This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3429416

Scene-aware Foveated Neural Radiance Fields

Xuehuai Shi, Lili Wang, Xinda Liu, Jian Wu, and Zhiwen Shao

GT *Ours* *FoV-NeRF*

Fig. 1: The ground truth image (*GT*, left) from the testing set of *classroom*, foveated images synthesized by our method (*Ours*, middle) and by the foveated neural radiance fields method (*FoV-NeRF*, right). Compared with *FoV-NeRF*, our method achieves 1.34× higher PSNR in the foveal region (*Ours* vs. *FoV-NeRF*, 32.46 vs. 24.29), and 1.21× higher PSNR in the overall screen space (*Ours* vs. *FoV-NeRF*, 22.47 vs. 18.50). Our method is 1.41-1.46× faster than *FoV-NeRF*.

Abstract—Foveated rendering provides an idea for improving the image synthesis performance of neural radiance fields (NeRF) methods. In this paper, we propose a scene-aware foveated neural radiance fields method to synthesize high-quality foveated images in complex VR scenes at high frame rates. Firstly, we construct a multi-ellipsoidal neural representation to enhance the neural radiance field's representation capability in salient regions of complex VR scenes based on the scene content. Then, we introduce a uniform sampling based foveated neural radiance field framework to improve the foveated image synthesis performance with one-pass color inference, and improve the synthesis quality by leveraging the foveated scene-aware objective function. Our method synthesizes high-quality binocular foveated images at the average frame rate of 66 frames per second (*FPS*) in complex scenes with high occlusion, intricate textures, and sophisticated geometries. Compared with the state-of-the-art foveated NeRF method, our method achieves significantly higher synthesis quality in both the foveal and peripheral regions with 1.41-1.46× speedup. We also conduct a user study to prove that the perceived quality of our method has a high visual similarity with the ground truth.

Index Terms—Virtual Reality, Foveated Rendering, Neural Radiance Fields

1 Introduction

Constructing complex scenes and presenting high-quality rendering results at high frame rates for users in virtual reality (VR) can enhance user experience. Conventional VR rendering methods heavily rely on 3D resources, which require significant design and production costs. It is difficult to construct new complex VR scene content and present them in real time using conventional VR rendering methods. Neural radiance fields methods utilize conventional rendering techniques and learning-based 3D scene representation techniques to achieve rapid scene content construction and rendering results synthesis from new views, such as neural radiance fields (NeRF) [35], D-NeRF [40], and Block NeRF [52]. These methods use a set of images from different views captured in a given scene as the training set to train a neural network, and then use the trained neural network to synthesize rendering results for new views.

To improve the synthesis performance of NeRF methods, Deng et al. [8] propose a foveated neural radiance fields method (FoV-NeRF), which integrates the neural radiance field into the framework of foveated rendering. But the radiance field representation of the FoV-NeRF loses the radiance details of salient regions in complex scenes due to the VR frame rate requirements, which poses a challenge to the foveated image synthesis quality. Moreover, the FoV-NeRF needs to train multiple networks to synthesize foveated images, which poses a challenge to the performance of foveated rendering.

In this paper, we propose the scene-aware foveated neural radiance fields method (SaF-NeRF) to address the above two challenges. To address the first challenge, we introduce the multi-ellipsoidal neural representation (MeNR), which improves the radiance field representation capability of complex VR scenes with high occlusion, intricate textures, and sophisticated geometries based on the scene content. To address the second challenge, we propose the uniform sampling based foveated neural radiance field framework (US-FNRF), which improves the foveated image synthesis performance by one-pass color inference, and improves the foveated image synthesis quality by the proposed foveated scene-aware objective function in complex VR scenes.

We compare the monocular images of the ground truth (*GT*), foveated images synthesized by our method (*Ours*) and by the foveated neural radiance fields method (*FoV-NeRF*) in the test scene *classroom*. The result shows that our method achieves 1.34× higher peak signal-to-noise ratio (PSNR) in the foveal region, and 1.21× higher PSNR in the overall screen space compared with *FoV-NeRF*. Our method achieves frame rates of 66 frames per second (*FPS*) in binocular rendering for head-mounted displays (HMDs) in complex VR scenes. Fig. 1 shows the comparison in *classroom*. The region in the yellow circle is the

- *Xuehuai Shi is with School of Computer Science, Nanjing University of Posts and Telecommunications, Jiangsu, China, 210003; and with State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. E-mail: xuehuai@njupt.edu.cn*
- *Lili Wang and Jian Wu are with State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, 100191 and Peng Cheng Laboratory, Shenzhen, Guangdong, China, 518000.*
- *Xinda Liu is with School of Information Science and Technology, Northwest University, Shanxi, China, 710127. E-mail: liuxinda@nwu.edu.cn*
- *Zhiwen Shao is with China University of Mining and Technology, Jiangsu, China, 221116. E-mail: zhiwen_shao@cumt.edu.cn*
- *Lili Wang is the corresponding author. E-mail: wanglily@buaa.edu.cn*

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

foveal region. Details in the rectangular regions are magnified to the right of each rendered image, the magnification of the green rectangular region in the foveal region is placed in the upper right, and the magnification of the red rectangular region in the peripheral region is placed in the lower right. Our method achieves better synthesis quality in both the foveal and peripheral regions. In the green rectangular region, the letters on the book cover have noticeable noise and cannot be recognized by users in *FoV-NeRF*. In the red rectangular region near fovea, *FoV-NeRF* cannot preserve the structure details of the book spines on the desk that are close to users.

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

- A scene-aware foveated neural radiance fields method, which synthesizes high-quality foveated images of complex scenes at high frame rates in VR;
- A multi-ellipsoidal neural representation, which enhances the neural radiance field representation capability by adjusting the ellipsoid layer density of different regions adaptively according to the scene content in complex VR scenes;
- A uniform sampling based foveated neural radiance field framework, which improves synthesis performance and quality by synthesizing foveated images using a single easily trainable network optimized by the foveated scene-aware objective function.

2 Related Work

In this section, we first introduce the prior work of foveated rendering in the past five years, then discuss the existing real-time neural radiance fields methods.

2.1 Foveated Rendering

Guenter et al. [15] firstly propose the foveated rendering framework to accelerate graphics rendering performance without sacrificing visual-perception rendering quality. It interpolates three eccentric layers with different resolutions to the final display resolution based on a visual acuity fall-off model. According to processed data types, the research of foveated rendering can be divided into 3D geometry and 2D image/video based foveated rendering.

2.1.1 3D Geometry based Foveated Rendering

In the research of foveated rendering for 3D geometry with mesh data, Meng et al. [34] propose a GPU-friendly two-pass foveated rendering pipeline to accelerate foveated rendering. It compresses the foveated pixel shading process in screen space into low-resolution log-polar space to reduce pixel shading quantity, and uses the inverse-log-polar transformation to transform the shading results back into screen space to output foveated images. Ye et al. [63] further improve the two-pass foveated rendering pipeline by changing the mapping scheme of the log-polar transformation to the rectangular mapping transformation, which preserves the structural details in the peripheral region with a similar shading quantity. Friston et al. [12] present a foveated rasterization pipeline to achieve foveated rendering and reduce aliasing artifacts in a single perceptual rasterization pass with per-fragment ray-casting. Tursun et al. [56] propose a luminance-contrast aware foveated ray tracer based on luminance contrast sensitivity function (CSF) to improve foveated rendering quality. Meng et al. [33] further accelerate foveated rendering without sacrificing the visual-perception rendering quality by leveraging the ocular dominance feature of HVS. Franke et al. [10] improve the performance of foveated rendering by reusing the pixels in the peripheral region according to the spatiotemporal reprojection technique. Jindal et al. [21] propose a variable-rate shading pipeline to control the pixel shading accuracy and the refresh rate of different regions based on CSF, thereby improving the performance of foveated rasterization. Shi et al. [47] integrate photon mapping into the foveated rendering framework to accelerate global illumination rendering. Then, they integrate stochastic lightcuts into the foveated rendering framework based on spatiotemporal-luminance CSF to accelerate many-lights rendering [48]. Liu et al. [31] propose a stochastic sampling scheme based on foveated depth of field to achieve longitudinal chromatic aberration and anti-aliasing at the same time. Kim et al. [24] combine the selective supersampling technique [20] with the foveated rendering scheme to accelerate real-time ray tracing for HMDs. Shi et al. [49] parametrically adjust the shading quantity in the peripheral region based on different locomotion modes to further accelerate foveated rendering.

Besides mesh data, many researchers focus on 3D geometry foveated rendering of volume data. Bruder et al. [6] accelerate foveated volume rendering based on Linde-Buzo-Gray sampling strategy [14] and neighbor interpolation based on the visual acuity fall-off model. Anan-piriyakul et al. [1] utilize face tracking to drive the smooth decrease in volume rendering resolution from fovea to periphery. Bauer et al. [4] construct a deep neural reconstruction network for reconstructing foveated sparse volume rendering results to obtain full-resolution volume rendering results. For accelerating point clouds rendering, Schutz et al. [45] propose a foveated point clouds rendering method to achieve the continuous level of detail effects from fovea to periphery.

Existing 3D geometry based foveated rendering methods rely on 3D resources to construct scene content, which makes it costly to present new complex scene content for users in VR. The proposed method in this paper achieves new complex scene exploration in real time based on captured scene images.

2.1.2 2D Image/Video based Foveated Rendering

In the research of foveated rendering for 2D image/video data, researchers mainly work on accelerating the streaming transmission by visual-perception models and enhancing the visual-perception quality of the given image/video data. In the research of foveated streaming transmission acceleration, Lungaro et al. [32] reduce the image quality of the peripheral region in 360° panorama video transmission to minimize the bandwidth requirements for panorama video streaming. Florian et al. [11] propose a collaborative foveated encoding method to reduce the overall video-streaming bandwidth required for large high-resolution displays with display walls located at different locations. Li et al. [29] introduce a log-linear encoding-decoding method that encodes full-resolution 360° video frames based on the visual acuity fall-off model on the server side and decodes frames on the client side to improve the image quality of foveated streaming video frames. Yang et al. [62] adaptively adjust the size of the foveal region in collaborative foveated rendering to maintain the VR performance requirements in mobile devices. In the research of the visual-perception quality enhancement, Kaplanyan et al. [22] use a generative adversarial neural network to reconstruct foveated rendering videos in the peripheral region to improve the peripheral rendering quality of foveated videos. Walton et al. [57] propose a real-time post-process method to filter the peripheral region to improve the visual-perception quality of foveated images. Tariq et al. [53] add procedural noise in the peripheral region with a specific range of frequencies based on image content and human perception to achieve more aggressive foveation without losing visual-perception quality. Deng et al. [8] propose the FoV-NeRF that uses multiple multilayer perceptrons (MLPs) to synthesize images in the foveal and peripheral regions based on neural radiance fields rendering [35], and blends these images together to generate foveated images in real time. Krajancich et al. [26] introduce the attention-aware contrast sensitivity model to accelerate foveated rendering when users allocate attention to the foveal region. Singh et al. [50] design a gaze-tracked foveated renderer to improve the runtime performance and energy efficiency when running on a mobile GPU.

The FoV-NeRF can synthesize foveated images for new scenes based on captured scene images. However, this method cannot perform foveated non-uniform radiance inference based on a single neural representation, and requires training multiple networks to synthesize images in the foveal, transitional and peripheral regions separately. The synthesized results generated by multiple networks will lead to image breakage between the foveal and peripheral regions, which affects the quality and performance of the foveated image synthesis, and increases the training cost of the neural radiance fields method. We propose the US-FNRF that uses a single end-to-end network to synthesize high-quality foveated images with one-pass color inference for complex VR scenes.

2.2 Real-time Neural Radiance Fields

Mildenhall et al. [40] first present NeRF to synthesize novel views of scenes by optimizing an underlying continuous volumetric scene function using a sparse set of input views. It recovers fine details in both geometry and appearance in new views and achieves better synthesis quality than prior 3D convolutional networks. Researchers accelerate NeRF to achieve real-time frame rates. Based on the scene representation modes, we categorize the research of real-time NeRF into explicit scene modeling-based NeRF and implicit scene representation-based NeRF. Real-time explicit scene modeling-based NeRF methods construct explicit structures such as volume, point clouds, octrees, etc., to store scene radiance and other features, and accelerate the rendering process by sampling these structures with ray marching. Real-time implicit scene representation-based NeRF methods describe scenes with functions, which can be understood as representing scenes in the function parameters. It achieves real-time frame rates by compressing the number of function parameters and reducing the number of samples.

2.2.1 Real-time Explicit Scene Modeling-based NeRF

In the research of real-time explicit scene modeling-based NeRF, Hedman et al. [18] construct a sparse 3D voxel grid data structure to store the learned opacity, diffuse color, and view-dependent effects feature vector in the training process, and perform ray marching through the sparse 3D voxel grid to accelerate radiance inference in the testing process. To improve the synthesis performance without sacrificing quality, Yu et al. [65] build a modified NeRF model that predicts spherical harmonic coefficients instead of color using the same optimization and volume rendering methods presented in NeRF [40], and samples the modified NeRF model to construct a sparse voxel-based octree named PlenOctree that is used for radiance inference in the test process. Wang et al. [59] extend the PlenOctree to support dynamic scenes by training the dynamic sequence's Fourier coefficients directly on the leaves of a union PlenOctree structure. To reduce the time cost of both the training and testing processes, Hu et al. [19] propose a valid and pivotal sampling strategy to accelerate the training process by decreasing the number of sampling points to eliminate unimportant sampling points, and construct a two-layer tree-based data structure to store the color and density for accelerating color inference in the testing process. Zhang et al. [66] construct a point clouds structure to store the local spherical harmonic coefficients, and propose an end-to-end differentiable rendering pipeline from point primitives and spherical harmonic coefficients to images for allowing coarse-to-fine first-order optimization by implementing a differentiable splat-based rasterizer. To accelerate image synthesis in dynamic scenes, Song et al. [51] decompose the dynamic scene into three fields: deformation, newness, and static fields, and introduce a sliding window scheme to represent deformation and newness features, which are combined with the static feature represented by a small MLP to infer the final color of the new view. Kerbl et al. [23] propose the 3D gaussian splatting method (3DGS) that uses 3D gaussian to represent the scene and develops a fast visibility-aware rendering algorithm to accelerate training and achieve real-time rendering. Although the 3DGS achieves excellent synthesis quality and performance, integrating the basic idea of the MeNR, that is, using scene saliency to guide point clouds construction, may further enhance the representation ability of the 3D gaussian radiance field in salient regions. And integrating the 3DGS into the framework of US-FNRF to reduce the number of samples in periphery for foveated rendering can further improve the synthesis performance without reducing visual perception quality.

2.2.2 Real-time Implicit Representation-based NeRF

Due to the scene representation being discrete in explicit scene modeling-based NeRF methods, overlapping and artifacts may occur when representing complex scenes if the number of parameters in explicit scene models is insufficient. However, if the number of parameters in explicit scene models is too large, the memory cost cannot afford high-resolution applications. Therefore, many researchers are dedicated to studying real-time implicit scene representation-based NeRF methods. In the research of real-time implicit scene representation-based NeRF, Neff et al. [38] propose a depth oracle network to predict and encode the locations of ray samples for each view, and a locally sampled shading network to accumulate ray samples. It improves the NeRF synthesis performance by reducing the number of samples per ray and the complexity of ray accumulation shading. Muller et al. [36] propose a small neural network augmented by a multi-resolution hash table of trainable feature vectors whose values are optimized through stochastic gradient descent to encode input for radiance inference, and implement the whole system using fully-fused CUDA kernels to accelerate image synthesis. Lin et al. [30] construct the cascade cost volume to output the 3D feature volume and the coarse scene geometry to guide sampling for accelerating volume rendering. To handle large motions of dynamic scenes, Wang et al. [58] introduce a residual radiance field to model the residual information between the adjacent timestamps in the spatial-temporal feature space. Deng et al. [8] propose an egocentric neural representation to represent VR scenes for better synthesizing scene images from new inside-out views. Then, they use multiple MLPs to synthesize the images from the egocentric neural representation for the foveal, mid-peripheral, and far-peripheral regions, and these images are blended to the final displayed frame.

Since the binocular resolution of VR applications reaches about 2880×1700, existing mainstream graphics cards struggle to meet the memory requirements of real-time explicit scene modeling-based NeRF methods when representing complex VR scenes. Existing implicit scene representations, e.g., the egocentric neural representation in the FoV-NeRF, use uniform density parameters to represent the radiance of any region in the scene, ignoring the need to improve the representation ability of the radiance field in salient regions. We propose the MeNR to represent the radiance field of complex VR scenes with limited structure complexity, which adaptively improves the ellipsoid layer density of salient regions to preserve the scene details. Our method utilizes the advantages of the MeNR and the US-FNRF to achieve the goal of synthesizing high-quality foveated images at high frame rates in complex VR scenes.

3 Scene-aware Foveated Neural Radiance Fields

The scene-aware foveated neural radiance fields method (SaF-NeRF) synthesizes high-quality foveated images efficiently with one-pass color inference by leveraging the US-FNRF, and uses the MeNR to accurately represent the radiance field of complex VR scenes based on the scene content. Fig. 2 visualizes the pipeline of the SaF-NeRF. There are three steps in the SaF-NeRF: the MeNR construction, the uniform-space foveated pixel sampling transformation, and the foveated image synthesis. The generation of the MeNR is performed in step 1. The construction of the US-FNRF is performed in steps 2 and 3.

Before the training process begins, step 1 constructs the MeNR based on the estimated saliency and depth of the captured scene images at all views of a complex VR scene in the training set. During the training process, step 2 transforms the foveated pixel sampling in the screen space to a uniform space and outputs the uniform-space foveated pixel samples based on the given input view and gaze. Then, step 3 gets the radiance features by traversing the uniform-space foveated pixel samples in the constructed MeNR, synthesizes the foveated image and outputs the radii offsets by encoding and decoding the radiance features through necessary modules. The foveated scene-aware objective function is formulated to optimize the network based on the synthesized foveated images, captured scene images, and radii offsets. During the testing process, the SaF-NeRF synthesizes foveated images of the scene based on the given view and gaze in real time.

In the SaF-NeRF, uniform space is a topological space that provides a consistent definition for all regions in a view. It utilizes a non-linear transformation to compress the peripheral region, ensuring consistent sampling rates for all regions within the uniform space. The radii offsets are trainable parameters, which will be accumulated into the three radii in x, y, z axes of each ellipsoid in the MeNR after each training session. It aims to optimize the layer density in the MeNR to enhance its radiance fields representation capability. This aims is achieved by optimizing the parameters in the network structure based on the foveated scene-aware objective function.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3429416

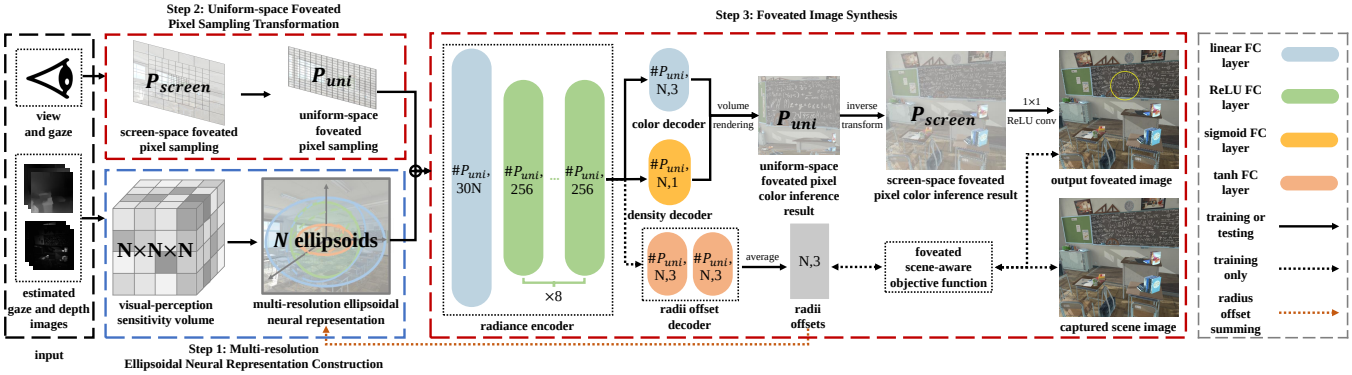


Fig. 2: *Overview of our method.* The MeNR construction is visualized in the blue dashed box, and the US-FNRF is visualized in the red dashed boxes. In the training process, our method first initializes the MeNR based on the saliency and depth estimation. During the forward propagation process of training, our method performs uniform-space foveated pixel sampling in the MeNR, then feeds the sampling points to the radiance coder and goes through other necessary modules to synthesize the foveated image and outputs the radii offsets of N ellipsoids in x, y, z axes in the MeNR. The radii offsets are then accumulated into the three radii in x, y, z axes of each ellipsoid in the MeNR to complete the forward propagation of this training session. The network is optimized by minimizing the loss calculated by the foveated scene-aware objective function. In the testing process, given the view and gaze position, our method synthesizes the corresponding foveated image in real time.

In the rest of this section, we propose the MeNR in section 3.1, which increases the radiance field representation capability in complex scenes based on the scene content to improve the synthesis quality of foveated images. Then, we introduce the US-FNRF in section 3.2, which builds a single network architecture optimized by the foveated scene-aware objective function to synthesize high-quality foveated images efficiently.

3.1 Multi-ellipsoidal Neural Representation

Algorithm 1: MeNR Initialization

Input: views in trainset *rays*, viewport of camera FOV, captured scene images in trainset *imgs*, width and height of captured images (W,H), number of ellipsoids N, the minimum and maximum distances from the originate of the scene (d_{min},d_{max})

Output: the initialized MeNR Ω

1 $Vol \leftarrow$ initVolume(N)
2 **for** $(img,ray) \in (imgs,rays)$ **do**
3 $salImg \leftarrow$ saliency(img)
4 $depthImg \leftarrow$ depth(img)
5 **for** $px \in img$ **do**
6 $o_{px},dir_{px} \leftarrow$ pxPosDir(px,W,H,FOV,ray)
7 $depth_{px} \leftarrow$ scaleDepth($depthImg[px],d_{min},d_{max}$)
8 $pos \leftarrow$ rayCast(o_{px}, dir_{px}, $depth_{px}$)
9 $p\hat{o}s \leftarrow$ round(pos,N)
10 $Vol[p\hat{o}s] \leftarrow Vol[p\hat{o}s]$ + $salImg[px]$
11 **end**
12 **end**
13 $points \leftarrow$ GaussianKernelSampling(Vol,N)
14 $sortedRadius_{x,y,z} \leftarrow$ sort($points$)
15 $\Omega \leftarrow$ construct($sortedRadius_{x,y,z}$)
16 return Ω

In complex scenes, salient regions are usually concentrated in some limited ranges rather than spreading over the whole scene [41]. Enhancing the radiance field representation capability within these regions for neural radiance fields methods can improve the synthesizing quality of foveated images. The FoV-NeRF uses a concentric sphere structure to represent the radiance field of a scene, and this structure has the same radiance field representation capability for all regions in the scene. Due to the real-time requirement, the number of sphere layers in the concentric sphere structure is limited in VR applications, and the radiance field in salient regions cannot be accurately represented due to the limited sphere layer density. Therefore, it is necessary to improve the radiance field representation capability of salient regions with a limited number of sphere layers. The MeNR adopts a concentric ellipsoid structure to represent the scene radiance field. It enhances the radiance field representation capability in salient regions of a scene by adaptively increasing the ellipsoid layer density of the structure in these salient regions.

In SaF-NeRF, firstly, we initialize the MeNR based on all views and captured scene images in the training set. In the MeNR initialization, we first construct a volume structure centered at the origin of the scene with a radius of d_{max} to indicate the saliency of the scene. The volume covers all regions of the scene. The value of each voxel in the volume indicates the saliency of the corresponding region in the scene. Then, we sample the radii of all ellipsoids in x, y, z axes through the importance sampling based on the saliency values in the volume. This results in more samples in voxels with high saliency values, and the layer density is higher in these regions, so the MeNR has better representation capability in these regions. The unevenness of saliency on the ellipsoidal surface in the representation does not affect the capability of radiance field representation. This is because the proposed representation adjusts the radiance field representation capability by controlling the ellipsoid layer density based on saliency in different regions. For a specific ellipsoid in the proposed representation, if the saliency of a certain region on its surface is high, the representation will increase the layer density around this region. Conversely, it will decrease the layer density.

Given the views *rays* in the training set, the viewport of the camera FOV, the captured scene images *imgs* in the training set, the width and height of the captured images (W,H), the number of ellipsoids in the MeNR N, and the minimum and maximum distances from the originate of the scene (d_{min},d_{max}), Algorithm 1 initializes the MeNR Ω.

Firstly, the visual-perception sensitivity volume Vol is initialized in line 1. In the initialization of Vol, we initialize a cube volume structure with a side length of $2d_{max}$ centered at the scene origin to indicate the saliency of the scene. The number of voxels in the volume is $N \times N \times N$, the length, width, and height of each voxel are all $\frac{2d_{max}}{N}$. The smallest coordinate of a voxel is $(\frac{(1-N)d_{max}}{N}, \frac{(1-N)d_{max}}{N}, \frac{(1-N)d_{max}}{N})$ and the biggest coordinate is $(\frac{(N-1)d_{max}}{N}, \frac{(N-1)d_{max}}{N}, \frac{(N-1)d_{max}}{N})$. This volume covers all regions of the scene. The value of each voxel in the volume indicates the saliency of the corresponding region in the scene.

Then, Algorithm 1 calculates the saliency value for each voxel (lines 2-12). For each image *img* in the captured image dataset *imgs*, we use the graph-based visual saliency method [16] to calculate the saliency image *salImg* of *img* (line 3), and use the monocular depth estimation

method [13] to calculate depth image *depthImg* of *img* (line 4). The saliency and depth estimation is only used in the MeNR initialization, which guides the volume construction for sampling radii in the MeNR. After the MeNR is initialized, the saliency estimation will no longer be required during training, testing or viewing process. For each pixel *px* in *img*, we calculate the ray origin o_{px} and ray direction dir_{px} (line 6). o_{px} is the same as its corresponding view position, and dir_{px} is the direction vector from the view position o_{px} to the pixel location of *px* on the view plane. The depth value in *depthImg* is in the range of [0,1], so we use Equation 1 to scale the depth value $depthImg[px]$ to obtain the estimated depth value $depth_{px}$ in the scene range (d_{min}, d_{max}) (line 7):

$$depth_{px} = depthImg[px] \cdot (d_{max} - d_{min}) + d_{min} \tag{1}$$

where $depthImg[px]$ is the depth value of *px* in *depthImg*. Then we use a ray casting method [42] to get the position *pos* of *px*'s hitpoint in the scene according to o_{px}, dir_{px}, and $depth_{px}$ (line 8). We use *round* function to calculate *pos*'s voxel index $\hat{pos}$ in *Vol* (line 9). Specifically, for a scene's position *pos* in the range of [($-d_{max}$, $-d_{max}$, $-d_{max}$), (d_{max}, d_{max}, d_{max})], the corresponding voxel index $\hat{pos}$ in *Vol* is $\lfloor \frac{pos+d_{max}}{2d_{max}+0.001} \cdot N \rfloor$, where $\lfloor\ \rfloor$ is the floor operation, and $\hat{pos}$ ranges from $[0,0,0]$ to $[N-1, N-1, N-1]$. Then, we add the saliency value of *px* in *salImg* to $Vol[\hat{pos}]$ to get the saliency value of the region in $Vol[\hat{pos}]$ (line 10).

After constructing the visual-perception sensitivity volume *Vol*, we use the Gaussian kernel density estimation method [27] to estimate the probability density function of the scene saliency, and sample *N* values based on the significance weight probability density function to get the sampling point set *points* (line 13). We sort all the points in *points* along x, y, z axes respectively to get the sorted value set $sortedRadius_{x,y,z}$ in x, y, z axes (line 14). $sortedRadius_{x,y,z}$ is regarded as the radii of all ellipsoids in the MeNR Ω, and we initialize Ω based on $sortedRadius_{x,y,z}$ and return Ω (lines 15-16).

In the MeNR initialization, we initialize the radii of all ellipsoids contained in the MeNR along x, y, z axes. In the rendering pipeline, for each pixel *px*, we intersect each ellipsoid in the MeNR based on the position and direction of *px* to obtain the sampling point set *PT*. Then, *PT* is fed into the radiance encoder to obtain the final output color of the pixel through forward propagation. Next, the parameters in the radiance encoder are optimized through loss function optimization in backpropagation. The neural radiance field of the scene is represented by the combination of the MeNR and parameters in the radiance encoder, which makes the neuralization of the MeNR. The parameters in the radiance encoder and the radii offsets are optimized through the proposed foveated scene-aware objective function (Section 3.2) in backpropagation. The radii offset optimization improves the layer density in salient regions of the scene, and the radiance encoder optimization enhances the precision of radiance field representation.

After the MeNR initialization, we optimize all ellipsoids' radii in the representation to adjust the ellipsoid layer density to further improve the radiance field representation capability. Specifically, after the radiance sampling of the MeNR for each pixel sample in the input view, the SaF-NeRF uses a radiance encoder to output the radiance features. In the MeNR, if the viewpoint moves outside the innermost ellipsoid, rays may not intersect with the innermost ellipsoid in the MeNR, and the sampling point with the innermost ellipsoid is set as (0,0,0) and is fed into the radiance encoder. The value of sampling point (0,0,0) indicates that the viewpoint does not sample radiance on the innermost ellipsoid. This is consistent with the conventional NeRF method of sampling radiance on voxel-based or egocentric radiance field representation. The radii offset decoder is connected with the radiance encoder to output the radii offsets, which utilizes the radiance features to guide the fine tune of all ellipsoids' radii. Then, the radii offsets are accumulated to the radii of all ellipsoids to achieve the scene-aware fine tune of ellipsoid layer density.

3.2 Uniform Sampling based Foveated Neural Radiance Field Framework

Conventional foveated neural radiance fields methods require building and training multiple synthesis networks to synthesize rendering images in the foveal, transitional and peripheral regions, and blending these images to obtain foveated images, which degrades the synthesis performance of foveated images. In addition, images from fovea to periphery synthesized by multiple networks in conventional foveated neural radiance fields methods need to be accurately aligned based on the current view. Inaccurate alignment results in breaks, which affects the synthesis quality of the final blended foveated image. We propose the US-FNRF, which uses a single network that is optimized by the foveated scene-aware objective function to synthesize foveated images. Compared with the conventional foveated neural radiance fields methods, the proposed framework synthesizes high-quality foveated images while reducing network complexity in complex VR scenes.

In order to improve the foveated image synthesis quality of our method, we use a saliency estimation method and a connected-component labeling method [17] to fully extract scene information from the training set in the process of network training. For the saliency estimation results of each captured scene image in the training set shown in the left side of Fig. 3 (a), we use a connected-component labeling method to determine the most likely gaze positions of that captured scene image according to the saliency estimation [16], as shown in the right side of Fig. 3 (a). Then, the US-FNRF synthesizes the foveated images based on those analyzed gaze positions for a single view, as shown in Fig. 3 (b), and optimizes the network based on the synthesized foveated images and captured scene images at the given views.

The network architecture of the US-FNRF includes three modules: the uniform-space foveated pixel sampler module, the inference module, and the inverse-transformed decoder module. The inference module has the radiance encoder, and three decoders: color, density, and radii offset decoders.

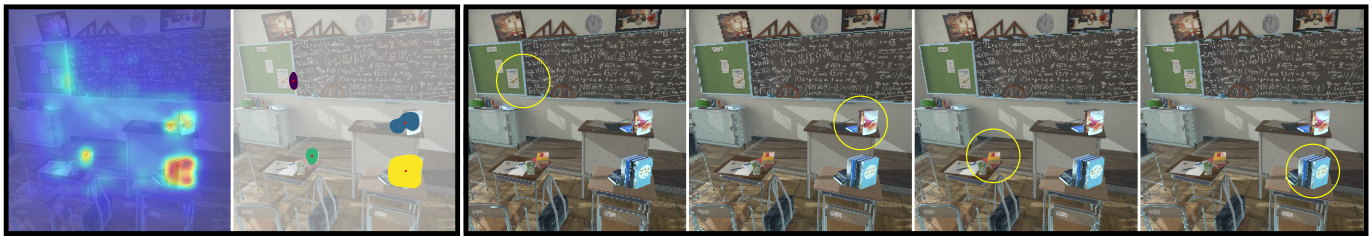
The uniform-space foveated pixel sampler module calculates the position *pos* and the direction *dir* of each pixel in the current view. The radiance encoder intersects *N* ellipsoids in the MeNR with *pos* and *dir* to get the sampling point set *PT* with the dimension of [$\#P_{uni}$, *N*, 3], where $\#P_{uni}$ is the number of pixels in the uniform space.

Then, *PT* is fed into the radiance encoder in the inference module. The inference module includes a fully connected layer using the linear activation function [46] connected with 8 fully connected layers using the ReLU activation function [2], which is the same as [8]. The first layer converts *PT* into a high-frequency feature tensor with the dimension of [$\#P_{uni}$, $N \times 3 \times 10$], and then the high-frequency feature tensor is fed into the remaining 8 layers to obtain the radiance feature tensor with the dimension of [$\#P_{uni}$, 256].

The radiance feature tensor is fed into the volume color, density, and radii offset decoders to get the volume color tensor with the dimension of [$\#P_{uni}$, *N*, 3], density tensor with the dimension of [$\#P_{uni}$, *N*, 1], and radii offsets tensor with the dimension of [$\#P_{uni}$, *N*, 3], simultaneously. The volume color decoder is a fully connected layer using the linear activation function, the density decoder is a fully connected layer using the sigmoid activation function [37], and the radii offset decoder has two fully connected layers using the tanh activation function [28]. We average the offsets based on the number of pixels $\#P_{uni}$ and add them to the corresponding radius of each ellipsoid to optimize the representation ability of the MeNR.

In the inverse-transformed decoder module, we first use the volume rendering method [35] to calculate the color with the dimension of [$\#P_{uni}$, *N*, 3] based on the color and density feature tensors, and then inversely transform the calculated color to the screen space with the dimension of [*W*, *H*, *N*, 3]. At last, we use a 2D convolutional layer with the ReLU activation function to denoise the artifacts in periphery and output the synthesized foveated image. The kernel size is [5,5] and the stride is [1,1] in the 2D convolutional layer.

The foveated image synthesis process in the US-FNRF is shown in Algorithm 2. Given the current view *ray*, the current gaze position (x_g, y_g), the width and height of output images (W, H), the camera



(a) (b)

Fig. 3: *Gaze position determination in network training.* We first estimate the saliency of captured scene images in (a) left, and determine the most likely gaze positions in (a) right based on the saliency estimation. We visualize the synthesized foveated images at the given view with four determined gaze positions in (b).

Algorithm 2: Foveated Images Synthesis in the US-FNRF

Input: view ray, gaze position (x_g, y_g), width and height of output images (W, H), viewport of camera FOV, compression coefficient of rectangular mapping σ, MeNR Ω

Output: synthesized foveated image of the current view COL

1 $P_{screen} \leftarrow \text{pxPosDir}(ray, W, H, FOV)$
2 $P_{uni} \leftarrow \text{uniformTransform}(P_{screen}, ray, \sigma, x_g, y_g)$
3 $P \leftarrow \text{ellipsoidSampling}(P_{uni})$
4 $RAD_{uni} \leftarrow \text{radEncode}(P)$
5 $volCol_{uni}, density_{uni} \leftarrow \text{ColDenEncode}(RAD_{uni})$
6 $COL_{uni} \leftarrow \text{render}(volCol_{uni}, density_{uni})$
7 $COL'_{fov} \leftarrow \text{inverseTransform}(COL_{uni}, x_g, y_g, W, H)$
8 $COL_{fov} \leftarrow \text{decodeConv}(COL'_{fov})$
9 return COL_{fov}

viewport FOV, the compression coefficient σ and the MeNR Ω, the proposed framework synthesizes the foveated image COL_{fov} of the current view and gaze position.

In the uniform-space foveated pixel sampler module, it first constructs the position and direction set of pixels in screen space P_{screen} based on the current view ray, the width and height of output images (W, H) and the camera viewport FOV (line 1). P_{screen} is a two-dimensional list that stores the ray position vector o_{px} and the ray direction vector dir_{px} of each pixel in screen space. Secondly, in order to reduce the number of parameters in the network, the uniform-space foveated pixel sampler module uses the rectangular mapping method [64] to compress the screen-space pixel features in P_{screen} into a low-resolution uniform space, and outputs the uniform-space pixel feature set P_{uni} (line 2). Specifically, for each pixel px whose coordinate is (x, y) in screen space, we map it to the uniform space coordinate (u, v) using Equation 2 and assign the pixel feature of $P_{screen}[x, y]$ into $P_{uni}[u, v]$, i.e., $P_{uni}[u, v] = P_{uni}[u, v] \cup P_{screen}[x, y]$:

$$\begin{cases} u = N_x\left(\frac{f \cdot x}{f + abs(x)}\right) \cdot \frac{W}{\sigma} \cdot (1 - \frac{x_g}{W}) + \frac{x_g}{\sigma} \\ v = N_y\left(\frac{f \cdot y}{f + abs(y)}\right) \cdot \frac{H}{\sigma} \cdot (1 - \frac{y_g}{H}) + \frac{y_g}{\sigma} \end{cases} \tag{2}$$

where x is in the range of $[0, W]$; y is in the range of $[0, H]$; u is in the range of $[0, \frac{W}{\sigma}]$; v is in the range of $[0, \frac{H}{\sigma}]$; f is the shading rate decrease control coefficient and is set to 0.38 [64]. $N_x()$ and $N_y()$ are shown in Equation 3:

$$\begin{cases} N_x(x) = \frac{x}{\frac{f \cdot (W - x_g)}{f + (W - x_g)}} \\ N_y(y) = \frac{y}{\frac{f \cdot (H - y_g)}{f + (H - y_g)}} \end{cases} \tag{3}$$

The uniform-space foveated pixel sampler module compresses the pixels in the screen space into the low-resolution uniform space. A pixel in the uniform space may correspond to multiple pixels in the screen space, i.e., there exists the rectangle space coordinate (u, v), such that $P_{uni}[u, v]$ contains multiple position and direction vectors. We average all position and direction vectors so that each coordinate position in P_{uni} contains only one position vector and direction vector.

Then, we take the ray origin o_{px} corresponding to all pixels px in P_{uni} as the position, intersect with the MeNR Ω along the ray direction dir_{px} to get the sampling point set p by Equation 4, and construct the sampling point set P based on the p of all pixels (line 3):

$$p = o_{px} + \frac{\sqrt{\Delta} - B}{2A} \cdot dir_{px} \tag{4}$$

where $\Delta = B^2 - 4AC$. A, B and C are calculated by Equation 5:

$$\begin{cases} A = \sum_{i \in xyz} \frac{dir_{px}[i]^2}{radius[i]^2} \\ B = \sum_{i \in xyz} \frac{2 o_{px}[i] \cdot dir_{px}[i]}{radius[i]^2} \\ C = \sum_{i \in xyz} \frac{o_{px}[i]^2}{radius[i]^2} - 1 \end{cases} \tag{5}$$

where $radius$ is the radius on x, y, z axes of each ellipsoid in Ω.

In the inference module, we feed the sampling point set P into the radiance encoder to obtain the radiance feature RAD_{uni} (line 4). Then, RAD_{uni} is fed into the volume color decoder and density decoder simultaneously to obtain the volume color feature $volCol_{uni}$ and density color feature $density_{uni}$ (line 5).

In the inverse-transformed decoder module, we use the volume rendering method [35] to estimate the color of all pixels COL_{uni} in the uniform space based on the volume color feature $volCol_{uni}$ and density color feature $density_{uni}$ (line 6). The color estimations of all pixels in RAD_{uni} are inverted into the screen space using Equation 6 (line 7), and the converted radiance features are fed into the 2D convolutional network for decoding and outputting the synthesized foveated image (lines 8-9):

$$RAD[\frac{f \cdot N_u u}{f - N_u u}, \frac{f \cdot N_v v}{f - N_v v}] = RAD_{uni}[u, v] \tag{6}$$

where (u, v) is the pixel coordinate in RAD_{uni}. $N_u()$ and $N_v()$ are shown in Equation 7:

$$\begin{cases} N_u u = \frac{u - \frac{x_g}{\sigma}}{\frac{W}{\sigma}(1 - \frac{x_g}{\sigma})} \cdot \frac{f(W - x_g)}{f + (W - x_g)} \\ N_v v = \frac{v - \frac{y_g}{\sigma}}{\frac{H}{\sigma}(1 - \frac{y_g}{\sigma})} \cdot \frac{f(H - y_g)}{f + (H - y_g)} \end{cases} \tag{7}$$

Since in foveated rendering, the rendering quality in the peripheral region can be reduced to some extent without sacrificing the perceptual rendering quality [49]. The foveated scene-aware objective function reduces the constraints on the synthesis quality from fovea to periphery based on the visual acuity fall-off model [15], and optimizes the radii of all ellipsoids in the MeNR structure to further improve the radiance field representation capability of the structure. The loss function is shown in Equation 8:

$$min\left((m\cdot e+\omega_0)MSE(COL[e],GT[e])+0.001\sum_{r\in rO}\frac{r}{\rho_r}\right) \quad (8)$$

where COL is the synthesized foveated image; GT is the ground truth image of the current view; rO is the radii offsets; ρ_r is the corresponding density of r in the MeNR structure; e is the eccentricity angle; m and ω_0 are the coefficients of the visual acuity fall-off model and are set to 1.65 and $\frac{1}{48}$ [15]. The foveated scene-aware objective function comprises a foveated image loss term and a density-based radii offset term. The foveated image loss term quantifies the quality degradation of the synthesized foveated image relative to the ground truth image, based on the visual acuity fall-off model. The density-based radii offset term minimizes the radii offsets according to the ellipsoidal layer density in the MeNR. The density-based radii offset term regards regions with higher layer density as salient regions, which require finer radii offset optimization. Consequently, it permits more aggressive fine-tuning of the radii in these regions, while imposing penalties for larger radii offsets in non-salient regions. Accurate radii offsets further reduce the foveated image loss term, thereby enhancing the overall quality of the synthesized foveated image.

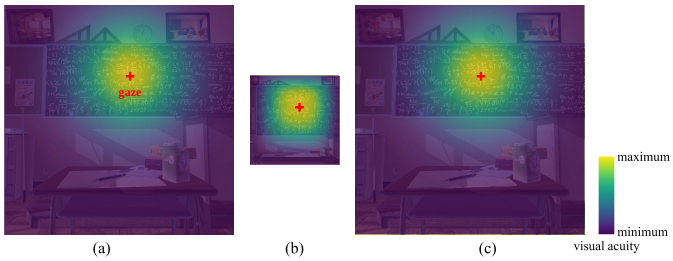


Fig. 4: *Stretching and squeezing visualization of synthesis.* We visualize the ground truth image in (a) where the gaze is at the center of the image. In (a), the visual acuity decreases from the maximum at the center of fovea to the minimum at the edge of periphery according to the visual acuity fall-off model. (b) is the output result of the uniform-space foveated pixel sampler module in the US-FNRF, and (c) is the final synthesized result of the US-FNRF.

Fig. 4 shows the stretching and squeezing visualization during the synthesis process of the US-FNRF. Fig. 4 (a) shows the ground truth image at a specific view, where the gaze is at the center of the image. According to the visual acuity fall-off model, the visual acuity decreases from fovea to periphery. First, the US-FNRF uses the uniform-space foveated pixel sampler module to perform foveated sampling at the MeNR. The sampling points are propagated forward into the inference module, and the inference module outputs the uniform-space foveated pixel color inference result, as shown in Fig. 4 (b). The output resolution of the inference result is only $\frac{1}{\sigma}$ of that of the ground truth, but the foveal region with high visual acuity is stretched to occupy most of the synthesis result, ensuring that the synthesis quality of the foveal region is not reduced. Then, the inference result is fed to the inverse-transformed decoder module, and the US-FNRF finally outputs the foveated image at the current view, as shown in Fig. 4 (c). The foveal region with high visual acuity is compressed back to the screen space, while the peripheral region with low visual acuity is stretched in screen space to synthesize the final foveated image, thus achieving synthesis performance improvement by reducing the peripheral synthesis quality.

4 Experimental Results and Discussion

In this section, we first give the implementation details of our method (Section 4.1), and then we compare the synthesis quality and performance of our method with *FoV-NeRF* by the quantitative quality experiment (Section 4.2) and the quantitative performance experiment (Section 4.3).

4.1 Implementation Details

We use an HTC Cosmos HMD with Droolon aGlass to track the user's gaze position. The HMD is connected to a graphics workstation with a 3.8GHz Intel Core(TM) i7-10700KF processor, 64GB of RAM, and an NVIDIA GeForce RTX 4090 graphics card. We test the foveated image synthesis quality and performance of our method in four test complex VR scenes and two real-world scenes. VR scenes contain two indoor scenes *classroom* and $office$, and two outdoor scenes *park* and *street*. Real-world scenes contain an indoor scene *playroom* and an outdoor scene *treehill*, and the dataset of real-world scenes are presented in [3]. To construct the datasets of VR scenes, we use a trajectory camera to roam VR scenes, and then use FFmpeg [55] to perform sparse sampling during the roam to generate 360 images. We use colmap method [44] to reconstruct the image set, automatically generating the camera position and direction corresponding to each image. We randomly select 300 images and their corresponding camera data as the training set, and the remaining 60 data as the testing set, finally obtaining the datasets of four complex VR scenes.

To synthesize the foveated images in real time, the network architecture of our method is built on CUDA-based PyTorch, and CUDA operators are utilized for MeNR-based sampling, radiance inference, and volume rendering to enhance the synthesis performance. We use the SIBR framework [5] for real-time view of synthesis results. In addition to denoising artifacts in periphery in the network architecture of the proposed method by the 2D convolutional layer, we implement a foveated anti-aliasing filter [39] in openGL to reduce the temporal flickers in periphery before presenting synthesis results to the screen.

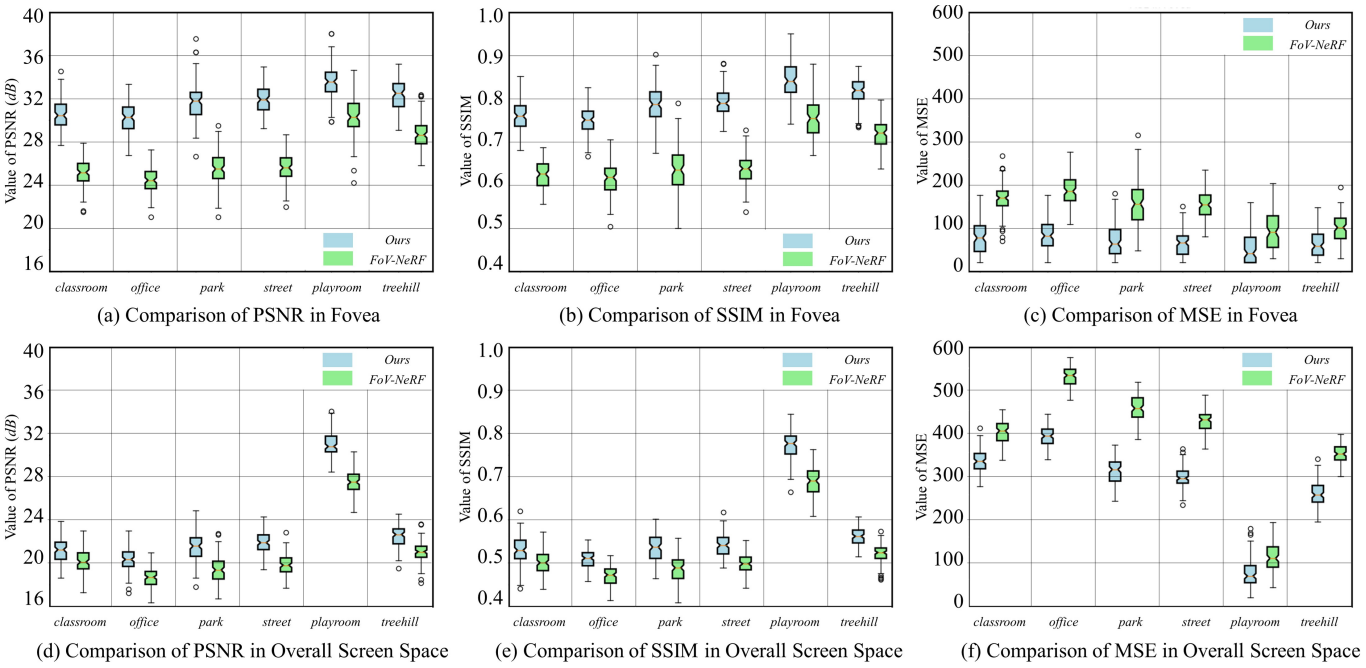
Our method adopts an end-to-end training approach. In experiments, consistent with *FoV-NeRF*, the optimizer of our method is the Adam optimizer, the learning rate of our method is 5×10^{-4}, the exponential decay function with an attenuation parameter of 0.9999954 being used to tune the learning rate after each iteration, and the number of ellipsoids in the MeNR N is 64. The compression coefficient σ is set to 2.6, which is the same as [63]. We train our method with 7000 epochs using the batch size of 4096, and the average training duration of all scenes is 6.09 minutes.

4.2 Quality

We use peak signal-to-noise ratio (PSNR), structure similarity index measure (SSIM), and mean squared error (MSE) to quantify the quality of foveated images synthesized by our method (*Ours*) and *FoV-NeRF*. PSNR [25] evaluates the quality of synthesized foveated images by comparing the noise between the synthesized images and the ground truth images. When PSNR is above 30dB, the HVS can hardly perceive the difference between the synthesized images and the ground truth images; when PSNR is in the range of 20-30dB, the quality of the synthesized image is poor compared with the ground truth image; PSNR below 20dB indicates severe image distortion [54]. SSIM [60] is an indicator for quantifying the structural similarity between the synthesized image and the ground truth image based on the theory of HVS's structural similarity. The range of SSIM values is 0 to 1, with larger values indicating higher structural similarity between the synthesized image and the ground truth image. MSE [7] calculates the squared intensity differences between the synthesized image and the ground truth image. The smaller the value of MSE, the smaller the difference between the synthesized image and the ground truth image.

Fig. 5 (a)-(c) visualize the PSNR, SSIM, and MSE statistic results of *Ours* and *FoV-NeRF* in the foveal region in the testing sets of six scenes. PSNR of our method is 1.22-1.25× higher than that of *FoV-NeRF* in the foveal region. We use p-value [61], Cohen's d and its $effect\ size$ [43] to evaluate the significance of our method's synthesis quality compared with *FoV-NeRF*. The significance evaluation results in the foveal region are shown in Table 1 columns 1-5. Both p-values and Cohen's d indicate that our method achieves significant synthesis quality improvement in PSNR compared with *FoV-NeRF*. The mean values of our method's PSNR are above 30 in all scenes, which indicates that the synthesized images of our method in the foveal region achieve no significant perceptual difference compared with the ground truth images. SSIM and MSE of our method are 1.22-1.25×

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3429416



(a) Comparison of PSNR in Fovea (b) Comparison of SSIM in Fovea (c) Comparison of MSE in Fovea

(d) Comparison of PSNR in Overall Screen Space (e) Comparison of SSIM in Overall Screen Space (f) Comparison of MSE in Overall Screen Space

Fig. 5: *Synthesis quality quantitative comparison results.* We compare the synthesis quality between *Ours* and *FoV-NeRF* in the foveal region and overall screen space in all test scenes by the quantitative metrics: PSNR, SSIM, and MSE.

higher and 2.18-2.42× smaller than those of *FoV-NeRF* in the foveal region. When evaluating the synthesis quality using SSIM and MSE, our method also achieves significant synthesis quality improvement compared with *FoV-NeRF* in the foveal region. This is because the proposed MeNR enhances the radiance field representation capability by improving the ellipsoid layer density of the structure in salient regions, and the inference network trained by the foveated scene-aware objective function preserves the structure details in the foveal region. Thus the image synthesis quality in the foveal region is improved compared with *FoV-NeRF*.

Fig. 5 (d)-(f) visualize the PSNR, SSIM, and MSE statistic results of *Ours* and *FoV-NeRF* in the overall screen space in the testing sets of six scenes. The significance evaluation results in the overall screen space are shown in Table 1 columns 6-8. Our method achieves better overall synthesis quality than *FoV-NeRF*. PSNR, SSIM and MSE of our method are 1.05-1.10× higher, 1.04-1.10× higher, and 1.19-1.48× smaller than those of *FoV-NeRF*. p-values of PSNR, SSIM and MSE between our method and *FoV-NeRF* are all below 0.01. Cohen's d values show that the overall synthesis quality of our method outperforms *FoV-NeRF* significantly in all scenes in PSNR. p-value and Cohen's d value indicate that our method achieves a significant improvement of synthesis quality than *FoV-NeRF* in both the foveal and peripheral regions. This is because the MeNR enhances the radiance field representation capability, and the US-FNRF controls the decrease in shading rates while aiming to minimize the loss of scene structural information in the peripheral region.

To detail the synthesis quality of our method, Fig. 6 shows the foveated images synthesized by our method (*Ours*, column 2) and *FoV-NeRF* (column 3) compared with the ground truth images (*GT*, column 1) at the views with high occlusion and objects with complex textures and geometries in six test scenes. The yellow circles on the image of *Ours* and *FoV-NeRF* indicate the foveal regions. We also crop and magnify the details in both the foveal and peripheral regions on the right of each rendering image for comparison (up: details in the green rectangle, down: details in the red rectangle). Our results are closer to the ground truth images than those of *FoV-NeRF*. Some artifacts are shown in the rectangle regions synthesized by *FoV-NeRF*. In *classroom*, mathematical formulas presented on the blackboard are

Table 1: *Significance evaluation of our method's synthesis quality compared with FoV-NeRF.* We use p-value, Cohen's d, and its $effect\ size$ to evaluate the significance of synthesis quality between Ours and FoV-NeRF measured by PSNR, SSIM and MSE.

Metric	Scene	Fovea			Overall Screen Space		
		p-value	Cohen's d	$effect\ size$	p-value	Cohen's d	$effect\ size$
PSNR	*classroom*	7.71×10^{-73}	4.08	*huge*	4.43×10^{-10}	0.93	*large*
	office	6.38×10^{-76}	4.26	*huge*	1.39×10^{-23}	1.62	*very large*
	park	4.14×10^{-61}	3.43	*huge*	2.46×10^{-22}	1.56	*very large*
	street	2.79×10^{-82}	4.66	*huge*	9.42×10^{-31}	1.95	*very large*
	playroom	7.55×10^{-31}	1.95	*very large*	1.02×10^{-49}	2.84	*huge*
	treehill	2.01×10^{-44}	2.59	*huge*	1.44×10^{-19}	1.43	*very large*
SSIM	*classroom*	3.08×10^{-72}	4.04	*huge*	2.48×10^{-9}	0.88	*large*
	office	6.50×10^{-70}	3.91	*huge*	5.21×10^{-25}	1.68	*very large*
	park	6.64×10^{-58}	3.26	*huge*	2.46×10^{-24}	1.65	*very large*
	street	4.78×10^{-84}	4.77	*huge*	3.37×10^{-26}	1.74	*very large*
	playroom	3.35×10^{-30}	1.92	*very large*	1.19×10^{-41}	2.46	*huge*
	treehill	1.76×10^{-51}	2.93	*huge*	1.58×10^{-26}	1.75	*very large*
MSE	*classroom*	1.33×10^{-41}	2.45	*huge*	1.50×10^{-44}	2.59	*huge*
	office	3.94×10^{-52}	2.96	*huge*	2.76×10^{-99}	5.84	*huge*
	park	1.38×10^{-29}	1.90	*very large*	1.37×10^{-84}	4.81	*huge*
	street	1.24×10^{-46}	2.69	*huge*	1.78×10^{-91}	5.27	*huge*
	playroom	2.33×10^{-11}	1.00	*large*	2.90×10^{-14}	1.16	*large*
	treehill	7.91×10^{-14}	1.14	*large*	4.72×10^{-64}	3.58	*huge*

not distinctly synthesized in the foveal region, and the textures of the book spine and milk carton lose their readability in the peripheral region adjacent to fovea. In *office*, the hands and time markings of the clock in the foveal region are excessively blurry, and there is noticeable loss of leaf veins on the leaves in the peripheral region adjacent to fovea. In *park*, the texture details on the door in the foveal region and on the window in the peripheral region are missing in *FoV-NeRF*. In *street*, the star on the door is blurred in the foveal region, and the outline of the rear of the car disappears in the peripheral region in *FoV-NeRF*. In *playroom*, the outline details of the keyboard and mouse are lost in the

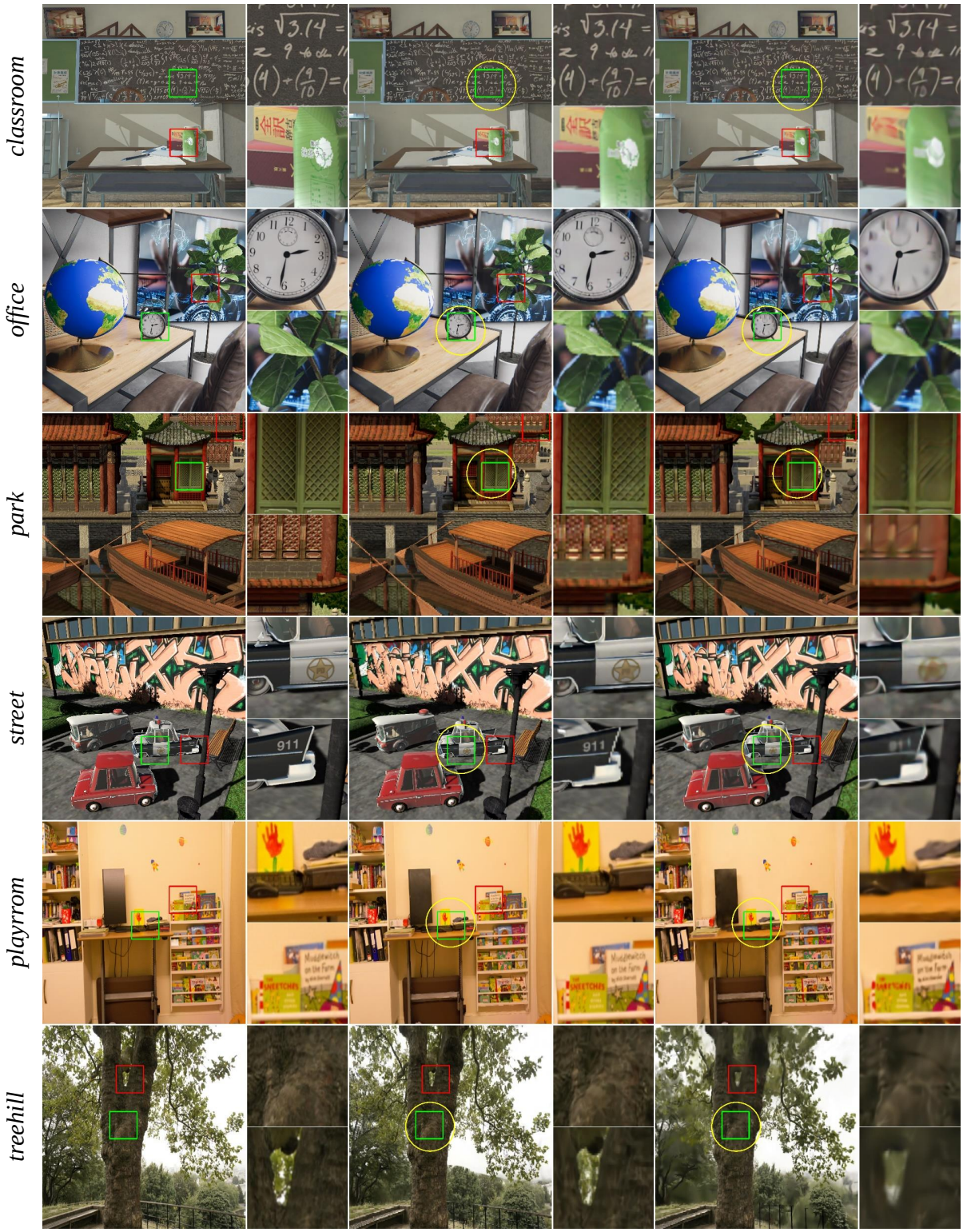


Fig. 6: *Visualization of synthesized results comparison.* We visualize the ground truth images *GT* (column 1), synthesized results of *Ours* (column 2), and *FoV-NeRF* (column 3) in all test scenes. Then, we magnify the details in the rectangular regions and place them on the right of each image for comparison.

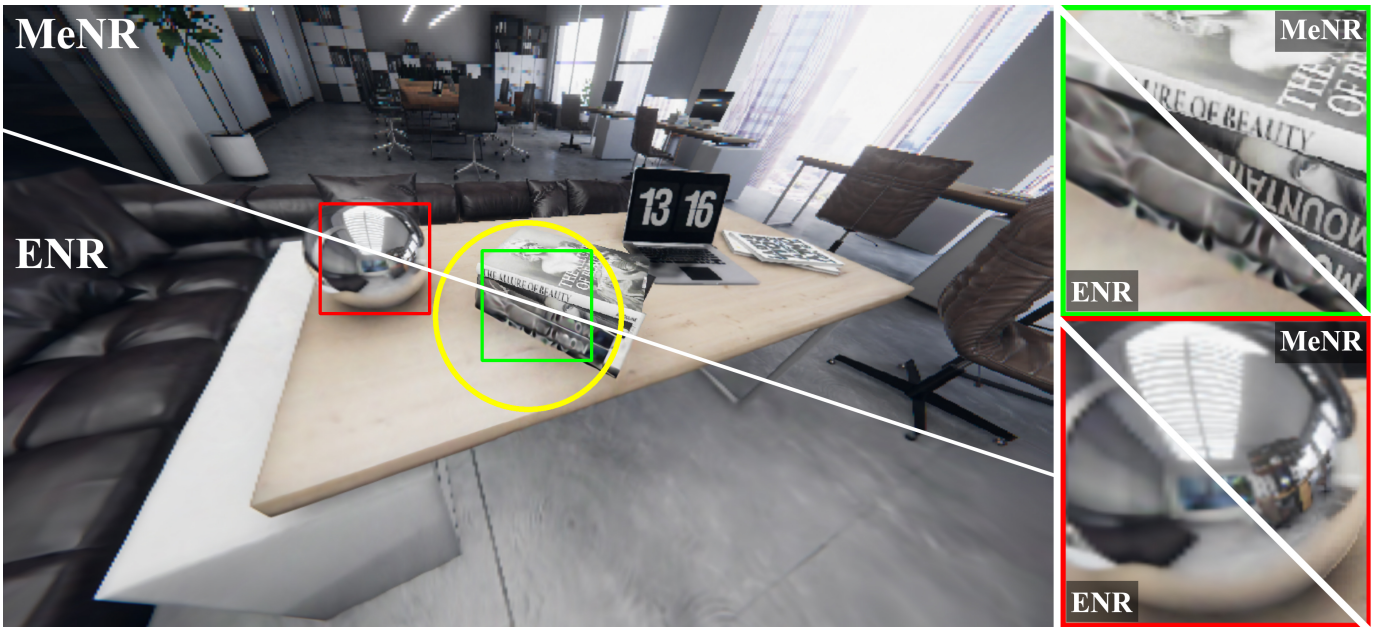


Fig. 7: *Synthesized results comparison of different neural representations.* We visualize the synthesized results of our method using the MeNR and the egocentric neural representation (ENR) in $office$, and magnify them to compare the details in the rectangular regions on the right side.

foveal region, and the letters on the book synthesized in the peripheral region are also too blurry in *FoV-NeRF*. In *treehill*, the details of the epidermis on the trunk in the foveal region cannot be retained, and the structural details of the distant tree tops through the branches are lost in the peripheral region in *FoV-NeRF*.

To verify the effectiveness of the MeNR, we also compare our method's synthesis quality when representing the neural radiance field using the MeNR and the egocentric neural representation (ENR) [8] in $office$, as shown in Fig. 7. The foveated image synthesized by the MeNR can better preserve the details of the book spine texture in the foveal region and the reflection details of the metal sphere in the peripheral region compared with the ENR, although the foveated image synthesized by the ENR preserves the scene structure information to some extent. Compared with the ground truth, PSNR, SSIM, and MSE of the MeNR are $[30.2, 0.76, 80.14]$ in the foveal region, and $[20.3, 0.51, 392.56]$ in the overall screen space. PSNR, SSIM, and MSE of the ENR are $[26.5, 0.66, 135.18]$ in the foveal region, and $[19.4, 0.49, 470.64]$ in the overall screen space. All quality assessment metrics indicate that the foveated image synthesis quality of the MeNR is superior to the ENR.

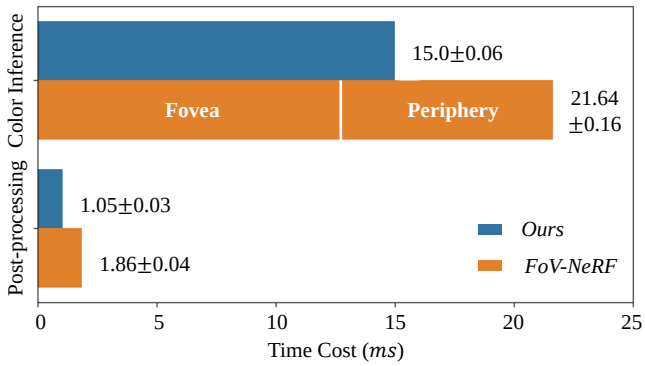


Fig. 8: *Performance comparison results.* We plot the average time cost of *Ours* and *FoV-NeRF* in color inference and post-processing steps in $office$.

4.3 Performance

Fig. 8 shows the time cost of color inference and post-processing on step 1 and step 2 separately using our method and *FoV-NeRF* to synthesize foveated images at novel views for both eyes in $office$. In step 1, *FoV-NeRF* needs to perform color inference for the foveal region and the peripheral region separately. In step 2, *FoV-NeRF* blends the foveal and peripheral synthesized images and anti-aliases the blended images. Our method performs color inference for both foveal and peripheral regions simultaneously in step 1, and only needs to perform anti-aliasing filtering to enhance visual fidelity in the peripheral region without image blending in step 2. Our method achieves the average frame rate of $62FPS$. Since our method only requires a single color inference in step 1, thereby avoiding the redundant synthesis of images in the foveal region compared with *FoV-NeRF*, so our method's performance in step 1 is $1.44\times$ better than *FoV-NeRF*. The proposed MeNR also significantly improves the image synthesis quality while maintaining our inference network complexity comparable to *FoV-NeRF*. In step 2, since our method doesn't need to blend images, it achieves a performance improvement of $1.77\times$ compared with *FoV-NeRF*. Ultimately, our method achieves a significant improvement in synthesis quality in both the foveal and peripheral regions, while also enhancing performance by $1.46\times$ compared with *FoV-NeRF*.

Fig. 9 shows the relationship between the performance and the foveal (a) and overall (b) synthesis quality in $office$ for our method and *FoV-NeRF*. For our method, when the foveal PSNR reaches $30.2dB$ and the overall PSNR reaches $20.3dB$, further enhancing the synthesis quality becomes challenging with a significant reduction in synthesis performance. Compared with *FoV-NeRF*, our method consistently achieves higher synthesis quality at similar performance levels. In *FoV-NeRF*, when the foveal PSNR reaches $24.5dB$ and the overall PSNR reaches $18.8dB$, increasing network complexity has a limited impact on improving synthesis quality while the performance sharply decreases.

5 User Study

We design a within-subject study [9] to evaluate the perceptual synthesis quality of our method in two indoor test scenes *classroom* and $office$, and two outdoor test scenes *park* and *street*.

Conditions We use our method as an experimental condition (*EC*), and *EC* uses our method to synthesize foveated images of four test

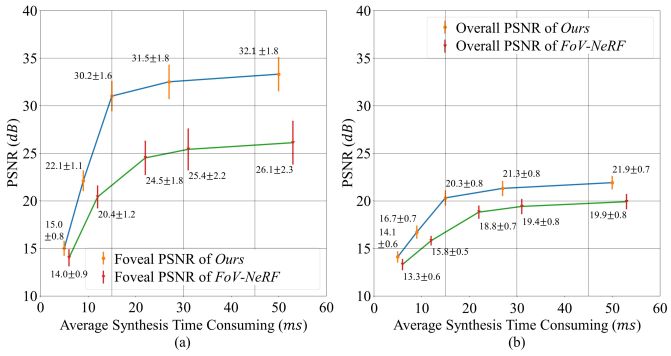


Fig. 9: *Ablation experiment results.* We visualize PSNR of *Ours* and *FoV-NeRF* as a function of the synthesis performance in $office$ in the foveal region (a) and in the overall screen space (b). The average synthesis time consuming of our method is 5.63ms, 9.23ms, 15.57ms, 27.81ms, 50.31ms, where σ is set to 3.9, 3.3, 2.6, 2.0, 1.4, and N is set to 32, 48, 64, 80, 96. The average synthesis time consuming of *FoV-NeRF* is 6.31ms, 12.47ms, 22.51ms, 31.73ms, 53.71ms, where the number of spheres in the egocentric neural representation is set to 16, 40, 64, 88, 112, and the parameter of each layer in the radiance encoder is set to 64, 128, 256, 512, 1024.

scenes: $classroom$, $office$, $park$, and $street$. The first control condition ($CC1$) is the ground truth rendering results of four test scenes. The second control condition ($CC2$) uses *FoV-NeRF* to synthesize foveated images of four test scenes.

Participants and Setup We recruit 20 participants (15 males and 5 females, aged between 21-30) in the user study, and 13 of them have had experiences in VR HMDs. The participants are asked to sign a consent form approved by the biology and medical ethics committee of Beihang University. We use an HTC Cosmos HMD with a Droolon F1 gaze tracker to track the gaze motion of the participants. The resolution of the HMD is 1440×1700 pixels for each eye, and the field-of-view is 97°. The HMD is connected to a PC workstation with a 3.8 GHz Intel(R) Core(TM) i7-10700KF CPU, 64 GB of memory, and an NVIDIA GeForce RTX 4090 graphics card.

Task 1 To ensure a fair perceived quality comparison of each method, we fix the animation view sequence, with a duration of 10 seconds for each scene's animation view sequence. In Task 1, we test various scenes from the $classroom$ to $office$. In each scene, we commence by presenting the participants with the animation of the fixed view sequence rendered by $CC1$ and telling the participants that this is the ground truth. Subsequently, we present the participants with the animation of the fixed view sequence generated by EC, $CC1$, and $CC2$ in randomized order. In the task process, the participants are asked to score the perceptual visual quality of the animation after each animation is presented. The visual quality score η [49] contains 5 confidence levels: 5 represents that they cannot perceive artifacts at all, 4 represents that they can perceive acceptable artifacts at a few very short moments, 3 represents that they can perceive acceptable artifacts, 2 represents that they can perceive noticeable artifacts, and 1 represents that they can perceive obvious artifacts. After this animation is scored, the next animation comes in. Each participant spends an average of 8 minutes. The data of 20 (participants) × 4 (scenes) × 3 (methods) = 240 trials are collected.

Task 2 In Task 2, we compare the perceptual synthesis quality between EC and $CC2$ in four test scenes. We pair the animations synthesized by EC and $CC2$ for all test scenes, and the order in the pair is randomized. The view sequence of each animation for each scene is the same as Task 1. Then, we present two animation sequences with a short interval of black in each pair to the participants. The interval duration is the same as Guenter et al. [15] (0.5s). In the task process, the participants are asked to press one of the two buttons (1 or 2) to answer the question "Which animation is synthesized with higher quality?" after being presented with each pair. After this, the next pair comes in. Each participant spends an average of 5 minutes. The data of 20 (participants) × 4 (scenes) = 80 trials are collected.

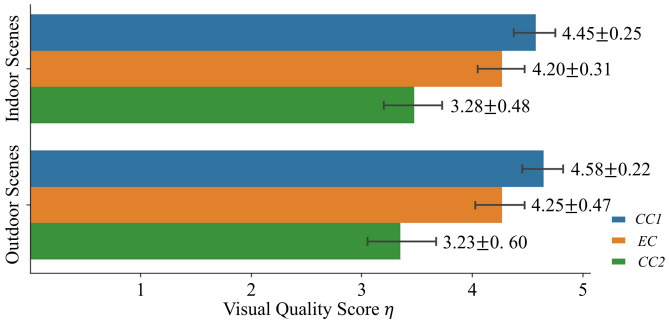


Fig. 10: *The visual quality score η of the participants in Task 1.* X-axis shows the average values and standard deviations of η of each pair. Y-axis lists the pair accordingly.

Results and Discussion Fig. 10 gives the average values and standard deviations of η under all conditions. The average values and standard deviations of η under [EC, $CC1$, $CC2$] are [4.45, 4.20, 3.28] and [0.25, 0.31, 0.48] in the indoor scenes, and [4.58, 4.25, 3.23] and [0.22, 0.47, 0.60] in the outdoor scenes. p-values of η under [EC, $CC1$, $CC2$] are [0.42, 0.80, 0.83] between the indoor and the outdoor scenes. The values of Cohen's d under [EC, $CC1$, $CC2$] are [0.18, 0.06, 0.05] between the indoor and the outdoor scenes where the corresponding $effect\ size$s are all $very\ small$. The statistical results of η indicate that the perceptual synthesis quality of [EC, $CC1$, $CC2$] has no significant difference between the indoor and the outdoor scenes.

p-values of η between EC and $CC1$ is 0.14, and Cohen's d is 0.33 where the $effect\ size$ is $small$. It indicates that the perceptual synthesis quality of our method is similar to GT. p-values of η between EC and $CC2$ is 4.99×10^{-9}, and Cohen's d is 0.96 where the $effect\ size$ is $large$. The probability that η is equal to or greater than 4 under EC exceeds 80%, while η has only 44% under $CC2$. According to the participants' feedback, in the animation of multiple scenes under $CC2$, participants do not pay much attention to the low-quality synthesized results in the peripheral region. However, they can obviously perceive blurs and artifacts in the foveal region at certain times, which makes it difficult for η under $CC2$ to reach 4. The statistical results of η show that our method achieves significantly better perceptual synthesis quality than *FoV-NeRF*.

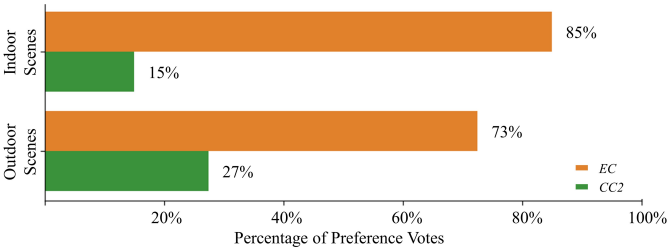


Fig. 11: *The participants' preference votes of EC and $CC2$ in Task 2.* X-axis shows the percentage of selecting the condition of each pair. Y-axis lists the pair accordingly.

Fig. 11 compares the participants' preference votes of EC and $CC2$ in the test scenes. The number of votes that the participants cast for EC exceeds those cast for $CC2$ by 3.8×. According to the participants' report, animations in EC are smoother while animations in $CC2$ are real-time, and animations in EC better preserve scene details in the foveal region. Therefore, the participants prefer EC. Additionally, the participants who choose $CC2$ regard that they will pay more attention to the synthesized details in the peripheral region because they are aware that this is foveated rendering. In some very short segments, animations

in *EC* have more noticeable flickering compared with *CC*2, so they tend to choose *CC*2.

6 Conclusion, Limitation, and Future Work

We have proposed a scene-aware foveated neural radiance fields method, which provides a high-precision neural radiance field representation in complex VR scenes and a single network to synthesize high-quality foveated images efficiently. The perceived quality of foveated images synthesized by our method shows no significant difference compared with the ground truth images. The frame rate of foveated images synthesized by our method in HMDs achieves 66*FPS*. Compared with *FoV-NeRF*, our method achieves a 1.41-1.46× speedup while significantly enhancing image synthesis quality in both the foveal and peripheral regions.

There are some limitations in our method. Firstly, our method does not consider the temporal information of the scene. It cannot synthesize high-quality foveated images for dynamic VR scenes. Therefore, a potential future work is to enhance the radiance field representation and the radiance field sampling algorithm based on the scene's temporal information, thus achieving foveated images synthesis for dynamic scenes in real time. Secondly, the US-FNRF is based on the rectangular mapping for encoding and decoding, which does not preserve the sampling density of salient parts in the peripheral region. Another potential future work is to propose a novel mapping strategy in the US-FNRF to adaptively identify the peripheral salient parts and preserve these parts' sampling density to improve the foveated image synthesis quality further. Our method does not set a limit on the range of viewpoint translation. The representation structure is learned from the images and corresponding camera parameters in the training set based on the idea of implicit representation-based NeRF methods. Thus, like the existing implicit representation-based NeRF methods, if the position of the viewpoint exceeds the coverage range of the camera translation in the training set, the radiance field representation structure lacks radiance samples from the relevant viewpoint, and the quality of the synthetic results will significantly decrease. In future work, in order to synthesize reasonable foveated images when the viewpoint moves out of the coverage range, we will try to use generative methods combined with features extracted from other scene datasets to complete the radiance beyond the coverage of the training set.

Acknowledgments

This work is supported by the Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (No.VRLAB2024C03), the National Natural Science Foundation of China through Project 61932003, 62372026, Beijing Science and Technology Plan Project Z221100007722004, National Key R&D plan 2019YFC1521102, National Natural Science Foundation of China (No. 62106268), the National Natural Science Foundation of China under Grant 62106268, and in part by the China Postdoctoral Science Foundation under Grant 2023M732223.

References

[1] T. Ananpiriyakul, J. Anghel, K. C. Potter, and A. Joshi. A gaze-contingent system for foveated multiresolution visualization of vector and volumetric data. *Electronic Imaging*, 2020. 2

[2] C. Banerjee, T. Mukherjee, and E. Pasiliao Jr. An empirical study on generalizations of the relu activation function. In *Proceedings of the 2019 ACM Southeast Conference*, pp. 164–167, 2019. 5

[3] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022. 7

[4] D. Bauer, Q. Wu, and K.-L. Ma. Fovolnet: Fast volume rendering using foveated deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 29:515–525, 2022. 2

[5] S. Bonopera, P. Hedman, J. Esnault, S. Prakash, S. Rodriguez, T. Thonat, M. Benadel, G. Chaurasia, J. Philip, and G. Drettakis. sibr: A system for image based rendering, 2020. 7

[6] V. Bruder, C. Schulz, R. Bauer, S. Frey, D. Weiskopf, and T. Ertl. Voronoi-based foveated volume rendering. pp. 67–71, 2019. 2

[7] T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014. 7

[8] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3854–3864, 2022. 1, 2, 3, 5, 10

[9] A. Field and G. Hole. *How to design and report experiments*. Sage, 2002. 10

[10] 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. 2

[11] F. Friess, M. Braun, V. Bruder, S. Frey, G. Reina, and T. Ertl. Foveated encoding for large high-resolution displays. *IEEE Transactions on Visualization and Computer Graphics*, 27:1850–1859, 2020. 2

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

[13] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3828–3838, 2019. 5

[14] J. Gortler, M. Spicker, C. Schulz, D. Weiskopf, and O. Deussen. Stippling of 2d scalar fields. *IEEE Transactions on Visualization and Computer Graphics*, 25:2193–2204, 2019. 2

[15] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. Foveated 3d graphics. *ACM transactions on Graphics (tOG)*, 31(6):1–10, 2012. 2, 6, 7, 11

[16] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. *Advances in neural information processing systems*, 19, 2006. 4, 5

[17] L. He, Y. Chao, K. Suzuki, and K. Wu. Fast connected-component labeling. *Pattern recognition*, 42(9):1977–1987, 2009. 5

[18] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. Barron, and P. Debevec. Baking neural radiance fields for real-time view synthesis. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5855–5864, 2021. 3

[19] T. Hu, S. Liu, Y. Chen, T. Shen, and J. Jia. Efficientnerf - efficient neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12892–12901, 2022. 3

[20] B. Jin, I. Ihm, B. Chang, C. Park, W.-J. Lee, and S. Jung. *Selective and adaptive supersampling for real-time ray tracing*. 2009. 2

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

[22] A. 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.*, 38:212:1–212:13, 2019. 2

[23] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 3

[24] Y. Kim, Y. Ko, and I. Ihm. Selective foveated ray tracing for head-mounted displays. *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 413–421, 2021. 2

[25] J. Korhonen and J. You. Peak signal-to-noise ratio revisited: Is simple beautiful? In *2012 Fourth International Workshop on Quality of Multimedia Experience*, pp. 37–38. IEEE, 2012. 7

[26] B. Krajancich, P. Kellnhofer, and G. Wetzstein. Towards attention–aware foveated rendering. *ACM Transactions on Graphics (TOG)*, 42:1 – 10, 2023. 2

[27] M. Kristan, A. Leonardis, and D. Skočaj. Multivariate online kernel density estimation with gaussian kernels. *Pattern recognition*, 44(10-11):2630–2642, 2011. 5

[28] M. M. Lau and K. H. Lim. Review of adaptive activation function in deep neural network. In *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 686–690. IEEE, 2018. 5

[29] D. Li, R. Du, A. Babu, C. Brumar, and A. Varshney. A log-rectilinear transformation for foveated 360-degree video streaming. *IEEE Transactions on Visualization and Computer Graphics*, 27:2638–2647, 2021. 2

[30] H. Lin, S. Peng, Z. Xu, Y. Yan, Q. Shuai, H. Bao, and X. Zhou. Efficient neural radiance fields for interactive free-viewpoint video. *SIGGRAPH Asia 2022 Conference Papers*, 2021. 3

[31] J. Liu, C. Mantel, and S. Forchhammer. Perception-driven hybrid foveated depth of field rendering for head-mounted displays. *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 1–10, 2021. 2

[32] P. Lungaro, R. Sjöberg, A. Valero, A. Mittal, and K. Tollmar. Gaze-aware streaming solutions for the next generation of mobile vr experiences. *IEEE Transactions on Visualization and Computer Graphics*, 24:1535–1544, 2018. 2

[33] X. Meng, R. Du, and A. Varshney. Eye-dominance-guided foveated rendering. *IEEE transactions on visualization and computer graphics*, 26(5):1972–1980, 2020. 2

[34] X. Meng, R. Du, M. Zwicker, and A. Varshney. Kernel foveated rendering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–20, 2018. 2

[35] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 5, 6

[36] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 3

[37] S. Narayan. The generalized sigmoid activation function: Competitive supervised learning. *Information sciences*, 99(1-2):69–82, 1997. 5

[38] T. Neff, P. Stadlbauer, M. Parger, A. Kurz, J. H. Mueller, C. R. A. Chaitanya, A. Kaplanyan, and M. Steinberger. Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. *Computer Graphics Forum*, 40, 2021. 3

[39] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics*, p. 1–12, Nov 2016. doi: 10.1145/2980179.2980246 7

[40] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10318–10327, 2021. 1, 3

[41] P. L. Rosin. A simple method for detecting salient regions. *Pattern Recognition*, 42(11):2363–2371, 2009. doi: 10.1016/j.patcog.2009.04.021 4

[42] S. D. Roth. Ray casting for modeling solids. *Computer graphics and image processing*, 18(2):109–144, 1982. 5

[43] S. S. Sawilowsky. New effect size rules of thumb. *Journal of modern applied statistical methods*, 8:597–599, 2009. 7

[44] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 7

[45] M. Schütz, K. Krösl, and M. Wimmer. Real-time continuous level of detail rendering of point clouds. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 103–110, 2019. 2

[46] S. Sharma, S. Sharma, and A. Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017. 5

[47] X. Shi, L. Wang, X. Wei, and L.-Q. Yan. Foveated photon mapping. *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1, 2021. 2

[48] X. Shi, L. Wang, J. Wu, R. Fan, and A. Hao. Foveated stochastic lightcuts. *IEEE Transactions on Visualization and Computer Graphics*, 28:3684–3693, 2022. 2

[49] X. Shi, L. Wang, J. Wu, W. Ke, and C. Lam. Locomotion-aware foveated rendering. *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 471–481, 2023. 2, 6, 11

[50] R. Singh, M. Huzaifa, J. Liu, A. Patney, H. Sharif, Y. Zhao, and S. Adve. Power, performance, and image quality tradeoffs in foveated rendering. *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pp. 205–214, 2023. 2

[51] L. Song, A. Chen, Z. Li, Z. Chen, L. Chen, J. Yuan, Y. Xu, and A. Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29:2732–2742, 2022. 3

[52] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8248–8258, 2022. 1

[53] T. Tariq, C. Tursun, and P. Didyk. Noise-based enhancement for foveated rendering. *ACM Transactions on Graphics (TOG)*, 41:1 – 14, 2022. 2

[54] Z. Tian, P. Qu, J. Li, Y. Sun, G. Li, Z. Liang, and W. Zhang. A survey of deep learning-based low-light image enhancement. *Sensors*, 23(18):7763, 2023. 7

[55] S. Tomar. Converting video formats with ffmpeg. *Linux journal*, 2006(146):10, 2006. 7

[56] O. T. Tursun, E. Arabadzhiyska-Koleva, M. Wernikowski, R. Mantiuk, H.-P. Seidel, K. Myszkowski, and P. Didyk. Luminance-contrast-aware foveated rendering. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2

[57] D. R. Walton, R. K. D. Anjos, S. Friston, D. Swapp, Kaan, Aksit, A. Steed, and T. Ritschel. Beyond blur: Real-time ventral metamers for foveated rendering. 2021. 2

[58] L. Wang, Q. Hu, Q. He, Z. Wang, J. Yu, T. Tuytelaars, L. Xu, and M. Wu. Neural residual radiance fields for streamably free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 76–87, 2023. 3

[59] L. Wang, J. Zhang, X. Liu, F. Zhao, Y. Zhang, Y. Zhang, M. Wu, L. Xu, and J. Yu. Fourier plenoctrees for dynamic radiance field rendering in real-time. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13514–13524, 2022. 3

[60] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[61] R. L. Wasserstein and N. A. Lazar. The asa statement on p-values: context, process, and purpose, 2016. 7

[62] Y. Yang, C. Xie, L. Liu, P. H. W. Leong, and S. Song. Efficient radius search for adaptive foveal sizing mechanism in collaborative foveated rendering framework. *IEEE Transactions on Mobile Computing*, 2023. 2

[63] J. Ye, A. Xie, S. Jabbireddy, Y. Li, X. Yang, and X. Meng. Rectangular mapping-based foveated rendering. *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 756–764, 2022. 2, 7

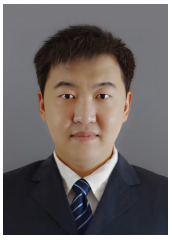
[64] J. Ye, A. Xie, S. Jabbireddy, Y. Li, X. Yang, and X. Meng. Rectangular mapping-based foveated rendering. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 756–764. IEEE, 2022. 6

[65] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5732–5741, 2021. 3

[66] Q. Zhang, S.-H. Baek, S. Rusinkiewicz, and F. Heide. Differentiable point-based radiance fields for efficient view synthesis. *SIGGRAPH Asia 2022 Conference Papers*, 2022. 3

Xuehuai Shi received his Ph.D. degree from the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. He is currently a lecturer at the School of Computer Science, Nanjing University of Posts and Telecommunications. His research interests include virtual reality, real-time rendering, and augmented reality.

Lili Wang received her Ph.D. degree from the Beihang University, Beijing, China. She is a professor with the School of Computer Science and Engineering of Beihang University, and a researcher with the State Key Laboratory of Virtual Reality Technology and Systems. Her interests include virtual reality, augmented reality, mixed reality, real-time rendering and realistic rendering.



Xinda Liu received the Ph.D. degree from the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. He is currently a lecturer in the School of Information Science and Technology at Northwest University. His research interests include virtual reality and artificial intelligence.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3429416

Jian Wu is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, Beihang University, Beijing, China. His current research focuses on virtual reality, augmented reality, and visualization.

Zhiwen Shao is currently an Associate Professor with the China University of Mining and Technology, China, as well as a Postdoctoral Fellow with the Shanghai Jiao Tong University, China. He received the B.Eng. degree and the Ph.D. degree in Computer Science and Technology from the Northwestern Polytechnical University, China and the Shanghai Jiao Tong University, China in 2015 and 2020, respectively. From 2017 to 2018, he was a joint Ph.D. student at the Multimedia and Interactive Computing Lab, Nanyang Technological University, Singapore. He has published more than 40 academic papers in popular journals and conferences. His research interests lie in computer vision and affective computing. He has been serving as a Publication Chair for CGI 2023, as well as a PC member for IJCAI and AAAI.