VPRF: Visual Perceptual Radiance Fields for Foveated Image Synthesis

Zijun Wang, Jian Wu, Runze Fan, Wei Ke, and Lili Wang

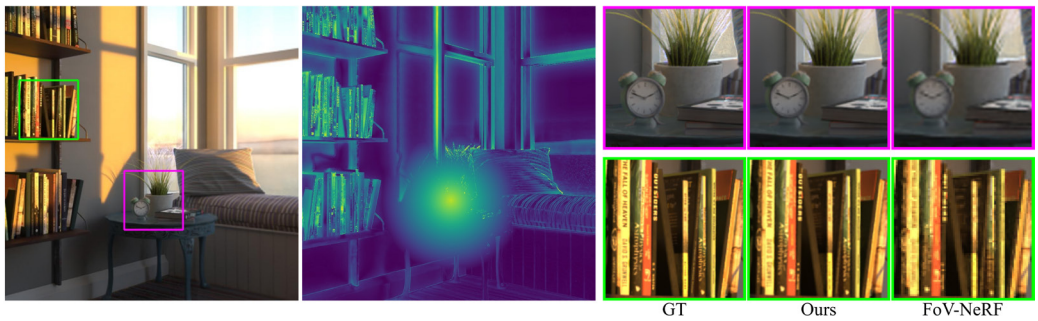


Fig. 1: Left: Foveated images synthesized by our method. Middle: Visual sampling rate map generated by our method. Right: Magnified images of our rendering results. Compared to the state-of-the-art FoV-NeRF [1] method, our results are closer to the ground truth. The PSNR of our method achieves $1.34\times$ in the foveal region (purple square) and $1.69\times$ in the salient region in the periphery (green square). While preserving more details in both regions, our method is about $2.6\times$ faster than FoV-NeRF.

Abstract— Neural radiance fields (NeRF) has achieved revolutionary breakthrough in the novel view synthesis task for complex 3D scenes. However, this new paradigm struggles to meet the requirements for real-time rendering and high perceptual quality in virtual reality. In this paper, we propose VPRF, a novel visual perceptual based radiance fields representation method, which for the first time integrates the visual acuity and contrast sensitivity models of human visual system (HVS) into the radiance field rendering framework. Initially, we encode both the appearance and visual sensitivity information of the scene into our radiance field representation. Then, we propose a visual perceptual sampling strategy, allocating computational resources according to the HVS sensitivity of different regions. Finally, we propose a sampling weight-constrained training scheme to ensure the effectiveness of our sampling strategy and improve the representation of the radiance field based on the scene content. Experimental results demonstrate that our method renders more efficiently, with higher PSNR and SSIM in the foveal and salient regions compared to the state-of-the-art FoV-NeRF. The results of the user study confirm that our rendering results exhibit high-fidelity visual perception.

Index Terms— Virtual reality, Foveated rendering, Visual perceptual, Contrast sensitivity

1 Introduction

Novel view synthesis task offers significant benefits for virtual reality (VR) by presenting high-quality rendering results for users. Neural radiance fields (NeRF) [2] leverages a learning-based approach to capture the geometry of scenes as well as view-dependent effects, achieving realistic novel view synthesis results through conventional volumetric rendering techniques, which is highly valuable for VR. NeRF utilizes images from various views of a given scene as input, using an implicit multilayer perceptron (MLP) to map the spatial information of 3D points to the color and density attributes.

However, employing NeRF directly in VR presents rendering speed bottlenecks. Although NeRF and its MLP-based variants exhibit exceptional quality in view synthesis, the training time for such representations spans 1-2 days, with nearly 30 seconds needed to render a single frame image, failing to meet the rendering performance requirements of virtual environments. Compared to MLP-based approaches, voxel-based methods offer superior rendering performance. Recently, Yu et al. [3] demonstrated the impressive efficiency of the voxel-based method (Plenoxels) by assigning spherical harmonics coefficients to each voxel for view-dependent appearance synthesis. While maintaining quality comparable to NeRF, Plenoxels achieves a rendering speed that is two orders of magnitude faster, with training times reduced to 27 minutes. To further enhance the inference performance of NeRF, existing work has focused on designing dedicated sampling networks to estimate appropriate sample locations, thereby reducing the number of samples required per view ray and accelerating inference. These sampling networks are typically optimized under supervision based on depth prediction [4] or the density distribution [5] of the radiance field.

Existing NeRF acceleration methods considered only the scene's geometric information (depth, opacity) without leveraging the perceptual characteristics of the human visual system (HVS) for optimization. Foveated rendering is an acceleration rendering technique based on the perceptual model of the HVS, allocating computational resources to render high-quality images for the foveal region while lower-quality for the peripheral region. Deng et al. [1] proposed FoV-NeRF, which implicitly represents the scene by allocating two networks with varying parameter quantities for the foveal region and peripheral region and renders images of different quality. Nevertheless, FoV-NeRF faces several challenges. Firstly, using implicit MLP-based representation demands more computational resources due to the dense network inferencing, which challenges the rendering performance. Secondly, studies have shown that many visual features of the peripheral region, such as saliency, have a significant impact on visual perceptual quality [6]. FoV-NeRF focuses exclusively on the rendering quality of the foveal region, neglecting the radiance details of peripheral salient

- *Lili Wang is with State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China; Peng Cheng Laboratory, Shengzhen, China; and Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing, China. Lili Wang is the corresponding author. E-mail: wanglily@buaa.edu.cn.*
- *Zijun Wang, Jian Wu, and Runze Fan are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. E-mail: 892710638@qq.com, lanayawj@buaa.edu.cn, by2106131@buaa.edu.cn.*
- *Wei Ke is with the Faculty of Applied Sciences, Macau Polytechnic University. E-mail: wke@mpu.edu.mo.*
- *Zijun Wang and Jian Wu are the co-first authors of this paper.*

Received 14 March 2024; revised 17 June 2024; accepted 1 July 2024.
Date of publication 11 September 2024; date of current version 4 October 2024.
This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2024.3456184, provided by the authors.
Digital Object Identifier no. 10.1109/TVCG.2024.3456184

regions. To further enhance rendering performance and overall perceptual quality, it is crucial to introduce visual perception based on scene content into the radiance field rendering model.

In this paper, we propose VPRF, a novel foveated radiance field representation method that integrates the visual acuity and contrast sensitivity models of HVS into the radiance field rendering framework. This method leverages input from multi-view RGB images and corresponding visual sensitivity images for end-to-end optimization, aiming to efficiently achieve high perceptual quality rendering results. First, VPRF is a view-dependent explicit voxel-based representation method, where each voxel stores optimizable feature vectors that encode the scene's geometric, appearance, and visual sensitivity information. This representation enables the visual sensitivity feature to be directly synthesized with novel views, the same as the scene appearance. Second, we propose a visual perceptual sampling strategy to improve the performance in radiance field rendering. Unlike the Fov-NeRF approach, which is based on network parameter quantity, our method controls the computational resource allocation for each pixel by regulating the sampling rate along each ray. Third, we propose a sampling weight-constrained training scheme for VPRF, which has a new loss function to limit the weight distribution of sampling points and ensure the effectiveness of the sampling strategy to improve the representation capability of the radiance field based on the scene content.

We compare our VPRF method with the state-of-the-art foveated neural radiance fields method and other NeRF acceleration methods on both real-world and synthetic datasets. Our method renders more efficiently (about 83FPS), with higher PSNR and SSIM in the foveal and salient regions. Compared to Plenoxels [3] and AdaNeRF [5], the efficiency of our method is improved by 5.1-13.1×, and the quality of synthesis in the foveal and salient regions is also enhanced. Compared to the FoV-NeRF method [1], our method achieves synthesis quality improvement in all regions. And at equal or superior quality, the efficiency of our method is improved by about 2.6×.

Figure 1 shows the comparison of the rendering results between our method, the ground truth, and the FoV-NeRF method. The foveal region is marked in green, and the salient region in the periphery is marked in purple. Details in these two regions are magnified on the right side. Our method achieves superior synthesis quality in both the foveal region and the salient region in the periphery. In the foveal region, we better preserve the details of the clock dial and the foliage. In the salient region, FoV-NeRF fails to retain the details of letters on the book spines and exhibits noticeable aliasing. We also conduct a user study to evaluate our method. The results show that the visual perceptual quality of our method has significantly increased compared to other methods.

In summary, the main contributions of our method are as follows:

- A voxel-based visual perceptual foveated radiance fields representation (VPRF). To the best of our knowledge, this is the first time that a visual sensitivity model and a visual contrast model have been combined into NeRF's rendering framework to improve computational efficiency.
- An adaptive sampling strategy based on visual perception in the NeRF inference process, which allocates rendering resources according to the HVS varying sensitivity across different scene regions, further improving perceptual quality and efficiency.
- A training scheme based on sampling weight constraints for end-to-end learning of our representation, which constrains the solution space of the scene geometry, provides support for the line-of-sight of the sampling strategy described above.

2 Related Work

In this section, we first introduce prior work related to foveated rendering and then discuss existing methods for NeRF acceleration. For a more exhaustive analysis of foveated rendering, we recommend the readers refer to the reviews [7, 8].

2.1 Foveated Rendering

Providing high-quality content to each location within head-mounted displays (HMD) is computationally expensive. Guenter et al. [9] proposed the first foveated rendering framework, leveraging a visual acuity fall-off model [10] to accelerate rendering performance without sacrificing perceptual synthesis quality. It rendered three nested image layers around the gaze point by rasterization, reducing the sampling rate with increasing eccentricity and fusing these layers into the final result through bilinear interpolation. Meng et al. [11] introduced a two-pass kernel foveated rendering pipeline that parameterizes foveated rendering by embedding polynomial kernel functions in a classic log-polar mapping, rendering results to a reduced resolution buffer and converting results back to full-resolution screen space via inverse log-polar transformation to output the final rendering results. Friston et al. [12] proposed a pipeline that achieves foveated rendering through ray casting for each fragment within a single rasterization process, overcoming the limitations of warping concerning disocclusions, object motion and view-dependent shading, as well as geometric aliasing artifacts. Tursun et al. [13] introduced a luminance contrast aware foveated ray tracing technique, demonstrating that significantly reducing the number of tracing rays is possible if the local spatial luminance contrast sensitivity function (CSF) [14] is considered in foveated rendering. Jindal et al. [15] proposed a variable-rate shading pipeline to accelerate rasterization rendering performance. They divided the full-resolution image into multiple 16×16 image blocks, which can be rendered with a different ratio of shader executions. Then, they adaptively adjust each image block's shading accuracy and refresh rate based on temporal and spatiotemporal luminance CSF. Murphy et al. [16] proposed a hybrid technique based on the visual acuity fall-off model and spatial CSF, utilizing ray casting to sample the geometry of scenes.

Beyond traditional ray-tracing [17–19] and rasterization-based foveated rendering [20, 21] methods, many researchers have concentrated on the application of deep learning in foveated rendering. Surace et al. [22] proposed a training procedure for a generative network designed for foveated image reconstruction. This procedure penalizes perceptually significant deviations in the output to preserve perceived over natural image statistics. Bauer et al. [23] used a two-pass deep neural reconstruction network derived from the W-Net model, which sparsely samples a volume around a focal point and reconstructs the full-resolution volumetric rendering result using a deep neural network. To further enhance the performance of foveated rendering, Kaplanyan et al. [24] utilized Generative Adversarial Networks (GANs) [25] to reconstruct peripheral regions, improving the peripheral rendering quality in foveated videos. This method reconstructs a plausible peripheral video from a small fraction of pixels provided in every frame by finding the closest matching video to this sparse input stream of pixels on the learned manifold of natural videos. However, conventional foveated rendering methods depend on 3D resources to reconstruct scene contents, and acquiring these resources in the real world often involves noise, which introduces challenges in presenting complex scene contents from a novel view in VR.

2.2 Neural Radiance Fields and Acceleration

To reconstruct 3D scenes and synthesize novel view images, the development of Neural radiance fields (NeRF) [2] has attracted extensive attention. NeRF uses fully connected neural networks to represent 3D scenes as an implicit function and optimizes it through differentiable volume rendering. It can recover the geometry and appearance information of a scene from input multi-view 2D images, training the network and rendering novel view images by sampling points in 3D space through ray marching. Although NeRF has achieved realistic synthesis quality, the rendering speed is considerably slow, which limits its practical applications.

To accelerate NeRF rendering, recent works [26–30] utilize explicit voxel grids to store the radiance and other features of a scene, avoiding the intensive inference of MLP during runtime. Hedman et al. [28] constructed a sparse 3D voxel grid to store the learned opacity, diffuse color, and view-dependent effects feature generated by a pre-trained NeRF. It accelerates the rendering performance by directly querying the voxel grid in the testing process. Yu et al. [26] proposed NeRF-SH, which uses the same optimization and volume rendering method as the original NeRF, predicts the spherical harmonics (SH) coefficients instead of color for each sampling point. These coefficients are used to synthesize view-dependent colors without the need for additional network inference. Finally, the SH coefficients and opacity are baked into a sparse voxel-based octree for real-time rendering. However, these methods even lengthen the overall training time by extracting features from a pre-trained NeRF and then baking them into explicit data structures. Yu et al. [3] proposed Plenoxels, which directly optimize the SH coefficients stored in the feature grid, achieving comparable rendering results to NeRF in only tens of minutes of

training time. This indicates that using an explicit feature grid can achieve fast optimization and inference without sacrificing quality. Müller et al. [30] proposed Instant-NGP, which optimizes features of sampling points adaptively, prioritizing the sparse areas with the most important fine scale detail through a hash table approach. During the training phase, the sampling points that significantly contribute to the final color will dominate. We use Plenoxels as our volume rendering backbone to implement the visual perceptual radiance fields representation with efficient performance. And unlike Instant-NGP, our adaptive sampling strategy is mainly applied in the testing phase to accelerate rendering efficiency according to the visual sensitivity.

Other works increased the rendering speed by improving the sampling efficiency of NeRF. The original NeRF employs a hierarchical sampling strategy, where the volume density distribution predicted by a coarse network is used to guide the sampling process of a fine network. Neff et al. [4] introduced a depth oracle network to replace the coarse network in original NeRF, designed to predict appropriate sample locations along each ray. By reducing the number of overall samples, this method improves the rendering performance. However, it does not support end-to-end training, and inaccurate depth information may affect the synthesis quality. Piala et al. [31] proposed to train a sampling network, which is supervised by the density predictions from a pre-trained NeRF. This network learns a mapping from camera rays to positions along the rays, selecting sampling points that are most likely to influence the final color of the pixels. Kurz et al. [5] introduced an end-to-end dual-network architecture, which learns sampling and shading networks simultaneously and only renders the output of the important sample by the sampling network to accelerate rendering. However, none of these methods considered the perceptual characteristics of HVS for the optimization of radiance field rendering performance. Deng et al. [1] presented FoV-NeRF, the first gaze-contingent neural radiance representation, and foveated synthesis approach, incorporating the psychophysics of human vision and stereo acuity into the egocentric neural representation of 3D scenes. Due to the degradation of visual acuity with increasing distance from the central line of sight [32], FoV-NeRF uses multiple MLPs to synthesize images for foveal, peripheral and far-peripheral regions. The foveal image is rendered with the highest quality, while the peripheral and far-peripheral images with lower quality, are then fused to generate the final foveated images in real time. Currently, no method combines the HVS's visual acuity with scene content contrast sensitivity to accelerate radiance field rendering. Our approach utilizes the advantages of explicit representations and further improves rendering performance by integrating scene awareness and human visual perception to achieve the goal of synthesizing high perceptual quality foveated images at high frame rates.

3 Method

Given a set of RGB images $\{\mathbf{I}_{rgb}\}$ and visual sensitivity images $\{\mathbf{I}_{vs}\}$ with corresponding 6-DoF calibrated camera parameters, our objective is to reconstruct the appearance and sensitivity representation of 3D scenes using feature grids, and synthesize high-quality foveated images from novel views. Fig. 2 shows the overview of our VPRF method. We introduce a new voxel-based visual perceptual foveated radiance fields representation, where each voxel encodes not only geometry but also appearance and visual sensitivity at its corresponding 3D location (Section 3.1). In runtime rendering, we propose a visual perceptual sampling strategy. A visual sampling rate (VSR) map is generated and used to guide adaptive sample selection along the ray, thus further enhancing the rendering efficiency without compromising visual perception quality (Section 3.2). In the training process, we propose a sampling weight-constrained training scheme for end-to-end learning of our representation. The feature values within each voxel are directly optimized by minimizing the discrepancy between rendered images and input images, as well as by predicting visual sensitivity maps and extracting visual sensitivity maps. Moreover, we introduce a new weight constraint loss to restrict the weight distribution of sampling points (Section 3.3).

3.1 Voxel-Based Visual Perceptual Foveated Radiance Fields Representation

Previous work [3] has already demonstrated the advantages of directly optimizing the spherical harmonic (SH) coefficients stored in explicit voxels for synthesizing novel view synthesis, which offers efficient computation speeds and higher synthesis quality compared to inferring density and color from MLPs [2]. Inspired by this idea, we propose our visual perceptual radiance field representation, which is based on the assumption that objects with high visual sensitivity within a scene retain their conspicuousness across different views, and visual sensitivity maps can be computed through volumetric rendering equations. We construct explicit feature grids of the density G_σ, the color G_c and the visual sensitivity G_s, storing density values σ, vectors of SH coefficients $\mathbf{k_c}$ and $\mathbf{k_s}$ for each color channel and a single visual sensitivity channel respectively. For a 3D point $\mathbf{q_i}$, its density and SH coefficient vectors are computed through trilinear interpolation from the nearest 8 voxels through Equation 1:

$$\mathbf{k_c}(x) = \mathscr{T}(G_c, x),\ \mathbf{k_s}(x) = \mathscr{T}(G_s, x),\ \sigma(x) = \mathscr{T}(G_\sigma, x) \quad (1)$$

where $\mathscr{T}$ denotes a trilinear interpolation, x is the 3D position of the sampling point. The SH coefficients $\mathbf{k}$ in each voxel are used for view-dependent evaluation. Given view direction $\mathbf{d}$, querying the predefined basis functions $Y_m^l(\mathbf{d})$, where l is the SH function degree and m is the order. The view-dependent color $\mathbf{c}$ and sensitivity value $\mathbf{s}$ of a sample are calculated by a weighted sum of $Y_m^l(\mathbf{d})$ for each channel and the corresponding optimized coefficients:

$$\mathbf{c}(x,\mathbf{d}) = f_{\mathrm{SH}}(\mathbf{k_c}(x),\mathbf{d}),\ \ \mathbf{s}(x,\mathbf{d}) = f_{\mathrm{SH}}(\mathbf{k_s}(x),\mathbf{d}) \quad (2)$$

$$f_{\mathrm{SH}}(\mathbf{k},\mathbf{d}) = S(\sum_{l=0}^{l_{\max}} \sum_{m=-l}^{l} \mathbf{k}_m^l Y_m^l(\mathbf{d})) \quad (3)$$

where $\mathrm{S} : \to (1+\exp(-x))^{-1}$ is the sigmoid function for normalizing the colors. We use spherical harmonics of degree 2 for color, allocating 9 coefficients per color channel. Given that visual sensitivity information in images is typically low-frequency, we opt to use spherical harmonics of degree 1 to model sensitivity, which requires 4 coefficients. Moreover, the decrease in the number of parameters allows for sensitivity to be computed quickly. Totally, each voxel contains a total of 32-dimensional vectors: 31 for SH coefficients and 1 for density σ.

The runtime rendering objective is mapping the 3D coordinates $(\mathbf{x},\mathbf{y},\mathbf{z})$ to a 32-dimensional feature vector $(\mathbf{k_c},\mathbf{k_s},\sigma)$, then combined with $\mathbf{d}$ through Equation 2 to synthesize view-dependent color and sensitivity. Following the volume rendering formula in NeRF, we calculate the color $C(\mathbf{r})$ and sensitivity feature $S(\mathbf{r})$ of ray $\mathbf{r}$ by:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1-\exp(-\sigma_i\delta_i))\mathbf{c}_i \quad (4)$$

$$\hat{S}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1-\exp(-\sigma_i\delta_i))\mathbf{s}_i \quad (5)$$

$$T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j\delta_j) \quad (6)$$

where δ_i is the intervals between samples and N is the number of the sampling points.

3.2 Visual Perceptual Sampling Strategy

In the foveated rendering technique, the most representative perceptual models include visual acuity and contrast sensitivity models [29, 33], which describe the human visual system (HVS) enhanced perceptual sensitivity to areas close to the retinal center and salient regions in the scene [34]. Existing works on NeRF rendering acceleration [5] calculate importance weights w_i for each sample along the ray $\mathbf{r}$, representing the contribution to the ray color $C(\mathbf{r})$, which is solely related to density. We calculate w_i by:

$$w_i = T_i(1-\exp(-\sigma_i\delta_i)) \quad (7)$$

where T_i is the accumulated transmittance at point $\mathbf{q_i}$ calculated by Equation 6. Next, samples with w_i below a threshold τ are discarded, not accumulated into the final color $C(\mathbf{r})$ to accelerate the rendering process. τ is typically a predefined constant. We propose a sampling strategy based on these two models of visual perception, dividing the rendering process into two stages: (1) Visual sampling rate map (VSR map) generation for current view. (2) Visual perceptual sampling for final foveated image synthesis. In the testing time, incorporating additional input from the user's gaze information, we use both visual acuity and contrast sensitivity

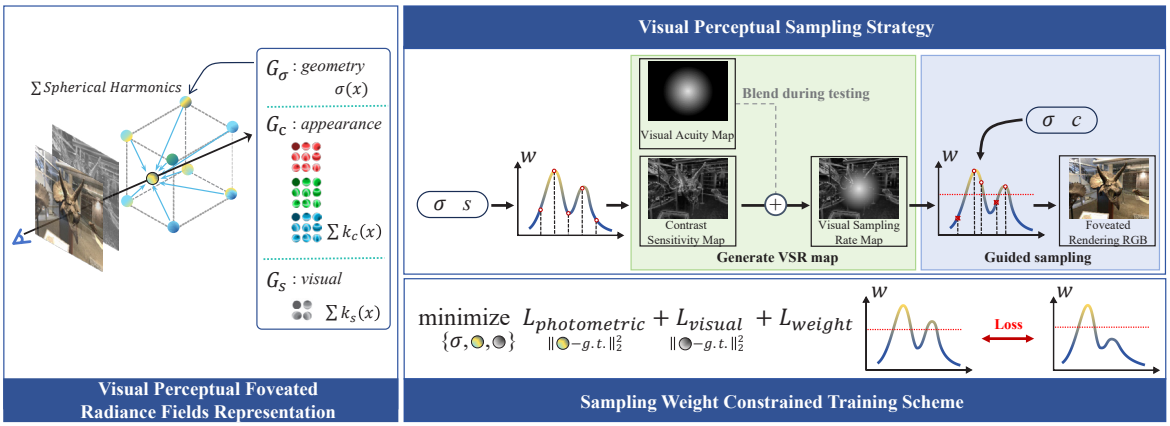


Fig. 2: The overview of VPRF method.

models to guide sampling, whereas in the training time, due to the absence of gaze information, only the contrast sensitivity model is used.

3.2.1 Visual Sampling Rate Map Generation

Definition Visual sampling rate map is a two-dimensional map of the same size as the input image, in which the value of each pixel represents the sampling rate $P([0,1])$ of the corresponding ray. The sampling rate P is calculated based on the visual sensitivity of the scene corresponding to the current pixel position. A higher P value indicates that we conduct dense sampling along the ray, whereas a lower value results in sparse sampling.

Generation To generate the VSR map, there are three steps as follows.

Step 1. Construct Visual Acuity Map.

The visual perception model is represented as a visual acuity map, in which the pixel value $V([0,1])$ indicates the user's visual acuity, calculated based on the gaze point. In the testing time, given a 2D gaze point $\mathbf{g}$, the acuity V for the pixel $\mathbf{p}$ is calculated as:

$$V = \omega_0 + m \cdot e(\mathbf{p}, \mathbf{g}) \tag{8}$$

where ω_0 represents the visual acuity limit at the gaze position, m represents the acuity slope, and e is the eccentricity function calculated based on $\mathbf{p}$ and $\mathbf{g}$.

Step 2. Construct Visual Sensitivity Map.

The contrast sensitivity model is represented as a visual sensitivity map(VS map), in which the value $S([0,1])$ indicates the sensitivity of HVS at this pixel. For single-image saliency detection, Yue et al. [35] proposed a co-saliency detection method that combines top-down and bottom-up approaches, where the backbone network is employed for co-saliency map prediction, and two branch networks are utilized to enhance the network's sensitivity to co-salient regions. Zhou et al. [36] introduced a hierarchical network structure that explores the role of foreground and background information in generating saliency maps. Our VS map is generated by employing the spatial contrast sensitivity function (CSF) to detect edges and salient features. Specifically, we consider areas that exceed the threshold of the spatial frequency function as significant areas, utilizing relative values of the spatial CSF rather than absolute values to indicate areas sensitive to the user.

Before training, we extract the VS map corresponding to each input image as the ground truth S for the predicted sensitivity feature $\hat{S}$, which is used to supervise the optimization of visual sensitivity grid G_s. In the training and testing time, for each sampling ray, G_s and density grid G_σ are sampled at uniform intervals, and the sensitivity value $\mathbf{s_i}$ and density $\mathbf{d_i}$ for each sample are calculated as mentioned in Section 3.1. Subsequently, through Equation 5. we compute the predicted sensitivity $\hat{S}$. To maintain the real-time performance of the entire rendering process, we first generate a low-resolution VS map for the current view, then upsample it using bilinear interpolation to restore the VS map to the same resolution as the output image. This ensures a one-to-one correspondence between each pixel on the VS map and each ray in the image synthesis stage.

Step 3. Generate Visual Sampling Rate Map.

Based on the visual acuity map and VS map, we calculate the final sampling rate $P = \max(V, S)$, referring to [20]. Next, P guides the sampling process of the ray during the image synthesis stage.

3.2.2 Visual Perceptual Sampling

During the sampling process, there are three important control parameters: the importance weight threshold, the number of sampling points and the ray marching step size. In existing NeRF acceleration methods [3,5,29], these parameters are typically predefined as constant values, uniformly accelerating across the overall image space. These methods neglect the characteristics of HVS: higher density samples are required in regions with high visual sensitivity, while fewer are needed elsewhere. To overcome this limitation, we propose using the value in each pixel of the VSR map to guide the computation of these three parameters during the sampling process, thereby allocating computational resources according to the sensitivity levels of HVS. Specifically, given sampled ray $\mathbf{r}_i$, we first get the sampling rate P_i from the VSR map. This process is comprised of three components.

Adaptive Importance Weights Threshold This parameter controls the threshold for filtering sampling points. Since the density grid G_σ has already been sampled during the visual perception stage, we have acquired coarse geometric information about the scene. For ray $\mathbf{r}_k$ at that stage, we additionally record the maximum w among all samples as $w^k_{\max}$, which is upsampled along with the predicted sensitivity feature $\hat{S}(\mathbf{r}_k)$ and stored in another channel of VS map. Assume that after upsampling, the pixel value $w^i_{\max}$ corresponding to pixel $\mathbf{p}_i$ approximates the actual maximum importance weight of samples on $\mathbf{r}_i$:

$$w^i_{\max} = pixel(\mathbf{r}_i) \approx w_{i,t},\ t = \underset{j\in[1,N]}{\arg\max}(\|w_j\|) \tag{9}$$

where $w_{i,t}$ indicates importance weight of the t-th sampling point along the ray $\mathbf{r}_i$. The threshold τ_i for $\mathbf{r}_i$ is defined as:

$$\tau_i = w^i_{\max} \cdot (1 - P_i) \tag{10}$$

Next, we sample the density grid G_σ once again, for each point $\mathbf{q}_{i,j}$ along $\mathbf{r}_i$, calculating the importance weight $w_{i,j}$. Only points where $w_{i,j} > \tau_i$ are sampled on the color grid G_c, and calculate the view-dependent color $\mathbf{c}_{i,j}$ through Equation 2. By reducing unnecessary color synthesis, the performance of image rendering is improved.

Sample Count Limitation This parameter controls the range of sample numbers. We limit the maximum number of allowed samples between $[N_{min}, N_{max}]$. The maximum sample count N_i for $\mathbf{r}_i$ is calculates based on the sampling rate P_i as:

$$N_i = \lceil P_i \cdot (N_{\max} - N_{\min}) \rceil + N_{\min} \tag{11}$$

where $\lceil\rceil$ is an upward rounding operation. When the number of samples exceeding τ_i surpasses N_i, we terminate the sampling process prematurely.

Adaptive Ray Marching Step This parameter controls each step size of ray marching. When $w_{i,j} > \tau_i$, it indicates that the current point $q_{i,j}$ contributes significantly to the final color $\hat{C}(\mathbf{r}_i)$. In practice, these points are mostly distributed on the surface, and we only need to focus on them with visible rays. Therefore, we adopt a conservative marching step to encourage sampling near the surface. Otherwise, an aggressive marching step will be used to encourage

skip empty voxels to reach the surface or boundary quickly. The next step, $j \rightarrow j+1$, is calculated by Equation 12.

$$STEP_{j \rightarrow j+1} = STEP_{base} \cdot e^{\beta \cdot (1 - \frac{w_{i,j}}{\tau_i})} \tag{12}$$

where $STEP_{base}$ is a predefined initial step size, and β is a hyperparameter.

3.3 Sampling Weight Constrained Training Scheme

Just using RGB images and extracted visual sensitivity maps to supervise the training process does not guarantee that the optimized scene geometry meets our sampling strategy. The weight threshold τ calculated in Equation 10 may filter out an excessive or insufficient number of sampling points, which results in artifacts in rendered results and a decrease in performance respectively. To address this problem, we introduce a sampling weight-constrained training scheme for VPRF. Our training scheme optimizes the feature vectors in G_σ, G_s and G_c end-to-end. We prioritize optimizing the geometry appearance features of the scene to provide an initial estimate of the scene. Gradually, we optimize the visual sensitivity features, which means that the number of samples required for each ray will progressively decrease as training progresses. We introduce a new importance weight constraint loss, forcing the distribution of learned density in the scene to meet the filtering criteria of threshold τ.

In addition, we generate VS maps at the same size as the input images rather than a low-resolution version. We need to perform dense sampling on the voxel grid, thereby removing the limitations on the number of samples and adopting a uniform step size instead of an adaptive one. This training scheme concentrates predictive capabilities on samples with actual contributions, thereby enhancing the representation of the radiance field based on the scene content. The entire training scheme is implemented using the following three loss functions.

Photometric Loss Given the ground truth for color, we also calculate the Mean Squared Error (MSE) loss between the rendered color $\hat{C}(\mathbf{r})$ and the ground truth color $C(\mathbf{r})$. Moreover, we propose the sensitivity mask by multiplying $C(\mathbf{r})$ with the color's MSE. This approach strengthens the constraint on the color in salient regions, enabling the model to concentrate its capabilities on actually salient areas. The photometric loss is used to optimize the density and color spherical harmonics coefficients, defined as:

$$L_{photometric} = S(r) \cdot \sum_{r \in R} \left\| \hat{C}(r) - C(r) \right\|_2^2 \tag{13}$$

Visual Perception Loss Given an input RGB image, the ground truth for the VS map is extracted using the method mentioned in Section 3.2.1, then calculate the MSE loss between the predicted sensitivity feature $\hat{S}(\mathbf{r})$ and the ground truth $S(\mathbf{r})$. The visual perception loss is defined as:

$$L_{visual} = (1-\alpha) \sum_{r \in R} \left\| \hat{S}(r) - 1 \right\|_2^2 + \alpha \sum_{r \in R} \left\| \hat{S}(r) - S(r) \right\|_2^2 \tag{14}$$

$$\text{where } \alpha = e^{-\beta \cdot (1 - \frac{current\_epoch}{target\_epoch})} \tag{15}$$

where R is a batch of ray samples and β is a hyperparameter that adjusts the balance factor α during training progresses. At the initial training phase, we force all predictions towards **1**, which means that the sampling rate P for all rays is set to **1**, without discarding any sampling points. This encourages the model to optimize an initial estimate of colors and geometry rather than visual perception, preventing the entire optimization process from collapsing due to an insufficient number of sampling points in the early stages of training. We apply the visual perception loss to supervise the training, optimizing the density and sensitivity spherical harmonics coefficients.

Importance Weight Constraint Loss For a ray with a sampling rate of P (during training $P = S$), assuming the number of sampling points is N, our goal is to select the top $N \cdot P$ points with the highest importance weights w. However, the cost of sorting is expensive. To ensure that the threshold τ calculated by Equation 10 satisfies the selection criteria above, we constrain w of sampling points. Specifically, during training, we can accurately get the maximum importance weights w_{max} along the ray, combined with P and calculate the threshold τ. The importance weight constraint loss is defined as follows:

$$L_{weight} = \sum \left\| \frac{1}{N} \cdot \sum^{N} f(w_i, \tau) - S(r) \right\| \tag{16}$$

$$f(w_i, \tau) = \frac{1}{1 + e^{-k \cdot (w_i - \tau)}} \tag{17}$$

where $f(\cdot)$ is a sigmoid function, which simulates the filtering process in a differentiable form, enabling gradient backpropagation: $f(w, \tau)$ approaches 0 when $w < \tau$, and approaches 1 when $w > \tau$. k is a hyperparameter that controls the rate of change around $w = \tau$. This loss ensures that the distribution of w among the sampling points satisfies $N : N_{w>\tau} = 1 : P$.

Finally, the total loss used for training VPRF is the combination of the above three losses:

$$L_{all} = \lambda_1 L_{photometric} + \lambda_2 L_{visual} + \lambda_3 L_{weight} \tag{18}$$

where λ is the balance weight.

4 Evaluation

In this section, we first provide a detailed description of our experimental settings, including the datasets, the specific parameters of the model optimization, and the hardware environments deployed. Subsequently, we conduct both qualitative and quantitative analyses of our experimental results.

4.1 Implementation

Datasets We evaluated the quality and performance of our method for foveated image synthesis on the Real Forward-Facing(LLFF) dataset [37] and FoV-NeRF dataset [1]. The LLFF dataset comprises 8 complex real-world physical scenes captured using handheld cameras, with a resolution of 1008×756. The FoV-NeRF dataset is a synthetic dataset that has both indoor and outdoor scenes. Each scene is accompanied by a periphery and a foveal dataset, both with resolutions of 400×400. For the forward data of real scenes, we sample in normalized device coordinates [2]. We follow the same training and test dataset splits as the original NeRF.

Implementation Details In all experiments, we set the loss weights as $\lambda_1 = 0.5$, $\lambda_2 = 0.5$, and $\lambda_3 = 0.015$ and use the RMSProp optimizer to train our model. We train our VPRF with 10 epochs, a total of 128,000 iterations each, with a batch size of 4,096 rays. We set a learning rate of SH coefficients to exponential decay, starting from 0.01 and decaying to 5×10^{-5} after 250,000 iterations. The volumetric density σ employs a delayed exponential update, decaying to 0.05 during 250,000 iterations. Our voxel grid resolution is initially set at (256, 256, 128), progressively upsampling to (512, 512, 128) and (900, 900, 256) after every 25,600 iterations. The base step size for sampling along rays, denoted as $STEP_{base}$ in Equation 12, is set at 0.5. For Visual Perception Loss (Equation 14), we adopt $\beta = 8$ and $target\_epoch = 10$. For Photometric Loss, we utilize k = 50. For the visual acuity parameters in Equation 8, refer to Guenter et al. [9], we set $\omega_0 = 1/48^\circ$ and $m = 0.0275$. In the VS map, we define regions where pixel values S exceed 0.4 as salient, which are used for calculating various metrics of salient regions in subsequent experiments. To ensure a fair comparison, monocular images are rendered for all methods. All our experiments are performed on a graphics workstation with a 3.8 GHz Intel(R) Core(TM) i7-10700KF CPU, 64 GB of memory, and an NVIDIA GeForce GTX 3090 graphics card.

4.2 Comparison

We compare our proposed VPRF method with the state-of-the-art method of foveated rendering, FoV-NeRF [1], and two NeRF accelerated methods for improving sampling efficiency, Plenoxels [3] and AdaNeRF [5]. For the Plenoxels method, we set the same grid resolution as Ours, with the sampling step size set at 0.5. For AdaNeRF and FoV-NeRF, we utilize the optimal parameters reported in their papers.

4.2.1 Quality

Figure 3 shows a comparison on two synthesized scenes, *Classroom*, and *Barbershop*, between foveated images synthesized by our method (column 1 & 3) and FoV-NeRF (column 4) compared with the ground truth (column 2). We adopt a trade-off approach (about 8 samples per ray) for experimentation. In the images, purple squares indicate the foveal regions and green squares indicate the salient regions in the periphery. Details of the magnified foveal and salient regions are presented on the right of each rendering image for comparison.

Our rendering results are more similar to the ground truth compared to FoV-NeRF, preserving better details in both the foveal

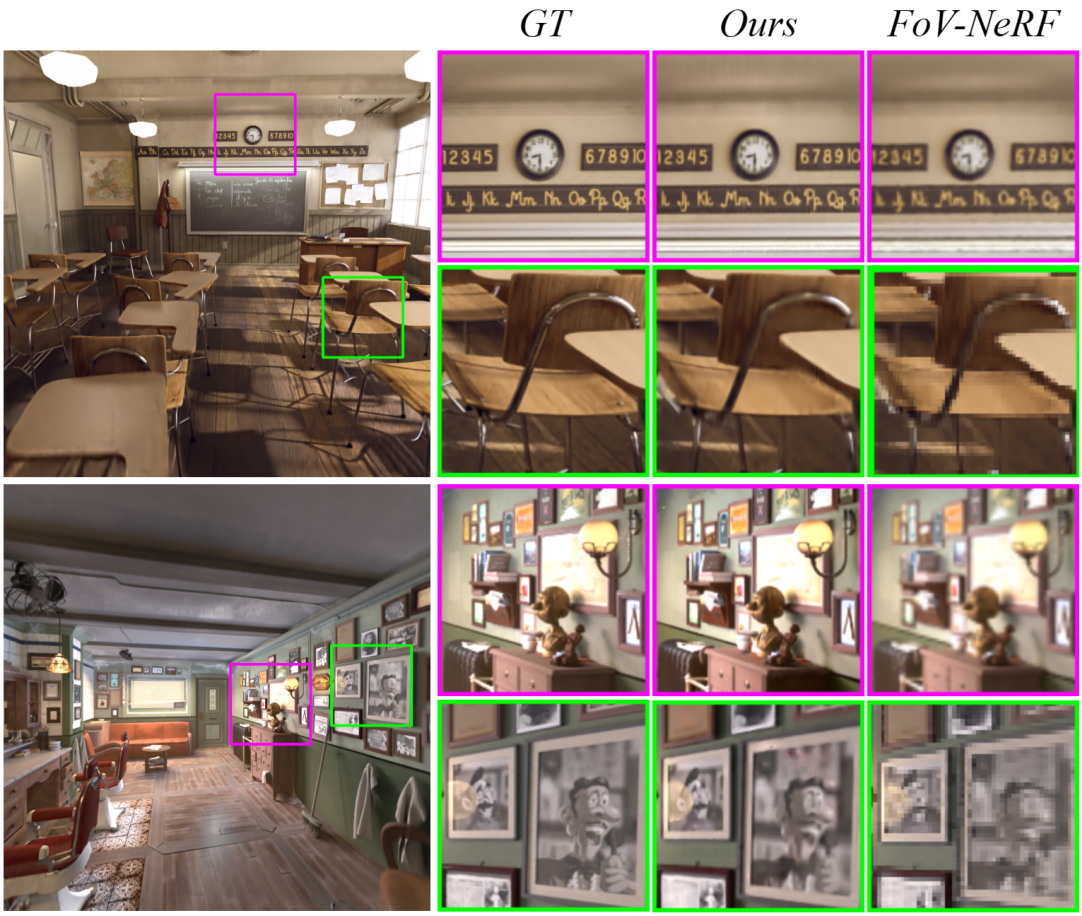


Fig. 3: Comparison of the rendered images with our VPRF method and FoV-NeRF.

region and salient region in the periphery. In *Classroom* and *Barbershop* Scene, for the foveal regions, our method enables the clear synthesis of characters on boards around the clock, the contours of the sculpture's head and the sconce, while those rendered by FoV-NeRF are blurred. This is because of our voxel-based representation method and dense sampling in the foveal regions, which further enhances the synthetic quality. Conversely, the MLP-based FoV-NeRF method can only process a fixed number of sampling points. For the salient regions in the periphery, our method renders the edges of chairs and desks more distinctly and the content within the paintings can be clearly recognized, while those rendered by FoV-NeRF are blurred and have significant serrations. This is because to accelerate rendering, FoV-NeRF maintains a rendering resolution of 400×400 in the peripheral regions, leading to information loss during the upsampling process to 800×800. Our method accelerates rendering by adjusting the sampling rates in different regions, rendering a resolution of 800×800 foveated image directly, without the need for additional upsampling.

Figure 5 shows a comparison on four real forward-facing scenes, *Horns*, *Flowers*, *Trex* and *Fern*, the foveated images synthesized by our method (Ours about 8 samples, column 2), Plenoxels (column 2), AdaNeRF (about 8 samples, column 3), and the real photos (ground truth, GT, column 1). Details in the foveal and salient regions are similarly magnified for comparison. Our results show a higher similarity to the ground truth, with clearer details both in the foveal regions and the salient regions in the periphery, while the comparison methods exhibit varying degrees of blurriness and artifacts in the rectangular regions. This is attributed to our method predicting higher and accurate sensitivity values at the salient regions, as shown in Figure 4. Therefore, we allocate more computational resources to these regions, enhancing the synthetic quality. Conversely, other methods uniformly distribute computational resources across the overall image. This also demonstrates the effectiveness of our visual perceptual sampling strategy, achieving foveated rendering for the radiance field.

We use peak signal-to-noise ratio (PSNR) and structure similarity index measure (SSIM) to quantitatively evaluate the synthesis quality. To validate the effectiveness of foveated rendering, we

(a) (b)

Fig. 4: Visualizing the visual sensitivity map extracted from the original RGB image (a) and generated by our method (b). Our method predicts higher sensitivity for the cracks on the horns and the skull model on the right side, which aligns with the actual results.

partition the image space into four regions for quality evaluation: foveal, salient, overall, and peripheral region, and compute PSNR and SSIM for each region. Table 1 presents a comparative analysis of PSNR and SSIM in different regions on the FoV-NeRF dataset between our method and FoV-NeRF. To achieve varying levels of synthesis quality, our method adjusts the average sample count $N = [4, 8, 12]$, whereas FoV-NeRF increases network complexity. The results of our VPRF method have higher PSNR and SSIM across all evaluated regions, which indicates that our method achieves better synthesis quality with similar time performance. Specifically, compared with FoV-NeRF, (PSNR, SSIM) of our method is $(1.37\times, 1.22\times)$ in the overall region, $(1.34\times, 1.17\times)$ in the foveal region, $(1.40\times, 1.26\times)$ in the peripheral region, and $(1.69\times, 1.40\times)$ in the salient region. Moreover, when the PSNR exceeds 30dB, the HVS can hardly perceive the difference between the synthesized images and the ground truth. Our method surpasses this threshold in both foveal and salient regions, indicating no perceptible between our synthesized images and the ground truth images in these regions.

Tables 2 shows the comparison of PSNR and SSIM across different regions of our VPRF method and other NeRF acceleration methods on LLFF datasets. Both our VPRF method and AdaNeRF

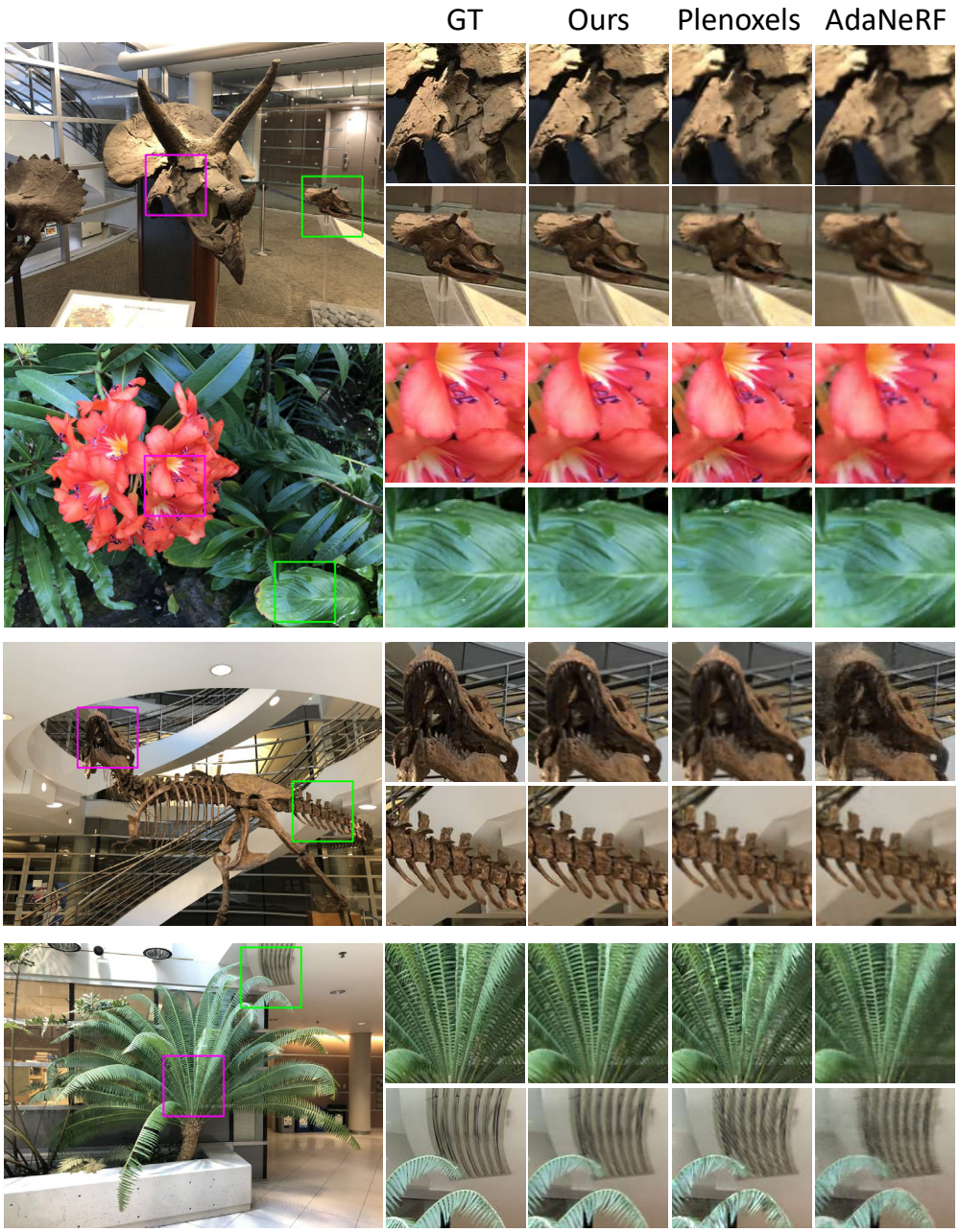


Fig. 5: Comparison of the rendered images with our VPRF method and the previous NeRF accelerated methods on LLFF dataset [37]

ing the trade-off between quality and performance, we set various average numbers of samples, $N = [4, 8, 12]$ as well for comparison. For the foveal region, all PSNR and SSIM computed by our method are higher than all comparison methods. For the salient regions in the periphery, except for the SSIM of ours-4, which falls behind Plenoxels, all other metrics are higher than the comparison methods. Compared to AdaNeRF, across various parameter settings, our method achieves superior quality in foveal and salient regions while achieving notable speed improvement. Particularly, at the highest quality setting (about 12 samples), PSNR and SSIM of our method are 1.15× higher and 1.13× higher in the foveal region, and 1.13× higher and 1.10× higher in the salient region than those of AdaNeRF. Compared to Plenoxels, at the fastest setting (about 4 samples), our method achieves comparable quality in foveal and salient regions. At the highest quality setting, PSNR and SSIM of our method are 1.13× higher and 1.08× higher in the foveal region, and 1.09× higher and 1.02× higher in the salient region than those of Plenoxels. For the overall and peripheral regions, our method achieves higher PSNR and SSIM compared to AdaNeRF across various parameter settings. And with the rendering performance is about 6.4×, PSNR and SSIM of our method are only slightly lower than those of Plenoxels. This is because we allocate more computational resources to the foveal and salient regions, sacrificing slightly the synthesis quality in the non-salient peripheral regions to significantly improve overall rendering performance. Totally, considering run-time efficiency, our method shows an optimal trade-off, allowing us to choose between fast rendering (about 4 samples) and high quality (about 12 samples).

4.2.2 Performance

Rendering Speed The last column ("Performance") of Table 1 and Table 2 show the performance comparison between our VPRF and other methods with different synthesis quality. Compared to FoV-NeRF, experimental results indicate that across all regions with equal PSNR, our method achieves superior time consumption. At equal or superior quality, our method enhances performance by about 2.6×. We utilize a voxel-based approach combined with a sampling strategy to accelerate rendering. Moreover, the proposed VPRF representation leverages scene content to enhance the prediction capabilities for salient regions. Ultimately, our method significantly outperforms FoV-NeRF in both synthesis quality and performance.

Table 1: Quantitative comparison between our VPRF method and FoV-NeRF on FoV-NeRF Dataset [1]

	Overall		Foveal		Peri		Salient		Performance	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	FPS	Speedup
Ours-4	22.34	0.73	25.28	0.82	22.29	0.73	24.88	0.81	**76**	
Ours-8	26.52	0.85	33.86	0.95	26.41	0.85	31.03	0.92	31	-
Ours-12	**27.03**	**0.86**	**35.59**	**0.96**	**26.91**	**0.86**	**33.27**	**0.93**	23	
FoV-NeRF-S	15.34	0.54	21.79	0.72	14.93	0.52	15.17	0.54	63	
FoV-NeRF-M	19.25	0.69	25.10	0.81	18.86	0.68	18.33	0.65	29	$\sim 2.6\times$
FoV-NeRF-L	20.48	0.71	25.87	0.83	20.15	0.71	19.86	0.70	17	

Table 2: Quantitative comparison between our VPRF method and other NeRF acceleration methods on LLFF Dataset [37]

	Overall		Foveal		Peri		Salient		Performance	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	FPS	Speedup
Ours-4	24.48	0.76	25.93	0.87	24.42	0.76	27.60	0.87	**83**	
Ours-8	25.27	0.79	28.38	0.90	25.22	0.79	29.06	0.89	56	-
Ours-12	26.08	0.82	**29.20**	**0.90**	25.97	0.82	**29.21**	**0.90**	45	
AdaNeRF-4	24.37	0.76	24.01	0.75	24.39	0.76	24.53	0.77	16	
AdaNeRF-8	25.02	0.78	24.82	0.77	25.15	0.79	25.31	0.78	9	$5.1 \sim 7.2\times$
AdaNeRF-12	25.91	0.81	25.35	0.80	25.90	0.82	25.89	0.82	6	
Plenoxels	**26.40**	**0.84**	25.74	0.83	**26.65**	**0.85**	26.87	0.88	7	$6.4 \sim 13.1\times$

Compared to other NeRF acceleration methods, our method achieves the highest rendering speed about 83 FPS, without sacrificing perceptual quality. Specifically, compared to Plenoxels, our method achieves 11.2-13.1× acceleration (about 4 samples) with similar quality in foveal and salient regions, and 6.4-8.6× acceleration (about 12 samples) with similar quality in peripheral and overall regions, but with superior quality in foveal and salient regions. This significant acceleration is attributed to our visual perceptual based sampling strategy, which efficiently filters for critical sampling points and allocates varying numbers of sampling points across different regions, leading to a massive speedup. Compared to AdaNeRF, our method achieves higher performance at approximately the same sample count. Specifically, at equivalent rendering performance levels, our method surpasses AdaNeRF in rendering quality across all regions. At equal or superior quality, our method enhances performance by 5.1-7.2×. This is attributed to our method utilizing the voxel-based scene representation, which avoids the substantial computational overhead associated with densely inferring implicit MLPs during runtime.

Table 3 shows the time cost for the individual step of synthesizing novel view foveated images using our method. The result illustrates that our upsampled VS map generation approach exhibits only minor overheads, with the computational bottleneck of the whole pipeline remaining at the stage of RGB image generation. This also indicates that our method is effective, which involves generating the VSR map with additional time to accelerate the synthesis process of RGB images.

Table 3: Time Proportion for the main stages of our VPRF method

Step	Time Proportion
Visual sampling rate map generation	17.4%
Visual perceptual sampling	9.5%
Final image synthesis	72.1%

Memory Consumption The entire model of our VPRF representation requires approximately 600MB of storage, with visual sensitivity information occupying only 28MB. This reduction is attributed to our method employs a voxel pruning strategy similar to Plenoxels during the training process. We applied a threshold to prune empty voxels that do not contain object and introduced a sparsity prior, encouraging the model to select empty voxels, further saving memory without reducing image quality. Compared to other methods based on explicit structures, we only store additional single-channel visual sensitivity information, which imposes little pressure on memory.

4.3 Ablation Studies

In this section, we conduct ablation studies to validate the efficiency of our importance weight constraint loss on synthesis quality and the visual perceptual sampling strategy on performance.

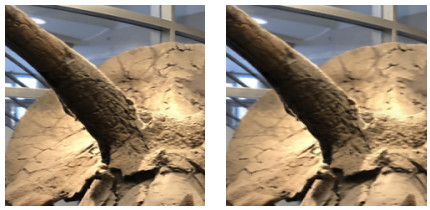
Quality related Ablation The effect of the weight constraint loss depends on the adoption of our visual perceptual sampling strategy. Figure 6 visualizes the results of a qualitative comparison of synthesis quality across four scenarios. The comparison results between 6(a) and 6(b) indicate that directly adding weight constraint loss does not detrimentally affect the synthesis quality. This is because the weight constraint loss encourages the model to optimize density by choosing solutions that satisfy the constraints rather than random distributions. The comparison results between 6(a) and 6(c) show that solely employing the visual perceptual sampling strategy leads to synthesis artifacts such as blurring and occlusion error. This is because the absence of constraints on the distribution of importance weights, falsely filters out sampling points on the surfaces of foreground objects, which leads to occluded objects being rendered. 6(d) shows the high-quality synthesis result of our full method, demonstrating that the importance of weight constraint loss ensures the effectiveness of our sampling strategy.

Performance related Ablation The main components that influence the performance of our method are: sample count limitation, adaptive ray marching step and adaptive importance weights threshold for filtering samples Table 4 shows the ablation of our sampling strategy, in comparison with *None*. The results indicate that each component contributes to varying degrees of improvement in performance, with our full method achieving a high performance of 83FPS. Without the sample count limitation, computational costs will significantly increase. This is because beyond a certain count, additional samples hardly improve the image quality. Without the adaptive ray marching step, it will prolong the duration rays spend marching through empty voxels, reducing the speed to reach the surface. Without the adaptive importance weights threshold, it will result in excessive sampling in regions with low visual sensitivity, reducing rendering performance without substantially enhancing the perceptual quality of the final image.

Hyperparameter Ablation For Equation 18, L_{weight} serves only as a constraint term, similar to other regularization terms. We experimented with a range of values $[0.005, 0.04]$ for the λ of L_{weight} and determined the best-performing values through testing, the results are presented in Table 5.

Table 4: Performance Ablation study

Method	Performance [FPS]
Our full method	83
w/o sample count limitation	23
w/o adaptive ray marching step	58
w/o adaptive importance weights threshold	41
None	7



(a) *None* (b) weight constraint *only*

(c) sampling strategy *only* (d) *Full*

Fig. 6: Qualitative comparison for quality related ablation. Our full method eliminates artifacts and occlusion errors while filtering out sampling points with low contributions to the final color for rendering acceleration.

Table 5: On the *Horns* scene of the LLFF dataset, the foveated rendering results of our method training with various lambda values

	Hyperparameter tuning							
λ of L_{weight}	0.005	0.01	0.015	0.02	0.025	0.03	0.035	0.04
PSNR ↑	25.98	26.54	**27.30**	26.73	26.56	26.12	25.85	25.33

5 USER STUDY

We design a within-subject study to evaluate the perceptual synthesis quality on the FoV-NeRF dataset [1] of our method compared with the previous method.

Participants and Setup We recruited 14 participants (7 males and 7 females, aged between 21-30 years), all of whom had experience using VR HMDs and had normal vision. Each participant wore an HTC Cosmos headset for the experiment.

Conditions The conditions included: the full-quality ground truth rendering results (*GT*), our method (*Ours*), FoV-NeRF, and Plenoxels. FoV-NeRF set the optimal parameters reported in their paper, and the common parameters between Plenoxels and Ours are kept consistent.

Task To ensure precise and fair comparisons, we fixed the view motion of the observation camera. Referring to the setup in FoV-NeRF, to avoid perceptual differences caused by varying gaze movements between individual trials, we enforced static gaze points in each method instead of free viewing. We used four test scenes from the FoV-NeRF dataset: *barbershop*, *classroom*, *lobby* and *stones*. For each scene, we generated an 8s animation sequence for each condition. Initially, we presented the *GT* animation sequence to the participants, informing them that this was the benchmark result. Subsequently, we displayed the animation results generated by *Ours*, *GT*, FoV-NeRF and Plenoxels from the same camera perspective in a random order, asking participants to rate the visual perceptual quality of each animation sequence. The viewing counts for all methods in the experiment were kept balanced. The visual perceptual quality rating included 5 confidence levels, with 5 indicating the highest quality (no perceptible artifacts) and 1 the lowest quality (noticeable artifacts perceptible). To mitigate the effects of visual fatigue, after completing the ratings, participants are given a 10 second rest before proceeding to the next animation.

Results Figure 7 shows the average score across all scenes for different conditions, utilizing the p-value and $Cohen'sd$ to estimate the differences between the comparison conditions and *Ours*. The results indicate a significant improvement in our average score compared to both FoV-NeRF and Plenoxels, which is closest to the GT. The p-value $= 0.19$ of scores between *Ours* and *GT*, with $Cohen'sd = 0.43$, indicating a *small* effect size. This suggests that our method has statistically perceptual similarity with the ground truth. The p-value < 0.001 of scores between *Ours* and FoV-NeRF, with $Cohen'sd = 2.33$, indicating a *huge* effect size. The p-value < 0.001 of scores between *Ours* and Plenoxels, with $Cohen'sd = 1.99$, indicating a *very large* effect size. These results demonstrate that compared to other methods, our method significantly enhances the visual perceptual quality of synthesized foveated images. This is because our method allocates more computational resources to both the foveal regions and the salient regions in the periphery.

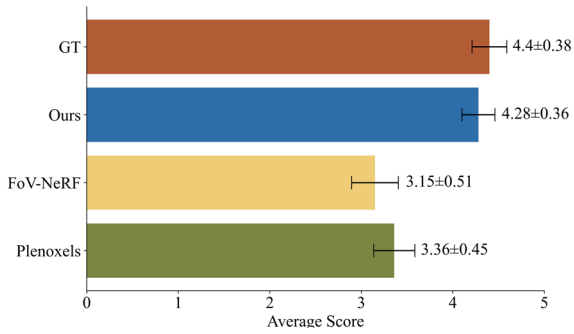


Fig. 7: The user's average scores and standard deviations for all conditions in our evaluation experiment.

6 CONCLUSION, LIMITATIONS AND FUTURE WORK

We have proposed a new visual perceptual radiance fields representation method, named VPRF, which integrates the HVS visual acuity and contrast sensitivity models into the radiance field rendering framework and can efficiently synthesize high-quality novel view foveated images at about 83FPS. We encode not only the scene appearance but also the visual sensitivity of the HVS to scene content. These encoded features are stored in our feature grid. For runtime rendering, we initially synthesize a visual sampling rate map for current view based on the sensitivity information, which is used to allocate rendering resources for different regions during appearance synthesis. We propose a visual perceptual sampling strategy that guides the sampling process based on the sampling rate associated with each ray to render the final foveated image efficiently. We also ensure the effectiveness of our sampling strategy by adding importance weight constraints to restrict the geometric distribution of the scene. We validate our method on both real and synthetic datasets, and the experimental results show that our method achieves superior synthesis quality in foveal and salient regions with significant acceleration. User study also demonstrates that our method significantly improves visual perceptual quality.

While our method achieves superior rendering performance and perceptual quality, there remain certain limitations. Firstly, in comparison to MLP-based methods, the explicit voxel grid consumes large memory during runtime, which indicates that the representation capacity is constrained by the grid resolution. A recent study [38] proposes factorizing the voxel into multiple compact low-rank tensor components. K-Planes [39] projects spatial points onto planes, utilizing d planes to represent a d-dimensional scene. These methods could potentially reduce the memory storage gap between pure MLP-based methods while retaining the advantages of explicit structures. Therefore, potential future work is needed to extend our voxel-based representation to these storage-efficient approaches. Secondly, our method does not incorporate the temporal information of the scene. Thus, it cannot synthesize foveated images for dynamic environments. In the future, we plan to model the radiance field based on temporal information and the perceptual characteristics of the HVS towards moving objects, enabling the synthesis of novel view foveated images in dynamic scenes. Furthermore, if the current frame rate is accessible, our method can dynamically adjust the sampling rate to meet a certain frame rate by incorporating simple strategies. Specifically, when the current frame rate does not meet the target, our method can reduce the number of sampling points less in the foveal and salient areas, and more in other areas. This allows us to maintain high visual perception quality in critical areas while improving the rendering frame rate. When the current frame rate exceeds the target, our method can increase the number of sampling points in the foveal and salient areas to enhance the rendering quality in these areas.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China through Projects 61932003 and 62372026, Beijing Science and Technology Plan Project Z221100007722004, and the National Key R&D plan 2019YFC1521102.

REFERENCES

[1] Nianchen Deng, Zhenyi He, Jiannan Ye, Budmonde Duinkharjav, Praneeth Chakravarthula, Xubo Yang, and Qi Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3854–3864, 2022. 1, 2, 3, 5, 8, 9

[2] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 5

[3] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 1, 2, 3, 4, 5

[4] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H Mueller, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. In *Computer Graphics Forum*, volume 40, pages 45–59. Wiley Online Library, 2021. 1, 3

[5] Andreas Kurz, Thomas Neff, Zhaoyang Lv, Michael Zollhöfer, and Markus Steinberger. Adanerf: Adaptive sampling for real-time rendering of neural radiance fields. In *European Conference on Computer Vision*, pages 254–270. Springer, 2022. 1, 2, 3, 4, 5

[6] Christian Vater, Benjamin Wolfe, and Ruth Rosenholtz. Peripheral vision in real-world tasks: A systematic review. *Psychonomic bulletin & review*, 29(5):1531–1557, 2022. 1

[7] Martin Weier, Michael Stengel, Thorsten Roth, Piotr Didyk, Elmar Eisemann, Martin Eisemann, Steve Grogorick, André Hinkenjann, Ernst Kruijff, Marcus Magnor, et al. Perception-driven accelerated rendering. In *Computer Graphics Forum*, volume 36, pages 611–643. Wiley Online Library, 2017. 2

[8] Bipul Mohanto, ABM Tariqul Islam, Enrico Gobbetti, and Oliver Staadt. An integrative view of foveated rendering. *Computers & Graphics*, 102:474–501, 2022. 2

[9] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. Foveated 3d graphics. *ACM transactions on Graphics (tOG)*, 31(6):1–10, 2012. 2, 5

[10] Wilson S Geisler and Jeffrey S Perry. Real-time foveated multiresolution system for low-bandwidth video communication. In *Human vision and electronic imaging III*, volume 3299, pages 294–305. SPIE, 1998. 2

[11] Xiaoxu Meng, Ruofei Du, Matthias Zwicker, and Amitabh Varshney. Kernel foveated rendering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–20, 2018. 2

[12] Sebastian Friston, Tobias Ritschel, and Anthony Steed. Perceptual rasterization for head-mounted display image synthesis. *ACM Trans. Graph.*, 38(4):97–1, 2019. 2

[13] Okan Tarhan Tursun, Elena Arabadzhiyska-Koleva, Marek Wernikowski, Radosław Mantiuk, Hans-Peter Seidel, Karol Myszkowski, and Piotr Didyk. Luminance-contrast-aware foveated rendering. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2

[14] Kil Joong Kim, Rafal Mantiuk, and Kyoung Ho Lee. Measurements of achromatic and chromatic contrast sensitivity functions for an extended range of adaptation luminance. In *Human vision and electronic imaging XVIII*, volume 8651, pages 319–332. SPIE, 2013. 2

[15] Akshay Jindal, Krzysztof Wolski, Karol Myszkowski, and Rafał K Mantiuk. Perceptual model for adaptive local shading and refresh rate. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 2

[16] Hunter A Murphy, Andrew T Duchowski, and Richard A Tyrrell. Hybrid image/model-based gaze-contingent rendering. *ACM Transactions on Applied Perception (TAP)*, 5(4):1–21, 2009. 2

[17] Erik N Molenaar. Towards real-time ray tracing through foveated rendering. Master's thesis, 2018. 2

[18] Matias Koskela, Atro Lotvonen, Markku Mäkitalo, Petrus Kivi, Timo Viitanen, and Pekka Jääskeläinen. Foveated real-time path tracing in visual-polar space. In *Proceedings of 30th Eurographics Symposium on Rendering*. The Eurographics Association, 2019. 2

[19] Matias Koskela. Foveated path tracing with fast reconstruction and efficient sample distribution. 2020. 2

[20] Michael Stengel, Steve Grogorick, Martin Eisemann, and Marcus Magnor. Adaptive image-space sampling for gaze-contingent real-time rendering. In *Computer Graphics Forum*, volume 35, pages 129–139. Wiley Online Library, 2016. 2, 4

[21] Xuehuai Shi, Lili Wang, Jian Wu, Wei Ke, and Chan-Tong Lam. Locomotion-aware foveated rendering. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 471–481. IEEE, 2023. 2

[22] Linus Franke, Laura Fink, Jana Martschinke, Kai Selgrad, and Marc Stamminger. Time-warped foveated rendering for virtual reality headsets. In *Computer Graphics Forum*, volume 40, pages 110–123. Wiley Online Library, 2021. 2

[23] David Bauer, Qi Wu, and Kwan-Liu Ma. Fovolnet: Fast volume rendering using foveated deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):515–525, 2022. 2

[24] Anton S Kaplanyan, Anton Sochenov, Thomas Leimkühler, Mikhail Okunev, Todd Goodall, and Gizem 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. 2

[25] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018. 2

[26] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2

[27] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 2

[28] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021. 2

[29] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. Efficientnerf efficient neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12902–12911, 2022. 2, 3, 4

[30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2, 3

[31] Martin Piala and Ronald Clark. Terminerf: Ray termination prediction for efficient neural rendering. In *2021 International Conference on 3D Vision (3DV)*, pages 1106–1114. IEEE, 2021. 3

[32] Frank W Weymouth. Visual sensory units and the minimal angle of resolution. *American journal of ophthalmology*, 1958. 3

[33] Lili Wang, Xuehuai Shi, and Yi Liu. Foveated rendering: A state-of-the-art survey. *Computational Visual Media*, 9(2):195–228, 2023. 3

[34] Andrew B Watson. A formula for human retinal ganglion cell receptive field density as a function of visual field location. *Journal of vision*, 14(7):15–15, 2014. 3

[35] Yuanhao Yue, Qin Zou, Hongkai Yu, Qian Wang, Zhongyuan Wang, and Song Wang. An end-to-end network for co-saliency detection in one single image. *Science China Information Sciences*, 66(11):210101, 2023. 4

[36] Wujie Zhou, Chang Liu, Jingsheng Lei, and Lu Yu. Rllnet: A lightweight remaking learning network for saliency redetection on rgb-d images. *Science China Information Sciences*, 65(6):160107, 2022. 4

[37] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 5, 7, 8

[38] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022. 9

[39] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 9