# Foveated Depth-of-Field Filtering in Head-Mounted Displays

MARTIN WEIER, Hochschule Bonn-Rhein-Sieg and Saarland University
THORSTEN ROTH, Hochschule Bonn-Rhein-Sieg and Brunel University
ANDRÉ HINKENJANN, Hochschule Bonn-Rhein-Sieg
PHILIPP SLUSALLEK, Saarland University and DFKI Saarbrücken

In recent years, a variety of methods have been introduced to exploit the decrease in visual acuity of peripheral vision, known as foveated rendering. As more and more computationally involved shading is requested and display resolutions increase, maintaining low latencies is challenging when rendering in a virtual reality context. Here, foveated rendering is a promising approach for reducing the number of shaded samples. However, besides the reduction of the visual acuity, the eye is an optical system, filtering radiance through lenses. The lenses create depth-of-field (DoF) effects when accommodated to objects at varying distances. The central idea of this article is to exploit these effects as a filtering method to conceal rendering artifacts. To showcase the potential of such filters, we present a foveated rendering system, tightly integrated with a gaze-contingent DoF filter. Besides presenting benchmarks of the DoF and rendering pipeline, we carried out a perceptual study, showing that rendering quality is rated almost on par with full rendering when using DoF in our foveated mode, while shaded samples are reduced by more than 69%.



CCS Concepts: • **Computing methodologies** → **Rendering**; **Ray tracing**; **Perception**; **Virtual reality**; *Antialiasing*; Image processing;

Additional Key Words and Phrases: Gaze-contingent depth-of-field, foveated rendering, ray tracing, eye-tracking

**ACM Reference format:**
Martin Weier, Thorsten Roth, André Hinkenjann, and Philipp Slusallek. 2018. Foveated Depth-of-Field Filtering in Head-Mounted Displays. *ACM Trans. Appl. Percept.* 15, 4, Article 26 (September 2018), 14 pages.
https://doi.org/10.1145/3238301

## 1 INTRODUCTION

Over the last few years, advancements in display and tracking technologies have led to the introduction of a wide range of Head-mounted Display (HMDs), opening up virtual reality (VR) to the consumer market. One of the key challenges when rendering to HMDs is maintaining low latencies, which are crucial for increased presence and reduced fatigue. Also, with the dramatic increase of pixel densities over the last two decades,

Authors' addresses: M. Weier, IVC, Hochschule Bonn-Rhein-Sieg, Grantham-Allee 20, 53757 Sankt Augustin, Germany; email: martin.weier@h-brs.de; T. Roth, IVC, Hochschule Bonn-Rhein-Sieg, Grantham-Allee 20, 53757 Sankt Augustin, Germany; email: thorsten.roth@h-brs.de; A. Hinkenjann, IVC, Hochschule Bonn-Rhein-Sieg, Grantham-Allee 20, 53757 Sankt Augustin, Germany; email: andre.hinkenjann@h-brs.de; P. Slusallek, DFKI GmbH, Saarland University, Campus D3 2, 66123 Saarbrücken, Germany; email: philipp.slusallek@dfki.de.

1544-3558/2018/09-ART26 $15.00
https://doi.org/10.1145/3238301

HMDs exceeding 18 megapixels are going to be available in the near future. Low-latency, high-quality rendering at such resolutions is still beyond the reach of current and foreseeable hardware and software solutions. Hence, several gaze-contingent rendering methods have been introduced that exploit the limitations of the human visual system (HVS) by adapting rendering quality to the user's retinal capabilities (foveated rendering) to reduce the computational workload. This is made possible by the availability of accurate eye-tracking devices and their integration into HMDs. Besides the retina's decreasing visual acuity, the HVS is also limited by optical effects. Among the most prominent ones is depth-of-field (DoF), occurring when focusing objects. Two adaptations of the eyes drive this process: The vergence movement, which is the rotation of the eyeballs for fusing a focused object into a single percept, and accommodation, describing the process of adjusting the optical power of the lens to create a sharp retinal image of the fixated object. Accommodation changes the focus distance to the point of regard (PoR). However, it is usually not possible to perceive all observed objects as one sharp image. While objects at the focused distance from the eye are perceived clearly, other objects appear increasingly blurry depending on their depth difference to the focused distance. This also reduces the required number of samples for such out-of-focus image areas. The goal of this article is to exploit the knowledge of the DoF while rendering in a foveated fashion. Foveated rendering is prone to artifacts, where the most prominent ones arise due to the eccentricity-based resolution reduction, also leading to temporal instabilities, which the HVS is highly sensitive to (Patney et al. 2016; Weier et al. 2017). We apply DoF as a post-processing step to conceal such artifacts and to remove high-frequency signals from the visual periphery by the inherent blur of the DoF. This allows for putting more computational effort into more important regions that are either in the central visual area or generally in focus. Using our method, DoF computation can be tightly coupled to the image reconstruction to improve the perceived visual quality of foveated rendering. In summary, this work contains the following contributions:

— A gaze-contingent rendering system with a tightly integrated DoF filter concealing potential visual artifacts
— A DoF model, incorporating knowledge about tracking inaccuracies when obtaining 3D gaze points
— A user study, evaluation, and benchmarks showing the potential of the proposed method

## 2 RELATED WORK

In the following section, we provide an overview of the main fields of research related to our system: foveated rendering, DoF simulation, and possible perceptual implications.

Recently, many methods have been introduced that try to optimize rendering techniques in a gaze-contingent manner, exploiting the eye's visual acuity limitations. A survey of various techniques is presented by Weier et al. (2017). The method presented in the work by Guenter et al. (2012) exploits the HVS's acuity fall-off by rendering three nested layers of decreasing resolutions and increasing diameters around the PoR as obtained by the eye-tracker. Stengel et al. (2016) and Patney et al. (2016) present foveated multi-rate shading approaches for GPU rasterized graphics. Both Weier et al. (2016) and Fujita and Harada (2014) use fast ray tracers to render gaze-contingent sampling patterns. Though the presented renderer is based on a GPU-accelerated ray-tracing core for maximum flexibility when scheduling pixels for sampling, it can easily be adapted to gaze-contingent rasterization pipelines.

Various techniques have been introduced that allow for simulating DoF. A survey on general rendering methods has been provided by Barsky and Kosloff (2008), while the surveys by Demers (2004) and McIntosh et al. (2012) put their focus on post-processing techniques to compute DoF in image space. When using DoF in a gaze-contingent manner, the method's performance is the critical factor—especially with the goal of using it for filtering rendering artifacts. Hence, for the introduced approach, methods with a low latency rather than a high visual quality were investigated. The presented methods most closely resemble the works by Bukowski et al. (2013) and Mulder and van Liere (2000). We use different filtering approaches for different parts of the visual field. Most similar to our proposed method is the work by Lindeberg (2016). The author uses DoF to conceal

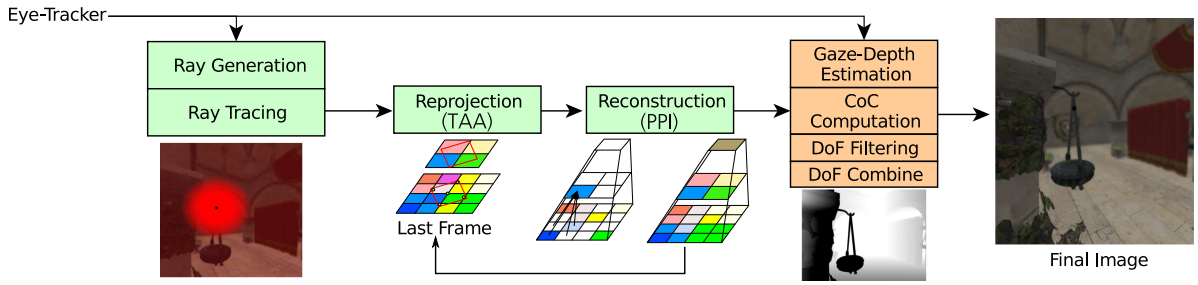



Fig. 1. Rendering pipeline of our approach. First, ray tracing is used to sample the image plane sparsely based on a visual acuity model. Next, reprojection is used to increase the temporal stability of the sparse samples. Afterward, a reconstruction kernel reconstructs a dense image from the sparse samples. Finally, depth-of-field is computed to conceal artifacts in the final image.

artifacts arising from gaze-contingent geometric simplifications utilizing a tessellation shader in the Unreal Engine. In contrast, we use such an approach to hide (under-)sampling artifacts in image space and investigate the effect on perception in a user study.

While the perceptual implications of rendering gaze-contingent DoF is still an active research topic, there has already been evidence for a positive influence on immersion and depth perception. In a study involving dynamic DoF in first-person shooters (Hillaire et al. 2008), half of the participants favored activating DoF rendering. Mantiuk et al. (2011) investigated gaze-contingent DoF using eye-tracking. In their study, participants mostly reported a more natural feeling when using the gaze-contingent blur effect. However, the success of DoF rendering strongly depends on tracking accuracy. Mantiuk et al. (2011) use only the depth at the PoR as obtained by the tracker to steer their DoF renderer. Due to the limited spatial accuracy of eye-trackers, this proved inaccurate and led to flickering, resulting from sudden depth changes. In a follow-up article by Mantiuk et al. (2013), they thus improved tracking accuracy and stability for DoF rendering by applying object and scene knowledge. A study by Mauderer et al. (2014) showed that gaze-contingent DoF did increase the perceived realism and resulted in better discrimination of object ordering. Therefore, the effect can likely improve relative depth estimation, while the accuracy of distance prediction is still limited. Duchowski et al. (2014) and Vinnikov et al. (2016) found evidence that gaze-contingent DoF may decrease the perceived image quality and increase fatigue in VR setups. However, they point out that this is likely happening due to a high system latency and imprecise depth estimation. Koulieris et al. (2017) showed that gaze-contingent DoF does not drive the actual accommodation reflexes and thus does not reduce mismatches like the vergence-accommodation conflict, known to increase visual discomfort and fatigue. Nonetheless, they also show that new generations of multifocal displays will likely improve on that aspect.

## 3 METHOD

To conceal perceptually disturbing artifacts, we combine foveated rendering with an approach to compute gaze-contingent DoF in image space to filter the final image. The entire rendering pipeline is depicted in Figure 1. First, a ray-tracing step samples the image sparsely, according to a visual acuity model. Afterward, temporal stability of peripheral image regions is improved by using reprojection-based temporal anti-aliasing (TAA). Next, the full image is reconstructed using pull-push interpolation (PPI) (Marroquim et al. 2007; Stengel et al. 2016). This approach provides a high degree of flexibility, allowing for use of arbitrary acuity models and (re-)sampling image regions based on image saliency (Stengel et al. 2016; Weier et al. 2016). Finally, to further improve the perceived image quality, gaze-contingent DoF is computed in a post-processing step. A more detailed description of each pipeline stage is presented in the following sections.

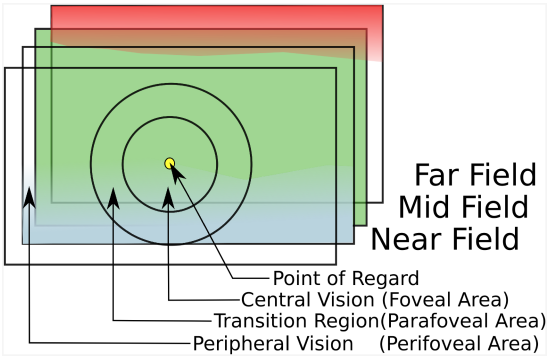



Fig. 2. Layered blur to compute the depth-of-field. For peripheral vision, a blur using a mipmap representation is computed. For central vision, a separable Gauss is used to blur the values for each layer. The transition region is blended between the blur approaches.

### 3.1 Foveated Rendering

At the core of the approach is a foveated rendering system using ray casting. To accelerate ray-geometry intersection, the ray caster makes use of irregular grids as presented by Pérard-Gayot et al. (2017), allowing for exceptionally high performance for coherent rays. All components are implemented using NVIDIA CUDA.

*Ray Generation and Tracing.* To render an image, rays are generated using a visual acuity model. Various methods and sampling patterns have been used to simulate the loss of acuity in the visual field (Weier et al. 2017). We use a model with a linear falloff in a transitional region between the area of central and peripheral vision (see Figure 2). The sampling points for this model are precomputed and stored as two binary lookup tables, $S_{in}$ and $S_{out}$. A ray is only cast for pixels with the associated bits set to one. For central vision and the transitional region, $S_{in}$ is used to mark the foveal and transitional sampling points. The addressing of this pattern is adapted based on the PoR obtained by the eye tracker to move the pattern. For peripheral vision, $S_{out}$ is referenced. This lookup table stores a static pattern to minimize temporal inconsistencies and flickering. The pattern contained in $S_{out}$ corresponds to a uniform distribution with a sampling probability of $p_{min}$. For $S_{in}$, there are two regions, limited by angular thresholds $r_0$ and $r_1$, $r_0 < r_1$. Angular distances $d < r_0$ are sampled with a probability of 1, while samples within the transitional region ($r_0 \leq d < r_1$) are generated with an importance sampling approach in polar coordinates:

$$r = r_0 + (r_1 - r_0)\frac{\sqrt{(p_{min}{}^2 - 1) * u + 1} - 1}{p_{min} - 1}, \tag{1}$$

$$\phi = 2\pi v. \tag{2}$$

Here, $u, v \in [0, 1]$ are random variables computed using a low discrepancy sampling scheme. The fractional part of the function for generating samples for $r$ (Equation (1)) is derived by applying the inversion method to $f(x) = 1 - (x \cdot (1 - p_{min}))$ with $\int_0^1 cf(x)dx = 1$. These values are then transformed to the range $[r_0, r_1]$. At runtime, a ray generation kernel looks up which pixels to sample by querying either $S_{in}$ or $S_{out}$ according to the pixel's distance to the current PoR. Eventually, CUDA threads are launched to compute pixel values for all generated rays.

*Reprojection and Reconstruction.* The ray-tracing process results in a sparse image with larger gaps between pixels for increasing eccentricities. These sparse regions result in the image brightness being vastly different from the fully sampled image. Besides, these sparsely sampled peripheral regions lead to temporal instabilities the eye is highly sensitive to Patney et al. (2016) and Weier et al. (2017). To counteract such artifacts and increase temporal

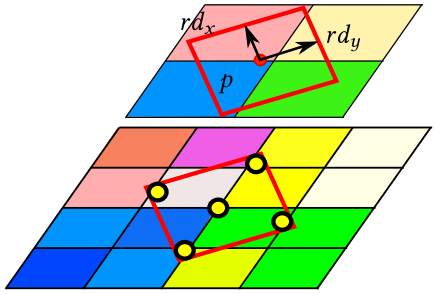



Fig. 3. Sampling of the pixel footprint ($rd_x$, $rd_y$) at the ray's hitpoint $p$ in a higher level of the mipmap using a Quincunx pattern (yellow dots).

stability, it is beneficial to re-use information from the previous frame by performing TAA. We introduce an approach that samples each active ray's reprojected pixel footprint of the previous frame. The pixel footprint described by each ray and its differentials (Igehy 1999) is adapted based on the eccentricity-dependent sampling probabilities, transformed into world space and reprojected to the previous frame. The color information of the old frame can then be evaluated by sampling the reprojected pixel's footprint extent. As described below, the old frame's information is not only a single image, but an image pyramid, similar to a mipmap with depths stored in the alpha channel. This allows for eliminating cache values in case of perspective-related occlusions (Yang et al. 2009). To further improve the precision of the reprojected colors and depths, five samples are evaluated in a Quincunx pattern for a higher mipmap level as illustrated in Figure 3. Now a running estimate combines the colors from the new and the old frame. The weights of the new and old pixel values in the estimate are adapted based on a user-defined minimum and the depth difference of each old Quincunx sample to the newly computed sample. Eventually, the temporally smoothed samples are used to reconstruct a new image using PPI (Marroquim et al. 2007). In a first phase, valid samples are pulled and combined upwards, level by level, to fill a mipmap pyramid. Afterward, the mipmap pyramid is processed from the coarsest to the finest level. Missing pixels are filled in by sampling the pixel value at the overlying coarser image. The mipmap pyramid becomes the input for the reprojection phase of the next frame and is also used as input for the gaze-contingent DoF filter presented in the next section.

### 3.2 Gaze-Contingent Depth-of-field

After the reconstruction phase, the image can be filtered using DoF to conceal potential artifacts. To model DoF, the focused gaze-depth has to be known. Based on this information a focal length and the circles of confusion (CoCs) can be computed. Ultimately, a layered blur filter inspired by the work of Mulder and van Liere (2000) and Bukowski et al. (2013) is used to compute the final image.

*Gaze-Depth Estimation.* As the focused depth will likely belong to a point on the surface of a fixated object, it is feasible to use a regular eye-tracker to derive a gaze depth. In an HMD using binocular eye-tracking, several approaches can be used to estimate the depth of the fixated object. Foremost, the depth can be obtained by sampling the scene's depth at the PoR. However, this is challenging due to the eye-tracker's lack of precision. If the measured PoR is only slightly different to the real PoR, the depth measure can be arbitrarily inaccurate. To improve depth estimates, we make use of our previously published work by training a support vector machine (SVM) with various depth measurements (Weier et al. 2018). These measurements include information about the eye's vergence and spatial depths at and around the PoR and depth variances as illustrated in Figure 4. At runtime, these measurements are used as input to the SVM to obtain the focused depth.

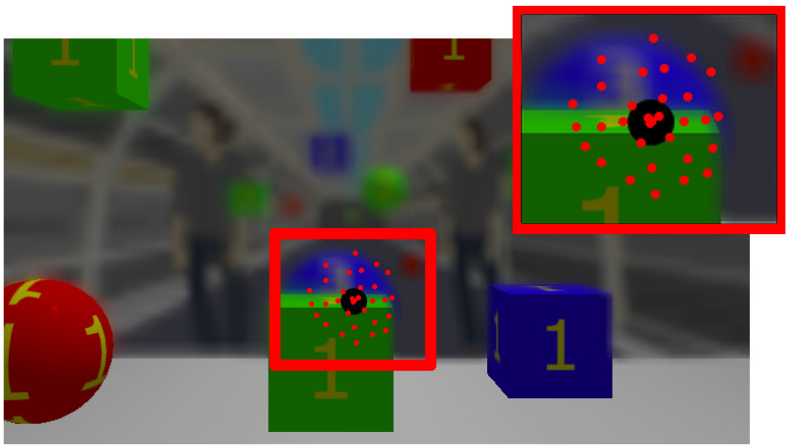


Fig. 4. Sampling pattern around the point of regard used to estimate the gaze depth. Depth samples (red dots) are drawn at the point of regard (black dot).

*Tracking-Aware Circles of Confusion Computation.* Though combining multiple measures in a single regression model improves gaze depth estimation, it is still suffering from inaccuracies (Weier et al. 2018). One way to tackle these is to employ temporal filters like higher-order Butterworth filters (Duchowski et al. 2011). In addition, we propose a conservative model that regards potential tracking inaccuracies by extending the focus range based on the accuracy of the depth estimate. The conservative model assumes a thin lens that can be expressed using the general lens equation $1/f = 1/d_o + 1/d_i$. We assume $d_i$, the distance to the image plane to be fixed with 22.4mm (Gross 2005, chap. 36.4). The object distance $d_o$ is obtained using the SVM-based gaze depth estimator. Using the mean depth estimation inaccuracy $t$ (Weier et al. 2018) to adapt $d_o$ as $d_n = d_o - t$ and $d_f = d_o + t$ to solve the general lens equation, two focal lengths $f_n$ and $f_f$ are computed. Increasing $t$ renders a larger part of the scene in focus. Finally, these values are used to compute the CoC as

$$CoC = \begin{cases} (V_d(f_n, g) - V_f(f_n, d_n)) \cdot (k \cdot V_d(f_n, g)) \cdot E & \text{if } g < d_n \\ (V_d(f_f, g) - V_f(f_f, d_f)) \cdot (k \cdot V_d(f_f, g)) \cdot E & \text{if } g > d_f \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

with $V_d(F, G) = (F \cdot G)/(G - F)$, $G > P$ and $V(F, D) = (F \cdot D)/(D - F)$, $D > F$. The distance to the unfocused object is denoted with $g$, $E$ is a measure of the retinal resolution, and $k$ is the pupil's diameter. Both, $E$ and $k$ can be adapted to control the intensity of the DoF effect. To compute the CoC, a CUDA kernel is launched that calculates a signed CoC value for each pixel. The sign marks if a pixel is located in the near or the far field. Finally, each thread stores the color information and the signed CoC's size encoded in the alpha channel in a new buffer.

*DoF Filtering.* As gaze-contingent rendering for HMDs requires low latencies to cope with fast eye movements and to match the displays' refresh rates (Albert et al. 2017), performance is a critical aspect when computing gaze-contingent DoF. Approaches that actually sample a lens model to compute physically correct DoF require casting a lot of rays, which is contradictory to the idea of foveated rendering to reduce the number of shaded samples. Though various more efficient techniques have been developed to approximate DoF (Demers 2004; McIntosh et al. 2012), they are still hard to use inside a performance critical rendering pipeline as even a simple adaptive Gaussian blur at the native HMD resolution can take several milliseconds. Therefore, we introduced a novel multi-layered filter with a single separable Gaussian blur that is used only for the foveal region, whereas a low-quality blur based on the mipmap pyramid generated in the *reconstruction phase* is used for those parts that

are located in the peripheral vision field. Similar to Bukowski et al. (2013), we process the image in layers that are eventually blended with different weights to obtain the final image (see Figure 2). For these layers, three different depth ranges are considered: Pixels in the far field, the focused mid field, and the near field. For a plausible DoF effect, the blur has to respect depth discontinuities. Blurry distant objects should not bleed over closer objects in the focused field. However, as long as the order of the layers is preserved, users are likely to tolerate potential blurring inconsistencies within each layer (Bukowski et al. 2013). To get highly blurry regions and to stay within the frame-time budget, Bukowski et al. (2013) work with buffers at a reduced resolution. However, the reduced resolution can be identified inside the HMD. To overcome this issue for highly blurry regions and the visual periphery, which already has a low visual acuity, we make use of the mipmap pyramid from the reconstruction. The blur kernel processes the image as follows: In the first pass of the filter, pixels are read horizontally. If a pixel is in a peripheral vision region, the mipmap is sampled. The radius of the CoC is used to select the appropriate mipmap level. As this can lead to artifacts, we decided to improve the blur by sampling the mipmap multiple times using the same Quincunx scheme as already used in the *reprojection phase* (see Figure 3). Based on the sign of the CoC, the pixel value is either stored in the near-field or far-field layer. To blend between the near and focused mid layer an additional coverage value is computed based on the size of the CoC and stored in the alpha channel. If the pixel is located in the central or transition region between central and peripheral vision, it is blurred using a Gaussian blur for the far field, and a blur for the near field, either computed using the mipmap or a using a simple box blur. For the latter, we rely on the coverage value computation for the near-field buffer in the horizontal pass as described by Bukowski et al. (2013). Again the coverage is stored in the alpha channel. Please note that the transition between the region of peripheral vision and the central/transitional region still might be visible in the final image. To overcome this issue, we blend both types of blur in the vertical pass of the separable filter as follows. Pixels that reside in peripheral vision are already blurred and are left untouched. Pixels residing in the central and transitional region are blurred in the vertical direction using a Gauss filter. However, for pixels that reside in the transitional region, the blur resulting from sampling the mipmap is also computed. This allows blending between the mipmap and the high-quality Gaussian blur in the near- and far-field buffer in a single pass. Having calculated two different blurred buffers, they can be combined to a final image.

*DoF Combine.* Finally, the different image layers can be blended. If foveated rendering is active, the quality degradation for peripheral vision can be counteracted by contrast enhancement (Patney et al. 2016). This enhancement is achieved by weighting each pixel's color with a blurred version of its surrounding with a kernel width adapted to the eccentricity. To get a high-quality blur, we again sample the mipmap multiple times. Combining the image is performed in a similar fashion as presented by Bukowski et al. (2013). First pixels are interpolated between the original unblurred pixels and the far-field buffer based on the CoC. Near-field values are blended over the image using alpha blending with the associated coverage information. Eventually, the combined image is presented to the user. A comparison of different rendering configurations using DoF for the presented foveated renderer in contrast to full rendering is shown in Figure 5. Please note how enabling DoF filters potential rendering artifacts.

## 4 EVALUATION

To evaluate the presented approach, we used a Fove 0 Headset,[1] natively equipped with a binocular eye-tracker running at up to 120Hz with a precision of 1° and a latency of 14ms. The benchmarks were performed on Windows using an Intel Core i7-3820 machine clocked at 3.6GHz equipped with 16GB RAM and two NVIDIA GeForce GTX 1080 Ti graphics cards with 11GB VRAM each. For the benchmarks, we modified the scene *space shooting range*.[2] The scene consists of a long tunnel extended by the different targets as illustrated in Figure 6. Our version

[1]Fove, Inc., Yuka Kojima. Retrieved March 29, 2018 from https://www.getfove.com.
[2]RossDaBoss, Space Shooting Range. Retrieved March 29, 2018 from https://3dwarehouse.sketchup.com/model/ffa37b029610b99185c07ed7ba9ce363/space-shooting-range.


Fig. 5. Different rendering configurations, showcasing the potential of the depth-of-field filter.

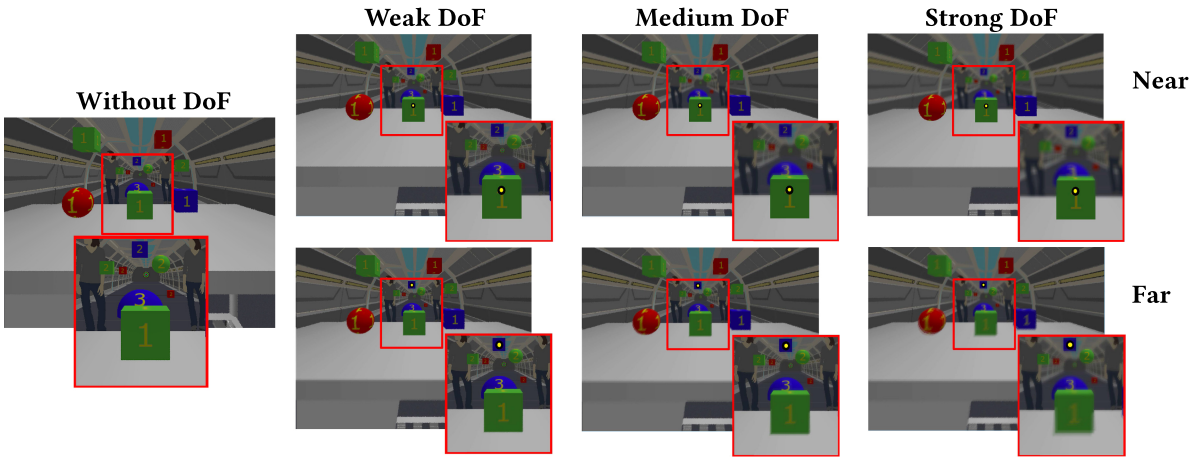



Fig. 6. The final renderings as presented in the gaze-contingent renderer. The images show the test scene that was used in the user experiment. The PoR is marked as a black dot with a yellow center, either fixating a target close to the user (near) or in the distance (far). Moreover, this image shows the different DoF modes as used in the user study. The scene itself consists of various targets at different depths. Targets labeled one range from 0.5m to 1m. Targets labeled with a two range from 1m to 6m. The big blue ball labeled three is located at a distance of 6m.

of the scene consists of 227,568 triangles. For foveated rendering, the inaccuracy of eye-tracking can be compensated by using a larger foveal area or by predicting saccadic movements (Arabadzhiyska et al. 2017; Stengel et al. 2016). However, determining the eye's focused depth to control the DoF can be inaccurate. Fortunately, the use of DoF is supported by the rather slow speed of accommodation (Temme and Morris 1989). Hence, the update rate of the focused depth is usually not critical when using eye tracking devices. A simple solution for counteracting inaccuracies is to render more of the scene in focus. However, in this case, filtering quality might be influenced negatively. Still, the proposed method to compute the CoC enables us to compensate for inaccurate

Table 1. Benchmarks of the Presented Pipeline in ms for a Single Eye Rendered at a Resolution 1280 × 1440 Averaged over 1,000 Frames

| Mode | Ray Tracing | Reprojection (TAA) | Reconstruction (PPI) | DoF | | | Total | # Samples |
|---|---|---|---|---|---|---|---|---|
| | | | | CoC Computation | Filter | Combine | | |
| **Foveated (Ours)** | 3.09 | 1.35 | 1.107 | 0.81 | 0.88 | 0.41 | 7.64 | **558945** |
| **Full Ray Tracing** | 5.52 | - | 1.09 | 0.81 | 0.88 | 0.38 | 8.68 | **1843200** |

Our approach reduces the number of shaded samples by 69% compared to full ray tracing.

depth estimates. As presented in Weier et al. (2018), we assume the mean depth estimation inaccuracy $t$ to be 0.2m over the entire critical depth range of 6m.

## 4.1 Benchmarks

In this section, we present benchmarks of our renderer. Note that the runtime of the kernel that performs the reprojection, reconstruction, and the DoF effect is independent of the scene's geometric complexity; only samples in image space are processed. The runtime of each pipeline stage is shown in Table 1. The scene was rendered for a single eye at the HMD's native resolution of 1280 × 1440. Runtimes were averaged over 1,000 frames with the same foveated configuration as for the user study. The region of central vision is specified with a radius of $r_0 = 10°$. The transition region is between $r_0 = 10°$ and $r_1 = 20°$. Samples in the peripheral vision are selected with a probability of $p = 0.2$. DoF was computed with the *medium* setting as illustrated in Figure 6. Although the total performance increase of about 1ms does not seem much, the scene was rendered using only primary rays and simple shading. No secondary contributions like shadows, ambient occlusion or even global illumination were computed. With an increasing computational complexity of each shaded sample, the difference between foveated and full rendering is expected to be much higher. Thus, the most important measure is the difference in the number of rendered samples between foveated and full rendering. This reduction is quite substantial, with the number of samples being reduced by 69%.

Below, we compare the given numbers to the state-of-the-art. Guenter et al. (2012) rasterize the image in three layers with different resolutions and render only 7% of the pixels. The image is strongly undersampled. Acceptable visual quality is achieved with hand-tuned anti-aliasing methods, limiting its applicability. Stengel et al. (2016) report that shaded pixels are decreased by 65% for the same resolution of 1280 × 1440. Patney et al. (2016), relying on Coarse Pixel Shading (Vaidyanathan et al. 2014), do not reduce the visibility rate (pixel writes) but the number of shading computations on the shaded quads to about 50% compared to the work by Guenter et al. (2012). Weier et al. (2016) report a reduction of shaded samples by 79%. In contrast to Coarse Pixel Shading (Vaidyanathan et al. 2014), our use of PPI provides high flexibility with a reasonable cost. Our approach using ray casting has a runtime of 7.64ms. To stay within the V-Sync limits of the HMD, two render threads were launched on two GPUs. Besides rendering most time is spent on reprojecting information from previous frames. As the visual quality of our approach in the foveal region mostly resembles the work by Bukowski et al. (2013) we achieve identical quality there. However, quality is reduced for parts in the visual peripheral as the high-quality Gaussian blur is replaced by the box blur from samples of the mipmap pyramid. Nevertheless, as the latter is sampled multiple times to compute a final color and coverage information, differences are hardly noticeable—especially, due to the general acuity loss at increasing eccentricities. Interestingly, in our setup, the difference in runtime between the various DoF settings is rather minimal. Once reprojection and reconstruction have been performed, the total time to compute the DoF (CoC, Filter, Combine) with *weak*, *medium,* and *strong* settings amounts to 2.08ms. We account the almost constant runtime of the DoF filter to the heavy usage of the mipmap pyramid in the visual periphery, making the amount of blurriness largely independent of the runtime.

However, slightly worse runtimes are to be expected if the focused region in the foveal and parafoveal region contains a higher amount of objects not in focus.

### 4.2 Perceptual Study

In this section, we present the results of a user study to evaluate the perceptual quality and implications of the presented gaze-contingent DoF framework.

*Procedure.* The user study consisted of three parts. First, users were asked to put on the HMD and do the spatial eye-tracking calibration provided by the Fove SDK. Following, the calibration of the gaze-depth estimation was performed. To collect training data for the SVM, users were asked to fixate, focus on and follow a tracking target as it is moving through the scene. More information is provided in Weier et al. (2018). After the SVM has been trained the main part of the study started. It was conducted as a within-subject study, employing a $4 \times 3 \times 2$ full factorial design with four DoF settings, three focus modes, and two rendering modes. Trials were generated with two repetitions and were randomly shuffled resulting in a total of 48 trials per participant. A single camera position was chosen for all trials using the scene shown in Figure 6. The scene was presented with four different DoF settings (DoFMode = NONE, WEAK, MEDIUM, or STRONG) (see Figure 6) and was designed to contain multiple labeled targets (spheres and boxes). Targets within a range of 0.5m to 1m were labeled with 1. Targets within a range from 1m to 6m were labeled with 2. For each trial, users were either asked to fixate targets labeled 1, 2, or to freely look around in the scene, corresponding to factor levels (FocusMode = NEAR, MID, or FREE). As the influence of the DoF is scene and focus point dependent, we wanted to make sure that a wide variety of objects at different depths were focused on by the user. Moreover, we tested the scene with full ray tracing (FoveatedMode = FULL) vs. the presented foveated mode (FOVEATED). For the foveated mode the same settings as used in Section 4.1 were selected, resulting in 558,945 updated samples per frame, regardless of the DoFMode. Each configuration was presented for 6 seconds. After each trial the participants were confronted with the following statements, where Q1 to Q3 had to be rated on a 7-point Likert scale from −3 (strongly disagree) to 3 (strongly agree), while Q4 had to be rated on a numerical scale from −3 (no depth perception) to 3 (strong depth perception):

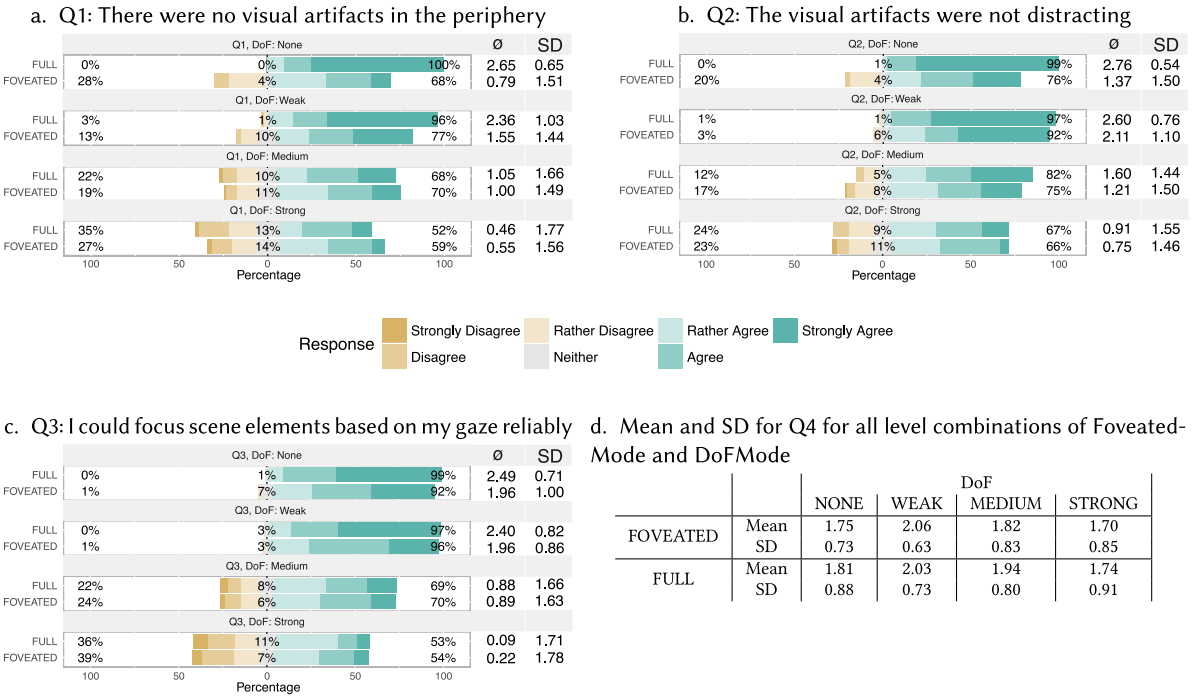
Q1 There were no visual artifacts in the periphery.
Q2 The visual artifacts were not distracting.
Q3 I could focus scene elements based on my gaze reliably.
Q4 Rate the intensity of depth perception.

*Evaluation.* The study was performed with 12 participants (seven male/female female, all with academic backgrounds) aged between 25 and 50 (M = 35, SD = 7.4), that reported to have normal or corrected-to-normal vision. To evaluate the visual quality of the presented approach, we define the following research questions:

RQ1 Did gaze-contingent DoF conceal visual artifacts?
RQ2 Did gaze-contingent DoF increase depth perception?

Plots of the ratings for Q1–Q3 are presented in Figure 7. Ratings for depth perception are illustrated in Figure 7(d). As Levene and Shapiro-Wilk tests have shown that the data's homoscedasticity and normality cannot be relied upon, we use the nonparametric, rank-based analysis of variance (ANOVA) approach from R's ARTool package.[3] Performing a three-way ANOVA with factors FoveatedMode, FocusMode, and DoFMode, also accounting for interactions, gave the results presented in Table 2. *Post hoc* tests have been performed using F-tests with Holm's method for $p$-value adjustment. The ratings presented in Q1 and Q2, as well as the depth perception

[3]Matthew Kay and Jacob O. Wobbrock. ARTool: Aligned Rank Transform. Retrieved May 23, 2018 from https://cran.r-project.org/web/packages/ARTool/index.html.




| | | DoF | | | |
|---|---|---|---|---|---|
| | | NONE | WEAK | MEDIUM | STRONG |
| FOVEATED | Mean | 1.75 | 2.06 | 1.82 | 1.70 |
| | SD | 0.73 | 0.63 | 0.83 | 0.85 |
| FULL | Mean | 1.81 | 2.03 | 1.94 | 1.74 |
| | SD | 0.88 | 0.73 | 0.80 | 0.91 |

Fig. 7. Likert scale ratings of Q1-Q3 and means and standard deviation for Q4.

rating in Q4 were filtered using the focus reliability rating illustrated in Figure 7(c). If the trial is rated below zero and users could not focus reliably (`Rather Disagree`) it was removed. By inspecting Figure 7(c) and by looking at the results from the ANOVA, it can be seen that there is an interaction between the focus reliability and the DoF mode. As the intensity of the DoF effect is increased, people are less likely to tolerate below-perfect gaze depth estimates. Even if the estimate is close to the focused depth, stronger DoF modes will likely blur the image in regions the user expects to be in focus. Moreover, users are quite rigorous when the DoF effect does not adjust with the correct speed. However, the speed of accommodation and its range are both user and age-dependent (Temme and Morris 1989). Incorporating more elaborate physiological models on those properties might improve the perceived accuracy as well. Nevertheless, taking a look at the filtered ratings for the noticeability and how distractive they were (Q1 and Q2) is of interest.

*(RQ1).* By looking at the results of Likert ratings presented in Figure 7(a), a shift in ratings between the various levels of DoFMode can be observed. Although the highest visual quality was reported for full rendering without using DoF, the ratings between foveated and full rendering become increasingly similar with a stronger DoF effect. This becomes apparent when comparing ratings for DoFModes WEAK, MEDIUM, and STRONG for both full and foveated rendering. Interestingly, the mean Q1 ratings for DoFMode STRONG were even higher for foveated than for full rendering. The existence of such differences is also apparent from the interaction between factors in the ANOVA shown in Table 2. While the results presented for Q2 in Figure 7(b) show that increasing DoFMode levels result in worse ratings for full rendering, foveated rendering clearly benefits from a WEAK DoF effect. This becomes especially apparent when comparing the means. Nonetheless, from the results shown in the figure we assume that both ratings are similar. Since visual artifacts were least noticeable and disturbing for DoFModes WEAK and MEDIUM with foveated rendering, we assume that our DoF mode successfully conceals artifacts. Unfortunately, enabling DoF seems to negatively influence the perceived quality using full ray tracing.

Table 2. Significant Results ($p < 0.05$) for the Performed ANOVA

| | | | |
|---|---|---|---|
| **Q1** | **FoveatedMode:DoFMode** | F(3, 484) = 11.78 | |
| | FOVEATED - FULL | NONE - WEAK | F(1, 484) = 13.7 |
| | | NONE - MEDIUM | F(1, 484) = 23.92 |
| | | NONE - STRONG | F(1, 484) = 25.16 |
| **Q2** | **FoveatedMode:DoFMode** | F(3, 484) = 7.72 | |
| | FOVEATED - FULL | NONE - WEAK | F(1, 484) = 19.16 |
| | | NONE - MEDIUM | F(1, 484) = 8.82 |
| | | NONE - STRONG | F(1, 484) = 12.82 |
| **Q3** | **FoveatedMode:DoFMode** | F(3, 484) = 2.73 | |
| | FOVEATED - FULL | NONE - WEAK | F(1, 484) = 3.87 |
| | | NONE - MEDIUM | F(1, 484) = 11.34 |
| | | NONE - STRONG | F(1, 484) = 15.88 |
| **Q4** | **DoFMode** | F(3, 484) = 5.6 | |

Main effects were left out if significant interactions were present. For Q1–Q4, significant results are shown as main effects or interactions together with their difference of differences (DoD) for significant factor levels.

In addition, perceived image quality does get worse with increasing DoF intensities for both full and foveated rendering. The loss in image quality coincides with the results by Duchowski et al. (2014) and Vinnikov et al. (2016).

*(RQ2).* With the ANOVA showing statistical significance for the differences in depth perception between the various levels of DoFMode ($p < 0.05$, $F(3, 484) = 5.6$), Figure 7(d) shows the corresponding means and standard deviations. While the lowest DoF setting shows the highest depth perception rating, differences between the various factor combinations are quite small. In a questionnaire after the study, we asked the participants for their agreement with the statement that they felt DoF could increase depth perception. Here we got mixed results (M = 1, SD = 1.13). While five of the participants rated their agreement to be neutral, eight users rather agreed with the statement. Surely depth-perception depends on the tracking accuracy. However, the response to the DoF effect seems to be highly individual. Depth perception is rated best for DoFMode = WEAK, with decreasing ratings in the order MEDIUM, NONE, STRONG. One possible explanation is that the introduction of a weak DoF effect supports depth perception, while stronger DoF settings produce over-blurring, no DoF fully reveals the rendering artifacts.

## 5 CONCLUSION AND OUTLOOK

In this article, we presented a gaze-contingent rendering and filtering approach exploiting knowledge of the retinal and optical abilities of the HVS to accelerate rendering. Samples can be reduced by 69%, and gaze-contingent DoF has shown to be a viable solution to conceal rendering artifacts. A user study showed that quality ratings between foveated and full rendering were almost on par using gaze-contingent DoF, although the visual quality was slightly reduced.

As DoF essentially is a guided low-pass filter applied to the image, it is useful to hide high-frequency artifacts that are challenging for peripheral vision. However, the influence of the DoF effect is scene dependent, as artifacts are only reduced for parts of the scene that are out-of-focus. Hence, for optimal results, several strategies for foveated rendering need to be combined. Filtering the image using DoF is another building block to perception-driven rendering. For future work, we want to computationally measure the influence of DoF on various artifacts with different scenes, e.g., by using a wavelet analysis as presented by Patney et al. (2016). Most recently, Meng et al. (2018) presented a log-space transform to synthesize images in a foveated fashion. Though they present

impressive results, their method is still suffering from temporal artifacts and view-dependent inconsistencies for glossy specular reflections. We are confident that using a DoF filter makes it possible to alleviate such artifacts, and we hope to entirely discard the use of TAA methods, commonly used in foveated rendering systems. The latter are prone to artifacts, especially in dynamic scenes. We are also confident that using knowledge about the DoF is crucial when scheduling potentially salient regions for (re-)sampling. Also, it should be possible to further reduce the number of cast rays by considering DoF progressively while rays are cast. Here our goal is to further investigate adaptive foveal sampling. Although we can already re-sample the scene multiple times employing knowledge of the CoC, we cannot achieve the necessary frame rates to meet the HMD's V-Sync limit, which is necessary to reduce fatigue and cope with fast eye movements (Albert et al. 2017). Knowing the size of the CoC allows for (re-)sampling the image in regions that are in focus. We think that the perceived visual quality using a DoF filter can ultimately exceed full rendering modes while achieving equal or even lower render times at lower sampling rates. Using PPI already allows for integrating samples based on perceptual requirements, e.g., image saliency. This way, DoF will become another important factor when deciding which regions or pixels need more computational effort. More flexible and faster ray casting and ray-tracing solutions like NVIDIA RTX[4] and HVVR,[5] as well as new hardware generations, will provide the necessary computing power. This also enables us to study fully dynamic scenes, which is still too slow for HMDs in our current implementation. Besides the computational complexity, better and more precise eye-tracking solutions or methods that apply scene or object knowledge, e.g., presented by Mantiuk et al. (2013), that allow for deriving more accurate gaze-depth estimates will extend the applicability of our approach, while more elaborate physiological models on the speed of accommodation might further improve perceived quality. Regarding the intensity of depth perception, it may be worthwhile to take a closer look at the influence of artifacts and over-blurring in the visual periphery.

## ACKNOWLEDGMENTS

The work is supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) funding the MoVISO ZIM-project under Grant No. ZF4120902.

## REFERENCES

Rachel Albert, Anjul Patney, David Luebke, and Joohwan Kim. 2017. Latency requirements for foveated rendering in virtual reality. *ACM Trans. Appl. Percept.* 14, 4 (Sep 2017), 25:1–25:13.

Elena Arabadzhiyska, Okan Tarhan Tursun, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. 2017. Saccade landing position prediction for gaze-contingent rendering. *Proceedings of ACM Transactions on Graphics (SIGGRAPH'17)* 36, 4, Article 50 (2017), 12 pages.

Brian A. Barsky and Todd J. Kosloff. 2008. Algorithms for rendering depth of field effects in computer graphics. In *Proceedings of the 12th WSEAS International Conference on Computers (ICCOMP'08)*. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 999–1010.

Mike Bukowski, Padraic Hennessy, Brian Osman, and Morgan McGuire. 2013. The skylanders SWAP force depth-of-field shader. In *GPU Pro 4: Advanced Rendering Techniques*. A K Peters/CRC Press, Wellesley, MA, USA, 175–184.

Joe Demers. 2004. Depth of field: A survey of techniques. *GPU Gems*. Vol. 1. Addison-Wesley, Chapter 23.

Andrew T. Duchowski, Donald H. House, Jordan Gestring, Rui I. Wang, Krzysztof Krejtz, Izabela Krejtz, Radoslaw Mantiuk, and Bartosz Bazyluk. 2014. Reducing visual discomfort of 3D stereoscopic displays with gaze-contingent depth-of-field. In *Proceedings of the ACM Symposium on Applied Perception (SAP'14)*. ACM, New York, NY, USA, 39–46.

Andrew T. Duchowski, Brandon Pelfrey, Donald H. House, and Rui Wang. 2011. Measuring gaze depth with an eye tracker during stereoscopic display. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization (APGV'11)*. ACM, New York, NY, USA, 15–22.

Masahiro Fujita and Takahiro Harada. 2014. Foveated Real-Time Ray Tracing for Virtual Reality Headset. Poster, SIGGRAPH Asia'14.

Herbert Gross. 2005. *Survey of Optical Instruments*, Vol. 4., *Handbook of Optical Systems*. Wiley-VCH.

Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D graphics. *ACM Trans. Graph. (SIGGRAPH Asia'12)* 31, 6, Article 164 (Nov. 2012), 10 pages.

[4]NVIDIA RTX Technology. Retrieved May 25, 2018 from https://developer.nvidia.com/rtx.

[5]Warren Hunt. 2017. Real-Time Ray Casting for Virtual Reality. In HPG 2017, Keynote. Oculus Research. Retrieved May 25, 2018 from http://www.highperformancegraphics.org/wp-content/uploads/2017/Hot3D/HPG2017_RealTimeRayCasting.pptx.

Sébastien Hillaire, Anatole Lécuyer, Rémi Cozot, and Géry Casiez. 2008. Using an eye-tracking system to improve camera motions and depth-of-field blur effects in virtual environments. In *IEEE (VR'08)*. IEEE, 47–50.

H. Igehy. 1999. Tracing ray differentials. In *Proceedings of SIGGRAPH'99*. ACM, New York, NY, 179–186.

George-Alex Koulieris, Bee Bui, Martin S. Banks, and George Drettakis. 2017. Accommodation and comfort in head-mounted displays. *ACM Trans. Graph. (SIGGRAPH'17)* 36, 4, Article 87 (2017), 11 pages.

Tim Lindeberg. 2016. *Concealing Rendering Simplifications using Gaze Contingent Depth of Field*. Master's thesis. KTH Royal Institute of Technology School of Computer Science and Communication, Sweden. https://kth.diva-portal.org/smash/get/diva2:947325/FULLTEXT01.pdf.

Radoslaw Mantiuk, Bartosz Bazyluk, and Rafal K. Mantiuk. 2013. Gaze-driven object tracking for real time rendering. In *Comput. Graph. Forum (Eurographics'13)*, Vol. 32. The Eurographs Association & John Wiley & Sons, Ltd., 163–173.

Radoslaw Mantiuk, Bartosz Bazyluk, and Anna Tomaszewska. 2011. Gaze-dependent depth-of-field effect rendering in virtual environments. In *Proceedings of the SGDA'11*. Springer, Berlin, Heidelberg, 1–12.

Ricardo Marroquim, Martin Kraus, and Paulo Roma Cavalcanti. 2007. Efficient point-based rendering using image reconstruction. *The Eurographics Association on Point-Based Graphics* 1 (2007), 101–108.

Michael Mauderer, Simone Conte, Miguel A. Nacenta, and Dhanraj Vishwanath. 2014. Depth perception with gaze-contingent depth of field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. ACM, New York, NY, USA, 217–226.

L. McIntosh, Bernhard E. Riecke, and Steve DiPaola. 2012. Efficiently simulating the bokeh of polygonal apertures in a post-process depth of field shader. In *Comput. Graph. Forum (Eurographics'12)* 31, 6 (2012), 1810–1822.

Xiaoxu Meng, Ruofei Du, Matthias Zwicker, and Amitabh Varshney. 2018. Kernel foveated rendering. In *Proc. ACM Comput. Graph. Interact. Tech. (ACM I3D'18)* 1, 1, Article 5 (July 2018), 20 pages.

Jurriaan D. Mulder and Robert van Liere. 2000. Fast perception-based depth of field rendering. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST'00)*. ACM, New York, NY, USA, 129–133.

Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. 2016. Towards foveated rendering for gaze-tracked virtual reality. *ACM Trans. Graph. (SIGGRAPH Asia'16)* 35, 6 (Nov. 2016), pp. 179:1–179:12.

Arsène Pérard-Gayot, Javor Kalojanov, and Philipp Slusallek. 2017. GPU ray tracing using irregular grids. *Comput. Graph. Forum (Eurographics'17)* 36, 2 (May 2017), 477–486.

Michael Stengel, Steve Grogorick, Martin Eisemann, and Marcus Magnor. 2016. Adaptive image-space sampling for gaze-contingent real-time rendering. In *Proceedings of the Eurographics Symposium on Rendering (EGSR'16)*, E. Eisemann and E. Fiume (Eds.). Eurographics Association, Goslar, Germany, 129–139.

L. A. Temme and A. Morris. 1989. Speed of accommodation and age. *Optom. Vis. Sci.* 66, 2 (Feb 1989), 106–112.

Karthik Vaidyanathan, Marco Salvi, Robert Toth, Tim Foley, Tomas Akenine-Möller, Jim Nilsson, Jacob Munkberg, Jon Hasselgren, Masamichi Sugihara, Petrik Clarberg, Tomasz Janczak, and Aaron Lefohn. 2014. Coarse pixel shading. In *ACM HPG'14*. Eurographics Association, Goslar, Germany, 9–18.

M. Vinnikov, R. S. Allison, and S. Fernandes. 2016. Impact of depth of field simulation on visual fatigue. *Int. J. Hum.-Comput. Stud.* 91, C (July 2016), 37–51.

Martin Weier, Thorsten Roth, André Hinkenjann, and Philipp Slusallek. 2018. Predicting the gaze depth in head-mounted displays using multiple feature regression. In *Proceedings of the ACM (ETRA'18)*. ACM, Warsaw, Poland, Article 19, 8 pages.

Martin Weier, Thorsten Roth, Ernst Kruijff, André Hinkenjann, Arsène Pérard-Gayot, Philipp Slusallek, and Yongmin Li. 2016. Foveated real-time ray tracing for head-mounted displays. In *Computer Graphics Forum (PG'16)*. Eurographics Association, Goslar, Germany, 289–298.

Martin Weier, Michael Stengel, Thorsten Roth, Piotr Didyk, Elmar Eisemann, Martin Eisemann, Steve Grogorick, Andre Hinkenjann, Ernst Kruijff, Marcus Magnor, Karol Myszkowski, and Philipp Slusallek. 2017. Perception-driven accelerated rendering. In *Comput. Graph. Forum (Eurographics'17)* 36, 2 (May 2017), 611–643.

Lei Yang, Diego F. Nehab, Pedro V. Sander, Pitchaya Sitthi-amorn, Jason Lawrence, and Hugues Hoppe. 2009. Amortized supersampling. *ACM Trans. Graph. (SIGGRAPH Asia'09)* 28, 5 (2009), 135:1–135:12.

Received May 2018; revised June 2018; accepted June 2018