Rendering Optimizations for Virtual Reality Using Eye-Tracking

Shawn Matthews and Alvaro Uribe-Quevedo
Faculty of Science
Ontario Tech University
Oshawa, Ontario L1G-0C5
shawn.matthews@ontariotechu.net alvaro.quevedo@uoit.ca

Alexander Theodorou
Neurofit VR Inc.
Toronto, Ontario
alex@neurofit.ca

***Abstract*—Optimizing rendering in virtual reality is an open problem in computer science. The nature of modern VR display technology (high refresh rate and increasing pixel density), coupled with the relatively slow growth modern compute capability, is leading to a bottleneck in VR performance. As we further research methodologies for improving rendering performance and accuracy for VR, it is important to understand the historical approaches and where they succeeded or failed in their approaches. Some implementations will double computing because of the need of stereoscopy, and thus have higher overhead for rendering. This can be improved with Multi-View Rendering where the GPU hardware can assist in duplicating rasterization for multiple views with differing projections. More recently, perception-based rendering has gained traction, which can be further accelerated using Variable Shading Rate or Multi-Rate Shading technology found on more recent GPUs. There has also been some success in using deep neural networks to assist with transmitting foveated content over a network. The advances in the field leave many open research questions, including sparse pixel rendering, driving user attention, and techniques and methodologies for combining variable shading rate images. This review focuses research associated with rendering optimizations for virtual reality using eye tracking, since it is becoming a feature present in consumer-level head-mounted displays. From our review, affordable off-the-shelf virtual reality and eye tracking are both leading to freeing up rendering resources towards improved performance and visual fidelity, as well as providing new and exciting opportunities for human-computer interaction.**

***Keywords*—virtual reality, perception, foveated rendering, multi rate shading, multi-view rendering**

I. INTRODUCTION

Virtual Reality (VR) is becoming a mainstream technology adopted in entertainment, education [1], and well-being [2], amongst others. With the rise of consumer level VR, and especially with standalone VR headsets with lower graphical compute capabilities, there rises the problem of increasing visual fidelity without increasing the rendering cost. This problem becomes more apparent when using Head Mounted Displays (HMDs), where framerates and response times must be kept high to minimize latency that can cause motion sickness or 'Cyber-Sickness' due to the mismatch between what is seen and what is felt [3]). Moreover, as visual fidelity increases, attention to detail within VR experiences can lead to higher cognitive load and overwhelming experiences where user struggle to figure out the next steps in the virtual world [4].

Early VR implementations duplicated all render commands at full resolution to provide the stereo pair needed for the left and right eye point of view to create the VR visual cues, thus doubling the required compute power to render a scene. As a result, research on diverse methodologies and techniques have been focusing on optimizing the rendering performance in VR, which we will explore in this paper. Furthermore, more recently, high-performance eye tracking is gaining momentum as a complementary tool to improve the user experience and reducing compute power employing foveated rendering. The current leading area of research in this field is perception-based rendering, where GPU compute resources are allocated to areas that have a higher impact on user perception, such as areas of high contrast or in the foveal region when eye-tracking hardware is used.

In this paper, we present a review of both the history of VR and its connection to eye tracking hardware, leading to novel approaches to VR rendering focusing on perception-based rendering techniques. The goal of this paper is to provide a basic understanding of the challenges and advances that have been made in the field of VR rendering. This review will allow us to identify some open areas of research and related applications aligned with VR hardware trends.

II. HISTORY OF VR AND EYE-TRACKING HARDWARE

This section presents a review on VR and eye-tracking hardware. Our search was conducted employing databases available at Ontario Tech University and Google Schoolar.

A. History of Virtual Reality

The understanding of depth perception led to identifying the stereopsis area that is fundamental to the development of VR Head Mounted Displays (HMD) [5]. Table II-A is by no means meant to be extensive, but rather to highlight the fact that research into virtual reality and related topics has been ongoing for over 150 years.

Looking at the information on Table II-A, the development of VR appears to come in waves, with significant waves in the pre-war period, 1960's, late 70's and into the 80's, and most recently in the 2010's. Interest in VR dates back significantly

978-1-7281-9231-4/20/$31.00 ©2020 IEEE
DOI 10.1109/SVR51698.2020.00066

TABLE I
GENERAL TIMELINE WITH SOME SIGNIFICANT ADVANCES IN VR.

Year	Event.
1838	Stereoscope [6].
1929	Link Trainer flight simulator, widely used in WW2.
1939	View Master is patented.
1954-62	Sensorama.
1960	Telesphere Mask.
1961	Headsight
1965	Ivan Sutherland - The Ultimate Display [7]
1969	First use of the term artificial reality.
1979	McDonnel-Douglas HMD for pilots.
1987	VPL popularizes the term virtual reality.
1987	VPL develops and sells dataglove, EyePhone.
1993/94	Sega introduces VR glasses to mass market.
1995	Nintendo virtual boy brings competition to the consumer VR marketplace.
1999	The matrix brings simulated realities into the mainstream eye.
2012	Oculus kickstarter begins.
2014	Sony announces PSVR, facebook buys occulus, Samsung Gear VR, Google Cardboard.
2016-17	VR boom, Rift, HTC Vive, Razer OSVR, PSVR, Fove, Pimax4k, etc...
2018	Standalone VR replaces mobile VR.
2019-20	Second wave of desktop VR (Rift S, Index, Vive Cosmos, Pimax8k, etc...), rise in high fidelity standalone VR.

though, with examples of the first VR experiences consisting of two stereoscopic images viewed through a viewfinder in 1832. Although simulation gathered traction in flight training as it allowed exposing trainees to scenarios otherwise impossible in real life. The earliest specialized hardware for was the Link Flight trainer used to train WW2 pilots. General purpose VR displays are theorized in 1965 by Ivan Sutherland [7], and initial versions developed by VPL in the 1980's (most notable being the EyePhone). The early 90's also saw an attempt at consumer-level VR with both Sega and Nintendo attempting to launch VR products, with Sega failing to launch its Sega VR system due to engineering issues, while the Nintendo Virtual Boy was a commercial failure. In terms of hardware, the Virtual Boy used a rotating mirror to convert a single line of pixels into a full field display, with each eye having its own line of pixels. Due to the relative novelty of LED technology, and the high cost of blue-light LEDs, the system launched with only red LEDs. The commercial failure of the Virtual Boy, as well as the lack of technology likely led to the lull in commercial VR development between 1995 and 2014, where most HMDs were expensive and aimed and industry and research, such as the Olympus LCD Eye-Trek ($500 USD in 2001), Daeyang cy-visor, ProView XL35 ($19,500 USD at launch), the NVIS virtual binocular ($19,000 USD at launch), the Boom 3C floor mounted display, the Windows VR monoscopic display, autostereoscopic, single CRT stereoscopic displays, workbench displays, and stereoscopic volume displays [8].

The advancement on mobile computing led to the development of cost-effective VR headsets such as the Google Cardboard that created an opportunity for everyone to experience the virtual worlds [9]. In 2012, Oculus started a Kickstarter for a new consumer VR headset, now using modern displays and tracking hardware. This indicated the start of a VR arms race, with Razer releasing the OSVR HMD for developers in late 2015. By 2016, an explosion in VR hardware can be seen, with products from multiple manufacturers, including industry leaders such as Valve, HTC, Sony, and Oculus. This time also has the rise of mobile VR with products such as Google Cardboard and Samsung VR hitting the consumer market. The first HMD with integrated eye tracking, the Fove 0 was launched in 2017, albeit with some limitations in platform support and field of view, resulting in only moderate commercial success. In 2018, standalone VR began to replace mobile VR, with Android VR emerging as a common platform for mobile VR development. 2019 brought the rise of VR headsets with integrated eye tracking, with the Vive Pro eye and Vive Cosmos arriving featuring Tobii eye trackers. 2019 also saw the first 8K VR headset (the Pimax8k), although bandwidth and processing limitations mean that the signal is up scaled from a 2460x1440 signal.

B. History of Eye Tracking and Gaze Tracking

Understanding the human gaze is an important tool for both data collection and optimizations within rendering systems. The ability to accurately track the human eye and gaze allows us to better understand the areas of interest or artifacts within our systems, and high latency gaze tracking can be used to improve our rendering performance using concepts such as foveation (as discussed later in this paper). Eye Tracking and Gaze Tracking are differentiated by what information is gathered. When discussing eye tracking in technology such as VR, we are usually referring to Gaze Tracking instead. Eye Tracking is the act of simply tracking the motion of a subject's eyes relative to their head, whereas gaze tracking additionally tracks the subject's head to determine where in the real world or virtual world the subject is looking.

Research into eye tracking and gaze detection started in the early 1900s, with the first rudimentary eye tracker implemented by Edmund Heuy, which relied on a physical apparatus affixed to the eye itself to track the user's gaze [10]. This was the first work aim at understanding the underlying mechanisms for human perception, and led to the discovery of saccades, and opened the door to further questions about the Human Visual System. Sometime between this point and 1937, photographs of a subject's eyes were used to track their gaze, although more accurate and less intrusive methods were sought after. In 1937, Guy Thomas Buswell reflected beams of light off subjects' eyes and recorded the reflections on a piece of film, creating the first non-intrusive eye tracking solution. [11]. 1947 saw the first proposal for head mounted eye trackers, with a microscope and light attached to the subjects head [12]. This style of eye tracker would become popular, and even modern desk or camera mounted eye trackers have difficulty competing with the precision of head-mounted eye trackers.

In 1967, Alfred Yarbus published his book "Eye Movements and Vision" [13], which forms the basis of much of the modern research into eye tracking and gaze detection, and lead to an explosion in eye tracking research in the 1970's [14].

Much of this work focused on improving the performance or accuracy of existing solutions. In the 1980's, focus shifted to uses for eye tracking as a method of human-computer interaction, a deviation from its traditional use as a research tool [15]. Trackers have since been developed to be less intrusive, often relying on a webcam, or dedicated IR cameras such as those found in the Tobii eye trackers. Most modern eye tracking solutions work by bouncing infrared light off the user's pupil and measuring the areas of highest contrast, much like Buswell's implementation from 1937. Additionally, due to the high processing cost for images as well as a desire to improve performance, Convolutional Neural Networks (CNNs) are now being used to process eye images to determine eye directions, with large datasets being collected by industry partners such as NVidia [16].

III. TRADITIONAL APPROACHES TO VR RENDERING

The simplest approach to rendering in VR is to simply duplicate the draw calls for each eye, transforming and rasterizing the geometry from each eye's perspective separately. Aside from doubling the amount of work that the GPU needs to do, this also introduces additional overhead in terms of scene traversal or state switches (see Algorithm 1). For instance, in order to render a mesh for two different viewports, this would require either switching the rendering target and view-specific shader parameters between each eye for each instance in the scene, or traversing the scene twice (once for each eye). Many of the more traditional approaches to VR optimization focus on reducing the amount of state changes or combining shading results for fragments that are visible from multiple viewports.

Algorithm 1: Psuedo Code of naive approaches.

```
begin Method 1:
    Bind Left Eye FBO and Uniforms;
    foreach Entity in Scene do
        Render Entity;
    end
    Bind Right Eye FBO and Uniforms;
    foreach Entity in Scene do
        Render Entity;
    end
end
begin Method 2:
    foreach Entity in Scene do
        Bind Left Eye FBO and Uniforms;
        Render Entity;
        Bind Right Eye FBO and Uniforms;
        Render Entity;
    end
end
```

A. Multi-View Rendering

A relatively early approach to bypass these limitations was to duplicate the geometry per-view on the GPU side. Originally this was achieved with geometry shaders and combining multiple views into a single output texture [17]. This allows for single-pass scene traversal with minimal state switches but still suffers from duplicated fragment shader calls. This approach was codified into the OpenGL standard (a widely used cross-platform open-source graphics library) with the introduction of OVR_Multiview in 2018 [18]. This extension to the OpenGL standard does not define a specific technique for implementation and leaves it open for the driver developers to implement. This extension was further expanded upon by NVidia with the introduction of Single Pass Stereo, which was introduced with the Pascal Architecture and provides hardware support for stereo rendering where both views are co-planar and share a common x-axis. This was again improved in the Turing architecture with the introduction of Multi-View Rendering (MVR), which adds support for more than two views as well as non-coplanar views [19]. This has also found uses outside of VR such as with cascading shadow maps [20], where there are similar requirements for geometry to be rendered from multiple perspectives.

IV. PERCEPTION-BASED RENDERING

Perception-based rendering has been a goal of graphics researchers for many years. It refers to a set of methodologies and techniques that aim to reduce the computational cost of rendering by leveraging the limitations of the human visual system. Focus on perception-based rendering has seen a resurgence with the introduction of consumer level VR, which has a high cost of rendering compared to traditional 2D displays. The goal of a perception-based rendering method is to generate an image with less computational resources that is imperceptible from an image generated with full computational resources.

This field of research borrows heavily from research into the psychophysical aspects of the human visual system, with some simplifications and generalizations made. It is well understood from previous research [21] that the human eye can be categorized into three main regions: the foveal, inter-foveal and peripheral regions (see Figure 1). The foveal region is approximately 5.2° from the center of the retina and has a high density of color-receiving cones, and a lower amount of rods (contrast sensitive photo-receptors), which leads to the fovea excelling at visual acuity and color accuracy. The inter-foveal region extends from approximately 5.2° to 17° for temporal vision, and 9° for nasal vision, and is marked by a sharp decrease in cone density, with a large increase in rod density. These two regions constitute 'central vision' and are responsible for most of the visual acuity. Beyond these regions there are no cones and rod density fall off steeply, referred to as the peripheral region. It is also worth noting that the periphery shows no decrease in ability to detect motion, which may have applications when attempting to guide user attention. Perception-based rendering aims to leverage these attributes and limitations of the human visual system to provide shortcuts for rendering techniques. The most promising of these fields is foveation, where rendering resources are allocated mainly in the foveal region, but there have been some promising results using contrast to guide rendering to higher-contrast areas. [22]

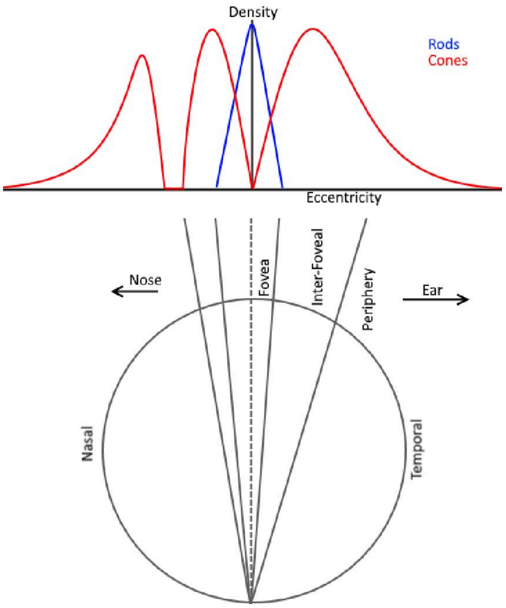

Fig. 1. Simplified Eye Model.

One area of research for alternative uses for perception-based rendering is imperceptible scene modifications, such as moving geometry to better suit a user's needs [23], or rotating the user within the scene during blinks [24], thus leveraging both perception-based rendering and gaze tracking technology, and is still an open area of research.

A. *Foveated Rendering & Multi-Rate Shading*

Some of the earliest work towards foveation focused on modifying the number of polygons rendered for a given area of the screen. One of the earliest references to foveated rendering comes from Irene Cheng from the University of Alberta. [25] Cheng's model used surface information obtained from a 3D scanner and allowed for a user to select a foveal point. Details were added or removed from a 3D model based on the distance from a given vertex to the foveal point. This was not intended as a real-time rendering optimization, but rather a way to simplify a mesh for transmission over a network. This had issues with performance and could lead to visual artifacts such as the silhouette of the object changing as the foveation point changed, leading to noticeable popping artifacts, making this approach unsuitable for gaze-based foveation. The concept of being able to reduce mesh complexity outside of foveal regions has remained appealing though, with research as late as 2020 [26] still attempting to achieve adequate results in performance and perceptual quality.

Some level of real-time foveation was achieved by Guenter et al in 2012 by using multiple layers of images, each with decreasing resolution to represent the concentric foveal regions [27]. This technique pioneered the concept of *Gaze Aware Foveation*, where the user's gaze data is captured and used to drive a foveated renderer which allocates more resources to the area around the point of focus. This model rendered to three layers of images at consecutively lower resolutions and larger viewport sizes, which are then blended together based on the distance from a given fragment to a foveal point (see Figure 3. This method suffers from the fact that the scene needs to be rendered multiple times, and fragments in the foveal region are evaluated multiple times with the results in lower-resolution layers being discarded (see Figure 2).

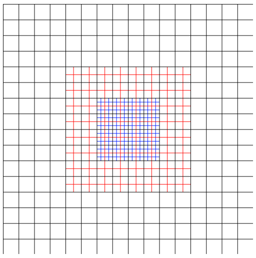

Fig. 2. Visualization of pixel density in Guenter et al.'s Implementation.

A major limiting factor to gaze-aware foveation is the latency requirements for the technique to be unnoticeable to the user. Research from Albert et al. (2017) shows that the eye-to-photon latency (that is the time between capturing an eye movement and the correct photon hitting the retina) must be below 100ms in order to be imperceptible [28]. Human eye motion tends to also be extremely quick, with saccades (rapid jumps between points of interest) exceeding 300° per second, which requires the simulation to respond quickly to eye movements. This requires modern compute capabilities as well as high-performance eye trackers, which have only recently entered the consumer market.

Another limitation of the Guenter et al. (2012) implementation was associated with the hardware. GPUs at the time only allowed fixed sample rates for an image, requiring them to render the image multiple times to multiple different images. This raises the computational overhead for scene traversal and state switches but would be resolved later with the introduction of Multi-Rate Shading. The eccentricities for the foveal and inter-foveal regions also tend to be larger than theoretically required, due to issues with contrast, aliasing, and the variability of the human eye. As well, the loss of contrast information in the periphery causes a *tunnel vision* effect that can disorient some users, and make the foveation obvious to the end user. The loss of contrast issue was mostly resolved by the introduction of a post-processing pass introduced by Patney et al in 2016, [29], which introduces variance sampling as a post-processing anti-aliasing effect. This method takes contrast, saccades, and temporal stability into account, and is applied after the scene is rendered to correct some of the issues caused by variable shading rates.

Another optimization that can be applied to gaze-aware foveation is considering the contrast in the scene, which was demonstrated by Tursun et al. in 2019. This builds upon the work done by Patney et al, and factors in contrast information

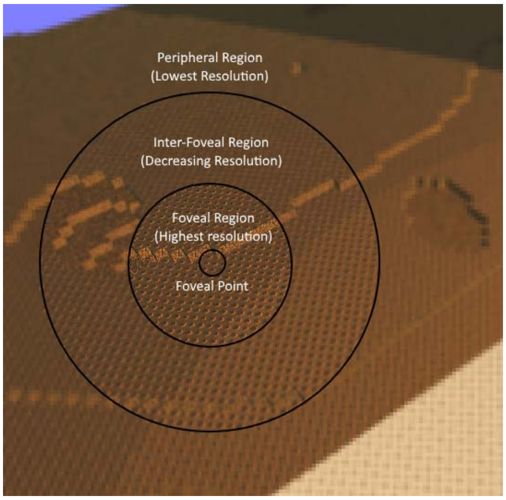

Fig. 3. Simplified Visualization of Gaze-Aware Foveation, generated using Gaussian blur in an image editing tool.

when determining the shading rate for a given region, reducing the number of samples needed for low-contrast regions within the foveal or inter-foveal regions. [30] This technique requires rendering the scene at a lower resolution prior to rendering at full resolution, but opens up the possibility of integrating other optimizations, such as motion aware variable rate shading, where object or camera motion will reduce the rendering quality for objects in motion relative to the eye. [31]

Many of the issues with Multi-Rate Shading were resolved with the introduction of hardware support for variable shading rate images. The implementation was originally proposed by He et al. in 2014, and provides a methodology for selecting how many samples are taken for a given fragment in the final image. The original model uses three levels of shading, corresponding to a single pixel, 2x2, or 4x4 block of pixels [32]. This was first implemented in hardware by Nvidia when they added support for multi-resolution shading on Turing GPUs with Variable Rate Shading. The NVidia implementation added more options for the sample arrangement, with 1x2, 2x1, 4x2 and 2x4 sample rates being supported. [19]

More recently, as of late 2019 to early 2020, some research has been done on providing additional supports for foveated rendering directly on the display driving hardware, as can be seen in the work done by Park et al. [33] Their implementation allows for a foveated image with reduced vertical resolution to be sent from the GPU to the display driver, where additional hardware will decode the compressed data into a full resolution image for display. Such advances in display technology allows for lower bandwidth between the GPU and display drivers, potentially allowing for an increase in response times or display resolution.

B. AI Powered Foveation

Recently, research has been presented that focuses on compress-ability and transmission of foveated content. The most promising of these leads is DeepFovea, which was developed by Facebook in 2019 [34]. Their approach uses a Deep Neural Network (DNN) to take sparse pixel sets which are generated from a video stream and uses the sparse data to create a temporally stable image. The density of the sparse pixel set varies across the image, with the highest density of pixels in a foveal region. The researchers claim that this results in a compression rate of approximately 14x, which opens the opportunity for streaming high framerate foveated content. It is important to note that this model was trained with natural images and would need further training or modification to support non-natural scenes such as those found in video games. Their approach also started with a full resolution video, and generated a sparse pixel set from that image. Further research will be required to determine if DNN powered foveation can be applied to real-time rendering techniques. Table IV-B presents a comparison of diverse perception-based approaches.

TABLE II
COMPARISON OF PERCEPTION-BASED APPROACHES

Tessellation Based Foveation	Works on vertices of a mesh as opposed to fragments. Can reduce rasterization time for complex meshes but produces noticeably perceptual artifacts that are hard to overcome. Also provides minimal benefits to fill rate limited applications.
Guenter et al	Uses the users gaze to allocate rendering resources. Requires multiple render passes/layers, lots of state switching, loss of contrast in periphery leads to a tunnel effect.
Patney et al	Improves upon Geunter et al to reduce tunnel effects by maintaining contrast in the periphery.
Tursun et al	Further improves upon Patney et al, can change resolution within a region based on changes in contrast, reducing artifacts and tunnel effect.
DeepFovea	Uses neural networks to decode sparse pixel sets into a foveated image, allowing for reduced bandwidth for foveated content.

V. OPEN AREAS OF RESEARCH

VR continues rapidly evolving field of research, and there are currently many open questions for improving rendering performance. A few of these questions have been selected to expand on below, but this is by no means a representation of the amount of open questions in the field.

A. AI Assisted Sparse Pixel Rendering

Another area of research that may yield interesting results in VR optimization is sparse pixel rendering, where a sparse pixel set is rendered in a traditional fashion, and the rest of the image is generated using neural networks. It may be possible to integrate aspects of Multi-Rate Shading and AI Powered Foveation to achieve this, where blocks of pixels are rendered at full resolution and used to generate the image using a DNN. It would also be an open problem to represent sparse pixel sets efficiently in GPU hardware, as current GPUs have a rigid and well defined layouts for images which lets them optimize their workflow using SIMD pipelines [35]. A sparse pixel image would need some method for effectively representing a given Texel location within an image, and in a way that can be processed efficiently by SIMD processors. Another approach might use a full resolution image but use ray casting to generate a sparse image instead of the traditional rasterization approach. This would also avoid the cost of unpacking a sparse pixel set into image space.

B. Eye-Dominance-Guided Foveated Rendering

Recent research done by Meng et al [36] suggests that it is possible to leverage ocular dominance to further improve performance with foveated rendering by increasing or maintaining visual quality for the user's dominant eye while reducing the quality for the non-dominant eye. In this model, each eye has foveation parameters specified, with the non-dominant eye receiving more foveation (lower accuracy) than the dominant eye. When foveation parameters are properly selected for the user the result is a perceptually identical image but requiring less rendering time for the non-dominant eye, resulting in performance increases of up to 47% in their implementation.

C. Driving User Attention

Another area for possible research is in leveraging the findings of perception-based rendering research to drive user attention in a way that does not impact immersion. Many of the approaches listed here focus on using gaze information to guide rendering resources to the user's area of focus, but there is very little research done on reversing this and having the simulation guide to user's attention. It may be possible to use contrast, aliasing, and foveation in a way that draws a user's attention to a given point of interest. One possible solution may be to artificially introduce aliasing or contrast in the inter-foveal region to induce a saccade response and move the artificially salient region until it aligns with the point of interest. This could provide guidance to the user in high-complexity virtual environments in video games and training simulations and may have a positive effect on immersion.

D. Combining Shading Rate Images

Using the techniques mentioned, it is possible to have multiple suggestions for shading rates. For instance, you may have a shading rate from contrast aware foveation, motion, and a fixed shading rate map for lens distortion. There is little research on how to combine these images and the effects that combining them can have on performance and user perception. It may be possible to further reduce the number of shaded fragments by evaluating multiple recommendations for shading rate from multiple streams, and examining these streams in relation to each other may provide deeper insight to how these perception-based techniques interact, both constructively and destructively.

VI. DISCUSSION

A. Applications and Industry Adoption

VR has been well received by multiple industries as of 2020 and continues expanding. The ability to create immersive environments for the purpose of training, therapy, entertainment, etc... provides many new opportunities. Perception based rendering and foveation allows for improved performance on low power mobile devices, such as those found on work sites or in hospitals, and can provide opportunities to increase visual fidelity where there is compute power available (such as with tethered VR solutions). The ability for lower-power hardware to maintain visual fidelity makes VR much more accessible to the mass market and can decrease costs for consumers.

For instance, one area that VR has been found to be effective is for therapeutic purposes, such as use in treating PTSD [37] or for reducing pain and anxiety for patients in hospitals [38]. Foveation and perception-based rendering can improve the fidelity of these simulations on low-power, more affordable hardware, allowing these treatments to become more accessible to patients.

VR also has a massive impact on entertainment, with VR games and experiences gaining popularity. For instance, Valve's recent title "Half Life: Alyx" (a VR only game), was highly praised and achieved a concurrent player count of over 42,000 players, a record for VR only games on that platform. Perception based rendering allows games and experiences to improve the visual fidelity, or to allocate more compute resources to more immersive gameplay or more accurate simulations. The impact is also producing significant research and development in VR user interfaces, where cross-sensory feedback and motion capture are enabling more natural user interfaces [39], and customized interactions [40].

As noted in our hardware review section, eye tracking hardware is starting to become more common in VR headsets, and we believe that this will also lead to further developments in presence and immersion. The popular VR social game "AltSpaceVR" has already implemented the ability for their users to enable eye tracking in VR, further improving the ability for players to send social queues to each other (for example, blinking, making eye contact, winking, etc...). The ability to track gaze also opens up significant opportunities for data collection and analytics. Previous work on attention in VR traditionally focuses on using headset orientation to estimate where a user is looking, which although can be helpful to determine approximate orientation, is not adequate to determine actual focal objects. In one of our previous studies for instance, we found that head-ray or head-cone tests correctly estimated a gaze target less than 10% of the time 4. The ability to track user's actual gaze targets may open the door to better understanding of user attention in games, training, simulations, etc...

VII. CONCLUSIONS

Research into eye tracking and foveation has opened the doors to a variety of novel technologies and methodologies. The understanding and exploitation of the human eye can be used to both free up rendering resources (leading to improved performance or visual fidelity), as well as providing new and exciting opportunities for human-computer interaction. Eye tracking allows for greater opportunities in understanding

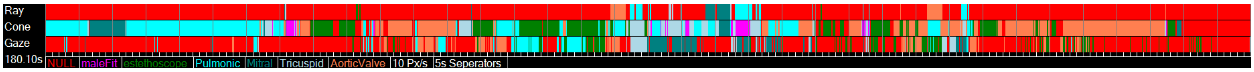

Fig. 4. Correlation between Gaze-Detection Algorithms in a previous study.

user attention and, when combined with knowledge of the human visual system, provides the opportunity to manipulate the environment or visualization to guide users through tasks. Research into foveation and perception-based rendering has proven that not all pixels the same, and they don't need the same attention as all others. By correctly estimating where rendering resources are required, and applying those resources effectively, we free up those resources to perform other tasks with.

As VR hardware is brought to the mass market with lower compute power standalone headsets, it will be difficult for engineers and developers to justify not including eye tracking hardware, given the potential for performance increases. The industry interest in eye tracking from this development will likely lead to further advances in eye tracking performance and accuracy, providing further gains for foveation and eye tracking solutions.

Further research should be conducted in the field of VR rendering and VR hardware design to find new ways of both exploiting the human visual system (foveation, perception based rendering, etc...), as well as what inferences can be drawn from gaze data in VR. Additionally, we should further exploit our knowledge of the human eye to develop novel ways to not only analyze, but guide user vision in virtual reality, and to analyze and understand the impacts this would have on immersion, tutorialization, and skill transfer.

ACKNOWLEDGMENT

The financial support of the *Natural Sciences and Engineering Research Council of Canada* (NSERC) in the form of a Discovery grant RGPIN-2018-05917 is acknowledged.

REFERENCES

[1] P. Smutny, M. Babiuch, and P. Foltynek, "A review of the virtual reality applications in education and training," in *2019 20th International Carpathian Control Conference (ICCC)*. IEEE, 2019, pp. 1–4.

[2] M. Peskin, B. Mello, J. Cukor, M. Olden, and J. Difede, "Virtual reality applications to treat posttraumatic stress disorder," in *Virtual Reality for Psychological and Neurocognitive Interventions*. Springer, 2019, pp. 85–102.

[3] S. Weech, S. Kenny, and M. Barnett-Cowan, "Presence and cybersickness in virtual reality are negatively related: A review," *Frontiers in Psychology*, vol. 10, p. 158, 2019. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2019.00158

[4] C. Merenda, C. Suga, J. Gabbard, and T. Misu, "Effects of vehicle simulation visual fidelity on assessing driver performance and behavior," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1679–1686.

[5] W. Richards, "Stereopsis and stereoblindness," *Experimental Brain Research*, vol. 10, no. 4, pp. 380–388, 1970.

[6] D. Brewster, *The stereoscope; its history, theory, and construction, with its application to the fine and useful arts and to education. By Sir David Brewster*. England: J. Murray, 1856, 1856.

[7] I. E. Sutherland, "The ultimate display," *Multimedia: From Wagner to virtual reality*, vol. 1, 1965.

[8] G. C. Burdea and P. Coiffet, *Virtual reality technology*. John Wiley & Sons, 2003.

[9] D. MacIsaac, "Google Cardboard: A virtual reality headset for $10?" *The Physics Teacher*, vol. 53, no. 2, pp. 125–125, 2015.

[10] E. B. Huey, *The psychology and pedagogy of reading; with a review of the history of reading and writing and of methods, texts, and hygiene in reading*. Cambridge, Mass: M.I.T. Press, 1910.

[11] G. T. Buswell, *How Adults Read*. Chicago, IL: University of Chicago Press, 1937.

[12] H. Hartridge and L. C. Thomson, "METHODS OF INVESTIGATING EYE MOVEMENTS," *British Journal of Ophthalmology*, vol. 32, no. 9, pp. 581–591, Sep. 1948. [Online]. Available: https://doi.org/10.1136/bjo.32.9.581

[13] A. L. Yarbus, *Eye Movements and Vision*. Plenum. New York., 1967.

[14] B. W. Tatler, N. J. Wade, H. Kwan, J. M. Findlay, and B. M. Velichkovsky, "Yarbus, eye movements, and vision," *i-Perception*, vol. 1, no. 1, pp. 7–27, Jan. 2010. [Online]. Available: https://doi.org/10.1068/i0382

[15] R. A. Bolt, "Eyes at the interface," in *Proceedings of the 1982 Conference on Human Factors in Computing Systems*, ser. CHI '82. New York, NY, USA: Association for Computing Machinery, 1982, p. 360–362. [Online]. Available: https://doi.org/10.1145/800049.801811

[16] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke, "Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–12. [Online]. Available: https://doi.org/10.1145/3290605.3300780

[17] F. De Sorbier, V. Nozick, and H. Saito, "Gpu-based multi-view rendering," in *CGAT 2010 - Computer Games, Multimedia and Allied Technology, Proceedings*, 2010, pp. 273–279.

[18] J. Carmack, T. Forsyth, M. Ribble, J. Dolan, M. Kilgard, M. Songy, Y. Uralsky, J. Hall, T. Lottes, J.-H. Fredriksen, J. Gustavsson, S. Holmes, N. Williams, T. Hector, D. Koch, J. Helferty, and O. Etuaho, "Ovr_multiview," https://www.khronos.org/registry/OpenGL/extensions/OVR/OVR_multiview.txt, 2018, accessed: 2019-12-01.

[19] NVIDIA, "NVIDIA Turing GPU," *White Paper*, 2018.

[20] A. Marrs, B. Watson, and C. Healey, "View-warped Multi-view Soft Shadows for Local Area Lights," *Journal of Computer Graphics Techniques*, vol. 7, no. 3, 2018. [Online]. Available: http://jcgt.orghttp//jcgt.org

[21] M. Weier, M. Stengel, T. Roth, P. Didyk, E. Eisemann, M. Eisemann, S. Grogorick, A. Hinkenjann, E. Kruijff, M. Magnor, K. Myszkowski, and P. Slusallek, "Perception-driven accelerated rendering," *Comput. Graph. Forum*, vol. 36, no. 2, pp. 611–643, May 2017. [Online]. Available: https://doi.org/10.1111/cgf.13150

[22] P. Longhurst, K. Debattista, and A. Chalmers, "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*, ser. AFRIGRAPH '06. New York, NY, USA: ACM, 2006, pp. 21–29. [Online]. Available: http://doi.acm.org/10.1145/1108590.1108595

[23] S. Marwecki, A. D. Wilson, E. Ofek, M. Gonzalez Franco, and C. Holz, "Mise-unseen: Using eye tracking to hide virtual reality scene changes in plain sight," in *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '19. New York, NY, USA: ACM, 2019, pp. 777–789. [Online]. Available: http://doi.acm.org/10.1145/3332165.3347919

[24] E. Langbehn, F. Steinicke, M. Lappe, G. F. Welch, and G. Bruder, "In the blink of an eye: Leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 66:1–66:11, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3197517.3201335

[25] I. Cheng, "Foveated 3d model simplification," in *Seventh International*

Symposium on Signal Processing and Its Applications, 2003. Proceedings., vol. 1, July 2003, pp. 241–244 vol.1.

[26] A. Tiwary, M. Ramanathan, and J. Kosinka, "Accelerated Foveated Rendering based on Adaptive Tessellation," in *Eurographics 2020 - Short Papers*, A. Wilkie and F. Banterle, Eds. The Eurographics Association, 2020.

[27] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder, "Foveated 3d graphics," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 164:1–164:10, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366145.2366183

[28] R. Albert, A. Patney, D. Luebke, and J. Kim, "Latency requirements for foveated rendering in virtual reality," *ACM Trans. Appl. Percept.*, vol. 14, no. 4, pp. 25:1–25:13, Sep. 2017. [Online]. Available: http://doi.acm.org/10.1145/3127589

[29] 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.*, vol. 35, no. 6, pp. 179:1–179:12, Nov. 2016. [Online]. Available: http://doi.acm.org/10.1145/2980179.2980246

[30] O. T. Tursun, E. Arabadzhiyska-Koleva, M. Wernikowski, R. Mantiuk, H.-P. Seidel, K. Myszkowski, and P. Didyk, "Luminance-contrast-aware foveated rendering," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 98:1–98:14, Jul. 2019. [Online]. Available: http://doi.acm.org/10.1145/3306346.3322985

[31] L. Yang, D. Zhdan, E. Kilgariff, E. B. Lum, Y. Zhang, M. Johnson, and H. Rydgård, "Visually lossless content and motion adaptive shading in games," *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 2, no. 1, pp. 6:1–6:19, Jun. 2019. [Online]. Available: http://doi.acm.org/10.1145/3320287

[32] Y. He, Y. Gu, and K. Fatahalian, "Extending the graphics pipeline with adaptive, multi-rate shading," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 142:1–142:12, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2601097.2601105

[33] S. Park, Y. I. Kim, and H. Nam, "Foveation-based reduced resolution driving scheme for immersive virtual reality displays," *Opt. Express*, vol. 27, no. 21, pp. 29 594–29 605, Oct 2019. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-27-21-29594

[34] A. Kaplanyan, A. Sochenov, T. Leimkühler, M. Okunev, T. Goodall, and G. Rufo, "Deepfovea: neural reconstruction for foveated rendering and video compression using learned statistics of natural videos," *ACM Transactions on Graphics*, vol. 38, pp. 1–13, 11 2019.

[35] E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym, "Nvidia Tesla: a Unified Graphics and Computing Architecture To Enable Flexible, Programmable Graphics and High-Performance Computing," *IEEE Micro*, pp. 39–55, 2008.

[36] X. Meng, R. Du, and A. Varshney, "Eye-dominance-guided foveated rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 5, pp. 1972–1980, 2020.

[37] S. Miyahira, R. Folen, H. Hoffman, A. Garcia-Palacios, and K. Schaper, "Effectiveness of brief vr treatment for ptsd in war-fighters: A case study," *Studies in health technology and informatics*, vol. 154, pp. 214–9, 01 2010.

[38] K. Arane, A. Behboudi, and R. D. Goldman, "Virtual reality for pain and anxiety management in children." *Canadian family physician Medecin de famille canadien*, vol. 63 12, pp. 932–934, 2017.

[39] M. V. Balderas, C. Carmichael, B. Ko, A. Nova, A. Tabafunda, and A. Uribe-Quevedo, "A makerspace foot pedal and shoe add-on for seated virtual reality locomotion," in *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*. IEEE, 2019, pp. 275–280.

[40] P. Kartick, A. J. U. Quevedo, and D. R. Gualdron, "Design of virtual reality reach and grasp modes factoring upper limb ergonomics," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2020, pp. 799–800.