VRS-NeRF: Accelerating Neural Radiance Field Rendering with Variable Rate Shading

Tim Rolff* Susanne Schmidt Ke Li Frank Steinicke Simone Frintrop

Universität Hamburg, Hamburg, Germany

Figure 1: Our variable rate shading technique for neural radiance fields using eye tracking to identify high- and low-resolution parts of the rendered image for foveated rendering (right) in comparison to the corresponding fully rendered image (left).

ABSTRACT

Recent advancements in Neural Radiance Fields (NeRF) provide enormous potential for a wide range of Mixed Reality (MR) applications. However, the applicability of NeRF to real-time MR systems is still largely limited by the rendering performance of NeRF. In this paper, we present a novel approach for Variable Rate Shading for Neural Radiance Fields (VRS-NeRF). In contrast to previous techniques, our approach does not require training multiple neural networks or re-training of already existing ones, but instead utilizes the raytracing properties of NeRF. This is achieved by merging rays depending on a variable shading rate, which reduces the overall number of queries to the neural network. We demonstrate the generalizability of our approach by implementing three alternative functions for the determination of the shading rate. The first method uses the gaze of users to effectively implement a foveated rendering technique in NeRF. For the other two techniques, we utilize shading rates based on edges and saliency. Based on a psychophysical experiment and multiple image-based metrics, we suggest a set of parameters for each technique, yielding an optimal tradeoff between rendering performance gain and perceived visual quality.

Keywords: neural radiance fields, variable rate shading, virtual reality, psychophysical experiment

Index Terms: Human-centered computing—Human computer interaction (HCI)—HCI design and evaluation methods—User studies; Computing methodologies—Computer graphics—Graphics systems and interfaces—Virtual reality

1 INTRODUCTION

Driven by the ever-increasing demand for high-quality graphics rendering in immersive applications, ranging from gaming and entertainment to Extended Reality (XR) and scientific visualization, computer graphics have advanced impressively. However, the complexity of today's virtual environments and the sheer number of geometric primitives and pixels involved in rendering them still pose significant challenges to real-time performance, especially on resource-constrained devices such as mobile phones, handheld devices, or portable head-mounted displays (HMDs) [41, 59]. To address these challenges, various techniques have emerged that reduce the required rendering cost [27, 70, 71]. These range from today's common techniques to upscale image resolution, such as NVIDIA's Deep Learning Super Sampling (DLSS) [2] or AMD's FidelityFX Super Resolution (FSR) [1], to methods that exploit the inherent properties of the human visual system. One of these methods is Variable Rate Shading (VRS) [71, 72], a rendering optimization technique that allows different shading rates to be applied to different parts of the image based on their importance and structure, thus reducing the overall shading workload. Another approach is Foveated Rendering (FR). This method exploits the limited human visual acuity in the peripheral region and only displays higher levels of detail in regions displayed to the fovea, resulting in significant performance gains [27].

In addition to the advancement of techniques that reduce computational costs with little or no visual impact, there has been continued progress in the field of deep learning-based 3D visualization, particularly in the area of neural rendering techniques. Many of these techniques have already shown impressive results, providing realistic, high-quality renderings [25, 47, 61, 64]. One of the most notable examples of these techniques is Neural Radiance Fields (NeRF) [16, 44, 47], a technique that utilizes a deep learning model to represent the volumetric density and transmittance of a 3D scene from a set of 2D images and camera poses. Building on the initial idea, NVIDIA recently released Instant Neural Graphics Primitives (instant-ngp) [47], a novel implementation of NeRF that significantly reduces training time while enabling real-time rendering for small to medium render volumes. This is achieved by using a hashing algorithm and an adapted neural network architecture. In this context, the demand for a photorealistic immersive experience has also led to several existing solutions for the rendering of NeRFs in immersive virtual reality (VR) [16, 40]. Despite the performance gains with instant-ngp, rendering a NeRF model still suffers from

*tim.rolff@uni-hamburg.de

high computational costs in interactive real-time systems, especially when applied to stereoscopic VR rendering, where stable high frame rates must be maintained for a comfortable visual experience [36].

To address this challenge, we propose VRS-NeRF, a novel approach to accelerate immersive NeRF rendering that exploits the ray-tracing properties of volumetric rendering. In our paper, we integrate a variable rate shading technique that merges quadratic pixels of size $n \times n$ into larger superpixels, significantly reducing the number of neural network queries and required rendering effort. We design our algorithm in such a way that the function for calculating the shading rate is exchangeable and can thus be flexibly adapted to the use case. We evaluate the proposed VRS techniques on different scenes, both through a psychophysical study and Structural Similarity (SSIM) [66] as well as Learned Perceptual Image Patch Similarity (LPIPS) [73] metrics. Through the proposed VRS techniques, we aim to improve the performance of NeRF-based rendering without lowering its perceived visual quality, thus enabling new applications and XR use cases that require real-time and interactive rendering. To summarize, the contributions of this work include:

- A variable rate shading mechanism for neural radiance field rendering (VRS-NeRF), merging up to 16×16 adjacent pixels into one superpixel. On average, this improves the performance of NeRFs rendered through instant-ngp by 94.69%. An advantage of our method over previous work, such as [16, 56], is that it does not require re-training or multiple models.
- A variety of different algorithms for the determination of the shading rate. We integrate three lightweight methods, based on (i) the user's gaze, (ii) edge detection, and (iii) saliency prediction. VRS-NeRF is not restricted to the three methods, but they rather serve as a baseline for more advanced methods, such as adaptive shading [71].
- An evaluation of all three shading rate determination algorithms through a psychophysical user experiment and the identification of thresholds for each method such that participants are unable to identify a difference between a scene rendered with and without VRS in 50% of the cases.
- A quantitative evaluation of our methods using SSIM [66] and LPIPS [73] through perception-based image metrics.

2 Related Work

2.1 Neural Radiance Fields

With the success of deep learning in other areas such as natural language processing or computer vision, there has been a growing interest in applying deep learning techniques to 3D graphics [44, 48, 57, 64] in the areas of 3D scene synthesis [75], robot teleoperation [3, 15] or immersive journalism [14, 33]. NeRF [44] has attracted considerable attention for its ability to represent complex, volumetric scenes with high visual fidelity, while requiring only a simple neural network architecture. But, despite using a small neural network, NeRF is still computationally expensive due to the need to query the network multiple times along a ray for each pixel. The original work by Mildenhall et al. [44] already proposed a solution to reduce the computational cost by training two neural networks, a coarse and a fine network. In their original work, the coarse model is used for hierarchical sampling of the underlying volume to generate a probability density function (PDF) along the ray that are likely to contain visible content. Although this approach already reduces computational complexity, it is still limited in its applicability in high-quality real-time interactive applications.

To address this challenge, several approaches have been proposed that aim to improve the efficiency of NeRF-based rendering [4, 16, 26, 29, 47, 49, 74]. One approach is the recently published instant-ngp [47]. In their work, the authors introduce the concept of a multi-resolution hash table, which serves as an input encoder for the neural network. In contrast, Neff et al. [49] utilize a depth oracle network for the determination of sample importance along each view ray, without spacial reduction of pixels, allowing for large open scenes. The technique by Zhang et al. [74], reduces the number of cast rays and network queries by employing a quadtree as acceleration structure, significantly reducing training times. In contrast, [29] utilizes a sparse voxel grid to accelerate ray marching, whereas [26] rephrases the formulation of the neural network for efficient caching. Another notable advance is FoV-NeRF [16], a technique designed for VR and HMDs. It uses two neural networks to render the scene at different resolutions. Here, they use two seperate models to render the peripheral and foveal parts of the image, using information about the current point of fixation from a built-in eye tracker of the HMD. Moreover, other attempts have been made to incorporate NeRF into popular game engines, such as Unity [40].

2.2 Adaptive Shading & Foveated Rendering

One recent adaptive shading technique is Variable Rate Shading (VRS), which takes advantage of hardware capabilities of modern graphics processing units (GPUs). VRS describes a method to reduce the shading workload per fragment by distributing the work across tiles [63, 72]. The general principle of VRS is to reuse shading calculations across multiple fragments or pixels, avoiding redundant calculations on visually similar fragments. The underlying algorithms and techniques have been researched for more than a decade, and several approaches to dynamic shading rates have been proposed [10, 55, 63, 70, 72]. These methods either directly exploit new hardware capabilities or propose novel rendering pipelines [72]. For example, [55] and [63] propose a coarse shading technique that decouples shading and visibility computations, reducing the workload by distributing $n \times m$ shading values over pixels. Additionally, with multi-resolution shading and lens-matched shading, there have been suggestions for VR-focused approaches for adaptive shading [37]. The core idea behind lens-matched shading is to avoid rendering the scene at parts of the screen that are invisible to the user, whereas multi-resolution shading reduces the shading rate at the distorted parts of the image. Independent of the methods already mentioned, there have been several approaches for adaptive sampling that take the human visual system into account (e.g., [7, 21, 72]). For example, Bolin and Meyer [7] proposed a model based on the human visual system to compute a Just Noticeable Difference (JND) map. This map is then used to guide the selection of important samples in rendering algorithms. Later, Yang et al. [72] focused on exploiting VRS hardware support by proposing an algorithm for determining shading rates through a perceptual quality loss, achieving real-time performance compared to previous work. In addition, due to its high computational requirements, there have been several suggestions for adaptive shading in the context of ray-tracing based applications to reduce the required sampling rate [12, 18, 45, 76].

Besides adaptive shading, other techniques have been developed, such as dynamic resolution rendering [6], checkerboard rendering [13, 20, 69], or projection-based approaches [43, 62]. Projection-based approaches have become more relevant recently due to their usefulness in VR applications [62]. Another technique that has gained interest is foveated rendering (FR) [16, 27, 30, 43]. FR increases rendering performance by exploiting the limited peripheral vision of the human visual system and applying higher levels of detail only in the foveal region. In their work, Hsu et al. [30] found that most users barely notice foveated rendering at an eccentricity of 7.5°. VRS and FR have also been explored as complementary techniques that can be combined to further improve rendering efficiency and quality. For example, Palswamy and Bhonde [51] proposed a VRS-based approach that uses FR to selectively apply higher levels of detail to the central region of interest while reducing shading workload in the periphery.

3 METHOD

3.1 Variable Neural Radiance Rendering

Rendering an image through a NeRF-based approach, like [44], require approximately $res_x \times res_y \times z$ queries to the neural network for an image with resolution $res_x \times res_y$. Each ray requires sampling z points along the cast ray, with the ray starting at position $o(x,y)$ in normalized device coordinates (NDC) using the pixel coordinates x,y and the viewing direction d of the camera. Hence, the ray can be described by:

$$r_{x,y}(t) = o(x,y) + t \cdot d \tag{1}$$

The color value $C(r)$ is then calculated by evaluating the accumulated transmittance T, volume density σ, and volume color c along the ray (cf. [44,47]), assigning exactly one color value per pixel:

$$C(r_{x,y}) = \int_{t_{\text{near}}}^{t_{\text{far}}} T(t) \cdot \sigma(r_{x,y}(t)) \cdot c(r_{x,y}(t), d)\, dt \tag{2}$$

In our approach, we utilize these ray properties of NeRFs and propose Variable Rate Neural Radiance Fields (VRS-NeRF). VRS-NeRF uses the idea of previous variable rate shading strategies [63,70,72], exploiting the fact that adjacent fragments may share the processing load if a certain criterion is met. Such a criterion may be, for example, similar color values within a tile, or whether the entire tile belongs to a non-salient region. However, since the concept of fragments does not exist in NeRF rendering, we shade the output pixels directly. In fact, our implementation does not use a conventional render rasterization or ray-tracing-based pipeline. We rely solely on CUDA, without VRS hardware support or fragment shaders. For the same reason, we do not decouple visibility and shading computation, as suggested by Ragan-Kelley et al. [55]. To control the shading rate $s(\mathscr{T})$ of these pixels, we define a tile $\mathscr{T}$ as a group of $n \times n$ square pixels, each sharing a shading rate. As shown in Fig. 2, this shading rate specifies the render resolution of all pixels in a tile, with all tiles having their independent rate. For our particular implementation, we set the tile size to be 16×16, following the previous approach of Yang et al. [72] for optimal workload distribution. We explicitly choose our shading rate to be a power of two for simplicity and performance, and only allow the values $s(\mathscr{T}) \in \{1,2,4,8,16\} = \mathscr{V}$. The estimation of a new pixel size $\hat{n}$ for all pixels in a tile is then given by dividing the tile size n by the shading rate, resulting in:

$$\hat{n}(\mathscr{T}) = \frac{n}{s(\mathscr{T})} \tag{3}$$

This allows us to reduce the number of rays for an image patch from n^2 to $(n/s(\mathscr{T}))^2$. Optimally, only one ray is needed to shade the whole tile, reducing the whole tile to a single superpixel. Note that our approach is not restricted to square image patches and could be generalized to reduce pixel sizes along the x- and y-axes independently. To store each shading rate, we use a shading rate buffer. This buffer has its resolution scaled down by the tile size, resulting in $\lceil res_x/n \rceil \times \lceil res_y/n \rceil$ shading rate values. To determine the values, we apply an exchangeable, freely definable function to each tile that either directly calculates the tile's shading rate or outputs a continuous value that can be scaled to the range [0,1]. We provide three example functions in Sec. 3.2 to demonstrate the generalizability of our approach.

With the shading rate calculated for each tile, we can determine the number of rays required to render the scene. Instead of constructing a ray per pixel, we use the calculated shading rate to determine new pixel positions. To estimate the superpixel color, we set the new ray position to the center of each superpixel in the tile, resulting in a set of rays $\mathscr{R}(\mathscr{T})$:

$$\begin{aligned} \mathscr{R}(\mathscr{T}) = \{ r_{u,v} \mid\ & u = p_x(\mathscr{T}) + \ell(\mathscr{T})(x+0.5), \\ & v = p_y(\mathscr{T}) + \ell(\mathscr{T})(y+0.5), \\ & 0 \le x,y < \hat{n}(\mathscr{T}) \} \end{aligned} \tag{4}$$

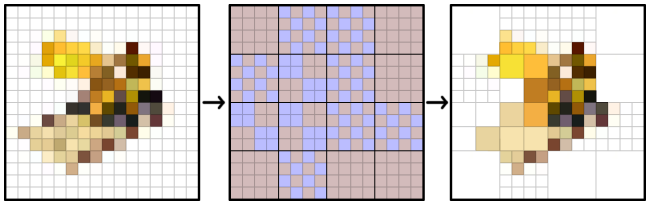

Figure 2: Schematic showing an image as it would have been rendered with all pixels (left) and the variable rate shading buffer covering multiple pixels (center) with a maximum tile size of 4×4 pixels, indicated through a black frame. The blue and red checkerboard pattern indicates the output pixel size of a tile. An example of the output rendered through our method is depicted on the right.

with $\ell(\mathscr{T}) = s(\mathscr{T})/n$ describing the size of a superpixel and $(p_x(\mathscr{T}), p_y(\mathscr{T}))$ the screen space position of a tile. This reduces the workload by spreading the work over several adjacent pixels, thus reducing the number of rays needed for the final output render. Therefore, it is only necessary to calculate $C(r)$ for all rays $r \in \mathscr{R}(\mathscr{T})$. The output color for each pixel is set depending on the color of the corresponding superpixel that covers it. Here, we use the shading rate to determine the tile size. We shade each pixel using the screen space origin of the ray at the tile's center and the tile size to set the corresponding pixel colors. With the above made adaptions to the original NeRF algorithm, our method does not require retraining the neural network as we do not change the rendering mechanism of NeRF itself. This allows us to reuse the same neural network and network weights for different VRS algorithms such that it is possible to change the VRS implementation while the application is running. To train a network for a specific scene, a loss evaluation against every pixel is still required to avoid introducing artifacts into our shading rate determination through noise.

3.2 Shading Rate Determination

To evaluate different approaches to shading rate estimation, we evaluated three different baseline algorithms, *Gaze*, *Edges* and *Saliency*. In our work, we focus on a stationary setup without movement or relative observer-scene motion. This decision was made based on previous work on classic VRS [72], which suggests that in motion, sample rates can be even more reduced without creating a perceivable loss in visual quality. It is therefore to be expected that our measured performance gains represent a lower limit. Further, the introduced algorithms should only provide a baseline showing the generalizability of our method in consideration of different applications, allowing for more sophisticated techniques in the future. The first algorithm uses the user's gaze to estimate the shading rate, effectively behaving like a foveated rendering method. The other two use the image's structural properties, with the first relying on edges and the second using a saliency prediction method to estimate salient regions of the image. In all three example cases, we output a real value that is normalized to the range $[0,1]$. Using a threshold mechanism, the real value $v(\mathscr{T})$ is mapped to a discrete shading rate $s(\mathscr{T})$. In the following, we will explain all the above methods in more detail.

Gaze: For foveated rendering, we use the gaze position captured by an eye tracker to determine the values $v(\mathscr{T})$ for each tile $\mathscr{T}$. In contrast to other approaches such as FoV-NeRF [16], we are able to render at multiple shading rates between 1 and 16. Instead of specifying different radii for the foveal and peripheral regions, we normalize the gaze position (g_x, g_y) by dividing it by the maximum

possible distance $d_{\max}$, which is the distance from the center to the corner of an image. Hence, we calculate $v(\mathscr{T})$ through:

$$v_{\text{gaze}}(\mathscr{T}) = \sqrt{\frac{(p_x(\mathscr{T}) - g_x/n)^2 + (p_y(\mathscr{T}) - g_y/n)^2}{(res_x/(2n))^2 + (res_y/(2n))^2}} \tag{5}$$

with $n = 16$ being the tile size. To estimate the shading rate $s(\mathscr{T})$ of a tile $\mathscr{T}$, we use the normalized distance $v_{\text{gaze}}(\mathscr{T})$ and thresholds $t_0, t_1, t_2, t_3 \in [0,1]$:

$$s_{\text{gaze}}(\mathscr{T}) = \begin{cases} 1 & \text{if} \quad 0 \le v_{\text{gaze}}(\mathscr{T}) < t_0 \cdot d_{\max} \\ 2 & \text{if } t_0 \cdot d_{\max} \le v_{\text{gaze}}(\mathscr{T}) < t_1 \cdot d_{\max} \\ 4 & \text{if } t_1 \cdot d_{\max} \le v_{\text{gaze}}(\mathscr{T}) < t_2 \cdot d_{\max} \\ 8 & \text{if } t_2 \cdot d_{\max} \le v_{\text{gaze}}(\mathscr{T}) < t_3 \cdot d_{\max} \\ 16 & \text{if } t_3 \cdot d_{\max} \le v_{\text{gaze}}(\mathscr{T}) \le 1 \end{cases} \tag{6}$$

As we have an increased number of thresholds compared to previous work [16], we need to re-estimate the optimal thresholds. We explain this process in detail in Sec. 4.

Edges: Edges have been shown to be an influential factor in the perception of an image [28, 50], as they are the most probable indicators of discontinuity in surface orientation, range, reflectance, or illumination [5]. Therefore, details can be preserved by rendering a more accurate representation of edges using high-resolution tiles, while low-frequency parts of the render can be combined into larger superpixels [72]. To achieve this, we first compute the luminance values of all pixels from the previous render and then apply an edge detector to each pixel to estimate high frequencies. In our case, we use a Sobel edge detector [34, 60] on the last frame to compute the image gradient I_x, I_y along the x-axis and the y-axis. Using both gradients, we then estimate the magnitude of the gradient by:

$$I = \sqrt{I_x^2 + I_y^2} \tag{7}$$

and compute the average differential values $v(\mathscr{T})$ over the whole 16×16 tile, via:

$$v_{\text{edges}}(\mathscr{T}) = \frac{1}{16^2} \sum_{x=0}^{15} \sum_{y=0}^{15} I(p_x(\mathscr{T}) + x, p_y(\mathscr{T}) + y) \tag{8}$$

Using the average magnitude of the image gradient, we calculate the maximum and minimum differential values and normalize the values across all tiles, using the following formula:

$$\hat{v}_{\text{edges}}(\mathscr{T}) = \frac{v_{\text{edges}}(\mathscr{T}) - \min\limits_{\mathscr{T}} v_{\text{edges}}(\mathscr{T})}{\max\limits_{\mathscr{T}} v_{\text{edges}}(\mathscr{T}) - \min\limits_{\mathscr{T}} v_{\text{edges}}(\mathscr{T})} \tag{9}$$

Using the normalized shading values for the edges, we then compute the shading rate s_{edges} from $\hat{v}_{\text{edges}}$ by using a different set of determined thresholds, similar to Eq. 6.

$$s_{\text{edges}}(\mathscr{T}) = \begin{cases} 16 & \text{if} \quad 0 \le \hat{v}_{\text{edges}}(\mathscr{T}) < t_0 \cdot d_{\max} \\ 8 & \text{if } t_0 \cdot d_{\max} \le \hat{v}_{\text{edges}}(\mathscr{T}) < t_1 \cdot d_{\max} \\ 4 & \text{if } t_1 \cdot d_{\max} \le \hat{v}_{\text{edges}}(\mathscr{T}) < t_2 \cdot d_{\max} \\ 2 & \text{if } t_2 \cdot d_{\max} \le \hat{v}_{\text{edges}}(\mathscr{T}) < t_3 \cdot d_{\max} \\ 1 & \text{if } t_3 \cdot d_{\max} \le \hat{v}_{\text{edges}}(\mathscr{T}) \le 1 \end{cases} \tag{10}$$

Note that we invert the shading rate, as edge magnitudes close to zero likely indicate low luminance variance of the tile.

Saliency: As a visual attention-based approach, we utilize saliency maps. These capture the regions of potential interest [53] and are often approximated through saliency models [8, 11, 32, 35, 39, 42, 52]. Current state-of-the-art models use deep learning techniques to predict human saliency. Usually, these are directly trained on datasets of human fixation positions captured by an eye tracker. However, they are often computationally expensive or require additional GPU resources that are already being used to render the neural radiance field.

Another approach explicitly designed for real-time prediction of saliency maps has been proposed by Katramados and Breckon [35], building on earlier work of Itti et al. [31,32]. They propose replacing the center-surround filters with Division of Gaussian (DIVoG) filters to improve the performance of the prediction. The algorithm generates a Gaussian pyramid from the input image U, down-sampling the image by half in each operation, and then applies a 5×5 Gaussian filter. Then the reverse operation is performed, up-sampling the image and again applying a Gaussian filter after each up-sampling step, with the last image D with the same resolution as the input. The saliency map S is then computed by the element-wise division at each pixel position (x, y) of the input image [35]:

$$S_{x,y} = 1 - \min\left(\frac{D_{x,y}}{U_{x,y}}, \frac{U_{x,y}}{D_{x,y}}\right) \tag{11}$$

Even though the saliency map is already in the desired range of 0 to 1, we still normalize it to avoid low-resolution images if the whole image is not salient. Here, we apply Eq. 5 to the saliency map and perform the thresholding operation (cf. Eq. 10), using a different set of thresholds. Before the full resolution rendering with saliency-based VRS, we render a pre-view image of the scene with the same resolution as the VRS buffer. This acts as a guidance for the saliency computation and avoids temporal artifacts due to temporal shifting of salient spots that occur because of different resolutions, allowing for interactive usage. Further, the low-resolution image of the pre-view results in a focus on larger objects in the scene, as the saliency predictor already allows small input resolutions.

4 Psychophysical Experiment

We performed a psychophysical experiment for two reasons. First, we aim to evaluate the practicality of our VRS implementation using the three exemplary methods mentioned above. Second, we want to derive for each of these methods a set of parameter values that yield an optimal trade-off between imperceptibility and computational performance gain, such that no perceived loss of quality occurs independent of the user. Here, the user analysis is a necessary step of a two-step process, of 1) identifying a parameter set for which no perceived loss of quality occurs, and 2) finding the pair of parameters within this set that maximize performance. Therefore, we presented participants with pairs of images, each consisting of two renderings (i) with and (ii) without one of the three proposed VRS methods, in a two-alternative forced-choice (2AFC) task.

4.1 Datasets & Visual Stimuli

To cover a wide range of applications, we chose our scenes to include (i) interior rooms, (ii) dioramas and (iii) single objects from three different datasets, shown in the supplementary. For the interior rooms, we used the FoV-NeRF dataset [16], containing four different settings. For the dioramas, we took one scene from the instant-ngp [47] and the synthetic NeRF dataset [44]. We define dioramas to show objects in a semi-open scene that is accessible from at least one side. Finally, we selected two singular objects from the synthetic NeRF dataset [44]. These are defined to show only the object itself, without any additional scenery. We included these synthetic scenes because they are commonly used when comparing and benchmarking NeRF approaches and provide a use case similar to an object viewer.

4.2 Target Parameters

As introduced in the previous section, VRS-NeRF can be applied to any method that assigns a value between 0 and 1 to each pixel of the rendered image. The three exemplary methods we implemented and tested in this experiment are all inspired by human perceptual processes and have in common that they map values $v(\mathscr{T})$ to all possible shading rates $s_i \in \mathscr{V}$ using thresholds $t_0, t_1, t_2, t_3 \in [0,1]$. To reduce the required number of parameters to be estimated down to two, we calculate these thresholds through an exponential function with method-dependent parameters c and k':

$$t_i = c \cdot e^{-k' \cdot (s_i - 1)} \tag{12}$$

For *Gaze*, the exponential relationship is due to the non-uniform distribution of cone photoreceptors on the retina, resulting in an exponential fall-off of the visual acuity starting from its maximum at the fovea [19]. For *Edges* and *Saliency*, we apply the Weber-Fechner law [22] as suggested by Yang et al. [72]. We compute exponential thresholds through Eq. 12 to avoid computing the logarithm of $\hat{v}$ due to lower computational complexity.

The goal of our psychophysical experiment was to determine values for c and k' that represent a good estimate for the specific display scenario, averaged across users, determining good choices for c and k' once per method. This is a necessary pre-step to determine possible c' and k' values where the majority of users will not notice the visual impact. Afterward, it is then possible to optimize in regard to a performance metric. To cover a wide range of possible values, we investigated a set of 9 values for the parameter space of c and additional 9 values for k'. To determine the range, we compared the SSIM scores (cf. Sec. 5.1) of generated images with the SSIM threshold determined by Flynn et al. [23]. In their work, they determined an SSIM threshold of 0.95 for image quality of 2D images through the JND paradigm such that 50% of participants were unable to tell the difference between the compressed and uncompressed images. We assumed that their findings for 2D images can be extrapolated to stereo images in VR, and therefore utilized their identified threshold as a guideline for the generation of our parameter range. Hence, for the range of c, we selected parameters to be equal to:

$$c \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\} \tag{13}$$

For the estimation of the range of k', we computed constant values through the following equation:

$$k' = \frac{\log(10^k \cdot c)}{16 - 1} \tag{14}$$

resulting in exponentially increasing distances between SSIM curves, as shown in Fig. 4. Here, we observed that a parameter range of

$$k \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\} \tag{15}$$

gave the best tradeoff between the number of parameters, the investigated SSIM scores, and duration required for performing the experiment (cf. Sec. 3).

4.3 Materials

The psychophysical experiment was conducted on a Meta Quest Pro with an integrated eye tracker. To implement our VRS approach, we used instant-ngp [47] as our starting implementation and adapted it as described in Sec. 3. However, we did not directly perform our psychophysical experiment through instant-ngp, as some evaluated values of our parameter space would have required frame timings that are too high for the participants to feel comfortable in performing the experiment, potentially inducing simulator sickness [36]. Therefore, we opted to use Unity[1], a 3D rendering and game engine, as our platform for the psychophysical experiment. We pre-rendered the images generated from our parameter space and displayed them in our Unity application as if they would have been directly rendered through instant-ngp. Since we focussed on evaluating and comparing the image quality for two identical viewpoints, we did not provide a possibility for changing the view to the NeRF scene. The stimuli shown in the study were displayed as screen space quads fully enclosing the view, resulting in a fixed stereoscopic viewpoint, displaying the scene. For the generation of these images, we used a fixed IPD, equal to 68 mm.

[1] https://unity.com/

4.4 Methods

We followed a within-subjects design, i.e., all participants performed 2AFC tasks with each of the three methods and all parameter values described in Section 3.2 and 4.2. To improve generalization across stimuli, we applied stimulus sampling [46, 67] with respect to the presented scenes (8 options) and viewpoints (2 options per scene), thus varying these aspects across participants and treating them as random factors in the analysis. Two combinations from the resulting pool of 16 scene/viewpoint stimuli were randomly assigned to every participant for each method and each pair of parameter values. With 3 methods (*Gaze*, *Edges*, *Saliency*) $\times$ 9 parameter values for $c \times 9$ parameter values for $k \times 2$ repetitions with random scene/viewpoint, this results in a total of 486 trials per participant.

4.5 Procedure & Participants

Before the main experiment, we conducted a preliminary user study to estimate salient regions in our dataset. For the main experiment, each participant was asked for their informed consent and filled out a demographic questionnaire. We invited 13 participants to the main experiment, 5 female and 8 male (aged between 18 and 64, with an average age of 40.38 ± 12.73 years). A screening question was asked for each participant prior to the experiment to ensure that all participants had a normal or corrected-to-normal vision. All participants were encouraged to read through the instructions, but were re-explained the task and materials regardless afterward. During the experiment, 486 configurations were displayed to the user (cf. Sec. 4.4) with each trial consisting of 5 phases. *Phase (1)*: A black screen with a red fixation cross was shown that required the user's eyes to fixate on the cross for 500 ms (without deviating more than 5°). *Phase (2):* A scene rendering for 500 ms, with or without VRS. *Phase (3):* A black screen for 500 ms. *Phase (4):* Display of the opposite condition of the previously displayed scene from Phase 2 for 500 ms (e.g., with VRS if the first display was without VRS). *Phase (5):* The final screen presented the 2AFC with 2 buttons labelled 'First' and 'Second' with a dwell time of 400 ms. In total, each study session took approximately 60 minutes. Further details on the entire process can be found in the supplementary.

4.6 Bayesian Analysis

The aim of our psychophysical experiment was to calculate the just noticeable difference (JND) for each of the three VRS-NeRF methods. In a 2AFC task, the two stimuli (in our case rendering S_{VRS_on} with and S_{VRS_off} without VRS) are subjectively equal when 50% of the responses favor S_{VRS_on} and 50% favor S_{VRS_off} (the equivalent of random guessing). Based on this notion, the JND is usually defined as the point halfway between random guessing and 100% preference of one of the stimuli. In our case, this corresponds to a 75% probability of participants choosing S_{VRS_off}, thus experiment participants noticed a difference in visual quality between stimuli in half of the cases. The probability ψ of choosing S_{VRS_off} given the stimulus parameters c and k is defined by

$$\psi(c, k; \theta, \gamma, \lambda) = \gamma + (1 - \gamma - \lambda) \cdot F(c, k; \theta) \tag{16}$$

with F being the underlying psychometric function [68]. γ and (1-λ) define the lower and upper bound of ψ, respectively, with γ

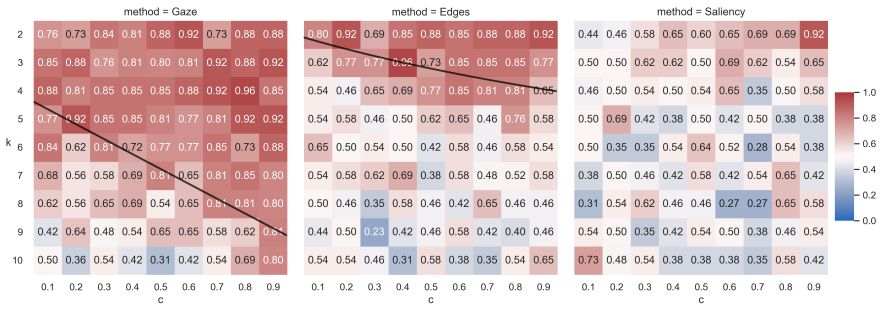

Figure 3: Pooled 2AFC responses for all combinations of the stimulus parameters c and k. The color represents the proportion of responses S_{VRS_off}, thus red corresponds to a preference of S_{VRS_off}, blue to a preference of S_{VRS_on}, and white to the guess level of 0.5. The black curves for methods *Gaze* and *Edges* represent the estimated JND, i.e., $F = 0.5$. Note that the individual cells show the raw probabilities measured in the 2AFC before fitting a psychometric function that takes into account the miss rate λ. Raw values can therefore be greater than 0.75 even though they are below the JND curve.

being the guess rate (fixed at 0.5 level) and λ being the miss rate (to be estimated). To account for observers lapsing, i.e., responding incorrectly regardless of the stimuli (e.g., due to fatigue effects), the JND is usually not assumed at the point $\psi = 0.75$ but at the slightly shifted midpoint of the unscaled psychometric function $F = 0.5$.

Since the two stimulus parameters c and k were manipulated in parallel, calculation of the JND was performed on a two-dimensional psychometric surface, with c and k on the x and z-axes, respectively, and the probability on the y-axis. As suggested by DiMattina [17], we chose the 2D psychometric function F to be sigmoidal, considering not only contributions of each individual parameter c and k but also their interaction:

$$F(c,k;\theta) = \sigma(\theta_0 + \theta_1 \cdot c + \theta_2 \cdot k + \theta_{12} \cdot c \cdot k) \qquad (17)$$

This results in five parameters per method, i.e., λ, θ_0, θ_1, θ_2, and θ_{12}. We estimated the parameters using Markov Chain Monte Carlo sampling with a No U-Turn Sampler (NUTS) with 4 chains, 2000 tune and 1000 draw iterations. For the priors, we assumed the standard normal distribution for θ_i ($i \in \{0,1,2,12\}$) and the beta distribution for λ. The 2AFC response was modeled as a Bernoulli variable with the success probability given by ψ.

4.7 Results

Estimation of the psychometric function using Bayesian inference yielded the following values (c,k) for the JND for *Gaze* (Eq. 18) and *Edges* (Eq. 19):

$$0.5 = \sigma(1.4048 + 1.8257 \cdot c - 3.4203 \cdot k/10 - 0.0216 \cdot c \cdot k/10) \qquad (18)$$

$$0.5 = \sigma(0.8957 + 1.7424 \cdot c - 4.5764 \cdot k/10 - 1.8652 \cdot c \cdot k/10) \qquad (19)$$

The divisor 10 was used to perform Bayesian inference with a value range of $[0,1]$ for both c and k.

For *Saliency*, the 2AFC responses could not be adequately modeled by the psychometric function F because (a) the probability of choosing S_{VRS_off} did not continuously fall for decreasing stimulus parameter values c and k, and (b) 40% of the parameter combinations yielded probabilities that were below the assumed guess rate of 0.5 (see Fig. 3, right). The latter corresponds to a preference for rendered images with VRS (i.e., partly reduced resolution) over images with the full resolution, and is therefore not reflected in our psychometric function F. An interpretation of these results will be provided in Section 6.

5 System Evaluation

In our system evaluation, we analyzed our proposed VRS-NeRF using two image-based metrics, SSIM and LPIPS, that account for human perception. Using the results of our psychophysical experiment, we analyzed the frame timings and performance improvements of our system. For the analysis, we chose the same scenes as in Sec. 4.1. Peak signal-to-noise ratio (PNSR) values can be found in the supplementary.

Table 1: Average frame timings on all rendering and shading methods at a resolution of 1800×1920 (left) and 1600×1440 (right) pixels on the FoV-NeRF [16], instant-ngp [47] and the NeRF [44] datasets. Lower values preferred and best timings in **bold**.

Enclosing scenes	Instant-npg w/o DLSS	*Gaze*	*Edges*	Instant-npg w/o DLSS	*Gaze*	*Edges*
Barbershop	375.43 ms	**17.20 ms**	22.31 ms	247.23 ms	**12.56 ms**	15.91 ms
Classroom	388.12 ms	**17.06 ms**	21.28 ms	257.25 ms	**12.49 ms**	14.95 ms
Lobby	684.47 ms	**19.73 ms**	33.64 ms	457.16 ms	**14.01 ms**	24.51 ms
Stones	517.59 ms	**17.75 ms**	20.86 ms	345.52 ms	**12.91 ms**	14.58 ms
Total	491.40 ms	**17.94 ms**	24.52 ms	326.79 ms	**12.99 ms**	17.49 ms
Diorama scenes						
Fox	343.73 ms	**17.27 ms**	27.53 ms	230.64 ms	**12.75 ms**	19.76 ms
Ship	202.60 ms	**17.02 ms**	21.22 ms	135.26 ms	**12.67 ms**	15.30 ms
Total	273.16 ms	**17.15 ms**	24.37 ms	182.95 ms	**12.71 ms**	17.53 ms
Single objects						
Chair	41.61 ms	**15.00 ms**	18.80 ms	28.79 ms	**10.83 ms**	13.75 ms
Lego	52.79 ms	**15.16 ms**	18.74 ms	35.38 ms	**11.00 ms**	13.82 ms
Total	47.20 ms	**15.08 ms**	18.77 ms	32.38 ms	**10.91 ms**	13.79 ms
Across all scenes						
Total	325.77 ms	**17.50 ms**	23.46 ms	217.15 ms	**12.40 ms**	16.57 ms

5.1 Metrics

The following section briefly describes the SSIM and LPIPS metrics used in our systematic analysis.

Structural Similarity: SSIM is a perception-based metric that measures the structural similarity of images [65, 66]. In their work, Wang et al. [66] define structural similarity as parts of images containing structural information about their content, representing the structure of an object in the scene. Furthermore, SSIM considers additional information by comparing three different properties of the input signal: the luminance, contrast, and structure of an image. When applying the metric, two images are more structurally similar if the output of the metric is close to one.

Learned Perceptual Image Patch Similarity: LPIPS is a recent metric that uses features extracted from deep learning models such as VGG [58] or AlexNet [38]. In their work, Zhang et al. [73] observed that deep learning features extracted from pre-trained models tend to mimic human perception. To estimate the similarity of two images, both images are fed to a neural network and their respective features are extracted and passed to another network for final regression. For the final prediction, a value closer to zero indicates more similar images.

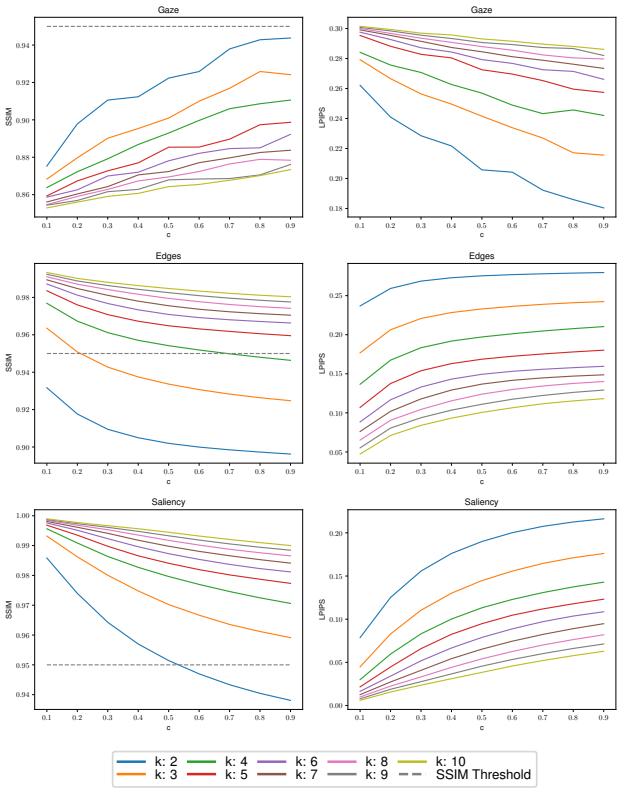

Figure 4: Graphs showing the SSIM (left) and LPIPS (right) curves of our images generated using the parameters chosen for our psychophysical experiment. We utilize these curves to determine the parameter space for our evaluation (cf. Sec. 4.2).

5.2 Results

We determined SSIM and LPIPS scores for images generated through the three tested VRS methods and plotted the results against the perceptual threshold reported by Flynn et al. [23] (see Fig. 4). From the images mentioned in Sec. 4, we computed the SSIM and LPIPS values by comparing the VRS-NeRF generated output with the reference rendered through instant-ngp. In addition, we computed the Pearson correlation [24] between the SSIM scores and the JND functions listed in Sec. 4.7 (cf. Eq. 19 and 18). This revealed a strong correlation between the reported SSIM scores and JDN both for *Edges* ($r = -0.9557$) and *Gaze* ($r = 0.8969$). An additional investigation of our parameter space (cf. Sec. 4.2) was performed to validate that the parameter combinations that were empirically determined through the JND boundary have a SSIM score close to the perceptual threshold of 0.95. As expected, this was not the case for *Gaze*, as this method highly degrades the images at the peripheral regions, therefore strongly influencing the SSIM score.

While the obtained LPIPS scores can serve as a basis for comparison with FoV-NeRF [16], we would like to explicitly note that we did not compare with FoV-NeRF directly due to its egocentric coordinates, that require all camera poses to be inside a sphere with the size of the near depth, which is not the case for all of our data samples. For the frame timings listed in Tab. 1, we rendered 1000 frames of each scene and changed the camera position to a different pose of the training set after 10 frames. We determined the viewpoints by iterating through the camera positions of the dataset. To get an overview of the performance on the Quest Pro and the HTC Vive Pro, we measured the timings at the per-eye resolutions of both devices, i.e. 1800×1920 (Quest Pro) and 1600×1440 (HTC Vive Pro). To

Table 2: Average frame timings (*FT*) and parameters from our JND analysis across all scenes at a resolution of 1800×1920 with the parameter values determined through the JND functions for *Edges* (Eq. 19), *Gaze* (Eq. 18), and *Saliency* (cf. Sec. 5.2) using the FoV-NeRF [16], instant-ngp [47] and the NeRF [44] datasets. Lower values preferred and best timings in **bold**.

instant-ngp	*Gaze*			*Edges*			*Saliency*		
avg. *FT*	c	k	avg. *FT*	c	k	avg. *FT*	c	k	avg. *FT*
325.77 ms	0.1	5	**16.49** ms	0.1	3	28.71 ms	0.5	2	32.76 ms
325.77 ms	0.2	6	16.65 ms	0.2	3	22.93 ms	0.9	3	**32.75 ms**
325.77 ms	0.3	6	17.06 ms	0.3	3	21.08 ms			
325.77 ms	0.4	7	16.98 ms	0.4	3	**20.42 ms**			
325.77 ms	0.5	7	17.27 ms	0.5	4	23.59 ms			
325.77 ms	0.6	8	17.08 ms	0.6	4	23.15 ms			
325.77 ms	0.7	8	17.27 ms	0.7	4	22.77 ms			
325.77 ms	0.8	9	17.11 ms	0.8	4	22.49 ms			
325.77 ms	0.9	9	17.31 ms	0.9	4	22.28 ms			

Table 3: Average number of tiles at a resolution of 1800×1920 on the entire dataset. For the full details, see the supplementary.

Method	Tile size 1	Tile size 2	Tile size 4	Tile size 8	Tile size 16
Instant-ngp	3456000	0	0	0	0
Gaze	256	569	5988	11886	10204
Edges	29539	21749	32990	24748	4772
Saliency	54680	70896	67852	21332	2621

select c and k used to determine our shading rates, we followed the analysis explained in Sec. 4. We chose the parameters closest but below our JND curves in Fig. 3 and computed the average across all frame timings. To render all images, we used the same neural network and network weights trained through instant-ngp and just varied our VRS implementation for direct comparability between the frame timings. This approach revealed an average performance improvement of 94.69%. Further investigation showed that optimal frame timings were achieved for the parameters $k = 3, c = 0.4$ for *Edges* and $k = 5, c = 0.1$ for *Gaze* (cf. Tab. 2). As *Saliency* could not be adequately modeled through the suggested psychometric function, we computed the potential performance gain using the perceptual SSIM threshold. The parameters $k = 2, c = 0.5$ and $k = 3, c = 0.9$ selected on this basis yielded frame timings of 32.76 ms and 32.75 ms, respectively, for a resolution of 1800×1920, with the saliency estimation taking 3.05 ms. Further investigating the origin of the system performance, we evaluated the number of tile sizes and rays across the dataset. Even without our method, the ray count is highly dependent on content, as instant-ngp already disables rays [47]. When rendering a fully enclosed scene at 1800×1920 it usually requires $3,456,000$ rays. Using our VRS method, we measured an average reduction by 99.16% (28,903 rays) for *Gaze*, 96,71% (113,798 rays) for *Edges*, and 93,71% (217,381 rays) for *Saliency*. This closely correlates with the performance increase of 94.6% for Gaze, 92.8% for Edges, and 89.9% for Saliency, indicating a correlation in runtime and number of rays. As shown in Tab. 3, we found that approximately 25.96% of the image is rendered in full detail and 21.75% of pixels belonged to a tile size of 8, when using *Edges* (cf. the supplementary for more details). We performed our evaluation on a RTX 3090 and an Intel i9-12900KF workstation with 64 GB of RAM.

6 Discussion and Limitations

For *Gaze* and *Edges* VRS, parameterized by the values closest to but below the perceptual threshold as determined in our psychophysical experiment, we found a significant average performance gain of 94.1%. In Tab. 2, we provide additional details on the average frame timings over all scenes when observed for a single parameter configuration. With these initial positive results on performance gain, and based on the proposed perceptually inspired mapping of gaze and edge values to shading rates, further studies can follow to investigate

(a) No VRS method applied (b) *Gaze* with $c = 0.1$, $k = 5$

(c) *Edges* with $c = 0.4$, $k = 3$ (d) *Saliency* with $c = 0.9$, $k = 3$

Figure 5: Rendering of the barbershop scene with overlaid shading rate using all three different approaches with parameters chosen from Tab. 2. We encourage the reader to zoom into the images. Here, a more reddish tone corresponds to a high shading rate of 1×1 pixels, whereas a blueish tone refers to tiles that are rendered 16×16 pixels. More examples can be found in the supplementary.

alternative mapping functions. For such follow-up studies, a suitable parameter space could be based not only on our results but also on the SSIM score. The strong correlation between the SSIM scores and our determined JDN functions (cf. Sec. 5.2) supports our initial hypothesis that the perceptual SSIM threshold of 0.95, which was determined by Flynn et al. [23], translates into VR applications. The results of our psychophysical experiment do not indicate a noticeable visual quality loss due to saliency-based VRS of the rendered NeRF scenes within the tested parameter value range (with one exception for the extreme values $c = 0.9$ and $k = 2$). While the measured parameter value range was not sufficient for a meaningful fitting of a psychometric sigmoid function, the empirical results are consistent with those of the SSIM benchmark, according to which the structural similarity of saliency-based VRS images for almost all parameter combinations is above a perceptual SSIM threshold of 0.95. These results warrant further investigation, as there is also a high potential for performance gain (e.g., 89.9% for parameter values $k = 2, c = 0.5$ and $k = 3, c = 0.9$, which yielded both a probability of less than 75% in the psychometric experiment and an SSIM value > 0.95). Moreover, frequent values below the guess rate of 0.5 in our psychometric experiment support the interpretation that saliency-based reduction of resolution may actually lead to an improvement in perceived visual quality, possibly by reducing visual noise [54]. This could lead to an even better performance, but would require the collection of more empirical data that would allow the fitting of a different psychometric curve.

As mentioned in Sec. 5.2, we deliberately did not compare against the ground truth used to train the neural network for two reasons. First, we do not aim to produce more accurate NeRF representations of scenes than previous approaches, but rather to provide an approach for improving the rendering performance of NeRF. Second, due to the design of our experiment, the original polygonal renderings were never shown to the participants, meaning that the control image known to the participants is the NeRF rendering from instant-ngp. Compared to previous VRS approaches, we also do not decouple visibility from shading. This results in visually perceivable pixel blocks that might even be noticeable in the peripheral region. One participant noted that it was possible to determine the quality of some images depending on the "roundness" of objects in the peripheral region. Further work on decoupling the shape of objects and their shading might be a promising direction to reduce these visual artifacts.

7 Conclusion and Future Work

In this paper, we proposed VRS-NeRF, a novel variable rate shading algorithm for neural radiance fields rendering. Our method builds on the idea of reducing queries to the neural network by merging multiple rays into one to compute superpixel tiles. These tiles allow us to share workload between multiple pixels to reduce the overall rendering time required to display NeRF scenes. Furthermore, we provided three different algorithms for the estimation of shading rates, which we evaluated through a psychophysical experiment. Using the thresholds determined in the experiment, we performed a system analysis computing the frame timings for renderings generated with VRS-NeRF. Here, we were able to improve the rendering time by 94.1% compared to instant-ngp.

In terms of future work, we believe that our approach can be further improved by considering the shape of the objects, similar to [55], and by allowing non-square superpixels to be processed. In addition, a more in-depth investigation into saliency as a predictor of shading rate might be needed, as it shows promising results that need to be further investigated. Future studies are also needed to clarify whether the results can be applied to dynamic scenarios where the user can explore a scene, potentially focusing on non-salient points. At these non-salient points, a merging of rays is performed, but due to the approach of the saliency predictor, the presence of fine structures is less likely, so that a merging may not be visually noticeable. We would also like to emphasize that there may be aliasing artifacts due to the reduced sampling. However, we believe a more in-depth analysis for prolonged exploration of virtual scenes, including relative movements between the user's head and the (potentially dynamic) objects in the scene to be a worthwhile research direction. Here, previous work has also shown that blur applied uniformly during head motion, while being perceivable, may be more suitable for VR as it significantly reduces induced motion sickness [9]. While an evaluation in terms of perceptibility and induced motion sickness is beyond the scope of this paper, we recommend a thorough investigation of the optimal transitions between our techniques for (nearly) stationary sequences and common motion blur strategies for dynamic sequences. Additional future extensions to the method may incorporate other factors to determine a variable shading rate. Inspired by work on classic rendering of dynamic scenes, such as Motion Adaptive Shading [72], rays could be merged for moving scene objects and/or when the user moves their head. However, we also believe more research is needed to address potential NeRF noise interference when using traditional VRS techniques for shading rate determination. For rendering in VR headsets, the specific lens optics and resulting image distortions could be considered to reduce the shading rate at the usually highly compressed image edges or to even stop rays completely if corresponding pixels would be discarded during the rendering pipeline anyway [37]. Compared to classical rendering, NeRF opens up another dimension for reduction, namely the number of samples along each individual ray. While VRS adjusts the local resolution of the rendered image, ray sampling would affect the color accuracy of individual pixels and thus could be used complementary to VRS. Finally, VRS-NeRF is not limited to immersive environments and, due to the flexibility of the shading rate determination, does not depend on the availability of an eye tracker. In the future, it could therefore also be investigated for common 2D applications.

REFERENCES

[1] AMD FidelityFX Super Resolution. https://gpuopen.com/fidelityfx-superresolution/. Accessed: 2023-08-10.

[2] NVIDIA DLSS Research. https://developer.nvidia.com/dlss/research. Accessed: 2023-08-10.

[3] L. Angrisani, P. Arpaia, D. Gatti, A. Masi, and M. D. Castro. Augmented reality monitoring of robot-assisted intervention in harsh environments at cern. *Journal of Physics: Conference Series*, 1065, 2018.

[4] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.

[5] H. G. Barrow and J. M. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial intelligence*, 17(1-3):75–116, 1981.

[6] D. Binks. Dynamic resolution rendering. In *Game Developers Conference (GDC)*, 2011.

[7] M. R. Bolin and G. W. Meyer. A perceptually based adaptive sampling algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 299–309, 1998.

[8] A. Borji. Saliency prediction in the deep learning era: Successes and limitations. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):679–700, 2019.

[9] P. Budhiraja, M. R. Miller, A. K. Modi, and D. Forsyth. Rotation blurring: use of artificial blurring to reduce cybersickness in virtual reality first person shooters. *arXiv preprint arXiv:1710.02599*, 2017.

[10] P. Clarberg, R. Toth, J. Hasselgren, J. Nilsson, and T. Akenine-Möller. Amfs: adaptive multi-frequency shading for future graphics processors. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.

[11] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, 2018.

[12] A. Dahlin and V. Sundstedt. Improving Ray Tracing Performance with Variable Rate Shading. In K. Xu and M. Turner, eds., *Computer Graphics and Visual Computing (CGVC)*. The Eurographics Association, 2021.

[13] G. De Carpentier and K. Ishiyama. Decima engine: Advances in lighting and aa. *ACM SIGGRAPH Courses: Advances in Real-Time Rendering in Games*, 3(8):11, 2017.

[14] N. de la Peña, P. Weil, J. Llobera, E. Giannopoulos, A. Pomés, B. Spanlang, D. Friedman, M. V. Sanchez-Vives, and M. Slater. Immersive journalism: Immersive virtual reality for the first-person experience of news. *PRESENCE: Teleoperators and Virtual Environments*, 19:291–301, 2010.

[15] A. Dehne, T. Hermes, N. Moeller, and R. Bacher. Marwin: A mobile autonomous robot for maintenance and inspection. In *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17)*, pp. 76–80, 2017.

[16] 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.

[17] C. DiMattina. Fast adaptive estimation of multidimensional psychometric functions. *Journal of Vision*, 15(9):5–5, 2015.

[18] P. Djeu, W. Hunt, R. Wang, I. Elhassan, G. Stoll, and W. R. Mark. Razor: An architecture for dynamic multiresolution ray tracing. *ACM Transactions on Graphics (TOG)*, 30(5):1–26, 2011.

[19] A. T. Duchowski. *Eye tracking methodology: Theory and practice*. Springer, 2017.

[20] J. E. El Mansouri. Rendering 'rainbow six — siege'. In *Game Developers Conference (GDC)*, p. 82, 2016.

[21] J.-P. Farrugia and B. Péroche. A progressive rendering algorithm using an adaptive perceptually based image metric. In *Computer Graphics Forum*, vol. 23, pp. 605–614. Wiley Online Library, 2004.

[22] G. T. Fechner, D. H. Howes, and E. G. Boring. *Elements of psychophysics*, vol. 1. Holt, Rinehart and Winston New York, 1966.

[23] J. R. Flynn, S. Ward, J. Abich, and D. Poole. Image quality assessment using the ssim and the just noticeable difference paradigm. In *Engineering Psychology and Cognitive Ergonomics. Understanding Human Cognition: 10th International Conference, EPCE 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I 10*, pp. 23–30. Springer, 2013.

[24] D. Freedman, R. Pisani, and R. Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

[25] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5501–5510, 2022.

[26] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14346–14355, 2021.

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

[28] T. Hansen and K. R. Gegenfurtner. Independence of color and luminance edges in natural scenes. *Visual neuroscience*, 26(1):35–49, 2009.

[29] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec. Baking neural radiance fields for real-time view synthesis. *ICCV*, 2021.

[30] C.-F. Hsu, A. Chen, C.-H. Hsu, C.-Y. Huang, C.-L. Lei, and K.-T. Chen. Is foveated rendering perceivable in virtual reality? exploring the efficiency and consistency of quality assessment methods. In *Proceedings of the 25th ACM international conference on multimedia*, pp. 55–63, 2017.

[31] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 40(10-12):1489–1506, 2000.

[32] L. Itti and C. Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

[33] S. Jones. Disrupting the narrative: immersive journalism in virtual reality. *Journal of Media Practice*, 18:171 – 185, 2017.

[34] N. Kanopoulos, N. Vasanthavada, and R. L. Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988.

[35] I. Katramados and T. P. Breckon. Real-time visual saliency by division of gaussians. In *2011 18th IEEE International Conference on Image Processing*, pp. 1701–1704. IEEE, 2011.

[36] E. M. Kolasinski. *Simulator sickness in virtual environments*. US Army Research Institute for the Behavioral and Social Sciences, 1995.

[37] M. Kraemer. Accelerating your vr games with vrworks. In *NVIDIAs GPU Technology Conference (GTC)*, 2018.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012.

[39] M. Kümmerer, L. Theis, and M. Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. In *International Conference on Learning Representations (ICLR 2015)*, 2014.

[40] K. Li, T. Rolff, S. Schmidt, R. Bacher, S. Frintrop, W. Leemans, and F. Steinicke. Immersive neural graphics primitives. *arXiv preprint arXiv:2211.13494*, 2022.

[41] K. Li, S. Schmidt, R. Bacher, W. P. Leemans, and F. Steinicke. Mixed reality tunneling effects for stereoscopic untethered video-see-through head-mounted displays. *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 44–53, 2022.

[42] A. Linardos, M. Kümmerer, O. Press, and M. Bethge. Deepgaze iie: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12919–12928, 2021.

[43] 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.

[44] 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.

[45] D. P. Mitchell. Generating antialiased images at low sampling densities. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 65–72, 1987.

[46] B. Monin and D. M. Oppenheimer. The limits of direct replications and the virtues of stimulus sampling. *Social Psychology*, 2014.

[47] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), 2022.

[48] T. Müller, F. Rousselle, J. Novák, and A. Keller. Real-time neural radiance caching for path tracing. *ACM Transactions on Graphics*, 40(4):36:1–36:16, 2021.

[49] 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. In *Computer Graphics Forum*, vol. 40, pp. 45–59. Wiley Online Library, 2021.

[50] P. Neri. Global properties of natural scenes shape local properties of human edge detectors. *Frontiers in Psychology*, 2:172, 2011.

[51] D. Palswamy and S. Bhonde. Nvidia vrss, a zero-effort way to improve your vr image quality, 2020. Accessed: 2023-03-24.

[52] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.

[53] H. E. Pashler. *The psychology of attention*. MIT Press, 1998.

[54] N. Ponomarenko, S. Krivenko, V. Lukin, K. Egiazarian, and J. T. Astola. Lossy compression of noisy images based on visual quality: a comprehensive study. *EURASIP Journal on Advances in Signal Processing*, 2010:1–13, 2010.

[55] J. Ragan-Kelley, J. Lehtinen, J. Chen, M. Doggett, and F. Durand. Decoupled sampling for graphics pipelines. *ACM Transactions on Graphics (TOG)*, 30(3):1–17, 2011.

[56] C. Reiser, S. Peng, Y. Liao, and A. Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14335–14345, 2021.

[57] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021.

[58] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*, pp. 1–14. Computational and Biological Learning Society, 2015.

[59] M. Smelyanskiy, D. R. Holmes, J. Chhugani, A. Larson, D. M. Carmean, D. P. Hanson, P. K. Dubey, K. E. Augustine, D. Kim, A. Kyker, V. W. Lee, A. D. Nguyen, L. Seiler, and R. A. Robb. Mapping high-fidelity volume rendering for medical imaging to cpu, gpu and many-core architectures. *IEEE Transactions on Visualization and Computer Graphics*, 15, 2009.

[60] I. E. Sobel. *Camera models and machine perception*. Stanford University, 1970.

[61] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Nießner, J. T. Barron, G. Wetzstein, M. Zollhöfer, and V. Golyanik. Advances in Neural Rendering. *Computer Graphics Forum (EG STAR 2022)*, 2022.

[62] R. Toth, J. Nilsson, and T. Akenine-Möller. Comparison of projection methods for rendering virtual reality. In *High Performance Graphics*, pp. 163–171, 2016.

[63] K. Vaidyanathan, M. Salvi, R. Toth, T. Foley, T. Akenine-Möller, J. Nilsson, J. Munkberg, J. Hasselgren, M. Sugihara, P. Clarberg, et al. Coarse pixel shading. In *Proceedings of High Performance Graphics*, HPG '14, pp. 9–18. Eurographics Association, 2014.

[64] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4688–4697, 2021.

[65] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.

[66] 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.

[67] G. L. Wells and P. D. Windschitl. Stimulus sampling and social psychological experimentation. *Personality and Social Psychology Bulletin*, 25(9):1115–1125, 1999.

[68] F. A. Wichmann and N. J. Hill. The psychometric function: I. fitting, sampling, and goodness of fit. *Perception & psychophysics*, 63(8):1293–1313, 2001.

[69] G. Wihlidal. 4k checkerboard in battlefield 1 and mass effect andromeda. In *Game Developers Conference (GDC)*, 2017.

[70] K. Xiao, G. Liktor, and K. Vaidyanathan. Coarse pixel shading with temporal supersampling. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 1–7, 2018.

[71] L. Yang, D. Zhdan, and M. Johnson. Nvidia adaptive shading overview. In *Game Developers Conference (GDC)*, 2019.

[72] L. Yang, D. Zhdan, E. Kilgariff, E. B. Lum, Y. Zhang, M. Johnson, and H. Rydgård. Visually lossless content and motion adaptive shading in games. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2(1):1–19, 2019.

[73] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

[74] W. Zhang, R. Xing, Y. Zeng, Y.-S. Liu, K. Shi, and Z. Han. Fast learning radiance fields by shooting much fewer rays. *IEEE Transactions on Image Processing*, 2023.

[75] S. Zollmann, C. Hoppe, S. Kluckner, C. Poglitsch, H. Bischof, and G. Reitmayr. Augmented reality for construction site monitoring and documentation. *Proceedings of the IEEE*, 102:137–154, 2014.

[76] M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon. Recent advances in adaptive sampling and reconstruction for monte carlo rendering. In *Computer graphics forum*, vol. 34, pp. 667–681. Wiley Online Library, 2015.