



## Perception-Driven Hybrid Foveated Depth of Field Rendering for Head-Mounted Displays

Liu, Jingyu; Mantel, Claire; Forchhammer, Søren

*Published in:*  
Proceedings of 2021 IEEE International Symposium on Mixed and Augmented Reality

*Publication date:*  
2021

*Document Version*  
Peer reviewed version

[Link back to DTU Orbit](#)

*Citation (APA):*  
Liu, J., Mantel, C., & Forchhammer, S. (Accepted/In press). Perception-Driven Hybrid Foveated Depth of Field Rendering for Head-Mounted Displays. In *Proceedings of 2021 IEEE International Symposium on Mixed and Augmented Reality* IEEE.

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Perception-Driven Hybrid Foveated Depth of Field Rendering for Head-Mounted Displays

Jingyu Liu\*

Claire Mantel†

Søren Forchhammer‡

Department of Photonics Engineering  
Technical University of Denmark

## ABSTRACT

In this paper, we present a novel perception-driven hybrid rendering method leveraging the limitation of the human visual system (HVS). Features accounted in our model include: foveation from the visual acuity eccentricity (VAE), depth of field (DOF) from vergence & accommodation, and longitudinal chromatic aberration (LCA) from color vision. To allocate computational workload efficiently, first we apply a gaze-contingent geometry simplification. Then we convert the coordinates from screen space to polar space with a scaling strategy coherent with VAE. Upon that, we apply a stochastic sampling based on DOF. Finally, we post-process the Bokeh for DOF, which can at the same time achieve LCA and anti-aliasing. A virtual reality (VR) experiment on 6 Unity scenes with a head-mounted display (HMD) HTC VIVE Pro Eye yields frame rates range from 25.2 to 48.7 fps. Objective evaluation with FovVideoVDP - a perceptual based visible difference metric - suggests that the proposed method gives satisfactory just-objectionable-difference (JOD) scores across 6 scenes from 7.61 to 8.69 (in a 10 unit scheme). Our method achieves better performance compared with the existing methods while having the same or better level of quality scores.

**Index Terms:** Computing methodologies—Rendering; Computing methodologies—Virtual reality

## 1 INTRODUCTION

A smooth experience in virtual reality (VR) requires an efficient rendering pipeline to achieve low latency. The requirement conflicts with the current scheme that is commonly used in rendering systems which allocates computing resources evenly across vertices and pixels. An ongoing research field is leveraging the limitations of the human visual system (HVS) to alleviate the computational bottleneck in rendering. To that end, the following characteristics of HVS can be used: 1. The visual acuity eccentricity (VAE) in both spatial resolution and color sensitivity, and 2. The vergence and accommodation mechanism drives the eyes for focusing, which causes an effect of depth of field (DOF) that only the objects in focus appear sharp on retina [70]. On the one hand, according to these HVS characteristics, eyes do not perceive pixels in different parts of the screen in the same way, especially for head-mounted displays (HMD) with a wide field of view (FOV) where most of the pixels fall in the periphery. Indeed, Patney et al. [47] estimated that only 4% of the pixels map to the central high visual acuity area on retina (i.e. *fovea*). On the other hand, the magnifying effect with lens to cover wide FOV makes HMD demanding in resolution. Thus conventional rendering methods that uniformly sample the pixels are not efficient for near-eye optic systems due to a high computational overheads. On the contrary, a scaled rendering method that is coherent with the characteristics of the HVS will improve the performance while maintaining the perceptual quality.

This paper proposes such a perception-driven hybrid rendering method that enables high-fidelity visual experience in virtual reality (VR). The core idea is to emulate some features of the natural visual process based on a human visual model (HVM). Our HVM first includes a transformation from screen to polar space, to simulate the angular spatial VAE as implemented in [29]. On that basis, we extend the VAE with a peripheral geometry simplification [53]. The second focus of our HVM is to build a mathematical model to simulate a physically accurate DOF using human eyes' parameters. Specifically, we account for the pupil diameter change [36] under HMD ambient lighting [40] and its influence on the DOF. We also account for longitudinal chromatic aberration (LCA) along with DOF [12] in our HVM.

The main contribution of this work includes:

- We build a novel mathematical model that emulates human eyes' DOF in VR.
- We propose a hybrid rendering method that simulates retinal blur by applying a DOF-based stochastic sampling on top of a foveated scaling.
- We extend the scope of HVM by accounting for LCA inline with Bokeh for DOF and anti-aliasing.

This article is structured in the following way: Section 2 introduces some features from the HVS that relate to visual perception in HMD, and reviews the literature that applies those features in rendering. Section 3 details the design of our hybrid perception-driven rendering method that features foveation, DOF, and LCA. Section 4 presents the implementation of our method in experiments with HTC VIVE Pro Eye, including an objective evaluation of our method along with representative existing methods. The paper ends with a discussion of the limitation and future work in Section 5.

## 2 RELATED WORK

This section reviews literature of features in HVS and their applications in rendering, with a focus on 2 aspects: VAE and DOF.

### 2.1 Features of the Human Visual System

Retinal ganglion cells (RGC) distribution is non-homogeneous across the retina. Curcio et al. [14] provides anatomical measurements of ganglion cell densities as a function of VAE. The major VAE effect appears as a retinal blur. In a total FOV of 160° only the central 2° has the highest-resolution called *fovea* [18]. In addition, similar disparity is found for the cone photoreceptors distribution, which determines the color sensitivity. Cones are the only photoreceptors in the fovea and with eccentricity their density decreases and their diameter increases, i.e. their resulting acuity decreases [70].

When human eyes look at a far away object, the binocular vergence behavior drives the gaze rays to diverge until they are almost parallel. In contrast, when focusing on a near object, the gaze rays converge to an intersection point. Simultaneously, the accommodation in eyes also focuses on that intersection point, which brings in the DOF effect. Stretching or relaxation of our ciliary muscle will change the shape and focal depth of our lens when focusing on different objects. DOF is the main effect for evaluating the depth and occlusion relationships among objects in the

\*e-mail: jing@fotonik.dtu.dk

†e-mail: clma@fotonik.dtu.dk

‡e-mail: sofo@fotonik.dtu.dk

real world [12] [37]. With the help of DOF, eyes can adapt to the depth [20] [21]. Furthermore, an accurately simulated DOF can help alleviate the vergence-accommodation conflict (VAC) [22] [51] in conventional stereoscopic displays [26].

## 2.2 Foveated Rendering

Foveated rendering is a rendering technique that leverages VAE of HVS, generally achieved by differentiating spatial resolution across the display. In this way, the perceived quality can be preserved while the computational workload decreased. Foveated rendering applies most resources within the fovea, while lowering the information density in the periphery. Weier et al. [70] did a thorough review on the gaze-contingent foveated rendering methods. Spjut et al. [54] introduced a taxonomy for HMD by a classification method designed towards foveation.

Guenter et al. [18] introduced a 3-layer foveated rendering with progressively lower resolution towards periphery. Vaidyanathan et al. [64] introduced a coarse pixel shading to decrease the shading rate in the periphery. A similar approach was applied in [19]. NVIDIA also published its variable rate shading (VRS) [46] that varies shading rate by blocks sizing from  $1 \times 1$  to  $4 \times 4$ . Stengel et al. [55] not only accounts VAE for the spatial resolution, but also for the contrast. Tursun et al. [62] proposed a content-dependent foveated rendering that is among a few foveated rendering methods that validate the result by objective metrics. Walton et al. [67] introduced a foveated rendering method that instead of simplifying VAE effect as a blur, it accounts for the pooling characteristics of HVS that render the periphery as metamers. Meng et al. [42] transformed the representation from the screen space to the polar space, as also done in [29], by mapping the Cartesian coordinates to the log-polar or visual-polar coordinates. The sampling distribution is more coherent with VAE model, and the mapping can be efficiently adapted to GPU's architecture. Later Meng et al. [41] improved the technique by adapting different foveation levels for different eyes. We adapt a similar approach as [29] did in our model. Besides the methods that focus on spatial foveated scaling, [16] introduced a foveated reprojection method that recycles pixels in the periphery with an adaption of the confidence map. Mueller et al. [44] proposed a similar approach, while instead of VAE model, it relies more on the characteristics of the scene itself and using the shading gradients to decide the pixels' re-usage. Ray tracing is a good fit to foveated rendering due to its flexibility with the sampling strategy. Hunt et al. [23] introduced a ray casting method that supports lens distortion, foveation and DOF at the same time. Koskela et al. [29] used path tracing with only 0.4 samples per pixel with a denoise module to achieve real-time foveated rendering. DeepFovea [24] is an AI-powered system that provides good perceptual quality for image and video rendering. An extension of DeepFovea to computer graphics can be a disruptor to the field. A hardware solution [56] did ray casting with importance sampling which only needs 16%-30% rays without compromising perceptual quality. Besides VR, a wide FOV augmented reality (AR) hardware solution [27] has been proposed, which uses a set of optical components that can travel at run time to achieve fovea tracking and focal depth adjustment.

**Aliasing in foveated rendering** Temporal aliasing is a common problem of the down-sampling methods applied in foveated rendering. While the peripheral vision has a low acuity in spatial resolution, its temporal sensitivity is high [50]. A rapid gaze movement from one location to another, called *saccade* [8], will also affect the perceptual quality. Patney et al. [47] alleviated temporal artifacts in periphery by using pre-filters and temporal anti-aliasing (TAA). Overall, pre-filters, TAA or deferred shading from [25] are usually applied to cover the temporal artifacts. Albert et al. [7] suggested a total system latency of 50 ~ 70 ms could be tolerated.

### 2.2.1 Peripheral Scene Simplification

Gaze-contingent scene simplification can be combined with foveated spatial resolution manipulation to further improve the performance. A general method for geometry simplification can be done through level of detail (LOD). Luebke et al. [33] introduced a perceptual based mesh simplification which decreases the complexity of meshes with a visibility check. When combined with VAE, an application can be seen in [15]. Textures can also be optimized from a mipmap structure [32] [9]. A typical usage of mipmaps is to apply a higher mip level when the object is far away. A potential adaption for VAE is to increase the mip level of the objects in the periphery.

## 2.3 Depth of Field Rendering

In computer graphics, DOF is achieved by differentiating the sharp pixels and Bokeh - or the out-of-focus blur - in the rendering. Bokeh is realized by calculating circle of confusion (COC) - an out-of-focus effect that projects points as discs on the retina. Since calculation of COC is pixel-wise in screen space, usually the design choice of DOF is implemented as a post-processing effect, which is an additional step after completing the lighting computation. Thus, realizing DOF has a negative effect on performance conventionally. A review by [39] introduced different ways to simulate DOF in the post-processing. Nowadays, in a standard rendering pipeline in engines like Unity, a built-in DOF simulation is optional in the post-processing stack [59]. While realizing DOF through post-processing will more or less add the shading overheads, the post-processed blur of DOF can conceal the temporal aliasing from the foveated rendering. An approach is introduced in [68]. Physically accurate DOF can be simulated by ray-tracing through a lens rather than a single point [13]. However, too many samples need to be convoluted for this method. It is therefore challenging to achieve ray-traced DOF in real-time. Xiao et al. [72] presented an application which applies a deep learning network to synthesize out-of-focus blur. An DOF simulation can be completed within 1-2 ms.

Previous works suggest that applying DOF as a synthetic blur in VR rendering will increase the realism and alleviate VAC [26]. Mantiuk et al. [34] investigated DOF with a user study, indicating a more natural feeling in rendering with a gaze-contingent synthetic blur. However, there is also literature [31] doubting the effectiveness of computational DOF on driving the accommodation of eyes to relieve VAC, a hardware solution with a display system of aligned multi-focal planes can provide correct accommodation cues [43].

### 2.3.1 Longitudinal Chromatic Aberration

Natural visual features such as LCA [12] can be applied along with the DOF simulation, since the effect comes from accommodation as well. One way to realize LCA is to differentiate the further or nearer out-of-focus objects, and account chromatic effects dependently, as introduced in [12]. In our approach we simplify their model with an adaption for the DOF Bokeh.

## 2.4 Hybrid Retinal Blur

Psychophysical experiments by [57] investigated the relationship between foveation and focus cues beyond the low-level anatomical vision model that determines VAE on the distribution of the RGC. The result disclosed that the blur discrimination threshold is lower than the theory. However, the discrimination threshold of depth perception has a clear dependency on VAE. Although these conclusions need more experimental data to support, a rendering method that combines the 2 retinal blurs can better support high-level analysis of perception and is beneficial for future research.

## 3 DESIGN

As reviewed in the last section, leveraging the limitations of HVS can achieve VR-ready performance while maintaining good perceptual quality. This section details the motivation and design principles of

our method which combines foveation and DOF as a hybrid retinal blur.

### 3.1 Visual Acuity Eccentricity

In our design, we simplify the binocular vergence for human eyes into one vector shooting from the center of left and right eyes' origin in a synthetic viewport, as illustrated in the right part of Figure 3. The intersection of the vector - or the combined gaze ray (CGR) - and the screen yields a *gaze* point. The pixel of the gaze point corresponds to the center of retina, and from that point going outwards, VAE happens. In [14] and [52], the distribution of RGC on the retina was introduced. Carreras et al. [11] concluded the relationship between RGC and receptive field: When single eye FOV  $\alpha_M$  is normalized to  $90^\circ$ , a normalized fovea angle  $\alpha_F$  is around  $5^\circ$ . This normalized receptive field distribution illustrates in Figure 1a is the underlying physiological structure for VAE, and is the distribution that we simulate in the following design.

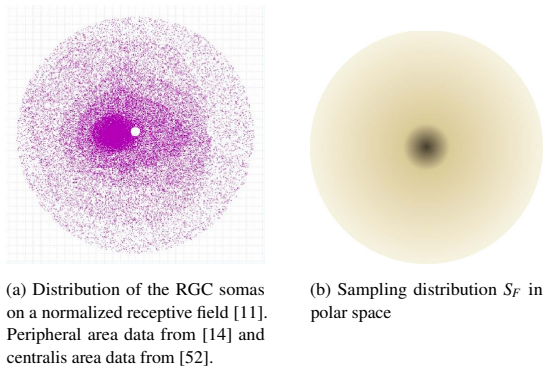


Figure 1: An illustration of the receptive field distribution and our polar space sampling distribution simulation.

Here we present a foveated scaling model which is coherent with a HVM in terms of VAE to save computational overhead. To simulate the distribution of receptive fields, we transform the sampling space from Cartesian coordinate to polar coordinate (with azimuthal angle axis  $\phi$  and polar angle axis  $\rho$ ). The origin of the coordinate adapts to the gaze point dynamically. This space conversion has been introduced in [42] and [29]. There are 3 main benefits of the coordinate transformation:

- Sampling distribution in polar space is more uniform as the gradient direction is now coherent with the receptive field.
- Foveated scaling realizes as the probability density function (PDF) differs across the polar space for sampling.
- It can outperform conventional tuning methods such as VRS since manipulation in polar space makes addressing more continuous which provides a better resource management.

We use an inversion method [48] to determine the sampling distribution in polar space. Equation 1 is the PDF that simulates VAE described in [49] with an adaption on the circumference of concentric circles [30]:

$$g(\alpha) = \begin{cases} 2\pi\alpha & \alpha \leq 5.79^\circ \\ \frac{14.98\pi\alpha}{(0.3\alpha+1)^2} & \alpha > 5.79^\circ \end{cases} \quad (1)$$

where  $\alpha$  is the eccentricity angle. For the peripheral area (i.e.  $\alpha > 5.79^\circ$ ), we have the cumulative distribution function (CDF) in Equation 2 as introduced in [29]. CDF integrates  $g(\alpha)$  over  $[0, \alpha_M]$ . Note if FOV is set to another angle, the integration interval adapts accordingly.

$$\left( \frac{1}{0.3\alpha(p)+1} + \ln(0.3\alpha(p)+1) \right) \times 166.4\pi - 612.3 \quad (2)$$

where  $\alpha(p)$  is the eccentricity angle of the sample point  $p$ . In practice we apply a least squares fitted polynomial as an approximation of the function. For the fovea (i.e.  $\alpha \leq 5.79^\circ$ ), we use linear coordinate mapping on the  $\rho$ -axis for a constant distribution of samples in the fovea.

When mapping back from the polar to Cartesian coordinate, the least squares fitted inverse of the CDF as in [29] is:

$$25.09\rho^4 + 0.1680\rho^3 + 27.61\rho^2 + 23.87\rho + 3.232 \quad (3)$$

While the backwards projection for  $\rho$ -axis is 1-to-1, for the  $\phi$ -axis many polar space pixels map to the same screen space pixel. Thus the backward projection for the  $\phi$ -axis is realized by applying a trilinear interpolation on the mipmaps.

The foveated sampling distribution  $S_F$  in polar space is illustrated in Figure 1b.

### 3.2 Depth of Field

In our approach, we propose a novel sampling strategy that achieves DOF by varying the shading rate. Compared with the traditional post-processing implementation, our approach helps improving the rendering performance. The idea is to form a sampling distribution  $S_D$  to simulate the DOF. Assuming that we know the focal depth  $F$  - which is the distance from eyes to the focused object - of the frame, we can bring it into the thin lens DOF model according to the following formula [3]:

$$D_F = \frac{f^2 F}{f^2 + \frac{cf^2}{\tau} - \frac{cfF}{\tau}} \quad (4)$$

$$D_N = \frac{f^2 F}{f^2 - \frac{cf^2}{\tau} + \frac{cfF}{\tau}} \quad (5)$$

where  $D_F$  is the far DOF limit and  $D_N$  is the near DOF limit,  $f$  is the focal length,  $\tau$  is the aperture size,  $c$  is the COC criterion - the largest blur spot that will still be perceived as a point.

We can simulate the DOF of human eyes using Equations 4 and 5 by inserting the parameters of human eye model. The characteristics of human eyes vary dynamically, depending on external factors such as ambient lighting and internal factors such as health and age of the individual. The image focal length of human eyes is about 22 to 24 mm, with  $f_e \approx 22.3$  mm for the standard European adults [17]. In addition, we apply an approximation that uses  $1^\circ$  as the angular resolution for human eyes, then multiplier  $\tan \frac{1}{60}$  by the focal length  $f_e$  to get the approximate  $c_e \approx 0.0065$  mm. Inserting the values into Equation 4 and 5, we get for the far and near DOF limits of human eyes ( $D_{Fe}$ ,  $D_{Ne}$ , unit in mm) as the following expressions:

$$D_{Fe} = \frac{497.29F}{497.29 + 3.23\tau_e - 0.14\frac{F}{\tau_e}} \quad (6)$$

$$D_{Ne} = \frac{497.29F}{497.29 - 3.23\tau_e + 0.14\frac{F}{\tau_e}} \quad (7)$$

For further approximation for VR, we simulate the pupil diameter  $\tau_e$ . In a VR headset, the ambient lighting is from almost 0 to the peak brightness of the VR display, with a maximum of around 190.5 nits for the HMD on the market [40]. We take the mean for approximation which is 85 nits (or  $\sim 300$  lux). Using the model of pupil diameter vs ambient lighting proposed in [36], we can do a table look-up and find the 2 endpoints: (5.22 mm, 150 lux), and (4.24 mm, 350 lux). With a linear fit the simulated pupil diameter

is 4.49 mm. With this approximation, Equations 6 and 7 can be rewritten as:

$$D_{Fv} = \frac{497.29F}{498.01 - 0.032F} \quad (8)$$

$$D_{Nv} = \frac{497.29F}{496.57 + 0.032F} \quad (9)$$

For VR, we use Equations 8 and 9 to propose a model to set the DOF scaling of the pixels  $p$  as a weight function  $w_d$ . The collection of  $w_d$  across the pixels forms the sampling distribution  $S_D$ :

$$w_d(p) = \begin{cases} 1 & \text{if } D_{Nv} \leq D_p \leq D_{Fv} \\ \frac{D_p}{D_{Nv}} & \text{if } D_p < D_{Nv} \\ \frac{D_M - D_p}{D_M - D_{Fv}} & \text{if } D_p > D_{Fv} \end{cases} \quad (10)$$

where  $D_p$  is the depth of the pixel from a look-up in the depth map,  $D_M$  is the maximum depth of the scene (if  $D_M$  of a certain scene is  $\infty$ , we set it to 1000 m). Since the DOF is calculated per-frame, the scaling is dynamic, with the rendering overheads inversely proportional to the range of DOF. A  $S_D$  example generated by Equation 10 is depicted in Figure 2. Note that if  $S_D$  is applied to a conventional display instead of HMD, Equations 6 and 7 should be used in Equation 10 instead.

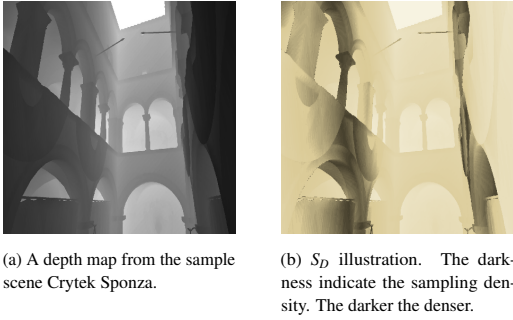


Figure 2: An illustration of DOF sampling distribution.

### 3.3 Hybrid Retinal Blur

In computer graphics, there are 2 mainstream rendering pipelines: forward rendering and deferred rendering. The latter provides a pipeline support for our hybrid scaling strategy. In deferred rendering, after completing the geometry pass, a geometry buffer (G-buffer) is allocated in the memory to store a collection of textures for the lighting pass. Once the G-buffer is generated, we can create an injection function for our scaling method: We grab the G-buffer from the memory, then apply the coordinate transformation on the textures in the G-buffer. After sampling in the polar space, the scaled G-buffer is then stored in another memory address which we call P-buffer. A similar flow has been applied in [42].

$S_F$  is in polar-coordinate, thus  $S_F$  can be applied on the P-buffer directly. As for a scaling on  $S_D$ , conventional scaling methods that adopt block-wise shading variation [64] [19] are no longer a satisfying option, considering that we have already used coordinate transformation to scale  $S_F$ , another pass of block-wise scaling for DOF would not be consistent from the perspective of the algorithm design. Thus we consider applying a DOF scaling that is different from the conventional options to realize a hybrid of the 2 sampling distributions  $S_D$  and  $S_F$  in a coherent manner: To filter the samples in  $S_F$  through a Russian Roulette on  $S_D$ . That is, after the coordinate transformation, instead of keeping all the samples on  $S_F$ , we apply a Russian Roulette to determine whether the sample is to undergo the

following lighting and shading passes through a filter  $\sigma(p)$  based on the  $w_d$  in the  $S_D$ :

$$\sigma(p) = \begin{cases} 1 & \text{if } \varepsilon \leq w_d(p) \\ 0 & \text{if } \varepsilon > w_d(p) \end{cases} \quad (11)$$

Where  $\varepsilon$  is a random float,  $\varepsilon \in [0, 1]$ . 0 means the sample will not be shaded in the following process.

#### 3.3.1 Bokeh: Blur and Anti-aliasing

While Gaussian filter is usually applied to simulate Bokeh [42] [29], we propose a physically correct approach with the same level of overheads. As we apply hybrid retinal blur in the same pass, no extra computation is needed to mitigate the aliasing introduced by invariant shading rate. Indeed, Bokeh for DOF can achieve anti-aliasing at the same time in the post-processing as a low-pass filter. The disc shape of the Bokeh kernel is pre-defined by a constant weight array from [63]. The COC size of the point in Bokeh for a thin lens model was derived in [45]. When parameterized with our HVM, we have:

$$C_e(p) = \left| \frac{f_e F D_p}{F D_p + f_e D_p - f_e F} - f_e \right| \cdot \left( \tau_e - \frac{f_e F D_p}{F D_p + f_e D_p - f_e F} \right) \quad (12)$$

where  $C_e$  is the COC of human eyes. We can bring in the parameters we derived in Section 3.2, where  $f_e \approx 22.3\text{mm}$ ,  $\tau_e \approx 4.49\text{ mm}$  in VR. A benefit of the physical simulation of DOF that emulates the natural visual process is that it can help relieve VAC through a computational approach without extra modification of the commercial HMD hardware [26] [34].

For scenes which are very sensitive to temporal artifacts, we can apply an approximation of COC as described in [1]:

$$C_{e'}(p) = \frac{D_p - F}{D_{Fv} - D_{Nv}} \quad (13)$$

Equation 13 applies an expanded Bokeh kernel size which strengthens the denoising effect. Note that if the display is not HMD, replace  $D_{Fv}$  and  $D_{Nv}$  by  $D_{Fe}$  and  $D_{Ne}$  accordingly.

At this point, the combination of the 2 scaling strategies and retinal blur presented above is achieved. Our method is a hybrid retinal blur, as illustrated in Figure 3.

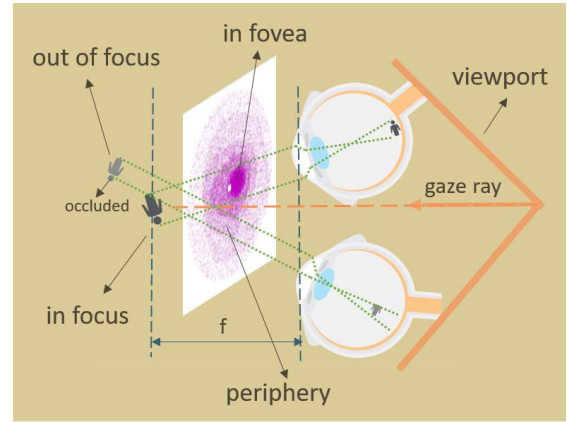


Figure 3: An illustration of our hybrid retinal blur, which is a combination of foveation and DOF.



### 3.3.2 Longitudinal Chromatic Aberration Realization

While prior art contains many discussions on the retinal blur, a perception-driven rendering focusing on color vision is a relatively new field. Our design adapt the concept of LCA [12] along with DOF: short wavelengths (i.e. blue) are more refracted through the cornea and lens of the eyes than long wavelengths (i.e. red), thus blue rays tend to focus in front of retina while red rays focus behind. Our approach is a concept adaption without the complex convolution in a physically correct simulation [12]. The realization of LCA is in the post-processing by varying the shading for different rows in the Bokeh kernel with a color filter: If  $D_p > D_{Fe}$ , the top 2 rows of kernel filter out blue, and the bottom 2 rows of kernel filter out red. The other way around if  $D_p < D_{Ne}$ . For  $D_p$  within DOF, no Bokeh kernel applies thus no LCA applies, which is also physically correct. In this way a simple chromatic shift is achieved to simulate LCA. LCA simulation is optional in our model.

### 3.3.3 Gaze-contingent Scene Simplification

In addition to the model introduced above, we also design a supplementary feature to fully utilize the gaze information, which is a gaze-contingent scene simplification on the level of object using LOD and texture mipmaps. The underlying geometry simplification method is from [53], which reduces the triangle number of an object by 50%-80% using quadrics. The method is applied offline to prepare the objects in the scene of 2 levels of LOD.

At run time we use the depth map and CGR to flag the object  $o$  as *in fovea* & *in focus* or otherwise. We conduct a weight function  $w_s$  across the objects  $o$  for the scene simplification as in Equation 14:

$$w_s(o) = \begin{cases} 1 & \text{if } \alpha_{(CEO-o, CGR)} \leq 5^\circ \text{ \& } D_{Nv} \leq D_o \leq D_{Fv} \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

The result of 1 or 0 means to apply object simplification or not. CEO stands for combined eye origin, which illustrated as the end point of the viewport in Figure 3.  $\alpha$  stands for the angle between the CGR and the ray from CEO to the object.  $D_o$  is the depth of the object that CGR intersect with (if there is any, otherwise the intersect function will return  $D_o$  as the clipping distance of the scene).

## 3.4 Pipeline Summary

Let us summarize the method for our hybrid retinal blur. The scaling strategies is described in Algorithm 1, where  $G_b$  refers to the G-buffer,  $M_d$  refers to the depth map,  $R$  refers to the screen resolution, and  $o$  refers to the gaze point coordinates on the display. The Bokeh post-processing is described in Algorithm 2.

**ALGORITHM 1:** Shader for foveation and DOF scaling

---

```

Inputs:  $G_b, M_d, R, o, D_{Ne}, D_{Fe}$ 
Output: color
while pixel do
     $(P_b, uv') = cartesian\_to\_polar(G_b, uv, R, o);$ 
     $\backslash\backslash$  Transformation G-buffer from Cartesian to polar coordinate by Eq 2
    and save to P-buffer
     $\sigma = dof\_russian\_roulette(uv, M_d, D_{Ne}, D_{Fe});$ 
     $\backslash\backslash$  Calculate whether to keep the sample according to  $S_D$  by Eq 11
     $color_p = lighting * sampler(P_b, uv', \sigma);$ 
     $\backslash\backslash$  Shading on P-Buffer and get the result in polar coordinate
     $color = polar\_to\_cartesian(color_p, uv', R, o);$ 
     $\backslash\backslash$  Transformation the result from polar to Cartesian coordinate by Eq 3
end
```

---

Overall, to shade a pixel and determine whether it is sharp or retinal blurred, there are 2 assessments to go through: *in fovea* and *in focus*. The result of the assessments will determine the shading scale of the pixel.

**ALGORITHM 2:** Shader for Bokeh post-processing

---

```

Inputs:  $M_d, F$ 
Output: color
while pixel do
     $coc = coc(uv, F, M_d);$ 
     $\backslash\backslash$  Calculate COC by Eq 12 or Eq 13
     $color = sampler(uv, coc, kernel);$ 
     $\backslash\backslash$  Calculate Bokeh and the blurred result
end
```

---

A pipeline of our design is presented in the Figure 4. Our custom rendering passes are applied in the deferred rendering pipeline. Gaze-contingent scene simplification is applied before shading. The scaling pass is described in Algorithm 1: Textures in G-buffer are transformed from Cartesian to polar coordinate, then a hybrid scaling is achieved by Russian Roulette  $S_D$  on top of  $S_F$ . After lighting, the result in polar space go through an inverse transformation to screen space. The post-processing pass includes realizing Bokeh with anti-aliasing (Algorithm 2), and an optional LCA simulation.

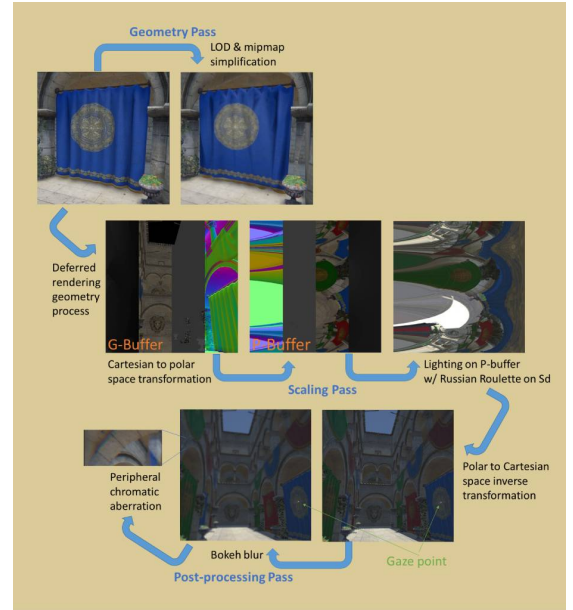


Figure 4: An overview of our perception-driven hybrid rendering method.

## 4 EXPERIMENT

We set up an experiment implementing the methods designed in Section 3. The aim is to verify that the method can work in VR with a satisfactory perceptual quality as well as performance. Besides implementing the designed method, the experiment also includes a preliminary test in which external participants provided feedback to our method. The section ends with an objective evaluation on the proposed method with 2 other representative methods.

### 4.1 Implementation

**Environment Setup** Our experiment runs on a system with NVIDIA GeForce RTX 2020 Ti. Our apparatus is HTC VIVE Pro Eye, which has a resolution of  $2 \times 1440 \times 1600$  and integrates an 120 hz eye tracker [4]. We use Unity engine as the experiment platform, utilizing its HDRP [5] pipeline which enables ray-traced and physically-based effects. However, if the frame rate is too low for a smooth VR experience for a certain scene, we have an option to switch to Unity’s conventional pipeline. VIVE Pro Eye has an

eye tracking SDK SRanipal [6] for Unity, providing data including the gaze origin & direction in 3D space and pupil diameter. Note that with a run time pupil diameter, instead of Equations 8 and 9 we use Equations 6 and 7 for a more precise DOF simulation.

**LOD manipulation** The SRanipal SDK provides a call to access the CGR, which can be brought into Equation 14 for the LOD manipulation. This process runs on CPU prior to shading, which adds some processing time. However, once the manipulation completes, an simplified scene with much less triangles - 50%- 80% depending on the underlying algorithm - will be passed to the shading phase. As [64] figured out that shading has often been found to dominate the cost for rendering, an overhead saving for shading has a potential to boost the performance.

**Shading** We implement the algorithms proposed in Section 3.4 with HLSL in Unity. The method is optimized towards deferred rendering pipeline to save overheads in the fragment shader. Our implementation is divided into 2 custom render passes [60]:

The first pass is injected right after the G-buffer is generated. The pass consists of the following steps as in Algorithm 1:

- Access the depth map and G-buffer. Calculate DOF related parameters including  $D_{Nv}$  and  $D_{Fv}$  (or  $D_{Ne}$  and  $D_{Fe}$  for the conventional display) by Equations 8 and 9.
- Apply forward transformation for G-buffer by Equation 2.
- Apply Russian Roulette filter  $\sigma$  (Equation 11) based on  $S_D$  (Equation 10). Compute lighting in P-buffer on top of Russian Roulette. So far we get the shaded result in polar space.
- Apply inverse transformation for the shaded content from polar coordinates to Cartesian coordinate by Equation 3.

The second pass is injected before post-processing in the rendering pipeline, and it follows Algorithm 2:

- Apply Bokeh kernel on the samples whose  $w_d < 1$  as the low-pass filters to simulate DOF blur.
- Or apply Bokeh with LCA simulation if enabled as Section 3.3.2 described for a color filtering.

To illustrate some features of our HVM, Figure 5 displays an LOD manipulation and an LCA simulation example in Crytek Sponza.

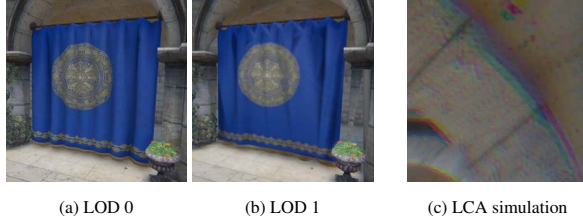


Figure 5: The LOD manipulation and LCA simulation

## 4.2 Preliminary Test

We also conduct a preliminary test. The test is to validate the pipeline. 4 people (age 24-35, 2 males and 2 females) participated in the test. They were naive regarding the aim of the experiment. All had self-reported normal or corrected-to-normal vision acuity (with corrective lenses for 3 of them), and normal color vision. During the experiment every user is guided through the eye-tracker calibration, plus lens and inter-pupillary distance adjustment towards their best comfortableness. The test includes 3 rendering stimuli: conventional VR rendering, default foveated rendering by VIVE [66] and our method. The test scene is Crytek Sponza [38], which features from rich occlusion and wide range of depth.

The preliminary test proved the feasibility of our method. Some observations include: 1. The participants did not point out the LCA effect until we asked them to pay attention to as a run time feedback.

2. The participants are highly tolerant with the simplified geometries even when the out-of-focus object is right on the occlusion edge. 3. The participants can sense the flickering which is due to the accuracy limitation of the eye tracker in tracking the saccade.

Besides, from the test, we get some feedback that we adopted to optimize the method, which increases the naturalness of the emulation and the perceptual quality. For example, Equation 13 tweaks the COC size to alleviate the temporal aliasing. This design choice is derived from the feedback of the preliminary test.

## 4.3 Evaluation

To evaluate the proposed method, we applied an experiment on 6 scenes: Crytek Sponza, San Miguel, Salle de Bain, Fireplace, Breakfast room, and Small office. The datasets are from the archive [38] except for Small office, which is a Unity HDRP sample scene [65]. A data format of .obj + .mtl makes the archive more friendly than their original source [10]. The test scenes' lighting complexity are calibrated to a similar level (i.e. a configuration of 3 real-time light sources).

We compare our method with Meng et al. [42] and Koskela et al. [29]. In which [42] has an open-source Unity project [71]. Note that the Unity implementation differs from the original approach as described in [42], however it still provides a fair comparison by adopting the same Unity HDRP pipeline as us.

### 4.3.1 Objective Assessment

General metrics, e.g. pixel-based metrics such as PSNR and structural-similarity metrics such as MS-SSIM or LPIPS, are taking blurriness, geometry distortion and chromatic aberration as artifacts. Although unsatisfactory blurriness or chromatic aberration should still be regarded as artifacts for our evaluation, more often blurriness and LCA are emulations of the limitation of HVS and should not be accounted as errors from the perspective of perceptual quality.

While to the best of our knowledge, no metric accounting for visual phenomenon such as DOF and LCA as features instead of artifacts for rendering, there are some structural-similarity metrics that taking foveation into account. Such as [58] presented an evaluation model that parameterize the foveal and peripheral area in the scene. However, the model is aimed at the conventional foveated rendering that applies several concentric circles with different shading rates, and is not applicable for the methods that apply a sophisticated scaling strategy. An emerging perception-based visible difference metric - named FovVideoVDP [35] - meets partial needs of our work as it accounts the angular spatial VAE with a HVM. Thus, for the foveated scaling, FovVideoVDP scores are of good reference value. The model regress the visual difference into just objectionable-difference (JOD) units with a 10 point scale (10 reports for no difference between a pair of content).

### 4.3.2 Procedure

**Test A** We first test on a conventional display to control the variables to ensure that the sequence used for evaluation is consistent in content across 3 different methods, which is important for a fair comparison. First we use VIVE Pro Eye to collect a set of pre-designed eye trajectories in the selected scenes. Then, in each scene, we preset the camera path coupled with the pre-recorded eye trajectories as a simulation of the head movement with fixation map into the rendering pipeline. With this setup we ensure that the rendering behavior is shared across methods. We render the result with a resolution of  $1440 \times 1600$ . In the end we record: 1. the frame rate within the interval of the preset rendering behavior, and 2. a sequence of the rendering outputs and apply FovVideoVDP to compare the scores among the methods. Note in this test, the pupil diameter  $\tau_e$  is simulated to 4.49 mm as introduced in Section 3.2 and Equations 8, 9 are applied for DOF parameters.

**Test B** Besides the test above. To reflect the actual latency in VR, we also setup an independent test for our method to record the frame rates with random camera paths and fixation maps on apparatus listed in the Section 4.1.

### 4.3.3 Results

**Visual inspection** Figure 6 displays the outputs of our hybrid blur rendering method for 6 different scenes. In which in fovea & in focus and peripheral / out-of-focus patches are highlighted. To make the selected results more representative, we chose different focuses on both diffuse & glossy / near & far / center & surrounding objects across scenes. Overall, we observed that the rendering scaling successfully adapts our model: The majority of budget is allocated to the in fovea & in focus pixels, thus the details are well preserved that complex phenomena such as refraction and reflection maintain high-fidelity.

Figure 7 illustrates the comparison of different methods on a view in Small office. Since Koskela’s method is path-traced, the result is more physically-correct in lighting / shadows plus refraction / reflection. For Meng’s method with the default setup ( $\sigma = 1.8$ ,  $\alpha = 4$ ), aliasing can be seen in the peripheral. In contrast, our result balanced the performance and the visual quality.

**Frame rates** Table 1 depicts the average frame rate for Test A across the HDRP scenes. Our method achieves better performance in comparison with Meng. Note in our test, fps for Koskela are significantly below the others ( $< 10$  fps). However, without an original open-source implementation, we are not positive about having all the optimization features.

Table 2 depicts the average frame rate for Test B: VR performance of our method. When coupled with the motion smoothing [2] from VIVE, 45 fps can be reprojected to achieve a 90 hz smoothness in general. 45 fps is achieved in 4 out of 6 test scenes, while the other 2 scenes have a video level (30 fps) of smoothness. As [7] suggested a total system latency of 50–70ms could be tolerant for foveated VR rendering. Thus our method is VR-compatible.

Table 1: Avg. fps of HDRP scenes with ours and Meng on the conventional display (w/ res. of  $1440 \times 1600$ ).

	Small office	Salle de Bain	Crytek Sponza	San Miguel	Fireplace	Breakfast room
Ours	47.1	40.6	36.8	31.9	51.1	47.1
Meng et al.	32.4	39.2	29.7	24.6	41.3	44.2

Table 2: Avg. fps of HDRP scenes with our method in VR (w/ full res. of  $2880 \times 1600$ ).

	Small office	Salle de Bain	Crytek Sponza	San Miguel	Fireplace	Breakfast room
in HTC VIVE Pro Eye	44.3	38.1	32.6	25.2	48.7	42.2

**FovVideoVDP** The references for ours and Meng are the default HDRP outputs of Unity while the references for Koskela are the path traced outputs from Unity (with a convolution period of 30 minutes). Table 3 and Table 4 depict the regressed JOD of the perceived difference. The JOD pool with all the frames within the period of the preset rendering behavior in the Test A. The robustness of JOD across scenes is represent by SD (i.e. standard deviation). Ours and Koskela have a same level of robustness which is better than Meng. Figure 8 derives from the PDF  $\Delta$  (i.e. JOD difference) in Table 3. (A  $\Delta$  of 1 means 75% of the population will choose ours over Meng [35].) A comparison of the CDF area of ours (blue) vs Meng (red) indicates the population is more probable to select our method over Meng across the test scenes. Besides, while we cannot make a direct numerical JOD comparison for Koskela due to the difference in the references, a cross-comparison with visual inspection indicates Koskela has least observable aliasing in peripheral.

Table 3: JOD of HDRP results with ours and Meng on the conventional display (w/ res. of  $1440 \times 1600$ ).

	Small office	Salle de Bain	Crytek Sponza	San Miguel	Fireplace	Breakfast room	SD
Ours	8.50	8.69	7.96	7.61	8.32	8.44	0.40
Meng et al.	8.35	8.12	7.74	7.08	8.41	8.47	0.54
$\Delta$	0.15	0.57	0.22	0.53	-0.09	-0.03	

Table 4: JOD of path traced result with Koskela on the conventional display (w/ res. of  $1440 \times 1600$ ).

	Small office	Salle de Bain	Crytek Sponza	San Miguel	Fireplace	Breakfast room	SD
Koskela et al.	9.12	8.98	8.46	8.29	9.00	9.31	0.40

Figure 9 illustrates the difference maps of JOD for some sample views in the selected scenes of our method, which discloses how the objectionable distortions are distributed within the frame. The redder, the higher objectionable in difference. As the foveated quality metric model in FovVideoVDP accounts for the angular spatial resolution changes with VAE, the foveal area is more sensitive to object the difference, which is reflected in the Figure 9 where the foveal area is more red. However, if the metric completely matches our foveation model which also accounts for DOF within the fovea, the foveal area should not be more red, unfortunately we cannot modify the metric model adaptively. The good thing is that the periphery is bluer, which means even if we do a scaling in the peripheral area, the quality loss has a low chance to be observed by the users.

## 5 DISCUSSION

As DOF relies on the depth buffer, our  $S_D$  does not account for translucent geometry (whose depth is omitted in the depth buffer). HDRP pipeline processes transparency after deferred lighting, which is out of the scope for this paper. A ray-tracing rendering pipeline [69] [29] can be a possible solution to integrate transparency to DOF scaling. Yet the ray-traced methods are computational intense.

Utilizing Unity’s HDRP improve the rendering fidelity compared to the result we get in the conventional rendering pipeline. However, the conventional pipeline does provide a good performance. San Miguel has the lowest average fps which is 260, while the simplest scene Fireplace has a average fps of 1090. A possible improving direction is to select the features from HDRP pipeline and implement them through scriptable rendering pipeline [61] to balance the quality and performance.

For the flickering from the inaccurate eye tracking for the saccade in the preliminary test, one possible solution can be adapting learning-based optimization method such as [28], which has an accuracy of 5 pixels and  $2.06 \pm 0.44^\circ$  accuracy across a  $30^\circ \times 40^\circ$  FOV. Unfortunately, the quarantine policy for Covid-19 restricts us from taking more periodic subjective tests to iterate the adapted modifications from the feedback of the preliminary test.

### 5.1 Future Work

We plan to have a comprehensive subjective test with an ablation study to model the relationship of perceptual quality and features (foveation, DOF, LOD and LCA) in our method. Besides, we can have a cross-comparison of the perceptual quality contribution of each module and the rendering time it saves / adds. Furthermore, with a subjective test we can also verify 1. whether our hybrid rendering has a contribution to relieve VAC, 2 whether our perception-driven rendering can drive users’ attention inversely.

To add new features to our HVM, we can model for VAE in color vision as discussed in [70] in addition to the LCA in DOF to expand the color vision emulation.

## 6 CONCLUSION

In this article, we develop a novel perception-driven hybrid rendering method that emulates the natural visual process. The physically



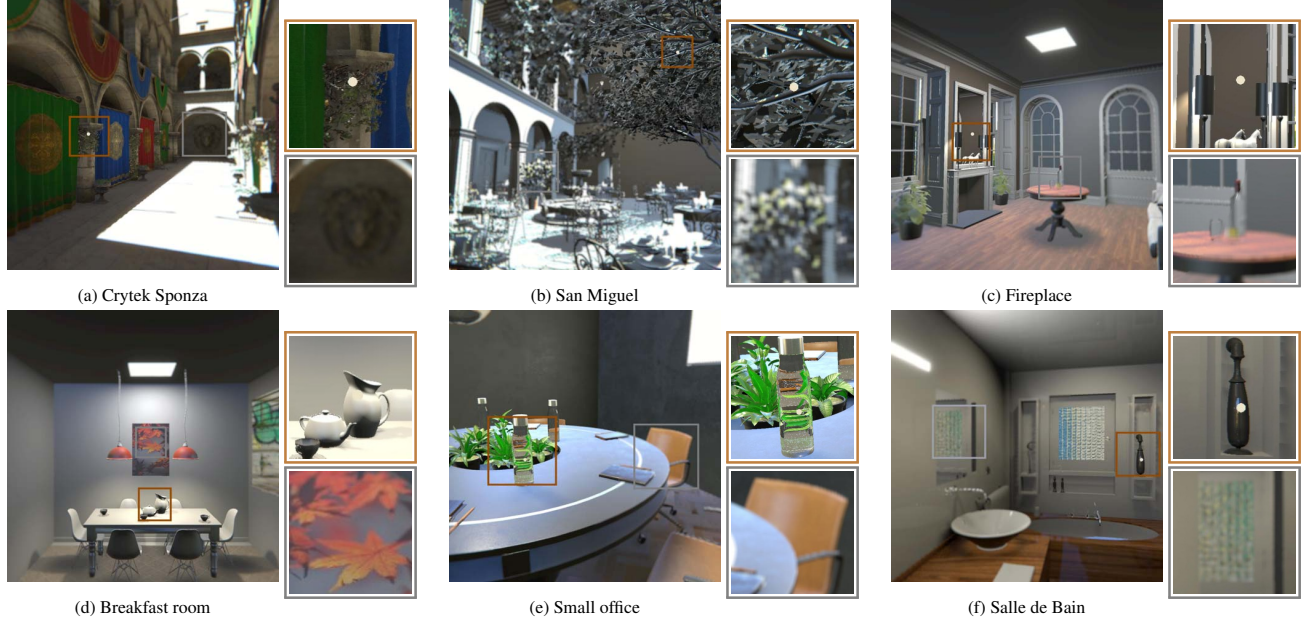


Figure 6: Retinal blur result with our hybrid method in Unity HDRP rendering pipeline

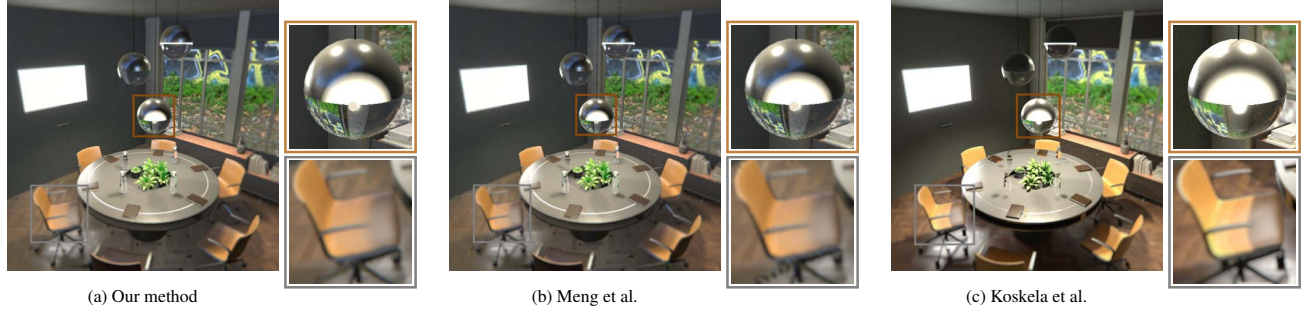


Figure 7: Different methods' rendering result.

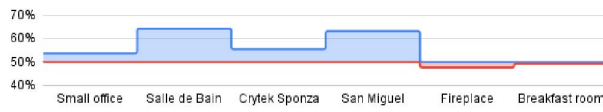


Figure 8: Probability of selecting ours over Meng based on JOD

accurate retinal blur is a combination of processes derived from the parameters in our hybrid HVM: peripheral geometry simplification, foveated scaling, stochastic DOF sampling, and Bokeh with LCA. The results validate that our method supports high-fidelity visual experience while being VR-ready in performance.

Our work also stimulates further research, such as 1. utilizing the method to explore the software-based solution to relieve VAC in the conventional VR systems, 2. using hybrid retinal blur to reversely guide users' behavior. These potential applications can be interesting topics for VR.

#### ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie Grant Agreement No. 765911 (RealVision).

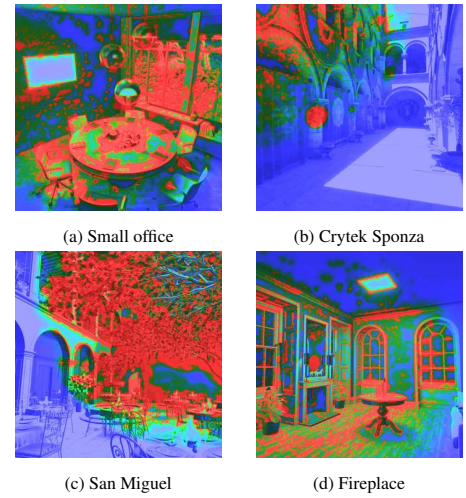


Figure 9: Difference maps on JOD of the selected scenes

## REFERENCES

- [1] Catlike-Coding/Depth of Field, 2018.
- [2] Steam :: SteamVR :: Introducing SteamVR Motion Smoothing, Nov. 2018.
- [3] Maple/Depth of field for optical lens - Wikibooks, open books for an open world, 2019.
- [4] VIVE Pro Eye | VIVE, 2019.
- [5] High Definition Render Pipeline overview | High Definition RP | 10.2.2, 2020.
- [6] VIVE Eye Tracking SDK (SRanipal), 2020.
- [7] R. Albert, A. Patney, D. Luebke, and J. Kim. Latency requirements for foveated rendering in virtual reality. *ACM Transactions on Applied Perception (TAP)*, 14(4):1–13, 2017.
- [8] A. T. Bahill, M. R. Clark, and L. Stark. The main sequence, a tool for studying human eye movements. *Mathematical biosciences*, 24(3-4):191–204, 1975.
- [9] A. C. Beers, M. Agrawala, and N. Chaddha. Rendering from compressed textures. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 373–378, 1996.
- [10] B. Bitterli. Rendering resources, 2016. <https://benedikt-bitterli.me/resources/>.
- [11] F. J. Carreras, J. Medina, M. Ruiz-Lozano, I. Carreras, and J. L. Castro. Virtual Tissue Engineering and Optic Pathways: Plotting the Course of the Axons in the Retinal Nerve Fiber Layer. *Invest. Ophthalmol. Vis. Sci.*, 55(5):3107–3119, May 2014. doi: 10.1167/iovs.13-13387
- [12] S. A. Cholewiak, G. D. Love, P. P. Srinivasan, R. Ng, and M. S. Banks. Chromabur: Rendering Chromatic Eye Aberration Improves Accommodation and Realism. *ACM Trans. Graph.*, 36(6):210:1–210:12, Nov. 2017. doi: 10.1145/3130800.3130815
- [13] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pp. 137–145, 1984.
- [14] C. A. Curcio and K. A. Allen. Topography of ganglion cells in human retina. *Journal of Comparative Neurology*, 300(1):5–25, 1990. doi: 10.1002/cne.903000103
- [15] A. T. Duchowski and A. Çöltekin. Foveated gaze-contingent displays for peripheral LOD management, 3D visualization, and stereo imaging. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(4):6:1–6:18, Dec. 2007. doi: 10.1145/1314303.1314309
- [16] 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.
- [17] Y. L. Grand. *Light, Colour and Vision: Approved translation from the French by R. W. G. Hunt, J. W. T. Walsh and F. R. W. Hunt*. Springer US, 2 ed., 1968.
- [18] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. Foveated 3D graphics. *ACM Trans. Graph.*, 31(6):164:1–164:10, Nov. 2012. doi: 10.1145/2366145.2366183
- [19] Y. He, Y. Gu, and K. Fatahalian. Extending the graphics pipeline with adaptive, multi-rate shading. *ACM Trans. Graph.*, 33(4):142:1–142:12, July 2014. doi: 10.1145/2601097.2601105
- [20] R. T. Held, E. A. Cooper, and M. S. Banks. Blur and disparity are complementary cues to depth. *Current biology*, 22(5):426–431, 2012.
- [21] R. T. Held, E. A. Cooper, J. F. O’Brien, and M. S. Banks. Using blur to affect perceived distance and size. *ACM transactions on graphics*, 29(2), 2010.
- [22] D. M. Hoffman, A. R. Girshick, K. Akeley, and M. S. Banks. Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of vision*, 8(3):33–33, 2008.
- [23] W. Hunt, M. Mara, and A. Nankervis. Hierarchical visibility for virtual reality. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–18, 2018.
- [24] 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):212:1–212:13, Nov. 2019. doi: 10.1145/3355089.3356557
- [25] B. Karis. High-quality temporal supersampling. *Advances in Real-Time Rendering in Games, SIGGRAPH Courses*, 1(10.1145):2614028–2615455, 2014.
- [26] P. Kellnhofer, P. Didyk, K. Myszkowski, M. M. Hefeeda, H.-P. Seidel, and W. Matusik. Gazestereo3d: Seamless disparity manipulations. *ACM Transactions on Graphics (TOG)*, 35(4):1–13, 2016.
- [27] J. Kim, Y. Jeong, M. Stengel, K. Akşit, R. Albert, B. Boudaoud, T. Greer, J. Kim, W. Lopes, Z. Majercik, et al. Foveated ar: dynamically-foveated augmented reality display. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.
- [28] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke. NVGaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI ’19, pp. 1–12. Association for Computing Machinery, May 2019. doi: 10.1145/3290605.3300780
- [29] M. Koskela, A. Lotvonen, M. Mäkitalo, P. Kivi, T. Viitanen, and P. Jääskeläinen. Foveated Real-Time Path Tracing in Visual-Polar Space. The Eurographics Association. doi: handle/10.2312/sr20191219
- [30] 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.
- [31] G. A. Kouliris, G. Drettakis, D. Cunningham, and K. Mania. Gaze prediction using machine learning for dynamic stereo manipulation in games. In *2016 IEEE Virtual Reality (VR)*, pp. 113–120, Mar. 2016. ISSN: 2375-5334. doi: 10.1109/VR.2016.7504694
- [32] S. Lee, G. J. Kim, and S. Choi. Real-time depth-of-field rendering using anisotropically filtered mipmap interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):453–464, 2009.
- [33] D. Luebke and B. Hallen. Perceptually driven simplification for interactive rendering. In *Proceedings of the 12th Eurographics conference on Rendering*, EGWR’01, pp. 223–234. Eurographics Association, June 2001.
- [34] R. Mantiuk, B. Bazyluk, and A. Tomaszewska. Gaze-Dependent Depth-of-Field Effect Rendering in Virtual Environments. In M. Ma, M. Fradinho Oliveira, and J. Madeiras Pereira, eds., *Serious Games Development and Applications*, Lecture Notes in Computer Science, pp. 1–12. Springer, Berlin, Heidelberg, 2011. doi: 10.1007/978-3-642-23834-5\_1
- [35] R. 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(6):19–32, Aug. 2021. doi: 10.1145/3450626.3459831
- [36] F. Maqsood. Effects of varying light conditions and refractive error on pupil size. *Cogent Medicine*, 4(1):1338824, Jan. 2017. Publisher: Cogent OA. eprint: <https://doi.org/10.1080/2331205X.2017.1338824>. doi: 10.1080/2331205X.2017.1338824
- [37] M. Mauderer, S. Conte, M. A. Nacenta, and D. Vishwanath. Depth perception with gaze-contingent depth of field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 217–226, 2014.
- [38] M. McGuire. Computer graphics archive, July 2017.
- [39] L. McIntosh, B. E. Riecke, and S. DiPaola. Efficiently simulating the bokeh of polygonal apertures in a post-process depth of field shader. In *Computer Graphics Forum*, vol. 31, pp. 1810–1822. Wiley Online Library, 2012.
- [40] A. Mehrfard, J. Fotouhi, G. Taylor, T. Forster, N. Navab, and B. Fuerst. A Comparative Analysis of Virtual Reality Head-Mounted Display Systems. *arXiv:1912.02913 [cs]*, Dec. 2019. arXiv: 1912.02913.
- [41] X. Meng, R. Du, and A. Varshney. Eye-dominance-guided foveated rendering. *IEEE transactions on visualization and computer graphics*, 26(5):1972–1980, 2020.
- [42] X. Meng, R. Du, M. Zwicker, and A. Varshney. Kernel foveated rendering. 1(1):5:1–5:20, July 2018. doi: 10.1145/3203199
- [43] O. Mercier, Y. Sulai, K. Mackenzie, M. Zannoli, J. Hillis, D. Nowrouzezahrai, and D. Lanman. Fast gaze-contingent optimal decompositions for multifocal displays. *ACM Transactions on Graphics*, 36(6):1–15, Nov. 2017. doi: 10.1145/3130800.3130846
- [44] J. H. Mueller, T. Neff, P. Voglreiter, M. Steinberger, and D. Schmalstieg. Temporally adaptive shading reuse for real-time rendering and virtual

- reality. *ACM Transactions on Graphics (TOG)*, 40(2):1–14, 2021.
- [45] J. D. Mulder and R. Van Liere. Fast perception-based depth of field rendering. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 129–133, 2000.
- [46] Nvidia. Nvidia/Variable Rate Shading, Jan. 2018. original-date: 2019-05-30T12:02:24Z.
- [47] 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):179:1–179:12, Nov. 2016. doi: 10.1145/2980179.2980246
- [48] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.
- [49] M. Reddy. Perceptually optimized 3d graphics. *IEEE computer Graphics and Applications*, 21(5):68–75, 2001.
- [50] J. Rovamo and A. Raninen. Critical flicker frequency and m-scaling of stimulus size and retinal illuminance. *Vision research*, 24(10):1127–1131, 1984.
- [51] T. Shibata, J. Kim, D. M. Hoffman, and M. S. Banks. Visual discomfort with stereo displays: effects of viewing distance and direction of vergence-accommodation conflict. In *Stereoscopic Displays and Applications XXII*, vol. 7863, p. 78630P. International Society for Optics and Photonics, 2011.
- [52] J. Sjöstrand, V. Olsson, Z. Popovic, and N. Conradi. Quantitative estimations of foveal and extra-foveal retinal circuitry in humans. *Vision Research*, 39(18):2987–2998, Sept. 1999. doi: 10.1016/S0042-6989(99)00030-9
- [53] Spacerat. sp4cerat/Fast-Quadric-Mesh-Simplification, 2015.
- [54] J. Spjut, B. Boudaoud, J. Kim, T. Greer, R. Albert, M. Stengel, K. Akşit, and D. Luebke. Toward standardized classification of foveated displays. *IEEE transactions on visualization and computer graphics*, 26(5):2126–2134, 2020.
- [55] M. Stengel, S. Grogorick, M. Eisemann, and M. Magnor. Adaptive image-space sampling for gaze-contingent real-time rendering. In *Computer Graphics Forum*, vol. 35, pp. 129–139. Wiley Online Library, 2016.
- [56] 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):192:1–192:13, Nov. 2017. doi: 10.1145/3130800.3130807
- [57] Q. Sun, F.-C. Huang, L.-Y. Wei, D. Luebke, A. Kaufman, and J. Kim. Eccentricity effects on blur and depth perception. *Optics express*, 28(5):6734–6739, 2020.
- [58] N. T. Swafford, J. A. Iglesias-Guitian, C. Koniaris, B. Moon, D. Cosker, and K. Mitchell. User, metric, and computational evaluation of foveated rendering methods. In *Proceedings of the ACM Symposium on Applied Perception*, pp. 7–14, 2016.
- [59] U. Technologies. Unity - Manual: Depth of Field.
- [60] U. Technologies. Unity - Manual: Injection Points.
- [61] U. Technologies. Unity - Manual: Scriptable Render Pipeline.
- [62] O. T. Tursun, E. Arabadzhyska-Koleva, M. Wernikowski, R. Mantiuk, H.-P. Seidel, K. Myszkowski, and P. Didyk. Luminance-contrast-aware Foveated Rendering. *ACM Trans. Graph.*, 38(4):98:1–98:14, July 2019. doi: 10.1145/3306346.3322985
- [63] Unity. Unity-Technologies/PostProcessing, 2020.
- [64] K. Vaidyanathan, M. Salvi, R. Toth, T. Foley, T. Akenine-Möller, J. Nilsson, J. Munkberg, J. Hasselgren, M. Sugihara, P. Clarberg, T. Janczak, and A. Lefohn. Coarse pixel shading. In *Proceedings of High Performance Graphics*, HPG ’14, pp. 9–18. Eurographics Association, Lyon, France, June 2014. doi: 10.5555/2980009.2980011
- [65] ViveSoftware. Unity-Technologies/SmallOfficeRayTracing, 2020.
- [66] ViveSoftware. ViveSoftware/ViveFoveatedRendering, Jan. 2020. original-date: 2019-05-30T12:02:24Z.
- [67] 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 Transactions on Graphics (TOG)*, 40(4):1–14, 2021.
- [68] M. Weier, T. Roth, A. Hinkenjann, and P. Slusallek. Foveated depth-of-field filtering in head-mounted displays. *ACM Transactions on Applied Perception (TAP)*, 15(4):1–14, 2018.
- [69] 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. *Computer Graphics Forum*, 35(7):289–298, 2016. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13026>. doi: 10.1111/cgf.13026
- [70] M. Weier, M. Stengel, T. Roth, P. Didyk, E. Eisemann, M. Eisemann, S. Grogorick, A. Hinkenjann, E. Kruijff, M. Magnor, K. Myszkowski, and P. Slusallek. Perception-driven Accelerated Rendering. *Computer Graphics Forum*, 36(2):611–643, 2017. doi: 10.1111/cgf.13150
- [71] M. X. Kernel foveated rendering in unity, 2019.
- [72] L. Xiao, A. Kaplanyan, A. Fix, M. Chapman, and D. Lanman. Deep-Focus: Learned Image Synthesis for Computational Displays. *ACM Trans. Graph.*, 37(6):200:1–200:13, Dec. 2018. doi: 10.1145/3272127.3275032