Foveated Rendering in Virtual Reality

Samuel A. Miller
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
mill5978@morris.umn.edu

ABSTRACT

Foveated rendering is a technique in which images are displayed at differing levels of detail dependent on where the user is looking. The question this paper addresses is how best to implement foveated rendering to reduce computational load when rendering three-dimensional environments in virtual reality. The main goal for these renderers is to provide an experience in which the user is unable to distinguish whether they are seeing environments that are using these optimizations or not. Ideally you should be able to implement this technology to render the same scene with the same impact on the user, while only needing a fraction of the computing power. At the current level of development, renderers have been designed that reduce the number of pixels shaded by up to 70 percent, without the user detecting the change. In this paper, we describe two studies that use these renderers to succesfully reduce the amount of computation done with negligible change in perception by their users.

Keywords

foveated rendering, gaze tracking, multiresolutional displays, virtual reality

1. INTRODUCTION

There is an abundance of modern computer programs and media that require the generation of complex visual scenes. From entertainment like video games, to more practical applications such as any computer program that uses 3D modeling, effectively rendering three dimensional scenes has become necessary for modern computers. Due to this mainstream need for rendering power, any technique that can make the process less costly has become more important as media increases in complexity. *Foveated rendering* is one such method.

The phrase foveated rendering refers to a process that exploits the way the human eye works in order to render only what is necessary in a three-dimensional scene. Normally, the entire display is rendered at the same resolution, regardless of where the user is looking. Because the human eye only perceives great detail at the center of vision [10], this uniformity in resolution regardless of user focus is a waste of valuable resources such as computing power. Foveated rendering uses a practice known as gaze tracking to determine the center of the user's focus, then chooses regions around that centerpoint to render at progressively lower quality. Because of the aforementioned inability of the human eye to detect detail toward the outer boundaries of their field of view, these regions of lower quality in the scene are perceived the same way they would be if there was no foveated rendering involved.

This process has been discussed for decades [4], however it was limited by the primitive nature of essential components such as gaze tracking. As it currently stands, the hardware required has caught up to make this process possible, just as higher pixel densities and refresh rates are becoming increasingly more in demand. With demanding technologies such as virtual reality becoming more mainstream, foveated rendering will be an essential tool for developers.

In this paper, we start with some background necessary to understanding foveated rendering, followed by a look at the efficacy of existing approaches. Section 2 begins with an explanation of the properties of the eye that are exploited by this process. It then introduces the hardware required for applying these renderers to virtual reality, as well as some components involved in renderer design. Section 3 delves into more specific facets of designing a foveated renderer by examining two implementations. Section 4 takes a look at user studies of these implementations, with a summary of their results. Section 5 then provides some concluding remarks.

2. BACKGROUND

In order to understand how foveated renderers are developed, some background concepts must be explained. We first look at the properties of human vision that are involved, as well as how to track the motion of the eye in an existing virtual reality setup. We then examine some components of the rendering process that play a key role in how effective foveated rendering can be.

2.1 Human Vision Overview

Determining the strengths and weaknesses of the human eye is essential when developing technology that is primarily perceived visually. Foveated rendering focuses on the differences between how humans perceive the fovea and periphery regions. The fovea is an area in the central region of vision approximately 5° in diameter that contains the highest spatial density of photoreceptors in the eye, while the periphery is the surrounding area [3]. This is depicted in Figure 1 with

UMM CSci Senior Seminar Conference, December 2017 Morris, MN.

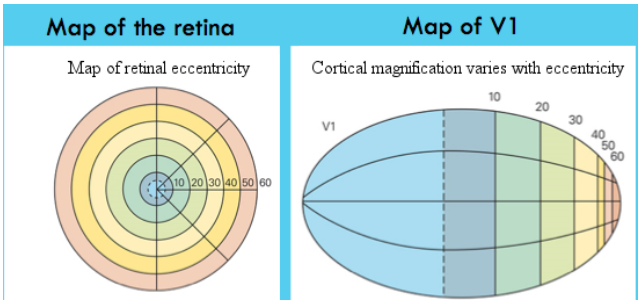

Figure 1: Representational map of the human retina and visual cortex. [5]

the fovea represented by a circle in the center of the diagram on the left with all other regions representing the periphery.

A key difference in how these two regions are perceived is that the optic nerve undersamples the retinal image from the periphery, while all data gathered in the foveal region is not undersampled. This means that the brain is fed more information by the photoreceptors at this center point, which leads the subject to build a mental image that has far more detail near the object in focus.

The difference in detail perception between the fovea and periphery is due to a concept known as Cortical Magnification Factor (CMF), where the section of the brain that is dedicated to visual processing (the visual cortex) is divided into subregions that each correspond to varying proportions of fovea and periphery. The further from the center point of vision you go, there is a progressively smaller portion of the visual cortex dedicated to processing information from that section [9]. This is shown in the diagram on the right in Figure 1, with almost half of the cortex dedicated to the fovea (the region in light blue). It is this facet of the human brain that leads to the visual acuity falloff that foveated rendering is based on.

Motion is different than detail in that it is perceived roughly the same throughout the field of view. Because of this, unintended motion anywhere in the scene can distract the user from the experience and break their immersion. If distortion causes flickering motion towards the edge of a user's field of view, they will be drawn to this new focal point and diverge from the experience designed by the developers. Because of the potential for immersion to be compromised, eliminating unintended motion in both fovea and periphery is a priority when designing renderers. [3, 10]

Additional important concepts for this subject matter are that of *retinal eccentricity* and *foveal layers*. Retinal eccentricity is the angular distance as you move away from the central focus of your vision [3]. This measure is represented in Figure 1; the units of the diagram on the left are degrees of retinal eccentricity. *Foveal layers* (also referred to as eccentricity layers) are multiple renders of a scene at different resolutions and sizes, that are then overlapped to create a composite image of varying quality. Figure 2 is a depiction of three foveal layers that are separated to show the size differences and what the overlap looks like. These layers are based on retinal eccentricity, as it is used to dictate how large each region should be according to photoreceptor density on the retina. The size of these layers varies depending on implementation, however the typical approach is to quantify CMF in some way and use that as a heuristic to determine

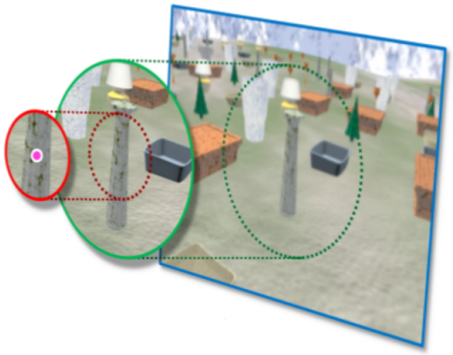

Figure 2: Visualization of nested foveal layers. [3]

which how large each subdivision should be based on the amount of visual cortex that is dedicated to processing that section.[9]

Saccadic eye movements (see Figure 3), are the simultaneous shift in focus of both eyes to a point that is a considerable distance from where they started. These movements are a critical challenge that foveated renderers must handle, as these movements are what make tracking the eye unpredictable. It is difficult to smoothly render a space based on the focal point of the eyeline if the user can shift his gaze somewhere else in an instant. Handling this weakness is one factor that held the advancement of this field back for so long, as *latency* proved to be an insurmountable restriction. Because the motion of the eyes is unpredictable, the user could make a saccadic movement and the system would exhibit a delay as the scene was re-rendered to suit the new focal point. This is demonstrated in Figure 3 by the concentric yellow circles representing the foveal layer positions of the current foveated render. In this instance, the system has not caught up to the new focal point of the user yet (as shown by the discrepancy between the render focus and visual focus in the second frame pictured).

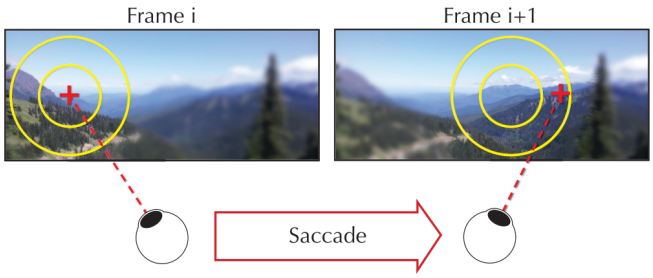

Figure 3: Visualization of a saccadic eye movement from one foveated rendered frame to another. [1]

This delay, or latency, has become less severe as hardware has advanced and computing speed has increased. Once an immersion breaking hurdle, latency has now become negligible enough that developers are able to implement foveated systems that are difficult to distinguish from normal ones (as discussed in the studies in later sections).

2.2 Virtual Reality Headsets

Virtual Reality (VR) displays have been a science fiction staple that developers have long tried to implement. There have been countless failures to make this technology feasible over the years, however it seems computing hardware has finally caught up. There has been a steadily increasing availability of affordable consumer options for VR, leading to a complementary increase in both public and corporate interest. This interest has yielded options for early adopters, such as the Oculus Rift and the HTC Vive, as well as the more casual consumer, such as Samsung's Gear VR and Google Carboard.

Figure 4: HTC Vive Virtual Reality Head Mounted Display. [2]

While all of these options have their niche in the market, in this paper we focus on the Head Mounted Displays (HMDs) with the most computing power available to them, e.g. the HTC Vive (see Figure 4). The Vive targets a market that wants applications with complex scenes as well as high resolutions. It is devices like these that have an immediate demand for any avenue of optimization in order to meet the needs of their consumer base,and will likely adopt techniques like foveated rendering first.

2.3 Gaze Tracking

Gaze tracking is the practice of using a combination of hardware and software to compute the trajectory of the user's eyeline to determine what the center of their view is focused on. This technology has been used for a variety of purposes over the years (e.g. accessibility services for those who are handicapped), and has now become increasingly in the public eye due to the commercial popularization of VR devices.

While gaze tracking hardware has been around for a suprisingly long time, it hasn't always been feasible for practical use. In 1990, Mark Levoy and Ross Whitaker [4] explored ways of utilizing gaze tracking to develop rendering optimization algorithms. Their results showed promise, with their software responding to the gaze tracking well, however the subjects were placed in bulky headwear that limited their practicality. At the time, the setup demanded an NAC Eye Mark eye Tracker (pictured in Figure 5), which covered a large portion of their head and had to be physically connected to rendering engine hardware. This hardware restriction limited what they could accomplish, as nearly the entire head was covered with technology purely devoted to gaze tracking. Integration with virtual reality would have been impossible at this time.

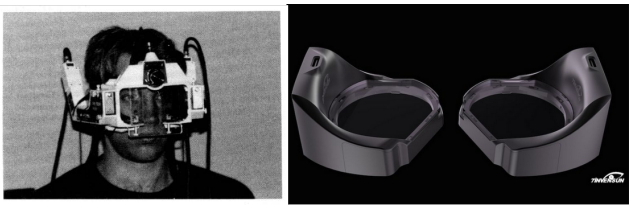

Figure 5: Pictured left: *NAC Eye Mark eye tracker* from the 1980s. Pictured right: *aGlass* eye tracker from 2017. [4] [8]

However, gaze tracking has advanced drastically over the years. 2017 is a landmark year in that there are several implementations being released for public purchase. "aGlass", developed by 7invensun from Beijing, is an affordable modular upgrade to the HTC Vive that will be available for purchase later in 2017. This product is notable in that it will be able to fit directly over the lenses of the Vive due to its compact size. Other emerging gaze tracking products, such as the Tobii Pro VR implementation for the Vive headset, are so compact that they come embedded in existing VR HMDs. The differences between these products and the previously mentioned NAC Eye Mark eye Tracker are significant enough for gaze tracking to be properly capitalized on as a technology. Developers no longer have to consider the hardware restrictions for the use of gaze tracking in foveated rendering, as soon the majority of the consumer base will already have gaze tracking functionality integrated into their devices.

2.4 3D Rendering Concepts

Some critical components involved in rendering a three dimensional scene are that of *aliasing* and *artifacting*.

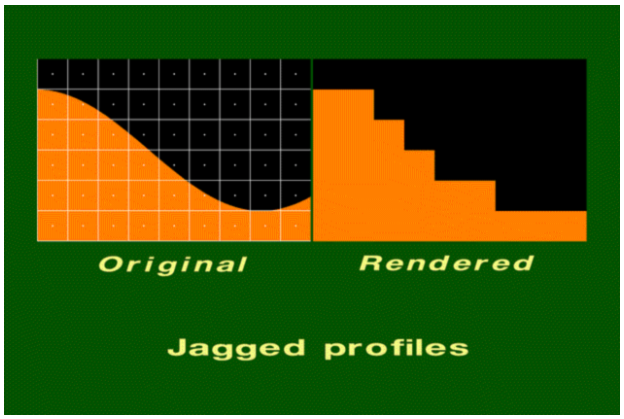

Figure 6: Depiction of a continuous analog picture converted into a discrete digital picture of separate pixels. [6]

Aliasing is a phenomenon that occurs in computer graphics due to the fact that the rendered images are discrete representations of analog constructs. This means that in order to represent an object, the rendered image is generated by picking individual sampling points and using them to create a model of the source image. The distances between these sample points typically depend on resolution/pixel position,

however if the renderer uses a sampling rate that doesn't accurately represent the complexity of the source, the frequency of the signals can become indistinguishable from one another and can then be interpreted falsely [6]. This ambiguity of signal interpretation manifests in the completed image as jagged edges, as seen in the rendered image on the right in Figure 6.

This effect is an example of artifacting, i.e. visual distortion due to complications during image processing. Artifacting and aliasing have proven troubling in the implementation of foveated renderers, as one of the methods used to lower quality in the periphery is to increase the distance between samples (otherwise known as *subsampling*). This process puts less computational strain on the system by having fewer points to shade, however it magnifies the negative effects associated with subsampling: spatial and temporal aliasing/artifacting.

Temporal aliasing is the effect of having consecutive frames of a rendered scene contain aliasing in the same general region, leading to consistant visual distortion over time. Areas containing this consistant distortion are referred to as *temporal artifacts*, and are essentially objects that can attract the user's attention and break their immersion by distracting them from the intended experience of the scene. Because the human eye detects motion uniformly throughout the field of view [7], these artifacts are problems no matter where they occur in the scene.

Due to the prevalence of this distortion in the process of foveated rendering, developers must strike a balance between lowering the quality in the periphery enough to yield savings on resources used, while avoiding these immersion breaking artifacts.

3. FOVEATED RENDERER DESIGN

As previously discussed, the main goals of foveated renderers are to save on computational resources and to keep the impact subtle enough that the user cannot detect it.

Perhaps the first notable work that actually claimed significant performance advantages in graphical rendering from this technology was that of Guenter et al. in 2012 [3]. They claimed that their renderer could produce results indistinguishable from full-resolution images, yet with around one fifth or sixth of the computational power previously required.

3.1 Desktop Display Foveated Rendering (2012)

In 2012 Guenter et al. attempted to validate previous research in the field by using what was cutting edge technology at the time to bridge the gap between theory and practice. Hardware had finally gotten to the point that they could take advantage of the work put forth by the likes of Levoy and Whitaker to make a renderer that capitalized on the opportunity for low cost optimization via foveated rendering.

3.1.1 Renderer Design Goals

Guenter et al. aimed to obtain considerable speedup in rendering times while also avoiding any noticeable spatial or temporal artifacts. They conducted numerous user studies in order to verify that users had difficulty distinguishing between the foveated and non-foveated renders. The team also had to deal with a self-imposed constraint in that they wanted their solution to have the capacity to be retrofitted to improve existing applications. This meant that their method of handling aliasing had to be effective while also requiring limited modification to how the native applications handled pixel shading. They had to strike a fine balance between being specific enough to address the artifacting problem, while also being applicable to a range of applications [3].

3.1.2 Techniques Used

Guenter et al. developed their foveated renderer for a desktop computer with a 1920 x 1080 resolution display, supported by a Tobii TX300 Eye tracker. Their system was tested on a virtual moving camera through a static threedimesnsional scene.

In this instance, successful optimization was achieved by minimizing system latency, handling aliasing in the periphery, and finding the ideal balance of size and resolution for their eccentricity layers. They were able to reduce the number of pixels shaded due to their implementation of the following:

- temporal jitter of the spatial sampling grid: the shifting back and forth of the eccentricity layer by small distances between frames to reduce artifacting.
- temporal reverse reprojection: blending of previous and current pixel values to increase the effective sampling.

Essentially, these two techniques amount to a smoothing of the jagged, aliased edges described before. The temporal jitter allows samples to cover a larger portion of the source image so that it can be more faithfully represented, However on its own it provides images that are constantly shifting, which triggers unwanted motion that is detectable by the user. This unintended motion of the image is eliminated by the second technique: temporal reverse reprojection. This process is basically just a way to average consecutive frames, so any pixels that were rapidly alternating colors due to temporal jitter would then be a constant intermediate shade. When combined, the resulting image has reduced aliasing, with no unwanted flickering motion.

Guenter et al. also used Multisample Anti-Aliasing (MSAA) to help combat distortion due to aliasing, however for the sake of brevity this will not be covered in this paper, more information on this subject can be found in their work. [3]

Guenter et al. became among the first developers to fully realize the potential of foveated rendering; they were able to reduce the number of pixels shaded by a factor of 10-15 while still presenting scenes of acceptible quality as determined by their user study [3]. Their system is even applicable to modern renderer setups, which is discussed later in the paper. However, it is worth noting that they were still limited by the technological constraints of their time:

> Ideally, the graphics card would render at full display resolution where the gaze is centered and continuously decrease resolution outward from there. A more efficient method on current graphics hardware is to approximate this ideal by rendering several overlapped rectangular regions called eccentricity layers. [3]

This hardware-imposed weakness to their approach is one of the areas improved upon by subsequent efforts by other developers building on their work.

3.2 Virtual Reality Foveated Rendering (2016)

Four years after the publication of Guenter et al. [3], Patney et al. [7] brought the technology of foveated rendering to the burgeoning field of virtual reality. Guenter's team proposed that the true benefits of this rendering technology would be most apparent when the display had a larger field of view (FOV), and virtual reality head-mounted displays (HMDs) boast the largest FOV in any display to date: true 360°. Virtual reality is a perfect match for foveated rendering as many headsets already have gaze-tracking built in, so implementing these algorithms yields significant gain for very low cost.

Patney et al. set out to bring the speedup of foveated rendering to a field where the computational demand is rapidly rising. Any potential for increased computational power would have significant impact on the capabilities of VR.

3.2.1 Renderer Design Goals

Patney et al. [7] specifically call out the aforementioned work from 2012 of Guenter et al. [3]:

> Prior foveated renderers like Guenter et al. ... focus on practical near-term techniques to reduce costs without explicitly identifying and minimizing perceptible artifacts introduced by foveation. Our studies show they often exhibit significant head- and gaze-dependent temporal aliasing, distracting users and breaking immersion. [7]

Having specified weaknesses in previous approaches, Patney et al. worked to find new methods that would minimize temporal aliasing and any sensation of *tunnel vision*: only perceiving visual stimuli from the central field of view due to reduction in peripheral detail. In order to identify which foveation technique best achieved these goals, Patney et al. developed a prototype to test different approaches. This prototype was referred to as a Perceptual Visual Target.

3.2.2 Perceptual Visual Target

Patney et al. differ in their design process from some of their predecessors in their use of a *Perceptual Visual Target* [7]. Essentially, their design team created a demonstrational model that emulated different versions of foveated rendering using pre-processed scenes. They then conducted user studies to confirm which of their intended optimizations would be effective. Once they had a user-approved renderer emulation, they began the design process with the goal of being able to render what was shown in the perceptual visual target, albeit in real time.

This process of guiding their development lead to impressive overall user satisfaction, with their renderer yielding superior results in a number of facets.

3.2.3 Techniques Used

Patney et al. built directly on the existing work of others by adding parameter tweaks and supplementary algorithms. For example, rather than render three nested foveal layers like the approach of the Guenter et al. (pictured in Figure 2), they utilize a piecewise linear variation of shading rate in a single layer.

In the graph shown in Figure 5, the foveal layers of Guenter et al. are represented by the solid orange lines at differing sampling factors. The dashed orange lines represent the transitional blend applied to turn multiple foveal layers into one smooth composite image. In contrast, the dashed blue line represents the single layer of Patney et al. with one transition from high to low detail starting right before 20° of retinal eccentricity.

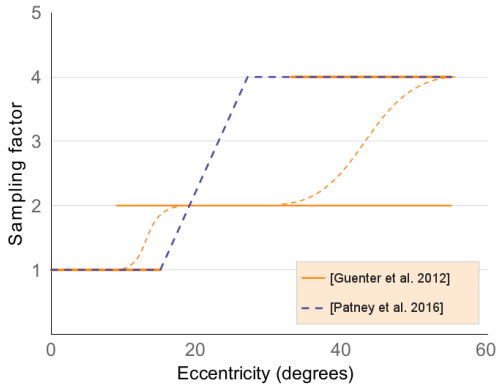

Figure 7: Top: visualization of nested foveal layers. [3]. Bottom: comparison of sampling approaches by Patney et al. and Guenter et al. [7]

Rather than having to transition between three layers with different sample rates, Patney et al. apply one filter that shifts pixel shading rate from their bounds of maximum detail (base sampling factor) to minimum detail (four times the distance between samples).

This alternate approach enables them to transition to a lower level of detail at smaller retinal eccentricities, which leads to lower computational demand due to fewer pixels being shaded. (See Figure 8).

Additionally, they differ from other approaches in that they are sampling at full resolution throughout the entire image rather than sampling different layers with multiple resolutions such as the approach of Guenter et al. This decision ended up requiring them to use a new algorithm, *variance sampling*, to improve visual fidelity. Variance sampling is a post-process image reconstruction algorithm developed by Patney et al. to reduce/eliminate temporal aliasing in their end product. For the sake of brevity, this technique is beyond the scope of this paper. More information can be found in the work of Patney et al. [7]

4. RESULTS

While Guenter et al. conducted a user study to gauge the efficacy of their product, they used hardware that is now outdated which makes it difficult to directly compare their results to more modern offerings such as Patney et al.

Fortunately, Patney et al. conducted a user study directly comparing their foveated renderer to that of Guenter et al.'s renderer, using the same hardware setup. This study effectively highlights the differences between the two approaches and demonstrates how far the field has progressed in such a short time frame. In this section we discuss the setup of the experiment and analyze the reasons for the difference in results.

4.1 User Study Setup

Patney et al. conducted a test of four users. They presented the user with two versions of a scene (one foveated

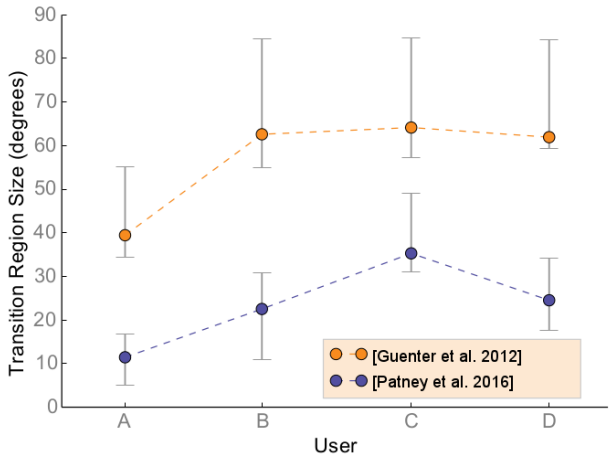

Figure 8: Comparison of the Transition Region Size used in Patney et al.'s user study [7]. The points here represent the degree of retinal eccentricity at which each of the four users could detect the presence of foveation. The error bars represent outliers in the trials, in which the user's ability to detect foveation was irregular when compared to the rest of their results. These outliers could have been due to identifying foveation by chance instead of when they noticed it, or if the user had cheated and looked away from the intended gaze point.

and one non-foveated) and asked the subject which scene looked better. The scenes were presented in a randomized order and were varied using different transition region sizes. This variation was accomplished by altering the size of the region between the foveal and the peripheral regions closer to or farther from the center. In Figure 8 this change would present as the shift towards larger sampling factors occuring at smaller or larger degrees of retinal eccentricity; the departures from base sampling factor would occur sooner or later depending on the intention of the trial. This was done to see which system could best present images with lower detail before the subject noticed the foveation.

The study applied both systems to a head-mounted display setup in order to confirm their hypotheses about the efficacy of foveated rendering on VR, and to keep results consistant. All four users were subjected to two hundred trials each which followed a one-up two-down staircase-method of variation, in that they would increase the transition region size for one trial, then decrease it for the next two to reduce training their subjects to expect constant increase. Reverse staircases of the same pattern were also used.

4.1.1 User Results

As seen in Figure 8, all of the users tolerated lower detail closer to the fovea in the scenes using Patney et al.'s setup. This validates their use of a perceptual visual target as well as their efforts towards minimizing temporal artifacts when lowering quality in the periphery. Not only did their system appear nearly indistinguishable from the non-foveated version when set to the right transition region, they also were able to shade 70 percent fewer pixels than the non-foveated scene.

Their study confirmed that they were able to use lower quality shading up to 30° closer to the fovea than Guenter et al.'s multi-resolution approach, implying that their piecewise linear shading rate is more effective than the three foveal layers of varying resolution used by Guenter et al. [7]

5. CONCLUSION

In this paper we discussed the efficacy of foveated rendering and gave a sense of how feasible integration with existing platforms could be. As the hardware required for such a system becomes more commonplace, this process will receive increasing amounts of attention and development. The most recent research implies that a temporally-stable foveated renderer implementation such as that developed by Patney et al. [7] is a proven method for reducing computational workload while still preserving the user's perception of normality when rendering scenes.

6. ACKNOWLEDGEMENTS

I would like to thank Nic McPhee, Kristin Lamberty, and Kristin Shirley for their excellent feedback and advice during the writing of this paper.

7. REFERENCES

[1] R. Albert, A. Patney, D. Luebke, and J. Kim. Latency requirements for foveated rendering in virtual reality. *ACM Trans. Appl. Percept.*, 14(4):25:1–25:13, Sept. 2017.

[2] D. Forrest. Tobiis integrated VR HMD eye tracking: Hands-on, 2017. [Online; accessed 8-October-2017].

[3] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. Foveated 3D graphics. *ACM Trans. Graph.*, 31(6):164:1–164:10, Nov. 2012.

[4] M. Levoy and R. Whitaker. Gaze-directed volume rendering. *SIGGRAPH Comput. Graph.*, 24(2):217–223, Feb. 1990.

[5] E. Morson. Your brain is made of maps, 2017. [Online; accessed 15-October-2017].

[6] G. S. Owen. Overview of aliasing in computer graphics, 1999. [Online; accessed 13-October-2017].

[7] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Trans. Graph.*, 35(6):179:1–179:12, Nov. 2016.

[8] A. Robertson. The HTC Vive will track eye movement with a 220 upgrade kit, 2017. [Online; accessed 10-October-2017].

[9] N. T. Swafford, J. A. Iglesias-Guitian, C. Koniaris, B. Moon, D. Cosker, and K. Mitchell. User, metric, and computational evaluation of foveated rendering methods. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '16, pages 7–14, New York, NY, USA, 2016. ACM.

[10] L. N. Thibos. Image processing by the human eye. In *Visual Communications and Image Processing IV*, volume 1199, 1989.