Locomotion-aware Foveated Rendering

Xuehuai Shi*
Beihang University

Lili Wang†
Beihang University
Peng Cheng Laboratory

Jian Wu‡
Beihang University
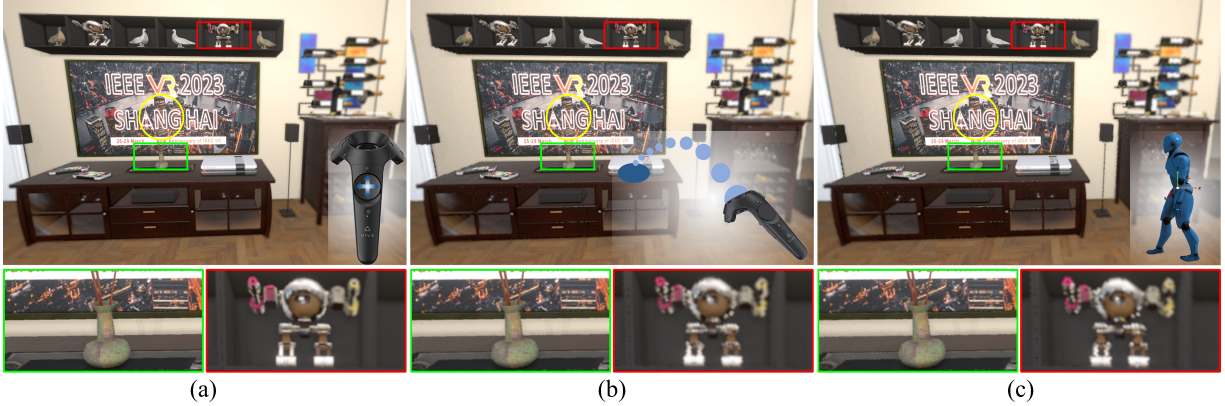
Wei Ke§
MPU

Chan-Tong Lam¶
MPU

Figure 1: Results of locomotion-aware foveated rendering when using (a) joystick moving rendered at 104 FPS, (b) teleportation rendered at 123 FPS, (c) redirected walking rendered at 134 FPS for scene exploration. In (a), (b) and (c), the foveal average shading rate control coefficient α and the peripheral shading rate decrease coefficient β are set to 0.85 and 0.7, the overall shading quantity control coefficient δ is set to 2.0, 2.2 and 2.4 respectively.

ABSTRACT

Optimizing rendering performance improves the user's immersion in virtual scene exploration. Foveated rendering uses the features of the human visual system (HVS) to improve rendering performance without sacrificing perceptual visual quality. We collect and analyze the viewing motion of different locomotion methods, and describe the effects of these viewing motions on HVS's sensitivity, as well as the advantages of these effects that may bring to foveated rendering. Then we propose the locomotion-aware foveated rendering method (LaFR) to further accelerate foveated rendering by leveraging the advantages. In LaFR, we first introduce the framework of LaFR. Secondly, we propose an eccentricity-based shading rate controller that provides the shading rate control of the given region in foveated rendering. Thirdly, we propose a locomotion-aware log-polar mapping method, which controls the foveal average shading rate, the peripheral shading rate decrease speed, and the overall shading quantity with the locomotion-aware coefficients based on the eccentricity-based shading rate controller. LaFR achieves similar perceptual visual quality as the conventional foveated rendering while achieving up to $1.6\times$ speedup. Compared with the full resolution rendering, LaFR achieves up to $3.8\times$ speedup.

Keywords: Virtual Reality, Foveated Rendering, Gaze-contingent Rendering, Perception

1 INTRODUCTION

The rendering performance significantly affects the user's immersion in virtual reality (VR) scene exploration. Users are able to detect

*e-mail: shixuehuaireal@buaa.edu.cn
†corresponding author, e-mail: wanglily@buaa.edu.cn
‡e-mail: lanayawj@buaa.edu.cn
§e-mail: wke@mpu.edu.mo
¶e-mail: ctlam@mpu.edu.mo

changes in smoothness in VR which lies at over 90Hz [26], posing a challenge to current rendering algorithms. Foveated rendering provides a possible solution to this challenge. Foveated rendering is a rendering acceleration technique that improves the performance of rendering algorithms without sacrificing the perceptual visual quality by leveraging the capabilities and limitations of the human visual system (HVS).

Users' viewing motion changes when they use different locomotion methods to explore scenes in VR. The sensitivity of HVS varies with different viewing motion. Thus, the requirements of the rendering quality vary with different locomotion methods. For example, if the frequencies of the user's saccade or head rotation reach a certain extent, the rendering quality of the foveal region can be reduced, i.e., the average shading rate in the foveal region can be reduced without sacrificing the perceptual visual quality. If the unconscious gaze movement occurs or the head rotates frequently, the expansion of the foveal region, i.e., slowing the decrease speed of shading rate in the peripheral region can improve the perceptual visual quality.

Log-polar mapping foveated rendering framework (LMFR) [30] supports the linear decrease of the shading rate from the fovea to the periphery. But LMFR cannot parametrically control the average shading rate in the foveal region, and cannot slow down the shading rate decrease speed in the peripheral region for different locomotion methods, so can't further improve the foveated rendering performance without reducing the perceptual visual quality.

In this paper, we propose the locomotion-aware foveated rendering method (LaFR) for scene exploration using different locomotion methods to further accelerate rendering performance without sacrificing the perceptual visual quality. Firstly, we analyze the viewing motion of scene exploration using different locomotion methods, which demonstrates the difference of viewing motion under different locomotion methods, and provides data support for LaFR. Secondly, we describe the effects of viewing motion on HVS's sensitivity, and the advantages that these effects may bring to LaFR, so as to provide the theoretical basis of LaFR. Thirdly, we propose the locomotion-aware foveated rendering method, which includes the framework

of LaFR based on LMFR, the eccentricity-based shading rate controller to provide the shading rate control for the given regions of the foveated rendering, and the locomotion-aware log-polar mapping method to control the foveal average shading rate, the peripheral shading rate decrease speed, and the overall shading quantity with the locomotion-aware coefficients. At last, we perform user studies to optimize the locomotion-aware coefficients of LaFR for minimizing computational resources without sacrificing the perceptual visual performance, and evaluate the perceptual visual quality of LaFR.

Figure 1 shows the rendering results of LaFR that achieve similar perceptual visual quality compared with the conventional foveated rendering method when exploring scenes using three different locomotion methods. α and β are set to 0.85 and 0.7 for all locomotion methods, and δ is set to 2.0, 2.2 and 2.4 under joystick moving, teleportation and redirected walking respectively. The yellow circles in the figure indicate foveal regions, and those green and red boxes are magnified at the bottom of the figure. The rendering quality in the green boxes at the edge of the fovea is comparable to that of the foveal regions for three locomotion methods. This is because LaFR reduces the foveal average shading rate and slows down the peripheral shading rate decrease speed, so as to maintain the details of objects at the edge of the fovea. But in the red boxes of the periphery, the rendering quality becomes lower from (a) to (c). This is because the sensitivity of HVS decreases gradually when using joystick moving, teleportation, and redirected walking for scene exploration. Thus LaFR can perform more aggressive foveated rendering without reducing the perceptual visual quality for the above locomotion methods. LaFR achieves 104-134 FPS under all three locomotion methods. Compared with LMFR, LaFR achieves 1.2-1.6$\times$ speedup with similar perceptual visual quality. Compared with the full resolution rendering, LaFR achieves 2.4-3.8$\times$ speedup.

In summary, the contributions of our method are as follows: 1) a viewing motion analysis for providing the data support and theoretical basis of locomotion-aware foveated rendering; 2) a locomotion-aware foveated rendering method (LaFR) that achieves up to 1.6$\times$ speedup compared with the conventional foveated rendering while maintaining similar perceptual visual quality; 3) user studies to optimize the locomotion-aware coefficients of LaFR, and evaluate the perceptual visual quality of LaFR.

2 RELATED WORK

In this section, we review the state-of-the-art research on multi-spatial resolution based foveated rendering and motion based rendering related to our work.

2.1 Multi-spatial Resolution based Foveated Rendering

The research on multi-spatial resolution based foveated rendering mainly includes multi-spatial resolution based ray tracing and multi-spatial resolution rasterization. For multi-spatial resolution based ray tracing, Fujita et al. [12] first implemented a foveated ray tracing system, which uses the pre-calculated sampling map to sample the sparse sampling of the current frame, and then uses the K-nearest neighbor algorithm to reconstruct the sparse sampling results for generating the final output image. Weier et al. [43] combined foveated ray tracing with the reprojection technique to reuse the ray samples of the previous frame to reduce the number of ray samples of the current frame. Koskela et al. [21, 22] proposed a foveated rendering method based on progressive Monte Carlo path tracing to accelerate the preview of the areas of interest. Tursun et al. [39] proposed a novel luminance-contrast-aware sampling scheme and integrated it into ray tracing to improve the computational savings. Kim et al. [19] proposed a perception-based sampling method for ray tracing in head-mounted displays (HMDs) by combining adaptive selective sampling technology [15] with foveated rendering.

For multi-spatial resolution based rasterization, Guenter et al. [13] proposed a foveated rasterization system, which rasterizes the images of the foveal region, transitional region and peripheral region with the guidance of a visual acuity fall-off model. Pattney et al. [33] proposed a foveated shading system that reduces 70% of shading calculation, and then introduced a new anti-aliasing algorithm to recover details in the peripheral region that can be resolved by HVS but are degraded by the filter. Meng et al. [30] proposed LMFR that can integrate foveated rendering into the shading pipeline efficiently. LMFR first transforms the coordinates of the screen space to the reduced-resolution log-polar space, and then performs lighting calculation in the log-polar space. Finally, the calculation results in the log-polar space are inversely transformed to the screen space to get the final output images. Turner et al. [38] aligned the rendered pixel grid in the peripheral region to the virtual scene during rasterization to reduce motion artifacts. To meet the field of view and real-time requirements of HMDs, Friston et al. [11] proposed a foveated rasterization pipeline that performs foveated rendering in one rasterization process by per-fragment ray casting. Meng et al. [29] reduced the rendering quality of non-dominant eye to further improve the foveated rendering performance without sacrificing the perceptual visual quality. Franke et al. [9] reused pixels in the periphery to improve the rendering performance by the re-projection technique. Walton et al. [41] proposed a real-time post-processing method to generate the foveated image that achieves similar perceptual visual quality as the ground truth image. Ye et al. [45] proposed a rectangular mapping-based foveated rendering method based on the deferred rendering pipeline to improve the foveated rendering quality.

Recently, research on foveated rendering has taken new directions, including new data types and new methods. Sun et al. [37] proposed a 4D light field foveated rendering method that compromises perceptual visual quality with fewer rays. Meng et al. [28] introduced a 3D-kernel foveated rendering method on the light field to improve rendering performance. Chakravarthula et al. [4] reduced perceived speckle noise by integrating the foveal and peripheral vision of HVS and retinal point spread function into the phase hologram computation. Li et al. [25] proposed a log-linear transformation method to encode and compress the original HD 360° video frame and transmit them to HMDs without degrading the perceptual visual quality. Deep learning methods are also used for foveated rendering. Kaplanyan et al. [17] proposed a generative adversarial neural network to improve the quality of foveated images/videos in the peripheral region. Deng et al. [8] proposed a NeRF-based foveated rendering method, which generates an egocentric neural radiance field representation of the scene and uses two multi-layer perceptrons to generate the rendering results in real time.

Since LMFR can be integrated into the shading pipeline efficiently, and LMFR supports the linear decrease of the shading rate from the gaze position to the periphery, we use it as the framework of our method. We propose an eccentricity-based shading rate controller based on LMFR to provide the average shading rate control and shading rate decrease speed control in foveated rendering.

2.2 Motion based Rendering

Motion based rendering can be divided into two categories. The first one is to enhance the rendering results through the motion information of objects and camera to improve the depth perception and realism of users in VR. The second one is to take advantage of HVS's features in motion perception to accelerate rendering.

For the first category, Konrad et al. [20] rendered gaze-contingent parallax based on the quantified thresholds for VR display systems. Xiao et al. [44] proposed a temporal neural network to reconstruct the low-resolution results in real time, which can reduce aliasing and restore scene details. Denes et al. [6] dynamically adjusted the resolution according to the motion quality of objects to achieve the balance between rendering performance and resolution. Briedis et al. [3] reconstructed the controllable free viewpoint video based on

the local neural radiance fields and the motion graph.

For the second category, Yee et al. [46] constructed a spatiotemporal error tolerance map that allows low-quality rendering without reducing the perceptual visual quality by leveraging that HVS's sensitivity will decrease when perceiving moving objects. Denes et al. [7] proposed a temporal resolution multiplexing technique that renders every even-numbered frame at low resolution and every odd-numbered frame at high resolution, and samples the high-resolution frames for compensating the high-sensitive regions in low-resolution frames. Mueller et al. [31] proposed a shading reuse method that chooses shading frequencies for each pixel based on the temporal motion information, which achieves 5× speedup without noticeable visual quality loss. Jindal et al. [16] proposed a variable rate shading method that adaptively adjusts the shading accuracy and refresh rate of different pixel blocks to improve rendering performance without sacrificing noticeable rendering quality.

The sensitivity of HVS varies with the viewing motion, and the user's viewing motion differs when using different locomotion methods for scene exploration. Thus, the rendering quality requirements of different locomotion methods are different. We propose a locomotion-aware log-polar mapping method that introduces the locomotion-aware coefficients to control the eccentricity-based shading rate, and integrates locomotion-aware coefficients into the eccentricity-based shading rate controller to improve the rendering performance without sacrificing the perceptual visual quality.

3 THEORETICAL BASIS

In this section, we describe the effects of viewing motion on HVS's sensitivity under different locomotion methods, and the advantages that these effects may bring to foveated rendering, so as to provide the theoretical basis of LaFR.

3.1 Gaze Motion

The sensitivity of HVS varies not only with the spatial frequency, but also with the gaze motion. Kelly [18] studied this effect by having users observe a varying sine wave and measuring their contrast sensitivity thresholds. In the experiment, Kelly applied a special technique to stabilize the imaging results of the retina, thus comparing the peak difference in contrast sensitivity of HVS at different gaze motion velocities. Kelly's experimental results showed that the sensitivity of HVS begins to decrease at gaze velocity above 0.15 °/sec. That is, in LaFR, if the user's gaze motion is not staying in static state ($\geq 0.15°/sec$) frequently when exploring the scene by a certain locomotion method, the rendering quality of the foveal region can be reduced to a certain extent without sacrificing the perceptual visual quality. In addition, Stengel et al. [36] have proved through user studies that when unconscious gaze movements are triggered in VR exploration, the foveal region needs to be expanded linearly to achieve better perceptual visual quality. Therefore, in LaFR, if the unconscious gaze movement occurs frequently during scene exploration by a certain locomotion method, the expansion of the foveal region can improve the perceptual visual quality.

3.2 Head Motion

In VR, head orientation is used as the interaction for perspective transformation and, in some cases, to assist in the locomotion for roaming VR scenes. The interactions of head and eye movements are complex and neurologically coupled [35]. When the head is free to move, changes in the direction and the position of the gaze point may involve both eye and head motion [10]. The acuity of HVS changes when the head moves with the eyes. Firstly, when the head rotation is large (i.e., greater than $40°$) during saccades, the vestibulo-ocular reflex fails [24]. Thus, in LaFR, if the user's head rotates frequently to obviously different directions when exploring the scene using a certain locomotion method, HVS cannot clearly perceive the foveal region. The rendering quality in the foveal region can be reduced while maintaining the perceptual visual quality.

Secondly, the velocity of the gaze motion increases linearly with the head rotation velocity [24], thus the unconscious gaze movement occurs more frequently. Therefore, in LaFR, if the user's head rotates frequently during scene exploration by a certain locomotion method, expanding the foveal region can improve the perceptual visual quality in foveated rendering.

In conclusion, for those locomotion methods in which the user's gaze is not in static state frequently, such as redirected walking, or head motion is not in static state frequently, such as teleportation and redirected walking, the rendering quality of the foveal region can be reduced to a certain extent without sacrificing the perceptual visual quality, and the expansion of the foveal region can improve the perceptual visual quality. Although expanding the foveal region directly can improve the perceptual visual quality, it will also reduce the rendering performance. Therefore, in pilot user study 2, we discuss the impact of expanding the foveal region on the perceptual visual quality in both keeping overall shading quantity and decreasing overall shading quantity cases. The experimental results show that under the condition of keeping overall shading quantity in the conventional LMFR, expanding the foveal region does not have significant impact on the perceptual visual quality. But after further reducing the overall shading quantity, expanding the foveal region can significantly improve the perceptual visual quality.

4 VIEWING MOTION ANALYSIS FOR DIFFERENT LOCOMOTION

In this section, we capture the user's viewing motion data, including gaze motion data and head motion data, when exploring VR scenes using different locomotion methods. Then we analyze the data to verify the difference of the user's viewing motion among different locomotion methods, so as to provide the data support for LaFR.

4.1 Viewing Motion Data Capture

Condition We choose the three representative locomotion methods [23]: joystick moving, teleportation, and redirected walking. Joystick moving is a locomotion mode that makes users move freely in the virtual scene through the controller's joystick. We use the joystick to control the user to move around in VR scenes. Teleportation is a locomotion mode that allows the user to specify the visible position in the field of view in VR and teleport the user to this position [2]. We use the parabola teleportation method provided by SteamVR [1] for scene exploration. Redirected walking is a locomotion mode that maps the path of the user walking in the real world to the virtual scene [34]. We implement a simple redirected walking with SteamVR [32] and use reset action to reverse the perspective through 'A' button of the controller.

Participants 10 participants (7 males and 3 females, aged between 22-30) from our university are recruited in this study, and 7 of them have had previous experience using VR HMDs.

Setup We use an HTC Cosmos HMD with a Droolon F1 gaze tracker to track the gaze motion and head motion of the user. The resolution of the HMD is 1440×1700 pixels for each eye, and the field-of-view is $97°$. The HMD is connected to a PC workstation with a 3.8 GHz Intel(R) Core(TM) i7-10700KF CPU, 64 GB of memory, and an NVIDIA GeForce GTX 2080 Ti graphics card. The program is developed with C# and is run in Unity 2021.3.6f1.

Procedure Participants are instructed to explore the scene freely using the above three locomotion methods, and explore all areas of the scene as much as possible. We record the participants' gaze positions and the rotation of their heads during their exploration. Before the experiment, we introduce how to explore a virtual scene using three different locomotion methods. All participants will perform gaze calibration before exploring scenes. Each participant explores 4 scenes that are randomly selected from the scenes visualized in Figure 5, and the starting positions are fixed in all scenes. The exploration time of each scene by using any locomotion method

is 30 seconds. The total experiment time of each user, including calibration and explanation time, is about 8 minutes.

4.2 Data Analysis

We generate the gaze and head motion data based on the collected gaze position and head rotation data. For generating the gaze motion data, we first get the current gaze position of the current frame in screen space, and the previous gaze position of the last frame in screen space. Then we subtract the previous gaze position from the current gaze position to obtain the direction vector of the current gaze movement, and use this direction vector to represent the gaze motion of the current frame. For generating the head motion data, we subtract the rotation vector of the current frame and the previous frame to obtain the direction vector of the current head rotation, and project the head rotation direction to the screen space to represent the head motion of the current frame.

We divide the gaze and head motion into five states: static ($\circ$), vertical motion ($\updownarrow$), horizontal motion ($\leftrightarrow$), positive oblique motion ($\nearrow$) and negative oblique motion ($\searrow$). Kelly [18] proved through experiments that the sensitivity of HVS begins to decrease at gaze velocity above 0.15 °/sec. Thus, we regard the gaze or head as static when its motion eccentric angle is within 0.15°/sec. Otherwise, in the screen space with x as the abscissa and y as the ordinate, for the motion vector $\vec{v}$ in the gaze motion data or the head motion data, if the included angle between $\vec{v}$ and the unit direction vector $\vec{u}$ $(0,1)$ is in $[-30°, 30°)$ or $[150°, 210°)$, $\vec{v}$ is regarded as vertical motion; if the included angle between $\vec{v}$ and $\vec{u}$ is in $[60°, 120°)$ or $[240°, 300°)$, $\vec{v}$ is regarded as horizontal motion; if the included angle between $\vec{v}$ and $\vec{u}$ is in $[30°, 60°)$ or $[210°, 240°)$, $\vec{v}$ is regarded as positive oblique motion; if the included angle between $\vec{v}$ and $\vec{u}$ is in $[120°, 150°)$ or $[300°, 330°)$, $\vec{v}$ is regarded as negative oblique motion.
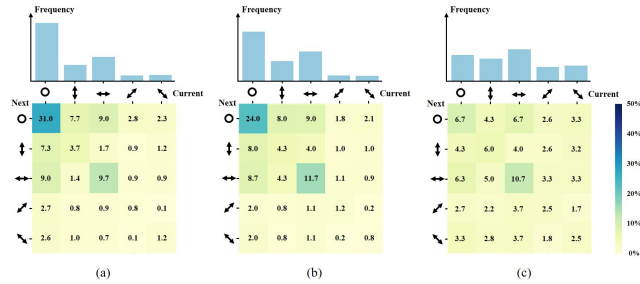

Figure 2: Frequency histograms and confusion matrices of the gaze motion under (a) joystick moving, (b) teleportation, and (c) redirected walking.

Figure 2 plots the frequency histogram of 5 gaze motion states under three different locomotion methods. The frequencies of static state account for 52.7% and 44.8% of all states for gaze motion under joystick moving and teleportation. But the frequency of static state accounts for 23.6% of all states under redirected walking, which is significantly lower than those of joystick moving and teleportation. It can be regarded that the gaze motion is not in static state frequently under redirected walking. Figure 2 further visualizes confusion matrices of the gaze motion directions between two adjacent frames for different locomotion methods. It shows that gaze tends to stay still, move horizontally, or switch between the two states.

Figure 3 plots the frequency histogram of 5 head motion states under three different locomotion methods. The frequencies of static state account for 37.3%, 19.9%, and 9.1% of all states for head motion under joystick moving, teleportation, and redirected walking. The frequencies of head staying still are much lower under redirected walking and teleportation than joystick moving. The head motion is not in static state frequently under teleportation and redirected walking. Figure 3 further visualizes confusion matrices of the head

motion directions between two adjacent frames. It can be seen from the confusion matrices that head tends to move in other states from horizontal motion, or back to horizontal motion from other states.
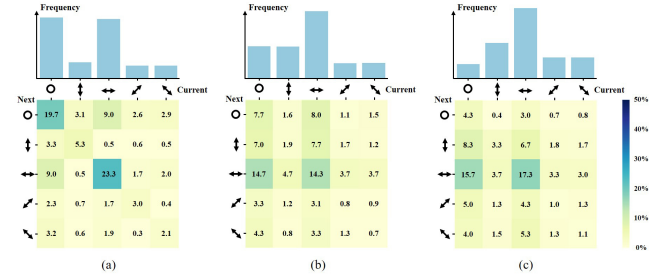

Figure 3: Frequency histograms and confusion matrices of the head motion using (a) joystick moving, (b) teleportation, and (c) redirected walking.

5 LOCOMOTION-AWARE FOVEATED RENDERING

In this section, we first introduce the framework of LaFR in Section 5.1, which is based on the log-polar mapping foveated rendering framework (LMFR) [30]. Then we propose the eccentricity-based shading rate controller in Section 5.2 for providing the shading rate control for the given region of the foveated rendering. Based on the eccentricity-based shading rate controller, we propose the locomotion-aware log-polar mapping method in Section 5.3 to parametrically control the foveal average shading rate, the peripheral shading rate decrease speed, and the overall shading quantity with locomotion-aware coefficients.

5.1 Framework of the Locomotion-aware Foveated Rendering

We first give a brief introduction of LMFR, and then introduce the framework of LaFR based on LMFR.

There are two steps in LMFR. The first step is to transform the input screen-space framebuffer pixel coordinate (x, y) in Cartesian coordinates to the reduced-resolution log-polar framebuffer pixel coordinate (r, θ) in the log-polar coordinates by Equation 1, and then render the reduced-resolution log-polar framebuffer.

$$r = K \left(\frac{\log \left(\sqrt{(x - x_g)^2 + (y - y_g)^2} \right)}{L} \right) \cdot w$$

$$\theta = \left(\arctan \left(\frac{y - y_g}{x - x_g} \right) \frac{1}{2\pi} + 1[(y - y_g) < 0] \right) \cdot h \tag{1}$$

where L is the log distance from the center to the corner of the screen, (x_g, y_g) is the position of the gaze point in the screen-space framebuffer, (w, h) is the width and height of the reduced-resolution log-polar framebuffer, $K(\cdot)$ is the kernel function, and $1[\cdot]$ is the indicator function.

The second step is to use the inverse log-polar transformation to transform the pixel coordinate (r, θ) in the reduced-resolution log-polar framebuffer back to the output screen-space framebuffer pixel coordinate (x', y') in Cartesian coordinates by Equation 2. Due to the non-uniform scaling property of the log-polar transformation, the output screen-space framebuffer preserves the details in the foveal region and reduces the details in the peripheral region.

$$x' = \exp \left(\frac{L}{w} \cdot K^{-1}(r) \right) \cos \left(\frac{2\pi}{h} \cdot \theta \right) + x_g$$

$$y' = \exp \left(\frac{L}{w} \cdot K^{-1}(r) \right) \sin \left(\frac{2\pi}{h} \cdot \theta \right) + y_g \tag{2}$$

where $K^{-1}(\cdot)$ is the inverse of the kernel function $K(\cdot)$.

Based on LMFR, we propose the framework of LaFR, as shown in Figure 4. Given the 3D Scene S, current viewpoint V, current gaze position (x_g, y_g), and the locomotion-aware coefficients, it outputs the rendering result. The locomotion-aware coefficients are

introduced to parametrically control the eccentricity-based shading rate in LaFR, including the foveal average shading rate control coefficient, the peripheral shading rate decrease coefficient, and the overall shading quantity control coefficient, which are detailed in Section 5.3.1. LaFR optimizes the locomotion-aware coefficients for different locomotion methods to further improve the performance of foveated rendering without reducing the perceptual visual quality.
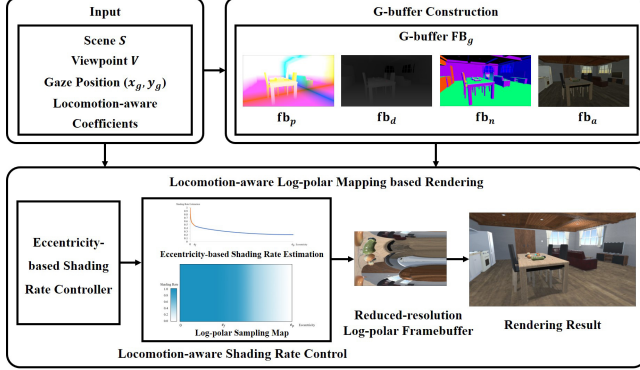

Figure 4: Framework of Locomotion-aware Foveated Rendering

In the first step, it constructs G-buffer FB_g, which includes the position map fb_p, the depth map fb_d, the normal map fb_n, and the albedo map fb_a. In the second step, it performs the locomotion-aware log-polar mapping based rendering. In the locomotion-aware log-polar mapping based rendering, it first uses the eccentricity-based shading rate controller to generate the eccentricity-based shading rate distribution, and converts the eccentricity-based shading rate distribution to the log-polar sampling map according to the guidance of the locomotion-aware coefficients, which determines the shading rate of pixels at different eccentric angles. Secondly, it performs illumination calculation to render the reduced-resolution log-polar framebuffer. Finally, it uses the inverse log-polar mapping to construct the output rendering result based on the reduced-resolution log-polar framebuffer.

5.2 Eccentricity-based Shading Rate Controller

The theoretical basis in Section 3 and the viewing motion analysis in Section 4 demonstrate that LaFR has the requirements of average shading rate control and shading rate decrease speed control for different regions. A direct way is to calculate the shading rate at each eccentric angle through $\frac{dr}{de}$ to adjust the kernel function $K(\cdot)$, so as to control the shading rate variation with the eccentric angle. But the direct calculation of the shading rate $\frac{dr}{de}$ requires taking the derivative of $K(\cdot)$ into account, which makes it very difficult to control the shading rate by adjusting the kernel function $K(\cdot)$ for LaFR. Thus, we first estimate the eccentricity-based shading rate without calculating the derivative of the kernel function. Then we parametrically control the average shading rate and the shading rate decrease speed for the given eccentricity range based on the eccentricity-based shading rate estimation result.

Given the eccentric angle e, the eccentricity-based shading rate estimation method outputs the average shading rate Ψ_e of all pixels in the output screen-space framebuffer whose eccentric angle is e. Firstly, we randomly select a pixel (x_e, y_e) among those pixels whose eccentric angle is e and calculate its radius r_e in the log-polar coordinates by Equation 1. We define a unit eccentric angle $|e|$ to represent the minimum angle of eccentricity for a single pixel in the log-polar coordinates. Since the shading rate of all pixels contained in the specific eccentricity is uniform, and the shading quantity of all pixels with eccentric angle e can be represented by $\pi r_e{}^2 - \pi(r_e - r_{|e|})^2$. Thus, Ψ_e can be represented by Equation 3.

$$\Psi_e = \frac{1}{\pi r_e{}^2 - \pi(r_e - r_{|e|})^2} \tag{3}$$

Since $r_{|e|}$ tends to 0, and the value of $r_{|e|}{}^2$ is negligible, Ψ_e can be estimated by Equation 4.

$$\Psi_e \approx \frac{1}{2\pi r_{|e|} r_e} = \frac{1}{2\pi r_{|e|} w} \cdot \frac{1}{K(z)} \tag{4}$$

where $z = \frac{log(\sqrt{(x_e - x_g)^2 + (y_e - y_g)^2})}{L}$.

According to [30], the kernel function in the conventional LMFR is set as $K(z) = z^a$. The eccentricity-based shading rate controller adjusts the average shading rate and the decrease speed of the shading rate for the specific region in the conventional LMFR flexibly based on the eccentricity-based shading rate estimation.

To adjust the average shading rate, we use

$$K_f(z) = \frac{z^{a'}}{-a' + 1} \tag{5}$$

as the kernel function, where $a' = f'(a, \sigma_f)$. The controller adjusts the average shading rate by tuning the average shading rate reduce parameter σ_f. There are two reasons for using the function K_f as a kernel function. The first one is that the shading rate estimation integration of this kernel function is easy to calculate. The second one is that the decrease of shading rate in the output screen-space framebuffer is consistent with the conventional LMFR by applying this kernel function. Given the eccentric angle range $[e_b, e_t]$ of the region, σ_f, Equation 6 makes the average shading rate in the region with eccentricity range. Eccentric angle range $[e_b, e_t]$ is σ_f times that of the conventional LMFR.

$$\int_{Z(e_b)}^{Z(e_t)} \frac{(-a' + 1)z^{-a'}}{2\pi r_{|e|} w} dz = \sigma_f \cdot \int_{Z(e_b)}^{Z(e_t)} \frac{z^{-a}}{2\pi r_{|e|} w} dz \tag{6}$$

Thus, a' can be calculated by Equation 7.

$$a' = 1 - \log_{Z(e_t)} \frac{\sigma_f \cdot Z(e_t)^{1-a}}{1 - a} \tag{7}$$

where $Z(e)$ is the function that transforms the eccentric angle e to its corresponding value of z in the kernel function. When σ_f is 1, the controller keeps the average shading rate consistent with the conventional LMFR in the region with the eccentric angle range $[e_b, e_t]$. The smaller σ_f is, the lower the average shading rate in the region with the eccentric angle range $[e_b, e_t]$.

We use the kernel function as shown in Equation 8 to adjust the decrease speed of the shading rate for the region with the eccentric angle range $[e_b, e_t]$. This kernel function maps the shading rate of the eccentricity range $[e_b, e_t]$ to the cosine function $cos(x^{f''(a, \sigma_p)})$ with the independent variable range $x \in [\frac{\pi}{2}, \pi]$, and the shading rate as the dependent variable range $\Psi \in [\Psi_{e_b}, \Psi_{e_t}]$. The controller adjusts the decrease speed of the shading rate by tuning the shading rate decrease parameter σ_p.

$$K_p(z) = \frac{1}{\cos\left(\frac{\pi}{2}\left(1 + \frac{z - Z(e_b)}{Z(e_t) - Z(e_b)}^{f''(a'', \sigma_p)}\right)\right) \cdot (\Psi_{e_b} - \Psi_{e_t}) + \Psi_{e_b}} \tag{8}$$

where Ψ_e is the shading rate estimation with the eccentric angle e, $f''(a'', \sigma_p) = \frac{a''}{\sigma_p}$ is the function which controls the shading decrease speed, and a'' is the empirical parameter to keep the shading decrease speed consistent with the conventional LMFR in the region with the eccentric angle range $[e_b, e_t]$ when σ_p is 1. The smaller σ_f is, the slower the shading rate will decrease in the region with the eccentric angle range $[e_b, e_t]$.

5.3 Locomotion-aware Log-polar Mapping

In this section, we first extract locomotion-aware coefficients based on the theoretical basis of Section 3. Then we propose the segmented nonlinear log-polar mapping method to integrate locomotion-aware coefficients into the eccentricity-based shading rate controller, thus achieving the locomotion-aware log-polar mapping.

5.3.1 Locomotion-aware Coefficients

According to the viewing motion analysis for different locomotion methods in Section 4, the gaze motion and head motion are different when exploring scenes in VR using different locomotion methods. According to the theoretical basis in Section 3, the rendering quality of the foveal region can be reduced for the locomotion methods in which the user's gaze is not staying in static state frequently or head rotates frequently to obviously different directions. Thus, we introduce the foveal average shading rate control coefficient α to control the average shading rate in the foveal region.

In addition, if the unconscious gaze movement occurs or the user's head rotates more frequently, the expansion of the foveal region can improve the perceptual visual quality. In this paper, the eccentric angle range $[0, e_f]$ of the foveal region is fixed according to [14]. We achieve the goal of expanding the foveal region by slowing down the decrease speed of the shading rate from the boundary of the foveal region e_f to the boundary of the peripheral region e_p. Therefore, we introduce the peripheral shading rate decrease coefficient β to control the decrease speed of the shading rate from e_f to e_p, so as to control the foveal region expansion.

The above two coefficients make the average shading rate in the foveal and peripheral regions vary among different locomotion methods for achieving the same perceptual visual quality. Compared with other locomotion methods, the overall shading quantity may also be less for those locomotion methods with lower average shading rates in the foveal or peripheral region. In LMFR, the shading rate of each pixel in the reduced-resolution log-polar framebuffer is uniform. We can control the overall shading quantity by controlling the total number of pixels in the reduced-resolution log-polar framebuffer. Therefore, we introduce the overall shading quantity control coefficient δ to control the total number of pixels in the reduced-resolution log-polar framebuffer.

5.3.2 Segmented Nonlinear Log-polar Mapping

We propose the segmented nonlinear log-polar mapping to integrate locomotion-aware coefficients into LaFR, control the foveal average shading rate through α, the peripheral shading decrease speed through β, and the overall shading quantity through δ.

For controlling the average shading rate in the foveal region, we use the kernel function $K_f(\alpha,z) = \frac{z^{f'(a,\alpha)}}{-f'(a,\alpha)+1}$ proposed in the eccentricity-based shading rate controller to transform the pixels in the foveal region of the screen-space coordinates to the reduced-resolution log-polar coordinates, and tune α to adjust the average shading rate in the foveal region. In order to be consistent with the average shading rate of the conventional LMFR in the foveal region, α is initialized to 1.0, and $f'(a,\alpha)$ is shown in Equation 9.

$$f'(a,\alpha) = 1 - \log_{Z(e_f)} \frac{\alpha \cdot Z(e_f)^{1-a}}{1-a} \tag{9}$$

where e_f is the boundary of the foveal region which is set as $9.78°$ [14], a is the parameter in the conventional LMFR which is set to $1/4$.

For controlling the decrease speed of the shading rate in the peripheral region, we use the kernel function proposed in the eccentricity-based shading rate controller to transform the pixels in the peripheral region of the screen-space coordinates to the reduced-resolution log-polar coordinates, which is shown in Equation 10.

$$K_p(\beta,z) = \frac{1}{\cos\left(\frac{\pi}{2}(1 + \frac{z-Z(e_f)}{Z(e_p)-Z(e_f)} \frac{f''(a'',\beta)}{})\right) \cdot (\Psi_{e_f} - \Psi_{e_p}) + \Psi_{e_f}} \tag{10}$$

where e_p is the boundary of the peripheral region which is set as $110°$, Ψ_{e_f} and Ψ_{e_p} are the shading rate estimation result of the eccentric angles e_f and e_p, and $f''(a,\beta) = \frac{a''}{\beta}$ is the function that controls the shading rate decrease speed. Since a is set to $1/4$ in the conventional LMFR, we set $a'' = 0.7$ empirically to make the shading rate decrease speed of the eccentricity range $[e_f, e_p]$

consistent with the conventional LMFR when β is 1. We tune β to adjust the decrease speed of the shading rate in the peripheral region.

For controlling the overall shading quantity of the output screen-space framebuffer, we set the width and height (w,h) of the reduced-resolution log-polar framebuffer as $(\frac{W}{\delta}, \frac{H}{\delta})$, where (W,H) are the width and height of the screen-space framebuffer. We tune δ to adjust the total number of pixels in the reduced-resolution log-polar framebuffer.

Thus, the kernel function $K_l(z,\alpha,\beta)$ in LaFR is shown as Equation 11.

$$K_l(z,\alpha,\beta) = \begin{cases} K_f(\alpha,z) & z \in [Z(0), Z(e_f)] \\ K_p(\beta,z) & z \in [Z(e_f), Z(e_p)] \end{cases} \tag{11}$$

The equation of log-polar transformation in our approach is shown as Equation 12.

$$r = K_l(z,\alpha,\beta) \cdot \frac{W}{\delta}$$
$$\theta = \left(\arctan\left(\frac{y-y_g}{x-x_g}\right)\frac{1}{2\pi} + 1[(y-y_g) < 0]\right) \cdot \frac{H}{\delta} \tag{12}$$

The equation of inverse log-polar transformation in our approach is shown as Equation 13.

$$x' = \exp\left(\delta \cdot K_l^{-1}(r)\right) \cos\left(\frac{2\pi}{h} \cdot \theta\right) + x_g$$
$$y' = \exp\left(\delta \cdot K_l^{-1}(r)\right) \sin\left(\frac{2\pi}{h} \cdot \theta\right) + y_g \tag{13}$$

6 PILOT USER STUDIES

We conduct two pilot user studies to optimize the foveal average shading rate control coefficient α, the peripheral shading rate decrease coefficient β, and the overall shading quantity control coefficient δ of LaFR, which can produce results similar to LMFR.

6.1 Pilot User Study 1: Evaluating δ

Pilot user study 1 divides LaFR into several foveation levels according to the overall shading quantity control coefficient δ. The higher foveation level indicates the bigger δ, which means more aggressive foveated rendering, i.e., better rendering performance and lower overall rendering quality. Pilot user study 1 aims to find the acceptable foveation levels for different locomotion methods. Users can't find significant differences between the rendering results of LaFR and LMFR.

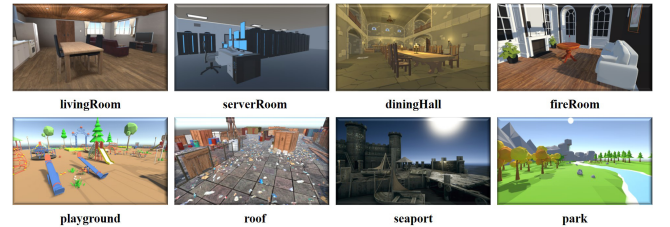| livingRoom | serverRoom | diningHall | fireRoom |
| playground | roof | seaport | park |

Figure 5: Scenes used for all user studies. *fireRoom* is from [27], and the other scenes are collected from [40]. All scenes are rendered with the Unity game engine.

6.1.1 Pilot User Study Design

Conditions Locomotion methods used for scene exploration in pilot user study 1 is the same as those in the viewing motion analysis in Section 4: joystick moving, teleportation and redirected walking. The experimental results of the conventional LMFR [30] show that when the overall shading quantity control coefficient δ is 1.8 and the kernel function is $K(z) = z^{1/4}$, LMFR is visually indistinguishable from the full resolution rendering. Thus, we fix the foveal average shading rate control coefficient α and the peripheral shading rate decrease coefficient β to 1.0 in pilot user study 1, which makes the foveal average shading rate and peripheral shading rate decrease speed consistent with the conventional LMFR. Then we divide LaFR into 5 foveation levels. From foveation level $L1$ to $L5$, The overall

shading quantity control coefficient δ is set to 1.8, 2.0, 2.2, 2.4, and 2.6. Higher foveation level means more aggressive foveated rendering, i.e., better rendering performance and lower overall rendering quality.

Participants and Setup 15 participants (10 males and 5 females, aged between 21-30) are recruited in this study, and 12 of them have had experiences in VR HMDs. The setup of this study is also the same as that of the viewing motion analysis.

Task and Procedure The study uses a 2×3×5 within-subjects design. We randomly select 1 indoor scene and 1 outdoor scene from 8 scenes that are visualized in Figure 5. Each scene uses 3 different locomotion methods to explore the scene, and 5 foveation levels are used for generating rendering results for each locomotion method, i.e., all participants are required to complete 30 trials. Before the experiment, we introduce how to explore scenes using three different locomotion methods. For each participant, the order of trials is random. The task is to explore the scene freely for 30 seconds in each trial, and participants are required to explore all areas of the scene as much as possible. After each trial, participants need to score the perceptual visual quality of the trial and have a rest of 10 seconds. The visual quality score η contains 5 confidence levels: 5 represents that they cannot perceive artifacts at all, 4 represents that they can perceive acceptable artifacts at a few very short moments, 3 represents that they can perceive acceptable artifacts, 2 represents that they can perceive noticeable artifacts, and 1 represent that they can perceive obvious artifacts. Each participant spends an average of 25 minutes. A total of 2 (scenes)×3 (locomotion methods)×5 (foveation levels)×15 (participants) = 450 trials are collected.

6.1.2 Results and Discussion

We calculate the average values and standard deviations of η of all conditions, and use the p-value [42] and Cohen's d [5] to estimate the difference between two conditions. Figure 6 gives the average values and standard deviations of η of all conditions under different locomotion methods at different foveation levels. When using any locomotion method for scene exploration at all foveation levels, the average η of indoor scenes is close to those of outdoor scenes and the decrease trend is consistent. It can be regarded that the selection of the acceptable foveation level is irrelevant to the scene for all tested locomotion methods. When the foveation level is $L1$, the average values and standard deviations of η under all three locomotion methods are close, but with the increase of the foveation level, the average values of η under joystick moving and teleportation decrease faster than that of redirected walking.
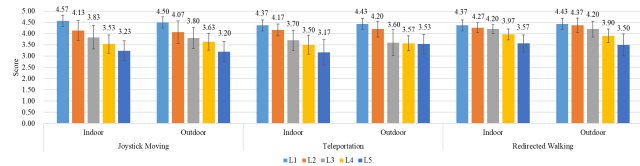

Figure 6: Average values and standard deviations of the visual quality score η in pilot user study 1 using different locomotion methods to explore different scenes over different foveation levels.

For joystick moving, when foveation level is $L2$, compared with $L1$, p-value = 0.03, Cohen's d = 0.58 and effect size is *medium*. Using joystick moving to explore the scene that is rendered at foveation level $L2$ with the fixed $\alpha = 1.0$ and $\beta = 1.0$ will not meet the requirements for exploring scenes with high perceptual visual quality. Thus, the acceptable foveation level $\hat{L}$ of joystick moving is $L1$ in pilot user study 1, in which the overall shading quantity control coefficient $\hat{\delta}$ is set to 1.8.

For teleportation, when the foveation level is $L2$, compared with $L1$, p-value = 0.14, Cohen's d = 0.39 and effect size is *small*. But when foveation level is $L3$, compared with $L1$, p-value = 0.00, Cohen's d = 0.93 and effect size is *large*. When using teleportation to

explore the scene, the experimental results show that significant difference is failed to be found between rendering scenes at foveation level $L2$ with the fixed $\alpha = 1.0$, $\beta = 1.0$ and foveation level $L1$. Thus, the acceptable foveation level $\hat{L}$ under teleportation is $L2$ in pilot user study 1, in which the overall shading quantity control coefficient $\hat{\delta}$ is set to 2.0.

For redirected walking, no significant difference of the perceptual visual quality is found when the foveation level rises from $L1$ to $L3$. p-value of $L3$ is 0.15, Cohen's d of $L3$ is 0.38 and effect size is *small*. Thus, the acceptable foveation level $\hat{L}$ we choose for redirected walking is $L3$ in pilot user study 1, in which the overall shading quantity control coefficient $\hat{\delta}$ is set to 2.2.

6.2 Pilot User Study 2: Optimizing α and β, Defining δ

Results of pilot user study 1 show that the acceptable foveation levels vary among different locomotion methods, i.e., the evaluated δ is different among different locomotion methods. Pilot user study 2 further divides LaFR into several acceptable foveation levels based on the evaluated acceptable foveation levels in pilot user study 1. The higher acceptable foveation level, the bigger δ, the more aggressive foveated rendering. Pilot user study 2 aims to find the maximum acceptable foveation level with the biggest δ by optimizing α and β. Users can't find significant differences between the rendering results of the maximum acceptable foveation level and the acceptable foveation level chosen for different locomotion methods in pilot user study 1.

6.2.1 Pilot User Study Design

Conditions Locomotion methods used for exploration in pilot user study 2 are the same as those in pilot user study 1. To achieve the maximum acceptable foveation level for different locomotion methods, we extend the acceptable foveation level $\hat{L}$, and tune the foveal average shading rate control coefficient α and the peripheral shading rate coefficient β in LaFR based on each acceptable foveation level. Firstly, we extend the acceptable foveation level $\hat{L}$ into 3 levels: $\hat{L}1$, $\hat{L}2$ and $\hat{L}3$. For each locomotion method, the overall shading quantity control coefficient $\hat{\delta}$ in $\hat{L}1$ is set to the value of δ obtained in pilot user study 1, $\hat{L}2$'s δ is set to $\hat{\delta}+0.2$, and $\hat{L}3$'s δ is set to $\hat{\delta}+0.4$. Under each acceptable foveation level, α is set to 1.0, 0.85 and 0.70, β is set to 1.0, 0.7 and 0.4. Figure 7 visualizes the eccentricity-based shading rate variation curves in the conventional LMFR and in LaFR with different values of locomotion-aware coefficients. Figure 7 also plots the shading rate estimation as a function of eccentricity in LMFR and in LaFR with different values of locomotion-aware coefficients. The figure shows that the smaller α, the lower average shading rate in the foveal region; the smaller β, the slower peripheral shading rate decrease speed.
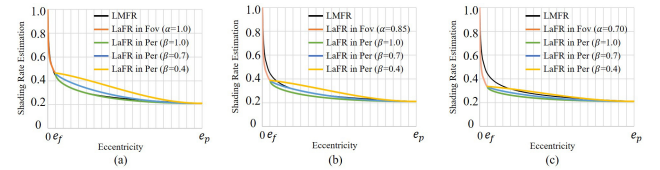

Figure 7: Shading rate falloff with eccentricity in LMFR, and (a) LaFR with $\alpha = 1.0$ and $\beta = 1.0$, 0.7, 0.4, (b) LaFR with $\alpha = 0.85$ and $\beta = 1.0$, 0.7, 0.4, (c) LaFR with $\alpha = 0.70$ and $\beta = 1.0$, 0.7, 0.4.

Participants and Setup The same 15 participants from pilot user study 1 participate in this study. The setup of this study is also the same as that of pilot user study 1.

Task and Procedure This study uses a 2×3×3×9 within-subjects design. As with pilot user study 1, we randomly select 1 indoor scene and 1 outdoor scene from 8 scenes. Each scene uses three different locomotion methods to explore the scene, and 3 acceptable foveation levels are used for generating rendering results for each locomotion

method. For each acceptable foveation level, we use 3 different values of α and 3 different values of β to adjust the eccentricity-based shading rate. All participants are required to complete 162 trials. Before the experiment, we let participants use three different locomotion methods to explore a random scene that is rendered by the full resolution rendering for 30 seconds. For each participant, the order of trials is random. The task is the same as pilot user study 1. After each trial, participants give the visual quality score η of the trial and have a rest of 10 seconds. Each participant spends an average of 85 minutes. A total of 2 (scenes)×3 (locomotion methods)×3 (acceptable foveation levels)×9 (locomotion-aware coefficients)×15 (participants) = 2430 trials are collected.

6.2.2 Results and Discussion

Figure 8 gives the average values and standard deviations of η of all conditions when exploring scenes under three different locomotion methods with different locomotion-aware coefficients. It shows that the perceptual visual quality can be affected by tuning the locomotion-aware coefficients. For all locomotion methods at any acceptable foveation level, the perceptual visual quality can be improved when α is set to 0.85 and β is set to 0.7.

For joystick moving, the group $\kappa_{\hat{L}2}$ has the highest average $\eta =$ 4.20 in acceptable foveation level $\hat{L}2$ when α and β are 0.85 and 0.7. Compared with foveation level $L1$ selected by pilot user study 1, i.e., the group κ_{L1} whose acceptable foveation level is $\hat{L}1$, $\alpha = 1.0$ and $\beta = 1.0$, p-value of $\kappa_{\hat{L}2}$ is 0.06, Cohen's d is 0.38 and effect size is *small*. That is, compared with κ_{L1}, no significant perceptual visual quality difference is found when LaFR with the coefficients of $\kappa_{\hat{L}2}$ is used to render scenes. However, the highest average η is 3.97 in $\hat{L}3$. p-value and Cohen's d both show significant difference. Thus, the maximum acceptable level chosen for joystick moving is $\hat{L}2$ in pilot user study 2, where $\alpha = 0.85$, $\beta = 0.7$, and $\delta = 2.0$.
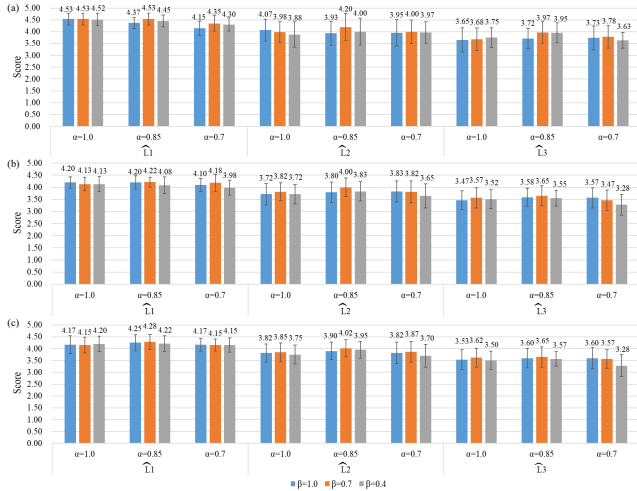

Figure 8: Average values and standard deviations of η in pilot user study 2 over different acceptable foveation levels with different locomotion-aware coefficients using (a) joystick moving, (b) teleportation, and (c) redirected walking to explore scenes.

For teleportation and redirected walking, the perceptual visual quality falloff with acceptable foveation level is consistent with that of joystick moving. The average values of η in group $\kappa_{\hat{L}2}$ under teleportation and redirected walking are 4.00 and 4.02. Compared with the group κ_{L1} under teleportation and redirected walking, p-values of $\kappa_{\hat{L}2}$ are 0.09 and 0.27, values of Cohen's d of $\kappa_{\hat{L}2}$ are 0.31 and 0.20, and effect sizes are both *small*. For all groups in $\hat{L}3$ under teleportation and redirected walking, the differences between these groups and κ_{L1} are significant. Thus, compared with κ_{L1}, significant perceptual visual quality difference is failed to be found when rendering scenes using LaFR with the coefficients of $\kappa_{\hat{L}2}$.

For teleportation, the optimized locomotion-aware coefficients α is 0.85, β is 0.7, and δ is 2.2. For redirected walking, the optimized locomotion-aware coefficients α is 0.85, β is 0.7, and δ is 2.4. Besides, the values of visual quality score η of pilot user studies 1 and 2 with the same locomotion-aware coefficients are slightly different because scenes are randomly assigned to 15 participants in these two pilot user studies.

In conclusion, LaFR can perform more aggressive foveated rendering gradually with the optimized locomotion-aware coefficients for joystick moving, teleportation, and redirected walking while no significant perceptual visual quality decrease is found.

7 USER STUDY

We conduct a user study to evaluate the perceptual visual quality of LaFR.

7.1 User Study Design

We formulate the hypotheses for the user study.

Hypotheses We formulate two hypotheses for the user study:

H1 For all locomotion methods, LaFR with optimized locomotion-aware coefficients achieves similar perceptual visual quality compared with the conventional LMFR.

H2 For all locomotion methods, LaFR with optimized locomotion-aware coefficients achieves acceleration compared with the full resolution rendering and the conventional LMFR.

Conditions Locomotion methods used to explore virtual scenes in the user study are the same as those in pilot user studies. The rendering methods used to generate rendering results include the full resolution rendering method (RR), the conventional log-polar foveated rendering method (LMFR) and LaFR with optimized locomotion-aware coefficients.

Participants 20 participants (15 males and 5 females, aged between 21-30) are recruited for the user study. No one has participated in pilot user studies, and 13 of them have had experiences in VR HMDs. The setup of this study is the same as those of pilot user studies.

Task and Procedure The whole experiment consists of 72 trials. All participants need to use 3 different locomotion methods to explore all 8 virtual scenes that are rendered by 3 different rendering methods. Before the experiment, we introduce how to explore the virtual scene using three different locomotion methods. The task is the same as those of pilot user studies. The order of trial is random. After each trial, users give the trial's visual quality score η to evaluate the perceptual visual quality and rest for 10 seconds. It takes an average of 48 minutes to complete all trials. The data of 20 (participants)×8 (scenes)×3 (locomotion methods)×3 (rendering methods) = 1440 trials are collected.

7.2 Results and Discussion

p-values between LaFR and LMFR under joystick moving, teleportation, redirected walking are 0.53, 0.34, and 0.56 respectively. Values of Cohen's d between LaFR and LMFR under joystick moving, teleportation, redirected walking are 0.10, 0.15, 0.09, and effect sizes are all *very small* under all locomotion methods. p-value and Cohen's d both show that the perceptual visual quality between LaFR and LMFR is similar under all locomotion methods. Table 1 compares the frequencies of η among RR, LaFR with optimized locomotion-aware coefficients and LMFR under different locomotion methods. We note that the probability of η greater than or equal to 4 for RR, LaFR and LMFR exceeds 85% under all locomotion methods, and the results indicate the comparable perceptual visual quality among RR, LaFR, and LMFR, and show the generalizability of LaFR. Therefore, the results support **H1**.

We calculate MSE of LaFR, the conventional LMFR, and LMFR with the same performance as LaFR (LMFR$_s$) compared with RR under three different locomotion methods in all eight virtual scenes, and compare p-value and Cohen's d between MSE of LaFR and LMFR, LMFR$_s$ and LMFR in Table 2. p-value and Cohen's d indicate that MSE of LaFR has no significant difference compared with

Table 1: Visual quality score η frequency of RR, LaFR, and LMFR under joystick moving (JM), teleportation (T), and redirected walking (RW).

	Methods	$\eta = 5$	$\eta = 4$	$\eta = 3$	$\eta = 2$	$\eta = 1$
	RR	82.5%	17.5%	0.0%	0.0%	0.0%
JM	LaFR	47.5%	43.8%	5.0%	3.7%	0.0%
	LMFR	30.0%	58.8%	6.2%	5.0%	0.0%
	RR	90.0%	8.8%	1.2%	0.0%	0.0%
T	LaFR	36.3%	51.2%	7.5%	5.0%	0.0%
	LMFR	23.8%	65.0%	6.2%	5.0%	0.0%
	RR	85.0%	12.5%	2.5%	0.0%	0.0%
RW	LaFR	33.8%	52.5%	8.7%	5.0%	0.0%
	LMFR	26.3%	63.8%	6.3%	3.8%	0.0%

that of LMFR. However, p-values between LMFR$_s$ and LMFR are above 0.05 and effect sizes of Cohen's d are *large* and *very large* under teleportation and redirected walking, which both show that MSE of LMFR$_s$ decreases significantly compared with that of LMFR.

Figure 9 shows the average FPS of RR, LaFR and LMFR under different locomotion methods in all test scenes. When using redirected walking for scene exploration, the average FPS of LaFR for all scenes is 135. Under redirected walking, LaFR achieves 3.2-3.8× speedup compared with RR, and achieves 1.6× speedup compared with LMFR. LaFR also achieves the average FPS of 122 under teleportation, and brings 2.9-3.4× and 1.4-1.5× speedup compared with RR and LMFR. Since higher rendering quality is required when using joystick moving for exploring scenes, the FPS of LaFR decreases. The average FPS of LaFR is 101 under joystick walking. Compared with RR and LMFR, the acceleration ratios of LaFR are 2.4-2.7 and 1.2 under joystick moving.

Table 2: Statistical MSE ($\times 10^{-2}$) comparisons among LaFR, LMFR and LMFR$_s$ under three locomotion methods.

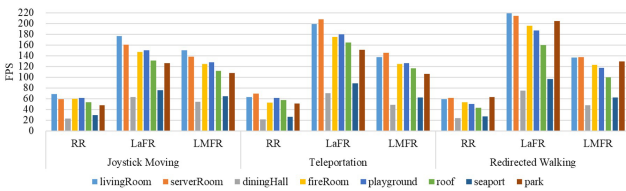|	Methods	avg.±std.	p-value	Cohen's d	effect size
	LMFR	3.44±0.79	/	/	/
JM	LaFR	3.61±0.86	0.71	0.20	*small*
	LMFR$_s$	3.78±0.89	0.45	0.41	*small*
T	LaFR	3.73±0.87	0.53	0.34	*small*
	LMFR$_s$	4.57±1.12	0.04	1.17	*large*
RW	LaFR	3.85±0.91	0.39	0.47	*small*
	LMFR$_s$	4.99±1.25	0.01	1.48	*very large*

Figure 9: Performance of RR, LaFR, and LMFR under joystick moving, teleportation, and redirected walking for all test scenes.

Table 3 shows the statistical performance difference of LaFR compared with RR and LMFR under all locomotion methods. Compared with RR, LaFR achieves 2.4-3.8× speedup under all locomotion methods. p-values between LaFR and RR are less than 0.05 and effect sizes of Cohen's d of LaFR are *huge* under all locomotion methods. Compared with LMFR under teleportation and redirected walking, LaFR achieves 1.4-1.6× speedup, p-values between LaFR and LMFR are less than 0.05 and effect sizes of Cohen's d of LaFR are *large* and *very large*. It indicates that LaFR achieves significant

Table 3: Statistical performance comparisons between LaFR and RR, LMFR under three locomotion methods.

	Comparisons	p-value	Cohen's d	effect size
JM	vs. RR	0.00	2.74	*huge*
	vs. LMFR	0.33	0.54	*medium*
T	vs. RR	0.00	2.97	*huge*
	vs. LMFR	0.04	1.15	*large*
RW	vs. RR	0.00	3.21	*huge*
	vs. LMFR	0.02	1.45	*very large*

performance acceleration compared with RR under all locomotion methods and LMFR under teleportation and redirected walking. When exploring scenes using joystick moving, the perceptual visual sensitivity HVS is higher. On the premise of maintaining similar perceptual visual quality as the conventional LMFR, LaFR also achieves 1.2× speedup. Therefore, the results support **H2**. But p-value between LaFR and LMFR is 0.33, and effect size of Cohen's d is *medium* under joystick moving, which indicates the performance acceleration is not significant.

8 CONCLUSION, LIMITATION, AND FUTURE WORK

We have proposed a locomotion-aware foveated rendering method (LaFR) to accelerate the foveated rendering performance when exploring VR scenes. By optimizing the locomotion-aware coefficients to reduce the foveal average shading rate and slow down the peripheral shading rate decrease speed, LaFR can perform more aggressive foveated rendering than the conventional foveated rendering LMFR with similar perceptual visual quality, and the degree of aggressiveness increases gradually with joystick moving, teleportation, and redirected walking. LaFR achieves 1.2-1.6× speedup while maintaining similar perceptual visual quality compared with LMFR. Compared with the full resolution rendering, LaFR achieves 2.4-3.8× speedup with comparable perceptual visual quality.

Since the rendering performance of LMFR is determined by the size of the reduced-resolution log-polar framebuffer, and the width and height of the reduced-resolution log-polar framebuffer are pre-defined in LMFR and can not be changed in rendering. Thus, the rendering performance of LMFR cannot be dynamically controlled in real time. LaFR is based on LMFR framework, so one limitation of LaFR is that it cannot dynamically control the rendering performance in real time according to the user's viewing motion. Thus, the first possible future work is to combine LaFR with the shading reuse technology to dynamically control the shading rate and refresh rate of each pixel on the reduced-resolution log-polar framebuffer, so as to further adaptively improve the rendering performance of the foveated rendering. The other limitation is that the object salient features of the VR scene is not considered in LaFR, so users may perceive the salient objects in the periphery at some moments, thus reducing the perceptual visual quality. In LMFR, the shading rate of a circle pixels corresponding to each eccentric angle is the same. Therefore, LaFR cannot be combined with the saliency map directly. So the other possible future work is to introduce a novel mapping method to transform the screen space coordinates into a new space, which can dynamically control the shading rate of the circle pixels corresponding to the specific eccentric angle. Then, it integrates the saliency map into the new space to increase the shading rate of pixels in the periphery with salient objects, thus improving the perceptual visual quality.

REFERENCES

[1] Htc vive teleportation system with arc pointer. `https://github.com/Innoactive/IA-unity-VR-toolkit-vive-teleporter`, 2017.

[2] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey. Point & teleport locomotion technique for virtual reality. CHI PLAY '16, p. 205–216. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2967934.2968105

[3] K. M. Briedis, A. Djelouah, M. Meyer, I. McGonigal, M. Gross, and C. Schroers. Neural frame interpolation for rendered content. ACM Transactions on Graphics (TOG), 40(6):1–13, 2021.

[4] P. Chakravarthula, Z. Zhang, O. Tursun, P. Didyk, Q. Sun, and H. Fuchs. Gaze-contingent retinal speckle suppression for perceptually-matched foveated holographic displays. IEEE Transactions on Visualization and Computer Graphics, 27(11):4194–4203, 2021.

[5] J. Cohen. Statistical power analysis for the behavioral sciences. Routledge, 2013.

[6] G. Denes, A. Jindal, A. Mikhailiuk, and R. K. Mantiuk. A perceptual model of motion quality for rendering with adaptive refresh-rate and resolution. ACM Transactions on Graphics (TOG), 39(4):133–1, 2020.

[7] G. Denes, K. Maruszczyk, G. Ash, and R. K. Mantiuk. Temporal resolution multiplexing: Exploiting the limitations of spatio-temporal vision for more efficient vr rendering. IEEE transactions on visualization and computer graphics, 25(5):2072–2082, 2019.

[8] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. IEEE Transactions on Visualization and Computer Graphics, 28(11):3854–3864, 2022.

[9] L. Franke, L. Fink, J. Martschinke, K. Selgrad, and M. Stamminger. Time-warped foveated rendering for virtual reality headsets. In Computer Graphics Forum, vol. 40, pp. 110–123. Wiley Online Library, 2021.

[10] E. G. Freedman. Coordination of the eyes and head during visual orienting. Experimental Brain Research, 2008.

[11] S. Friston, T. Ritschel, and A. Steed. Perceptual rasterization for head-mounted display image synthesis. ACM Trans. Graph., 38(4):97–1, 2019.

[12] M. Fujita and T. Harada. Foveated real-time ray tracing for virtual reality headset. Light Transport Entertainment Research, 2014.

[13] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. Foveated 3d graphics. ACM Transactions on Graphics (TOG), 31(6):1–10, 2012.

[14] B. Guenter, M. Finch, S. M. Drucker, D. S. Tan, and J. Snyder. Foveated 3d graphics. international conference on computer graphics and interactive techniques, 2012.

[15] B. Jin, I. Ihm, B. Chang, C. Park, W. Lee, and S. Jung. Selective and adaptive supersampling for real-time ray tracing. In Proceedings of the Conference on High Performance Graphics 2009, pp. 117–125, 2009.

[16] A. Jindal, K. Wolski, K. Myszkowski, and R. K. Mantiuk. Perceptual model for adaptive local shading and refresh rate. 40(6), dec 2021. doi: 10.1145/3478513.3480514

[17] 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 Transactions on Graphics (TOG), 38(6):1–13, 2019.

[18] D. H. Kelly. Motion and vision. ii. stabilized spatio-temporal threshold surface. Journal of the Optical Society of America, 1979.

[19] Y. Kim, Y. Ko, and I. Ihm. Selective foveated ray tracing for head-mounted displays. In 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 413–421. IEEE, 2021.

[20] R. Konrad, A. Angelopoulos, and G. Wetzstein. Gaze-contingent ocular parallax rendering for virtual reality. ACM Transactions on Graphics (TOG), 39(2):1–12, 2020.

[21] M. Koskela, K. Immonen, T. Viitanen, P. Jääskeläinen, J. Multanen, and J. Takala. Foveated instant preview for progressive rendering. In SIGGRAPH Asia 2017 Technical Briefs, pp. 1–4. 2017.

[22] M. K. Koskela, K. V. Immonen, T. T. Viitanen, P. O. Jääskeläinen, J. I. Multanen, and J. H. Takala. Instantaneous foveated preview for progressive monte carlo rendering. Computational Visual Media, 4(3):267–276, 2018.

[23] E. Langbehn, P. Lubos, and F. Steinicke. Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking. In Proceedings of the Virtual Reality International Conference-Laval Virtual, pp. 1–9, 2018.

[24] V. P. Laurutis and D. A. Robinson. The vestibulo-ocular reflex during human saccadic eye movements. The Journal of Physiology, 1986.

[25] D. Li, R. Du, A. Babu, C. D. Brumar, and A. Varshney. A log-rectilinear transformation for foveated 360-degree video streaming. IEEE Transactions on Visualization and Computer Graphics, 27(5):2638–2647, 2021.

[26] A. Maimone, A. Georgiou, and J. S. Kollin. Holographic near-eye displays for virtual and augmented reality. ACM Transactions on Graphics (Tog), 36(4):1–16, 2017.

[27] M. McGuire. Mcguire computer graphics archive. `https://casual-effects.com/data`, 2017.

[28] X. Meng, R. Du, J. F. JaJa, and A. Varshney. 3d-kernel foveated rendering for light fields. IEEE Transactions on Visualization and Computer Graphics, 27(8):3350–3360, 2020.

[29] X. Meng, R. Du, and A. Varshney. Eye-dominance-guided foveated rendering. IEEE transactions on visualization and computer graphics, 26(5):1972–1980, 2020.

[30] X. Meng, R. Du, M. Zwicker, and A. Varshney. Kernel foveated rendering. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 1(1):1–20, 2018.

[31] J. H. Mueller, T. Neff, P. Voglreiter, M. Steinberger, and D. Schmalstieg. Temporally adaptive shading reuse for real-time rendering and virtual reality. ACM Trans. Graph., 40(2), apr 2021. doi: 10.1145/3446790

[32] J. W. Murray. Building Virtual Reality with Unity and SteamVR. CRC Press, 2020.

[33] 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 Transactions on Graphics (TOG), 35(6):1–12, 2016.

[34] S. Razzaque, D. Swapp, M. Slater, M. C. Whitton, and A. Steed. Redirected walking in place. In Proceedings of the Workshop on Virtual Environments 2002, EGVE '02, p. 123–130. Eurographics Association, Goslar, DEU, 2002.

[35] 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, 2018.

[36] M. Stengel, S. Grogorick, M. Eisemann, and M. Magnor. Adaptive image-space sampling for gaze-contingent real-time rendering. eurographics, 2016.

[37] Q. Sun, F.-C. Huang, J. Kim, L.-Y. Wei, D. Luebke, and A. Kaufman. Perceptually-guided foveation for light field displays. ACM Transactions on Graphics (TOG), 36(6):1–13, 2017.

[38] E. Turner, H. Jiang, D. Saint-Macary, and B. Bastani. Phase-aligned foveated rendering for virtual reality headsets. In 2018 IEEE conference on virtual reality and 3D user interfaces (VR), pp. 1–2. IEEE, 2018.

[39] O. T. Tursun, E. Arabadzhiyska-Koleva, M. Wernikowski, R. Mantiuk, H.-P. Seidel, K. Myszkowski, and P. Didyk. Luminance-contrast-aware foveated rendering. ACM Transactions on Graphics (TOG), 38(4):1–14, 2019.

[40] Unity. Unity assets store. `https://assetstore.unity.com/`, 2022.

[41] D. R. Walton, R. K. Dos Anjos, S. Friston, D. Swapp, K. Akşit, A. Steed, and T. Ritschel. Beyond blur: Real-time ventral metamers for foveated rendering. ACM Transactions on Graphics, 40(4):1–14, 2021.

[42] R. L. Wasserstein and N. A. Lazar. The asa statement on p-values: Context, process, and purpose. The American Statistician, 70(2):129–133, 2016. doi: 10.1080/00031305.2016.1154108

[43] M. Weier, T. Roth, E. Kruijff, A. Hinkenjann, A. Pérard-Gayot, P. Slusallek, and Y. Li. Foveated real-time ray tracing for head-mounted displays. In Computer Graphics Forum, vol. 35, pp. 289–298. Wiley Online Library, 2016.

[44] L. Xiao, S. Nouri, M. Chapman, A. Fix, D. Lanman, and A. Kaplanyan. Neural supersampling for real-time rendering. ACM Transactions on Graphics (TOG), 39(4):142–1, 2020.

[45] J. Ye, A. Xie, S. Jabbireddy, Y. Li, X. Yang, and X. Meng. Rectangular mapping-based foveated rendering. In 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 756–764. IEEE, 2022.

[46] H. Yee, S. Pattanaik, and D. P. Greenberg. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. ACM Transactions on Graphics (TOG), 20(1):39–65, 2001.