Emulating Foveated Path Tracing

Andrew Polychronakis
apolychronakis@isc.tuc.gr
Technical University of Crete
Chania, Greece

George Alex Koulieris
georgios.a.koulieris@durham.ac.uk
Durham University
Durham, United Kingdom

Katerina Mania
amania@isc.tuc.gr
Technical University of Crete
Chania, Greece

ABSTRACT

At full resolution, path tracing cannot be deployed in real-time based on current graphics hardware due to slow convergence times and noisy outputs, despite recent advances in denoisers. In this work, we develop a perceptual sandbox emulating a foveated path tracer to determine the eccentricity angle thresholds that enable imperceptible foveated path tracing. In a foveated path tracer the number of rays fired can be decreased, and thus performance can be increased. For this study, due to current hardware limitations prohibiting real-time path-tracing for multiple samples-per-pixel, we pre-render image buffers and emulate foveated rendering as a post-process by selectively blending the pre-rendered content, driven by an eye tracker capturing eye motion. We then perform three experiments to estimate conservative thresholds of eccentricity boundaries for which image manipulations are imperceptible. Contrary to our expectation of a single threshold across the three experiments, our results indicated three different average thresholds, one for each experiment. We hypothesise that this is due to the dissimilarity of the methodologies, i.e., A-B testing vs sequential presentation vs custom adjustment of eccentricities affecting the perceptibility of peripheral blur among others. We estimate, for the first time for path tracing, specific thresholds of eccentricity that limit any perceptual repercussions whilst maintaining high performance. We perform an analysis to determine potential computational complexity reductions due to foveation in path tracing. Our analysis shows a significant boost in path-tracing performance ($\geq 2x - 3x$) using our foveated rendering method as a result of the reduction in the primary rays.

CCS CONCEPTS

• **Computing methodologies → Rendering**; **Perception**; Image processing.

KEYWORDS

Foveated rendering, path-tracing, gaze detection

ACM Reference Format:
Andrew Polychronakis, George Alex Koulieris, and Katerina Mania. 2021. Emulating Foveated Path Tracing. In *Motion, Interaction and Games (MIG '21), November 10–12, 2021, Virtual Event, Switzerland.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3487983.3488295

MIG '21, November 10–12, 2021, Virtual Event, Switzerland

ACM ISBN 978-1-4503-9131-3/21/11…$15.00
https://doi.org/10.1145/3487983.3488295

1 INTRODUCTION

The demand for highly realistic, real-time, physics-based rendering systems is more prevalent than ever in fields such as gaming, simulations, cultural heritage reconstructions, etc. The need for high quality content as well as the emergence of Virtual Reality (VR) as a commodity, requires significant computational power for rendering to achieve realistic, physically-accurate lighting effects. State of the art rendering systems produce a uniform, high quality image. A fundamental characteristic of the Human Visual System (HVS) is that visual acuity decreases rapidly as the angular distance away from central gaze direction (eccentricity) increases. The fovea supporting the highest visual acuity is an area in the central retina corresponding to only 5° of the visual field. Foveated rendering algorithms improve real-time performance by progressively decreasing rendering quality towards the periphery of vision while maintaining high fidelity in the fovea, based on tracking gaze motion in real-time.

Foveated rendering techniques suffer from visual artifacts such as aliasing in the periphery [Guenter et al. 2012]. In order to correct visual artifacts, progressive blur has been applied inducing tunnel vision, corrected by enhanced contrast [Patney et al. 2016]. Initial efforts of foveated ray tracing employed computational cost reductions using adaptive ray tracing and re-projection from previous frames [Weier et al. 2016]. Such techniques have not been evaluated with moving objects, glossy materials and dynamic light sources. Rendering techniques such as path tracing (PT) allow the rendering of complex effects such as reflection, refraction, soft shadows and diffuse inter-reflection [Kajiya 1986]. Despite performance gains of previous foveated, rasterisation-based rendering approaches [Patney et al. 2016], at full resolution, path tracing cannot be deployed in real-time on current graphics hardware. This is due to slow convergence times and noisy outputs despite recent advances in denoisers [Chaitanya et al. 2017a; Schied et al. 2017, 2018].

In this paper, we develop a perceptual sandbox emulating a foveated path tracer to determine, for the first time, the eccentricity angle thresholds that enable imperceptible foveated path tracing. In a foveated path tracer, the aim is to reduce the total number of rays fired, and this, in principle, can be achieved in three ways: (i) reducing the total number of samples-per-pixel (SPP), (ii) reducing the total number of secondary ray bounces, and, (iii) reducing the total pixel count of the output buffer. Due to current hardware limitations prohibiting real time path-tracing, we pre-render image buffers using different parameters and emulate foveated rendering as a post-process, guided by an eye tracker capturing gaze motion. We then perform three experiments to estimate conservative thresholds of eccentricity where image foveation is detectable for path traced images. The first experiment compared pairs of sequences rendered with different foveation eccentricities. The second experiment investigated the lowest foveal eccentricity perceived to be indistinguishable compared to the non-foveated reference. Finally, the last experiment measured a single eccentricity per user deemed to be imperceptible, based on a slider test. Finally, we perform an analysis of the expected computational complexity reduction when such a system will be able to run in real time. Our specific contributions include:

- We develop an emulated foveated path tracer. The foveation parameters are drawn from a visual acuity psychophysical model for the entire visual field. We pre-render image buffers and emulate foveated rendering as a post-process by selectively blending pre-rendered content, guided by an eye tracker detecting eye motion in real-time.
- We perform perceptual calibration, comparing and contrasting three different experimental protocols to investigate how the methodology might affect path tracing being noticed. We determine perceptibility thresholds for path traced images.
- We perform an analysis to determine the potential to reduce computational complexity via foveation, by reducing the rays fired as a result of zoning the output buffer to varied resolutions. We estimate thresholds of eccentricity that limit perceptual artifacts whilst maintaining high performance.

2 RELATED WORK

In this section, we include an overview of the eye physiology, analyze past research in foveated rendering for rasterization & ray-tracing as well as path tracing and denoising.

2.1 Physiology of the eye

There are two types of photoreceptors located in the photosensitive retina of the eye: $6 \cdot 10^6$ *cones* enabling colour, high resolution vision and approximately 20 times as many *rods* supporting monochromatic peripheral vision. The fovea is located in the center of the posterior portion of the retina measuring 1.5 mm in diameter, corresponding to 5.2° of the visual field consisting entirely of cones.

Foveal vision is defined as the central 1.5–2° of the visual field [Strasburger et al. 2011]. *Eccentricity* refers to the angular distance from the centre of the visual field. Cone density decreases with increasing eccentricities. Ring-shaped regions surrounding the fovea are the parafovea and the perifovea. The parafovea has an outer diameter of 2.5 mm which covers around 5.2° to 9° of the visual field, and the perifovea covers a 9° to 17° radius around the fovea. Vision outside of the fovea is called peripheral, or indirect vision [Curcio et al. 1990]. The highest density of *rods* is approximately 15–20° around the *fovea*.

Visual acuity signifies clarity of vision dependent not only on optical but also on neural factors such as the ability to form a sharp retinal image, how healthy is the optical functioning of the retina and on the quality of interpretation of the visual stimulus by the brain. The visual acuity of the eye drops significantly outside the small foveal region generating a high resolution image. Acuity decreases rapidly as eccentricity increases, first steeply and then more gradually, in an inverse-linear fashion, i.e., the decline follows approximately a hyperbola [Strasburger et al. 2011]. At an eccentricity of 6°, it is already reduced by 75%. Visual acuity is often measured according to the size of letters viewed on a Snellen chart.

2.2 Foveated Rasterization

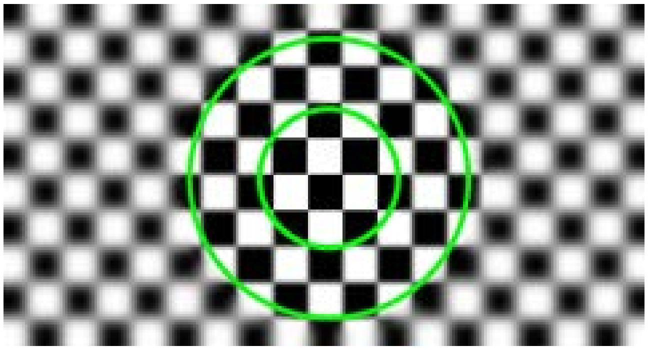


Figure 1: Visualisation of the foveal (inner circle), middle and outer peripheral zones. The placement of the boundaries of the zones (two green circles) are parameters of our experiment. Foveation exaggerated here, for visualisation.

Early work on foveated rendering achieved performance gains but was constrained by inherent aliasing and system lag [Luebke et al. 2000; Ohshima et al. 1996]. Such approaches manipulated geometric level-of-detail (LOD) based on eccentricity but not rendering resolution which is not efficient for speeding up scenes of high per-pixel shading cost. State-of-the-art reports detail early work on gaze-contingent displays [Baudisch et al. 2003; Duchowski and Çöltekin 2007; Koulieris et al. 2019].

Early attempts of foveated rendering for Level-of-Detail (LOD) was based on real-time gaze tracking using an eye tracker [Loschky and McConkie 2000] also manipulating LOD on areas of simple tasks [Cater et al. 2003]. Such attempts could not maintain steady display update rates often resulting to visual artifacts. Foveated rendering could also take input from saliency models without the use of an eye tracker resulting in gaze predictions [Longhurst et al. 2006]. Saliency models were based on low level global image features such as luminance and contrast, on task maps [Hillaire et al. 2010; Lee et al. 2008] or even on high-level context [Koulieris et al. 2014]. Low-level saliency models alone often fail to predict gaze direction because of context [Sundstedt et al. 2008].

Foveated rendering presented by Guenter et. al. [2012] renders three eccentricity layers (inner/foveal, middle, and outer layer), corresponding to the angular distance away from the central gaze direction around the tracked gaze point (Figure 1). The inner layer has the smallest angular diameter being rendered at the highest resolution and finest LOD. The middle and outer layers span larger angular diameters, rendered at a progressively lower resolution and LOD. These layers are updated at half the temporal rate of the inner layer. The output is composed by smoothly blending and interpolating the three layers up to the native display resolution. This approach suffers from two issues: aliasing in the periphery due to reduced point sampling density and system latency.

The approach of Guenter et al. [2012] focused on reducing computational cost without explicitly identifying and minimizing perceptible artifacts, resulting in temporal aliasing due to head and gaze motion which is distracting, breaking immersion. Patney et al. [2016] attempted to solve this problem by employing a post process Gaussian blur with a progressively increasing filter-width based on eccentricity. This resulted in another issue, as blur with a large radius induces a sense of tunnel vision. Tunnel vision arises from reduced contrast, as filtering typically reduces image contrast. The authors alleviated this problem by enhancing contrast in the periphery which restored most of the missing details. This allowed them to double the amount of blurring in the periphery before inducing any noticeable tunnel vision.

Previous work has shown that temporally stable approaches, both blurred and aliased, were less noticeable than temporally volatile approaches when down-sampling an image [Hoffman et al. 2018]. The boundaries between zones in the periphery region were found to have low visibility. The replacement from a low to a high-resolution image was not detectable after testing, if presented for less that 40 ms. The processing time of our foveated method does not surpass the time limit of 40 ms, in order for the drop in quality in the periphery region not to be perceptible after the rendering process. In our work, the rendering zones, e.g. fovea, periphery and outer, are blended.

2.3 Foveated Ray Tracing

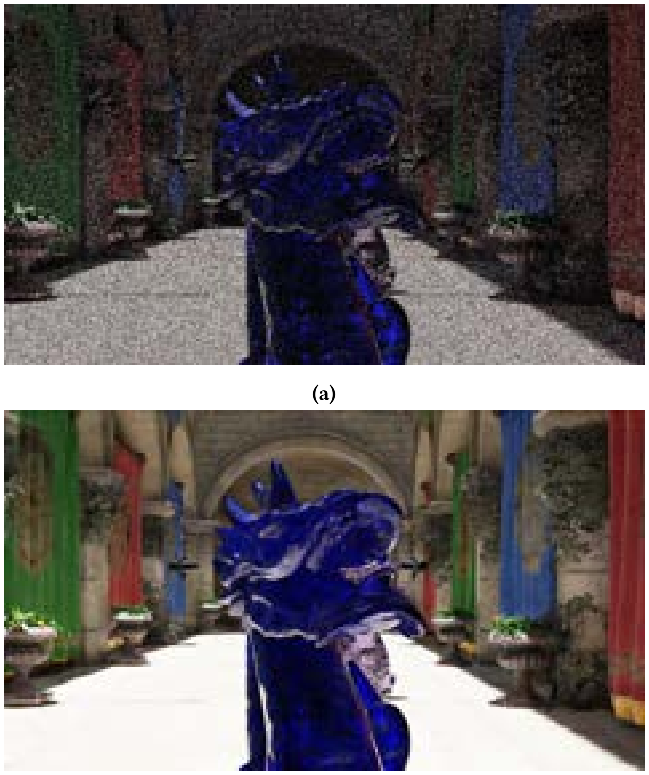
Ray tracing and high visual fidelity techniques such as path tracing cannot be deployed in real-time with current graphics hardware. Gaze-contingent rendering based on ray tracing has been proposed based on increased sampling near object silhouettes and high contrast areas [Murphy et al. 2009]. Another system also resulted to visual artifacts without taking into account the HVS's contrast sensitivity, not evaluated by users [Fujita and Harada 2014]. In our work, we sample at high resolution only where the user is fixating and we reduce the samples as eccentricity increases. We perform a user study to validate that our foveated rendering is undetectable.

Though the principles enabling foveated rendering remain the same irrespective of the rendering method, it is unclear how techniques developed for rasterization can be applied to ray tracing. Recent approaches proposed foveated ray tracing that employs specific computational cost reductions targeted specifically to ray tracing [Roth et al. 2016; Weier et al. 2016]. The total sample density is reduced by adapting the ray generation pattern to the foveal receptor density. To limit the detection of visual artifacts, image reconstruction is performed at higher quality using two methods. The first uses a *support image* that is guaranteed to sample the full scene using a lower, but uniform resolution. The second method draws information from re-projected frames to further improve the quality of the reconstructed image. The use of re-projected frames can produce temporal artifacts [Weier et al. 2016]. In our work, we use our proposed probability model to generate rays. We generate an extra ray for a set of pixels for the periphery and outer zones. Then, we reconstruct the frame using spatial data of the rendering frame and apply a post-processing effect to create an imperceptible frame based on the user's gaze. Our method does not suffer from temporal artifacts. The aliasing issues in the middle periphery and outer periphery are eliminated employing a blur effect.

2.4 Path-tracing & Denoising

Path-tracing (PT) [Kajiya 1986] is a rendering technique producing photo-realistic images by simulating light transport. PT has a high computational cost as a large number of rays shot form each pixel have to be tracked along their path through a virtual scene to calculate the light reaching a virtual camera.

Tracking a ray continues until a user-defined maximum number of bounces is reached. For some effects, such as reflection and refraction, PT requires setting the maximum number of bounces to high values. For each pixel sample, PT creates a slightly different image. Pixel samples need then be averaged in a single image to eliminate the noise stemming from the fact that each ray for each sample starts from a different position inside a pixel. In addition, the BDRF itself introduces some randomness to the reflected direction of the ray. Taking only a single sample, though computationally effective, produces noisy outputs.



(a)

(b)

Figure 2: (a) Sponza palace image rendered with 1SPP. (b) Image with noise removed using the Optix Denoiser.

A theoretical study has shown that foveated path-tracing rendering in VR is potentially feasible as total ray counts could be reduced down to 94% [Koskela et al. 2016]. To demonstrate that the potential ray reduction could be significant, the authors, based on a human visual acuity approximation function, estimate how many rays need to be generated. The study does not describe how the non-calculated (empty) pixels will be re-constructed but only focuses on the potential performance gains from this reduction, without evaluating their results in a user study. In our work, we first develop an emulator to measure detectability thresholds for foveated path tracing, then provide a method to reconstruct missing pixels due to down-sampled images, and, finally, we calculate potential performance gains in desktop displays.

Normally, to render a high-quality image, the SPP are set to a high value and the total bounces are set to at least 16, depending on the scene content, which leads to increased rendering time. For example, for a resolution of 1920x1080 and an SPP of 4, ~8.3 million primary rays must be tracked, and for each ray, one must further perform several intersection calculations with the geometry of the virtual scene depending on the total bounces allowed. Each ray intersection and bounce generates more rays. The total number of such rays cannot be determined prior to rendering. Current generation GPUs, despite having specialised hardware ray tracing units that can significantly accelerate intersection calculations, can only support real time path tracing at 1 SPP. Using 1 SPP creates a very noisy image, as shown in Figure 2.

To eliminate such noise, many denoisers [Chaitanya et al. 2017a; Schied et al. 2017, 2018] have been proposed. Particularly effective in reducing noise, is Nvidia's Spatiotemporal Variance-Guided Filtering (SVGF) [Schied et al. 2017]. SVGF reprojects previous frames to increase the number of samples in the filter input. This both improves temporal stability and is an indicator of the per pixel variance. The size of the filter is determined based on the per-pixel variance. SVGF's main limitation is that it only works on the primary rays and as a result, noise from e.g., reflections remains, inducing temporal blur in the denoised frame.

An extension of the SVGF, attempting to eliminate temporal blur, is the Adaptive-Spatiotemporal Variance-Guided Filtering (A-SVGF) [Schied et al. 2018]. Temporal blur appears when, for example, a light source or light from a reflection, that has ceased to exist, appears still in subsequent frames, usually as a *trail of light.* To eliminate temporal blur the A-SVGF filter adapts the per-pixel temporal accumulation factor based on an estimation and reconstruction of the sparse temporal gradients.

A deep learning-based approach for denoising PT by Nvidia is based on recurrent autoencoders (RA) to improve temporal stability [Chaitanya et al. 2017a]. The RA considers information from the normal & depth channels of neighboring pixels to both increase the performance and the image quality of the output (Figure 2). Another approach to speed-up path tracing is to render at a much lower resolution and then use Nvidia's deep learning super-sampling technique (DLSS) [Hasselgren et al. 2020]. DLSS uses machine learning to upscale a low resolution frame to a higher resolution one.

3 EMULATING FOVEATED PATH TRACING

3.1 Overview

In an ideal foveated path tracing system, the sum total of rays can be significantly reduced, based on the distance from the fixated point. Special care has to be taken to eliminate popping artifacts due to e.g., reduced ray bounces. Such a system does not currently exist.

In our work, inspired by [Patney et al. 2016], we develop a perceptual sandbox, by emulating a foveated path tracer. In a foveated path tracer, the aim is to reduce the total number of rays fired, and this, in principle, can be done in three ways: (i) reducing the total number of samples-per-pixel, (ii) reducing the total number of ray bounces, and, (iii) reducing the total pixel count of the output buffer.

Reducing the total number of SPP was not a subject of this study, as modern denoisers perform exceptionally well with 1SPP, and as a result the main limiting factor for real time path tracing remains the enormous amount of rays required for high resolution buffers (even at 1SPP) and the secondary rays instantiated at geometry intersections to track ray bounces.

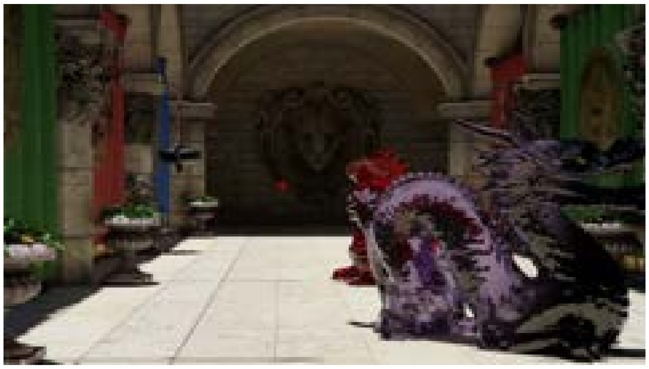


Figure 3: FPT with less secondary ray bounces in the middle and outer peripheral zones. Strong artifacts can be noticed, for example, the half transparent - half black dragon.

Ray bounce reduction enables significant performance gains. However, by reducing the bounces, quality drops remarkably especially for phenomena such as reflection and refraction (Figure 3). In a pilot study (see Sec. 4.2) we found that reducing the number of bounces for secondary rays is not a viable option for foveated path tracing as such ray elimination generates popping artifacts when, for example, transparent objects change colour as they pass through zones corresponding to different eccentricities.

In this study we investigated the perceptual repercussions and the potential to reduce computational complexity, by reducing the total rays fired by zoning the output buffer to different resolutions, similar to [Guenter et al. 2012]. We develop a model for ray generation in PT based on the acuity of the HVS. Lanczos resampling [Duchon 1979; Komzsik 2003] is then applied to smoothly interpolate between different zones.

Due to current hardware limitations prohibiting real time path-tracing, we pre-render the three buffers at different resolutions and emulate foveated rendering as a post-process based on eye tracking. We then perform three experiments to estimate conservative thresholds of eccentricity where the image zoning is detectable. Finally we perform an analysis of the expected performance gains for when such a system exists and can run in real time.

3.2 Model

The fovea has the highest acuity at determining visual detail. In the periphery and away from the fovea, the acuity of the HVS falls rapidly. We devise a probability model based on eccentricity (ω), from the fixated x, y location. Our model is based on the work of Mandelbaum et. al. [1947] that measured the visual acuity for different values of eccentricity of the human eye. We fit to Mandelbaum's data an exponential function closely matching the measured dataset with an accuracy of over 95%:

$$V(\omega) = 0.90964e^{\frac{\omega}{2.9661}} + 0.00792 \tag{1}$$

Based on equation 1 we then construct a probabilistic method:

$$P(\omega) = \begin{cases} 1 & , \omega <= \omega_o \\ 0.90964e^{\frac{\omega}{2.9661}} + 0.00792 & , \omega > \omega_o \end{cases} \tag{2}$$

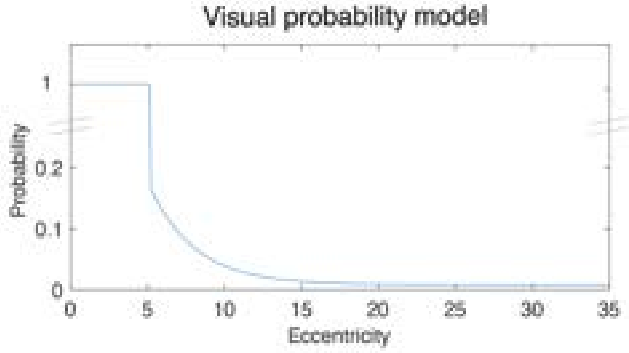


Figure 4: Our visual probability model, demonstrated for a foveal angle of 5 degrees. The function showcases how the lessening visual acuity of the human eye, would require a significantly smaller number of traced rays as we move from the fovea to the periphery.

where ω_o is the eccentricity value set to represent the size of the fovea of the human eye in degrees.

We visualise in Figure 4, this visual probability function. The function showcases how the lessening visual acuity of the human eye, results in spawning a significantly smaller number of rays as we move away from the fovea to the periphery.

When ω is less or equal than ω_o the probability is equal to one. Otherwise, a probability is estimated for ω, as seen in Figure 4. This model can be used to decide whether or not to spawn a new ray at a certain pixel based on the visual angle that pixel spans from the central fovea (ω is the eccentricity angle of each pixel from central fovea). To calculate ω for all pixels we determine the Euclidean distance (eq. 3) between the pixel being fixated (p) and all other pixels (q). Having calculated the Euclidean distance, we can then determine visual angle (eq. 4) to find the ω.

$$d(p,q) = \sqrt{(q.x - p.x)^2 + (q.y - p.y)^2} \tag{3}$$

$$V = 2\arctan\left(\frac{d}{2D_m}\right) \tag{4}$$

where D_m is the distance from the display.

Due to hardware limitations prohibiting real-time path-tracing we cannot use our model in real-time. Similarly to [Guenter et al. 2012] & [Patney et al. 2016], we discretise visual acuity in three distinct zones (Figure 1). Inner (foveal), middle (peripheral) and outer (extra-peripheral), with progressively lower resolution. Three zones were deemed to be enough both during our pilot experiments and according to previous work [Guenter et al. 2012; Patney et al. 2016]. Having more than three zones approximates the human visual system more closely, but increases the rendering overhead. Based on our discretisation, in the inner region, we path trace at full resolution. In the middle zone we trace a single ray for every 4 pixels. In the outer region we PT a single ray for every 16 pixels. To classify the pixels into regions we use our probability model (eq. 2).

To generate the frame buffers for the three zones we start by pre-rendering full-frame high resolution path-traced images (inner) and then down-sampling them to a quarter (middle) and one-sixteenth of the resolution (outer) using nearest neighbor interpolation to drop extra pixels. We then use Lanczos re-sampling to upscale the middle and outer images back to full-frame buffers enabling easy blending between them. We found that Lanczos filtering produces high quality outputs for its complexity. Lanczos re-sampling is also very fast in modern GPUs using a compute shader.

At runtime, centered around the pixel that the eyes are fixating as detected by an eye tracker, we dynamically blend pixels from all three buffers based on our probability model (eq. 2) that describes the acuity fall-off in the HVS. The central/inner region samples from the high resolution frame buffer, the middle from the medium resolution buffer and the outer from the low resolution buffer. To determine the boundaries of the zones we sample from the visual probability model, rather than, for example, doubling the foveal region size to obtain the middle zone size as in [Guenter et al. 2012]. We linearly interpolate a short overlapping boundary to avoid sharp transitions between zones.

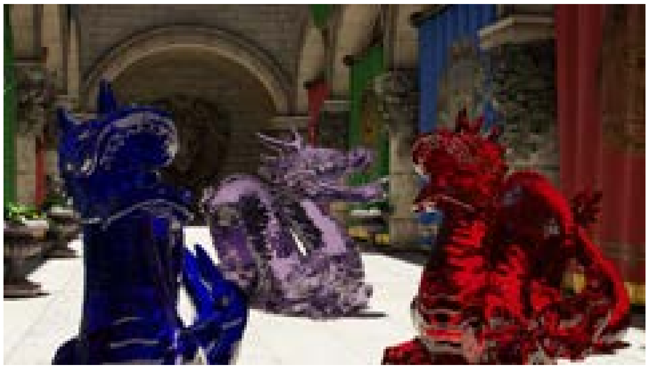


Figure 5: The Sponza palace scene, 16th century, reconstruction, with embedded transparent Stanford dragons enabling experimentation with many complex effects such as diffuse inter-reflection, shadows, transparency and more.

4 USER STUDIES

We conduct a series of experiments to evaluate the potential of foveated PT (FPT). In all experiments the highest visual quality setting (reference) is a full-resolution frame buffer where every pixel has been estimated using path tracing with 128 samples and 32 bounces. Because FPT produces a lower quality image compared to full-blown PT we are interested in determining conservative visual angle thresholds for the three foveation zones, where the drop in visual quality is *undetectable*. The thresholds we estimate for zoned FPT could then be employed in an actual FPT, once such a PT can actually run in real time.

Hypothesis. We hypothesize that there will be a single, average threshold, where the reduced ray-count buffers will be indistinguishable from non-foveated rendering.

Scene. For all experiments we use the Sponza scene [McGuire 2017], as seen in Figure 5, an iconic archaeological monument. To study the effect of FPT on complex reflection and refraction, we embedded in the virtual environment three Stanford dragons [Stanford 2021] which are made of glass of different colours.

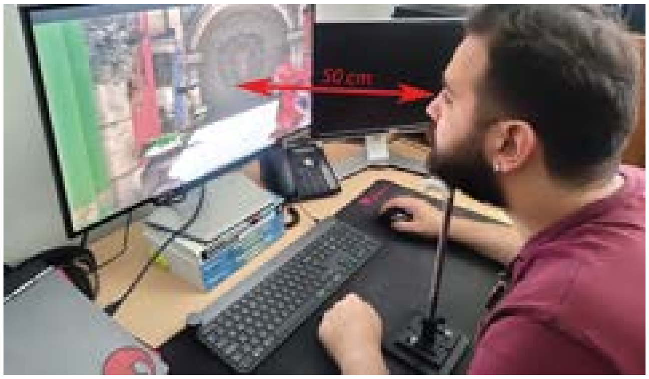


Figure 6: The experimental setup and a participant, set at a fixed distance from the monitor.

Basing our approach on [Guenter et al. 2012], we conducted three experiments: a pair test, a ramp test, and a slider test. Each subject participated in all three tests. In all experiments, the dependent variable is the size of the foveal region bordering with the middle region. The border of the middle zone with the outer zone is set based on our probability model. We use eye tracking to dynamically re-construct simulated images of the FPT according to gaze as explained in section 3.2. This enabled us to accurately measure conservative thresholds without relying on instructing our subjects to fixate on specific areas of the display; they were rather allowed to look anywhere on the screen.

4.1 Experiments: Hardware Setup

We use the 27" DELL S2719H monitor, with a resolution of 1920 x 1080 and a refresh rate of 60 Hz. We employed the Tobii eye tracker 4c to track user gaze. The eye tracker has a sampling rate of 90Hz, however, our latency bottleneck was the limited refresh rate of the monitor (>16.6ms). See Sec. 5.2 for additional discussion on latency. The computer running the FPT emulator was a Lenovo Legion y740 Laptop with a CPU i7-9750H, 16 GB RAM, and a single Nvidia GPU RTX-2060 with 6GB VRAM.

4.2 Pilot experiment: Ray Bounces

During the design phase of our study, we ran pilot experiments serving as a sensitivity analysis for the independent variables of the main experiments. One such experiment, regarding the effect of ray bounces, eliminated any ray bounce modification for our main experiments. We run a pair-test comparing non-foveated rendering to foveated images with reduced secondary ray bounces in the middle and peripheral zones. These secondary bounces generate soft shadows, reflections, refraction, diffuse inter-reflections and more. Reducing the secondary bounces to values low enough to measurably increase performance, generated strong motion signals in the periphery as transparency effects appeared or disappeared when objects transitioned between zones, and shadows & light leaks were flickering among other artifacts (see Fig. 3). We decided to exclude any variation in ray bounces for the main experiments, as such variation was perceived by all pilot subjects, unanimously.

4.3 Participants

20 subjects participated (8 female, average age 25.45, SD 2.11), only with normal or corrected-to-nornal vision. Participants were placed at 50cm from the screen as seen in Figure 6. Each participant performed the standard calibration procedure for the Tobii eye tracker 4c. Following calibration, the experiment sequence was initiated. The average time it took for each participant to finish the three experiments and the calibration process, was around 25.2 minutes.

4.4 Experiment 1: Pair

During the pair test, participants were presented with pairs of dynamic (moving) sequences of identical camera trajectories separated by a short (0.5s) black interval. At random, one member of the pair was the non-foveated path-traced rendering and the other part used foveated path-traced rendering at foveal eccentricity levels spanning from 5° to 30°, sampled using 5 discrete steps. Corresponding middle peripheral eccentricity levels spanned 57.5° to 82.5°. Pairs of all levels were presented three times with the foveated rendering first and three times with the non-foveated rendering first, at random. After showing each pair to participants, they were asked to report whether the first or the second rendering was better, or the two were of the same quality. The experiment was designed to investigate the foveation angles where the quality level was deemed similar to non-foveated rendering.

Procedure & Data recording: For each pair of images, we record the answers provided. To collect the data, we use a user interface that pops up following the presentation of the image pair. The user is then asked to report with an on-screen cursor which sequence was better or if they appeared to be the same.

4.5 Experiment 2: Ramp

During the ramp test, each participant was presented with a set of dynamic sequences of identical camera trajectories. An increasing ramp started with a non-foveated sequence and then the foveal and peripheral zones ramped down from foveal eccentricities of 30° down to 5°. A decreasing ramp started with a foveated sequence at 5° and then the foveal and peripheral zones ramped up from a foveal eccentricity of 5°, up to 30°, to non-foveated. Participants were then

Table 1: Performance boost expected from ray reduction. Even with conservative thresholds of 15° foveal and 65° middle, a performance boost of over 3x is expected.

	Middle Zone (deg)					
Foveal Zone (deg)	**50**	**55**	**60**	**65**	**70**	**75**
5	4.98x	4.56x	4.19x	3.95x	3.91x	3.91x
10	4.6x	4.23x	3.91x	3.7x	3.67x	3.67x
15	4.07x	3.78x	3.52x	3.35x	3.32x	3.32x
20	3.49x	3.28x	3.08x	2.95x	2.93x	2.93x
25	2.95x	2.79x	2.65x	2.55x	2.53x	2.53x
30	2.47x	2.36x	2.25x	2.18x	2.17x	2.17x

asked whether the quality had increased, decreased, or remained the same across each sequence step. Each ramp was sampled using 6 discrete steps. Each sequence was 7 seconds long and separated by a short interval of black, aiming to find the lowest foveal eccentricity perceived to be indistinguishable to the non-foveated reference.

Procedure & Data recording: While the foveation eccentricity level ramps up/down, we record the subject's response in each change of eccentricity level. We use a similar user interface to the previous experiment to collect user data. The user interface pops up after each sequence ends, asking the participant if the quality has increased, decreased, or remained the same.

4.6 Experiment 3: Slider

The slider test allows participants to change the foveal and peripheral eccentricity levels. They are first presented with a non-foveated animation as a reference. Starting at a low level of foveation eccentricity (foveal = 5° and periphery = 57.5°), they can increase the level, see the non-foveated reference again or decrease the level, reporting a quality level equivalent to the non-foveated reference.

Procedure & Data recording: Users first received instructions that they could change the foveation eccentricity using the keyboard. Having determined the quality level where the foveated sequence quality is indistinguishable from the non-foveated reference, users could save their preference, again using the keyboard.

4.7 Results

Raw data recorded for all subjects and experiments are shown in Figure 7. In the pair experiment, we have identified the lowest average foveation level for each subject where users reported that FPT looked better or similar to the non-foveated reference. The foveation eccentricity threshold for the pair experiment had a mean of 10.25° (SD 6.38°). In the ramp experiment, we identified the lowest average foveation level, when the users incorrectly report that the quality has changed based on the ramp direction or that the quality was the same across the entire ramp. For the ramp test, the foveation level had a mean of 21° (SD 6.85°). In the slider experiment, we estimate an average preferred value. For the slider experiment, the foveation level had a mean of 16.95° (SD 6.9166°).

5 DISCUSSION

We observed significantly different foveation thresholds between the three experiments. For the pair test, we obtain the lowest mean threshold (foveal ≈ 10°) where the foveated sequence appears to be of equivalent quality to the non-foveated sequence. For the slider test, we obtain a medium mean threshold (foveal ≈ 15°). Finally, for the ramp test, we obtain a high mean threshold (foveal ≈ 20°). This was a surprise. Contrary to our hypothesis of a single foveation threshold, we hypothesize that the varying thresholds estimated, arose for a few potential reasons. First, the methodologies of the three experiments (pair, ramp, slider) may have affected the subjects' sensitivity to peripheral blur, as the way that comparisons were made between images were inherently different, i.e., A-B testing vs sequential presentation vs custom adjustment of eccentricities. In addition, we hypothesize that operating on a 60 Hz monitor led us estimate very conservative eccentricity thresholds. If our subjects performed quite rapid saccades, the alterations due to foveated rendering were visible at times; we hypothesize that subjects selected a much higher angle threshold than they would have otherwise (see also Sec. 5.2). Eccentricity thresholds were shown to be based on the user's ability to detect sudden visual changes in their periphery rather than a specific value.

5.1 Computational Complexity Gains

Our method enables an extreme reduction of spawned rays for each frame. We can calculate the reduction of primary rays, and thus performance gains. However, the reduction of secondary rays is unpredictable. The amount of such rays spawned for shadows, reflections, etc., depends also on scene geometry and the current view-port. We can calculate the reduction of primary rays based on the *Gauss circle problem* [Guy 2004], i.e., determine how many pixels (lattice points) there exist in the maximum resolution foveal circle, the reduced resolution middle zone, and the further reduced resolution outer periphery (Figure 1). The two boundaries of the three zones parametrise the problem (green circles in Figure 1).

For our calculations we assume a 27" full-HD monitor at a distance of half a meter. Without foveation, 1920 * 1080 = 2 073 600 rays are normally required per frame. In Figure 8 we can observe, as expected, that the number of total rays fired is a function of both the foveal and middle zone boundaries. The maximum value in this plot corresponds to a foveal region that spans the entire field of view (∼ 70°), i.e., non-foveated rendering. For any other combination of foveal and middle zone angles we obtain significant ray count reductions and a corresponding performance boost as shown in Table 1. We note that to avoid sharp transitions between the three rendering zones, the circles must overlap a few pixels. This has a negligible effect in the total speedup. The same applies for the cost of Lanczos upscaling that has a negligible cost when running in a compute shader.

In Table 1, we note that even with a very large foveal region of 30° and a very large middle region of 75° (practically the middle zone spans the rest of the screen) we can still expect at least a 2x speedup. Based on a more conservative threshold of 15° degrees foveal and 65° middle zone (the rest of the screen would be rendered using the outer zone quality), the expected speed-up is over 3x. Even for exterior scenes with daylight, containing pre-rendered skyboxes, where as much as half of the image would not require multiple inter-reflections, we expect that even though the potential savings

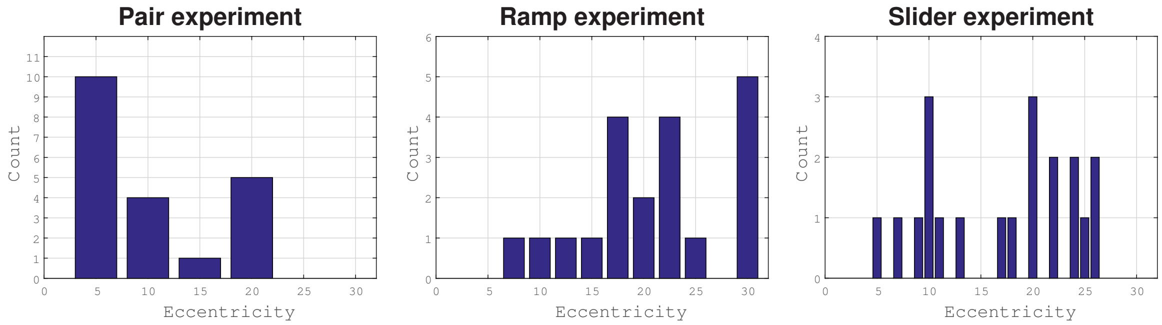


Figure 7: The histograms of the raw data for each experiment, as a function of foveal eccentricity.

would be less, they will still be significant, due to the pruning of primary rays for the rest of the scene.

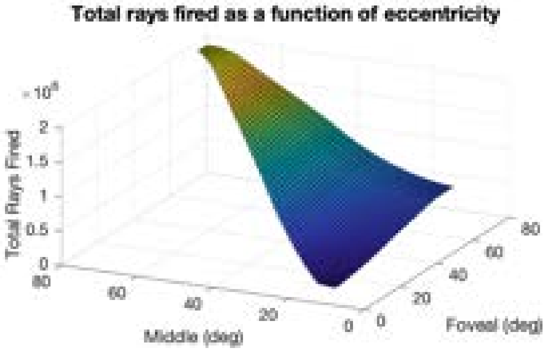


Figure 8: Total rays fired as a function of foveal and middle zone eccentricities. The peak of the plot corresponds to a foveal eccentricity spanning the entire field of view, which equates to non-foveated rendering.

5.2 Limitations

The refresh rate of our monitor, set to 60Hz, settles our minimum end-to-end latency up to 16.6ms. We measured the end-to-end latency of our pipeline and we found that, the bottleneck is the monitor, as the eye tracker and processing pipeline can function at 90 and over 200Hz respectively without any delays. The total latency of 16.6ms, although not high, might have slightly biased the measured eccentricity thresholds to higher than actual values, as any potentially perceptible manipulation due to latency, for quite rapid saccades, would have skewed the detection threshold to larger eccentricities. Certain subjects mentioned that in the occasion they blinked several times during the experiment, they were able to detect the image manipulation due to eye tracking error. A similar issue has also been reported in previous work [Patney et al. 2016] that is based on an eye tracker with the same refresh rate. Another potential limitation is the full-HD resolution of our monitor. We opted to perform our experiment on the most common resolution and screen size, currently [Anderson 2021]. In 4K or 5K high resolution monitors, the perceptibility thresholds might vary slightly. Nevertheless, a proportional, or even higher performance boost is to be expected in such high resolution monitors.

6 CONCLUSION

We develop a perceptual sandbox emulating a foveated path tracer to determine the eccentricity angle thresholds that enable imperceptible foveated path tracing. The foveation parameters are drawn from a psychophysical model describing the human visual acuity in the entire visual field. Due to current hardware limitations prohibiting real time path-tracing, we pre-render image buffers and emulate foveated rendering as a post-process. We then perform three experiments to estimate conservative thresholds of eccentricity where any image manipulation is detectable for path-traced images. We determine perceptibility thresholds for path-traced images. Three thresholds have been produced for foveal eccentricity, one from each study. The pair study has produced a mean eccentricity value of 10.25° for the foveal size; the ramp study a mean of 21° and the slider test a mean of 16.95°. We estimate that the different methodologies adopted for testing may have affected perceptual sensitivity to blur as well as the refresh rate (60 Hz) of our monitor.

We performed an analysis of the expected computational complexity reduction when such a system runs in real time. Our analysis shows a significant boost in path-tracing performance using our foveated rendering due to the reduction in the primary rays. The expected boost is 2x speed-up (foveal region of 30°, middle region 75°) or 3x speed-up (foveal region of 15°, middle region of 65°).

Future work may include the implementation of our method into a rendering system, hardware permitting, to be evaluated in real-time and further measure the effectiveness of our technique when dealing with high-polygon models or scanned point clouds of large areas acquired by drones. Investigating the latest denoisers [Chaitanya et al. 2017b; Schied et al. 2018] combined with DLSS [Hasselgren et al. 2020] and our foveated path tracing model might also further increase the efficiency & effectiveness of our approach. A potential future avenue would be to investigate the effect of latency on the visual angle where foveation in the periphery is perceivable. Finally, there is a strong potential for FPT in VR, as indicated in [Koskela et al. 2016].

ACKNOWLEDGMENTS

This research forms part of the project 3D4DEPLHI, co-financed by the European Union and Greek funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call 'Specific Actions, Open Innovation for Culture' (project code: T6YBΠ-00190)

REFERENCES

Shaun Anderson. 2021. Top Ten Most Common Screen Resolutions. https://www.hobo-web.co.uk/best-screen-size/. [Online; accessed 27-May-2021].

Patrick Baudisch, Doug DeCarlo, Andrew T Duchowski, and Wilson S Geisler. 2003. Focusing on the essential: considering attention in display design. *Commun. ACM* 46, 3 (2003), 60–66.

Kirsten Cater, Alan Chalmers, and Greg Ward. 2003. Detail to attention: exploiting visual tasks for selective rendering. In *ACM International Conference Proceeding Series*, Vol. 44. 270–280.

Chakravarty R Alla Chaitanya, Anton S Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017a. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.

Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017b. Interactive Reconstruction of Monte Carlo Image Sequences Using a Recurrent Denoising Autoencoder. *ACM Trans. Graph.* 36, 4, Article 98 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073601

Christine A Curcio, Kenneth R Sloan, Robert E Kalina, and Anita E Hendrickson. 1990. Human photoreceptor topography. *Journal of comparative neurology* 292, 4 (1990), 497–523.

Claude E. Duchon. 1979. Lanczos Filtering in One and Two Dimensions. *Journal of Applied Meteorology and Climatology* 18, 8 (1979), 1016 – 1022. https://doi.org/10.1175/1520-0450(1979)018<1016:LFIOAT>2.0.CO;2

Andrew T Duchowski and Arzu Çöltekin. 2007. Foveated gaze-contingent displays for peripheral LOD management, 3D visualization, and stereo imaging. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 3, 4 (2007), 1–18.

Masahiro Fujita and Takahiro Harada. 2014. Foveated real-time ray tracing for virtual reality headset. *Light Transport Entertainment Research* (2014).

Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D Graphics. *ACM Trans. Graph.* 31, 6, Article 164 (Nov. 2012), 10 pages. https://doi.org/10.1145/2366145.2366183

Richard Guy. 2004. *Unsolved problems in number theory.* Vol. 1. Springer Science & Business Media.

Jon Hasselgren, Jacob Munkberg, Marco Salvi, Anjul Patney, and Aaron Lefohn. 2020. Neural temporal adaptive sampling and denoising. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 147–155.

Sebastien Hillaire, Anatole Lecuyer, Tony Regia-Corte, Remi Cozot, Jérome Royan, and Gaspard Breton. 2010. A real-time visual attention model for predicting gaze point during first-person exploration of virtual environments. In *Proceedings of the 17th acm symposium on virtual reality software and technology.* 191–198.

David Hoffman, Zoe Meraz, and Eric Turner. 2018. Limits of peripheral acuity and implications for VR system design. *Journal of the Society for Information Display* 26, 8 (2018), 483–495.

James T. Kajiya. 1986. The Rendering Equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143–150. https://doi.org/10.1145/15886.15902

Louis Komzsik. 2003. *The Lanczos method: evolution and application.* SIAM.

Matias Koskela, Timo Viitanen, Pekka Jääskeläinen, and Jarmo Takala. 2016. Foveated path tracing. In *International Symposium on Visual Computing*. Springer, 723–732.

George Alex Koulieris, Kaan Akşit, Michael Stengel, Rafał K Mantiuk, Katerina Mania, and Christian Richardt. 2019. Near-eye display and tracking technologies for virtual and augmented reality. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 493–519.

George Alex Koulieris, George Drettakis, Douglas Cunningham, and Katerina Mania. 2014. C-LOD: Context-aware material level-of-detail applied to mobile graphics. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 41–49.

Sungkil Lee, Gerard Jounghyun Kim, and Seungmoon Choi. 2008. Real-time tracking of visually attended objects in virtual environments and its application to LOD. *IEEE Transactions on Visualization and Computer Graphics* 15, 1 (2008), 6–19.

Peter Longhurst, Kurt Debattista, and Alan Chalmers. 2006. A gpu based saliency map for high-fidelity selective rendering. In *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa.* 21–29.

Lester C Loschky and George W McConkie. 2000. User performance with gaze contingent multiresolutional displays. In *Proceedings of the 2000 symposium on Eye tracking research & applications.* 97–103.

David Luebke, Benjamin Hallen, Dale Newfield, and Benjamin Watson. 2000. *Perceptually driven simplification using gaze-directed rendering.* Technical Report. Tech. Rep. CS-2000-04, Department of Computer Science, University of . . . .

Joseph Mandelbaum and Louise L Sloan. 1947. Peripheral Visual Acuity*: With Special Reference to Scotopic Illumination. *American Journal of Ophthalmology* 30, 5 (1947), 581–588.

Morgan McGuire. 2017. *Computer Graphics Archive.* https://casual-effects.com/data

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

Toshikazu Ohshima, Hiroyuki Yamamoto, and Hideyuki Tamura. 1996. Gaze-directed adaptive rendering for interacting with virtual space. In *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*. IEEE, 103–110.

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.* 35, 6, Article 179 (Nov. 2016), 12 pages. https://doi.org/10.1145/2980179.2980246

Thorsten Roth, Martin Weier, André Hinkenjann, Yongmin Li, and Philipp Slusallek. 2016. An analysis of eye-tracking data in foveated ray tracing. In *2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS).* IEEE, 69–73.

Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. 2017. Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics.* 1–12.

Christoph Schied, Christoph Peters, and Carsten Dachsbacher. 2018. Gradient estimation for real-time adaptive temporal filtering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 2 (2018), 1–16.

Stanford. 2021. The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/

Hans Strasburger, Ingo Rentschler, and Martin Jüttner. 2011. Peripheral vision and pattern recognition: A review. *Journal of vision* 11, 5 (2011), 13–13.

Veronica Sundstedt, Efstathios Stavrakis, Michael Wimmer, and Erik Reinhard. 2008. A psychophysical study of fixation behavior in a computer game. In *Proceedings of the 5th symposium on Applied perception in graphics and visualization.* 43–50.

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. *Computer Graphics Forum* 35, 7 (Oct. 2016), 289–298. https://doi.org/10.1111/cgf.13026 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13026