping it to 3D (Section 3.3). This paper focuses on combining foveation and saccade as the main perceptual mechanisms to optimize spatial quality and temporal smoothness (Equation (3)). Other possibilities include applying other perceptual phenomena to compute 2D importance, such as color, saliency/attention [14, 55], and masking [10, 11, 23], as well as applying our 2D importance estimation for streaming 2D and 360-degree videos [8]. The modeling, implementation, and evaluation of all these possibilities are promising future works, beyond the scope of a single paper.

In our system and analysis, we presumed an ideal and stable network environment without bandwidth sharing. Moreover, the data transmission was considered as individual packages of 100KB without compression and decomposition between the cloud and the edge [7, 24]. As shown in Section 6.2, all these factors may introduce additional latencies of the timely gaze tracking data sharing between the cloud and the edge, causing approximation errors. There are several super parameters in our method like the spatio-temporal balancing λ and the Weber's law adjustment for low intensities ω. One interesting future work is to adaptively optimize these super parameters for each scene.

The neural network's prediction precision is shown to support local 6DoF motions, as shown in Section 6.2. However, its capability of supporting very large motions such as in a flight simulation may require larger training data volume. Similarly, they are currently trained with static scenes. Predicting high accuracy results would also increase data samples due to introducing time as an additional input dimension.

To address the quality drop due to the gaze information latency, we foresee gaze motion prediction with machine learning may shed light on further reducing the perceptual latency in low bandwidth and unstable networks. In our current method, the rendering updates immediately upon receiving the relevant assets. Our framework *mitigates* the perceived flickering. For instance, during saccadic movement, we reduce the future (after the gaze lands) perceived flickering by taking advantage of the change blindness. However, targeting at progressive streaming systems, the temporal artifacts are not to be completely eliminated. Instead, a locally adaptive rendering may increase temporal consistency at the cost of delayed full quality. Recent research on balancing quality and latency [25] could make the system adaptable to various users.

One important application scenario that may require 3D assets (instead of 2D frames) is AR. High resolution and high FoV AR displays are yet to be available. Thus, we conducted the experiments with VR. In the future, considering not only virtual assets but also its interaction with physical surroundings may inspire more efficient transmission.

References

[1] R. Albert, A. Patney, D. Luebke, and J. Kim. Latency requirements for foveated rendering in virtual reality. *ACM Trans. Appl. Percept.*, 14(4), Sept. 2017. doi: 10.1145/3127589

[2] E. Arabadzhiyska, O. T. Tursun, K. Myszkowski, H.-P. Seidel, and P. Didyk. Saccade landing position prediction for gaze-contingent rendering. *ACM Trans. Graph.*, 36(4), July 2017. doi: 10.1145/3072959.3073642

[3] P. G. J. Barten. Evaluation of subjective image quality with the square-root integral method. *J. Opt. Soc. Am. A*, 7(10):2024–2031, Oct 1990. doi: 10.1364/JOSAA.7.002024

[4] R. Brown, V. DuTell, B. Walter, R. Rosenholtz, P. Shirley, M. McGuire, and D. Luebke. Efficient dataflow modeling of peripheral encoding in the human visual system, 2021.

[5] A. Bruckert, M. Christie, and O. L. Meur. Where to look at the movies : Analyzing visual attention to understand movie editing, 2021.

[6] M. A. Cohen, T. L. Botch, and C. E. Robertson. The limits of color awareness during active, real-world vision. *Proceedings of the National Academy of Sciences*, 117(24):13821–13827, 2020. doi: 10.1073/pnas.1922294117

[7] D. Cohen-Or, Y. Mann, and S. Fleishman. Deep compression for streaming texture intensive animations. In *SIGGRAPH '99*, pp. 261–267, 1999. doi: 10.1145/311535.311564

[8] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski. Viewport-adaptive navigable 360-degree video delivery. In *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, 2017. doi: 10.1109/ICC.2017.7996611

[9] D. B. Elliott, K. C. Yang, and D. Whitaker. Visual acuity changes throughout adulthood in normal, healthy eyes: seeing beyond 6/6. *Optometry and vision science : official publication of the American Academy of Optometry*, 72 3:186–91, 1995.

[10] J. A. Ferwerda, P. Shirley, S. N. Pattanaik, and D. P. Greenberg. A model of visual masking for computer graphics. In *SIGGRAPH '97*, pp. 143–152, 1997.

[11] W. Griffin and M. Olano. Evaluating texture compression masking effects using objective image quality assessment metrics. *IEEE Transactions on Visualization and Computer Graphics*, 21(8):970–979, 2015.

[12] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. Foveated 3d graphics. *ACM Trans. Graph.*, 31(6), Nov. 2012. doi: 10.1145/2366145.2366183

[13] C. Gutterman, B. Fridman, T. Gilliland, Y. Hu, and G. Zussman. Stallion: Video adaptation algorithm for low-latency video streaming. In *Proceedings of the 11th ACM Multimedia Systems Conference*, MMSys '20, p. 327–332. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3339825.3397044

[14] S. Hillaire, A. Lecuyer, T. Regia-Corte, R. Cozot, J. Royan, and G. Breton. Design and application of real-time visual attention model for the exploration of 3d virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):356–368, 2012. doi: 10.1109/TVCG.2011.154

[15] J. Hladky, H.-P. Seidel, and M. Steinberger. The camera offset space: Real-time potentially visible set computations for streaming rendering. *ACM Trans. Graph.*, 38(6), Nov. 2019. doi: 10.1145/3355089.3356530

[16] P. Hu, Q. Sun, P. Didyk, L.-Y. Wei, and A. E. Kaufman. Reducing simulator sickness with perceptual camera control. *ACM Trans. Graph.*, 38(6), Nov. 2019. doi: 10.1145/3355089.3356490

[17] M. Ibbotson and B. Krekelberg. Visual perception and saccadic eye movements. *Current Opinion in Neurobiology*, 21(4):553–558, 2011. Sensory and motor systems. doi: 10.1016/j.conb.2011.05.012

[18] 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), Nov. 2019. doi: 10.1145/3355089.3356557

[19] J. Knoell, P. Binda, M. C. Morrone, and F. Bremmer. Spatiotemporal profile of peri-saccadic contrast sensitivity. *Journal of vision*, 11(14):15–15, 2011.

[20] R. Konrad, A. Angelopoulos, and G. Wetzstein. Gaze-contingent ocular parallax rendering for virtual reality. *ACM Trans. Graph.*, 39, 2020.

[21] G. A. Koulieris, K. Akşit, M. Stengel, R. K. Mantiuk, K. Mania, and C. Richardt. Near-eye display and tracking technologies for virtual and augmented reality. *Computer Graphics Forum*, 38(2):493–519, 2019. doi: 10.1111/cgf.13654

[22] B. Krajancich, P. Kellnhofer, and G. Wetzstein. Optimizing depth perception in virtual and augmented reality through gaze-contingent stereo rendering. *ACM Trans. Graph.*, 39, 2020.

[23] G. Lavoué, M. Langer, A. Peytavie, and P. Poulin. A psychophysical evaluation of texture compression masking effects. *IEEE transactions on visualization and computer graphics*, 25(2):1336–1346, 2018.

[24] M. Levoy. Polygon-assisted jpeg and mpeg compression of synthetic images. In *SIGGRAPH '95*, pp. 21–28, 1995. doi: 10.1145/218380.218392

[25] M. Li, Y. Wang, and D. Ramanan. Towards streaming perception. In *ECCV*, 2020.

[26] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner. *Level of Detail for 3D Graphics*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[27] L. R. Luidolt, M. Wimmer, and K. Krösl. Gaze-dependent simulation of light perception in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3557–3567, 2020. doi: 10.1109/TVCG.2020.3023604

[28] 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), July 2021. doi: 10.1145/3450626.3459831

[29] H. Murphy and A. Duchowski. Gaze-contingent level of detail rendering. *EuroGraphics*, 2001, 01 2001.

[30] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Trans. Graph.*, 35(6), Nov. 2016. doi: 10.1145/2980179.2980246

[31] E. Peli. Contrast in complex images. *J. Opt. Soc. Am. A*, 7(10):2032–2040, Oct 1990. doi: 10.1364/JOSAA.7.002032

[32] S. Petrangeli, G. Simon, H. Wang, and V. Swaminathan. Dynamic adaptive streaming for augmented reality applications. In *2019 IEEE International Symposium on Multimedia (ISM)*, pp. 56–567, 2019.

[33] C. Párraga, T. Troscianko, and D. Tolhurst. The effects of amplitude-spectrum statistics on foveal and peripheral discrimination of changes in natural images, and a multi-resolution model. *Vision Research*, 45(25):3145 – 3168, 2005. doi: 10.1016/j.visres.2005.08.006

[34] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan. Beyond throughput, the next generation: A 5g dataset with channel and context metrics. In *MMSys '20*, pp. 303–308, 2020. doi: 10.1145/3339825.3394938

[35] S. Rashidi, K. Ehinger, A. Turpin, and L. Kulik. Optimal visual search based on a model of target detectability in natural images. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds., *Advances in Neural Information Processing Systems*, vol. 33, pp. 9288–9299. Curran Associates, Inc., 2020.

[36] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen. Commute path bandwidth traces from 3g networks: Analysis and applications. In *Proceedings of the 4th ACM Multimedia Systems Conference*, MMSys '13, pp. 114–118. Association for Computing Machinery, New York, NY, USA, 2013. doi: 10.1145/2483977.2483991

[37] M. F. Romero-Rondón, L. Sassatelli, F. Precioso, and R. Aparicio-Pardo. Foveated streaming of virtual reality videos. In *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, pp. 494–497, 2018. doi: 10.1145/3204949.3208114

[38] S. Rusinkiewicz and M. Levoy. Qsplat: A multiresolution point rendering system for large meshes. In *SIGGRAPH '00*, pp. 343–352, 2000. doi: 10.1145/344779.344940

[39] M. Schütz, G. Mandlburger, J. Otepka, and M. Wimmer. Progressive real-time rendering of one billion points without hierarchical acceleration structures. *Computer Graphics Forum*, 39(2):51–64, 2020. doi: 10.1111/cgf.13911

[40] M. Schwarz and M. Stamminger. On predicting visual popping in dynamic scenes. In *APGV '09*, pp. 93–100, 2009. doi: 10.1145/1620993.1621012

[41] A. Serrano, I. Kim, Z. Chen, S. DiVerdi, D. Gutierrez, A. Hertzmann, and B. Masia. Motion parallax for 360° rgbd video. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1817–1827, May 2019. doi: 10.1109/TVCG.2019.2898757

[42] A. Serrano, V. Sitzmann, J. Ruiz-Borau, G. Wetzstein, D. Gutierrez, and B. Masia. Movie editing and cognitive event segmentation in virtual reality video. *ACM Trans. Graph.*, 36(4), July 2017. doi: 10.1145/3072959.3073668

[43] C. Serrano, B. Garriga, J. Velasco, J. Urbano, S. Tenorio, and M. Sierra. Latency in broad-band mobile networks. In *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, pp. 1–7, 2009. doi: 10.1109/VETECS.2009.5073642

[44] S. Shi, V. Gupta, and R. Jana. Freedom: Fast recovery enhanced vr delivery over mobile networks. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, p. 130–141. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3307334.3326087

[45] Y. Shuai, G. Petrovic, and T. Herfet. Olac: An open-loop controller for low-latency adaptive video streaming. In *2015 IEEE International Conference on Communications (ICC)*, pp. 6874–6879, 2015. doi: 10.

1109/ICC.2015.7249421
[46] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein. Saliency in vr: How do people explore virtual environments? *IEEE transactions on visualization and computer graphics*, 24(4):1633–1642, April 2018. doi: 10.1109/TVCG.2018.2793599
[47] M. Stengel, Z. Majercik, B. Boudaoud, and M. McGuire. A distributed, decoupled system for losslessly streaming dynamic light probes to thin clients. In *Proceedings of the 12th ACM Multimedia Systems Conference*, MMSys '21, p. 159–172. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3458305.3463379
[48] Q. Sun, F.-C. Huang, J. Kim, L.-Y. Wei, D. Luebke, and A. Kaufman. Perceptually-guided foveation for light field displays. *ACM Trans. Graph.*, 36(6), Nov. 2017. doi: 10.1145/3130800.3130807
[49] Q. Sun, A. Patney, L.-Y. Wei, O. Shapira, J. Lu, P. Asente, S. Zhu, M. Mcguire, D. Luebke, and A. Kaufman. Towards virtual reality infinite walking: Dynamic saccadic redirection. *ACM Trans. Graph.*, 37(4), July 2018. doi: 10.1145/3197517.3201294
[50] O. T. Tursun, E. Arabadzhiyska-Koleva, M. Wernikowski, R. Mantiuk, H.-P. Seidel, K. Myszkowski, and P. Didyk. Luminance-contrast-aware foveated rendering. *ACM Trans. Graph.*, 38(4), July 2019. doi: 10.1145/3306346.3322985
[51] D. R. Walton, R. K. D. Anjos, S. Friston, D. Swapp, K. Akşit, A. Steed, and T. Ritschel. Beyond blur: Real-time ventral metamers for foveated rendering. *ACM Trans. Graph.*, 40(4), jul 2021. doi: 10.1145/3450626.3459943
[52] L. Wang, R. Li, X. Shi, L.-Q. Yan, and Z. Li. Foveated instant radiosity. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 1–11, 2020. doi: 10.1109/ISMAR50242.2020.00017
[53] Y.-Z. Wang, A. Bradley, and L. N. Thibos. Aliased frequencies enable the discrimination of compound gratings in peripheral vision. *Vision Research*, 37(3):283–290, 1997. doi: 10.1016/S0042-6989(96)00160-5
[54] A. B. Watson. A formula for human retinal ganglion cell receptive field density as a function of visual field location. *Journal of Vision*, 14(7):15–15, 06 2014. doi: 10.1167/14.7.15
[55] A. B. Watson, A. J. Ahumada, and J. E. Farrell. Window of visibility: a psychophysical theory of fidelity in time-sampled visual motion displays. *J. Opt. Soc. Am. A*, 3(3):300–307, Mar 1986. doi: 10.1364/JOSAA.3.000300
[56] M. Weier, M. Stengel, T. Roth, P. Didyk, E. Eisemann, M. Eisemann, S. Grogorick, A. Hinkenjann, E. Kruijff, M. Magnor, et al. Perception-driven accelerated rendering. In *Computer Graphics Forum*, vol. 36, pp. 611–643, 2017.
[57] W. Zhang, F. Qian, B. Han, and P. Hui. Deepvista: 16k panoramic cinema on your mobile device. In *Proceedings of the Web Conference 2021*, WWW '21, p. 2232–2244. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3442381.3449829

A Spatial Visual Acuity

The human vision is foveated. Watson [54] proposed a formula that approximates midget ganglion cells density as a function of retinal eccentricity $r=\sqrt{x^2+y^2}$, where x and y are the eccentricity degree in horizontal and vertical direction respectively, for the meridian type m:

$$\rho(r,m) = 2\rho_{cone}\left(1+\frac{r}{41.03}\right)^{-1}\times \left[a_m\left(1+\frac{r}{r_{2,m}}\right)^{-2}+(1-a_m)\exp\left(-\frac{r}{r_{e,m}}\right)\right], \tag{9}$$

where $\rho_{cone}=14804.6deg^{-2}$ is the density of cone cell at fovea and $a_m, r_{2,m}, r_{e,m}$ are all fitting constants along the four meridians of the visual field. And the cell-wise spacing is calculated by [48]:

$$\sigma(x,y)=\frac{1}{r}\sqrt{\frac{2}{\sqrt{3}}\left(\frac{x^2}{\rho(r,1)}+\frac{y^2}{\rho(r,2)}\right)} \tag{10}$$

Then, given the eccentricity, the importance function E of a given eccentricity $\mathbf{x}=(x,y)$ is modeled as [48]:

$$E(\mathbf{x})=0.5\sigma(\mathbf{x})^{-1}. \tag{11}$$

B Perceiving Static Stimuli

The visual sensitivity s of a certain frequency f and illumination L is determined by [3]:

$$s(f,L)=afe^{-bf}\sqrt{1+ce^{bf}}, \tag{12}$$

where $a=\frac{540(1+0.7/L)^{-0.2}}{1+\frac{1}{1+f/3}}$, $b=0.3(1+100/L)^{0.15}$ and $c=0.06$.

Because illumination of commodity displays is largely insensitive to the displaying content, we assume a constant illumination L for all frequencies. Given an image I, its visual sensitivity is the integral of all frequencies weighted by their amplitude,

$$\int s(|\mathbf{f}|,L)\,|A(\mathbf{f})|\mathrm{d}\mathbf{f} \tag{13}$$

where $\mathbf{f}=(f_x,f_y)$ is the two-dimensional frequency of I, and A is the amplitude of $\mathbf{f}$.

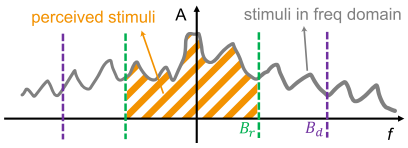


Fig. 14. *Perceived static visual stimuli.* The perceived stimulus (crossed area) is an integration of frequency amplitudes. Its integral domain is bounded by the densities of display pixels (B_d) and retinal receptors (B_r).

As shown in Figure 14, because of the finite display angular resolution and the finite retinal receptor densities, the non-zero frequency in Equation (13) lies in a finite domain as well,

$$\int_{|\mathbf{f}|<\min(B_d,\sup B_r)} s(|\mathbf{f}|,L)\,|A(\mathbf{f})|\mathrm{d}\mathbf{f}, \tag{14}$$

where B_d is the display band from the pixel density and eye-panel distance (0.5 cycle per pixel (CPP) in our display); $\sup B_r$ is the supremum of the foveated retinal band.

The amplitude A in Equation (14) will become spatially-variant if combined with foveated vision (Equation (9)). That requires the evaluation of spatially-variant spectrum, such as windowed Fourier or wavelet, which is prohibitively expensive in a real-time system. Therefore, we evaluate local contrast c, which can be more efficiently computed than A, of every pixel:

$$\Phi(\mathbf{g},I)\triangleq\int_{\mathbf{x}\in I}\overbrace{\int_{|\mathbf{f}|<\bar{B}(\mathbf{g},\mathbf{x})} s(\mathbf{f},L)\,c(\mathbf{x},\mathbf{f},I)\mathrm{d}\mathbf{f}}^{\hat{\Phi}(\mathbf{g},\mathbf{x},I)}\mathrm{d}\mathbf{x}, \tag{15}$$
$$\bar{B}(\mathbf{g},\mathbf{x})=\min\left(B_d,B_r(\mathbf{g},\mathbf{x})\right)$$

where c is the local contrast of $\mathbf{x}=(x,y)$ of I under the frequency $\mathbf{f}$. $\mathbf{g}=(g_x,g_y)$ is the tracked gaze position on the screen space. $\hat{\Phi}$ is the corresponding sensitivity value for a spatial position $\mathbf{x}$, thus the frequency is integrated within a narrower foveated retinal band $B_r(\mathbf{g},\mathbf{x})=E(\mathbf{g}-\mathbf{x})$ according to the gaze position and spatial posotion as shown in Equation (11). By applying clamping with $B_r(\mathbf{g},\mathbf{x})$, the eccentricity variances on the contrast sensitivity function are considered. Note that our model is for importance-based decision than pixel rendering so we assume the visibility of certain spatial frequencies is defined by receptor density. The peripheral vision can "reason" (other than directly perceive) higher-than-receptor-frequency information from the aliasing on the receptors [53]. Therefore, fully approximate the value by clamping the integration with the receptor frequency may cause loss of importance in the periphery. Thus, we introduced a constant ω in Equation (2) that compensates the peripheral region with low importance.

C BANDPASS FILTERING

To analytically compute the temporal consistency considering not only the content but the retinal receptors and display capability, we discretize the $\hat{\Phi}$ in Equation (15). Specifically, we perform a series of bandpass filtering of I to first obtain the gaze- and content-aware pixel-wise sensitivity:

$$\hat{\Phi}(\mathbf{g},\mathbf{x},I) \approx \sum_{i=0}^{b-1} s\left(\mathbf{f}_i,L\right) c(\mathbf{x},\mathbf{f}_i,I), \tag{16}$$

where we divide the frequency domain of integral in Equation (15) into b bands, where $\mathbf{f}_i = 2^{i\frac{\hat{p}(\mathbf{g},\mathbf{x})}{b}}$ are the representative frequencies located at the mid-point of every band (following [33]). I_i is the $\mathbf{f}_i$-filtered version of I. Figure 2 visualizes $\hat{\Phi}$ across different bands $\mathbf{f}_i$. The contrast c at point $\mathbf{x}$ is defined as

$$c(\mathbf{x},\mathbf{f},I) = \frac{a_{\mathbf{f}}(\mathbf{x},I)}{a_0(\mathbf{x},I)}. \tag{17}$$

Here $a_{\mathbf{f}}$ is the approximated local frequency of the $\mathbf{f}$-filtered I, as detailed in [31], and a_0 is $a_{\mathbf{f}}$ with $\mathbf{f}_i = 0$.