Impact of Foveated Rendering on Procedural Task Training

Rafael Radkowski(✉) and Supriya Raul

Virtual Reality Applications Center, Iowa State University, Ames, IA, USA
rafael@iastate.edu

Abstract. Foveated rendering (FR) is a technique for virtual reality (VR) that adapts the image quality to the user's eye fixation. The content within the user's eye fixation appears in high quality, the peripheral area in lower quality. The technique exploits the fact that the eye, the retina, in particular, has its highest density of light-sensing cells in the center, the so-called fovea. All other areas are covered with less density. Eye tracking is essential for foveated rendering since one needs to be able to determine the user's fixation. Although FR is a promising technique to reduce computer performance requirements, it is unclear whether it has an impact on the user's task performance, primarily when one uses VR as a training tool. Theoretically, the technique is invisible. However, its implementation depends on several parameters and technical hardware limitations. We conducted a study to see whether or not these limitations distract the user and reduce his/her training performance. The results indicate that the user notices the technique. However, s/he does not care, and the performance difference is insignificant, except for some outliers caused by technical eye tracking limitations.

Keywords: Virtual reality · Foveated rendering · Procedural tasks · Training

1 Introduction

Foveated rendering (FR) is an upcoming rendering technique for virtual reality (VR) application that adapts the quality of the rendered image with respect to the user's eye focus. Human eyes perceive the physical world with different fidelity and attention due to the density distribution of light-sensitive nerve cells on the retina. The density, thus the visual resolution, is at a maximum in a central pit of the retina, the so-called fovea. It decreases with increases distance to the fovea; so does the perceivable resolution or fidelity. Foveated rendering exploits this fact by decreasing the rendering quality depending on the user's eye focus, also called fixation. The visual rendering quality is high at the fixation point, and lower in the visual periphery. Figure 1 illustrates an example; Fig. 1(a) shows a regular computer graphics scene, and (b) the same scene with a fixation kernel. A round area in which content appear in high quality. The peripheral portion of the scene is rendered with lower fidelity. Here, every second pixel is discarded. FR relies on eye tracking to determine the user's fixation point in real-time so that that renderer can adapt the kernel's position depending on the user's fixation.

J. Y. C. Chen and G. Fragomeni (Eds.): HCII 2019, LNCS 11574, pp. 258–267, 2019.
https://doi.org/10.1007/978-3-030-21607-8_20

Head-mounted displays (HMD) need to be equipped with eye tracking hardware and sample the eye position with a high sampling rate so that the user's eye cannot outrun the tracking device.

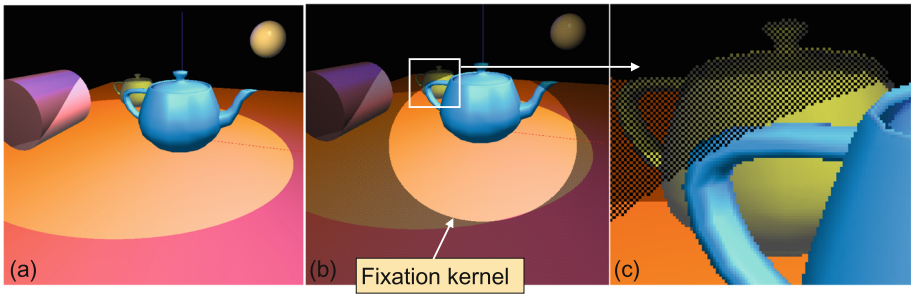

Fig. 1. (a) A regularly rendered scene. (b) with foveated rendering, the bright area shows the fixation kernel (too large here). The remaining image appears darker since the example approach (c) uses a pattern to reject every second pixel. The colors were not reconstructed for this example.

FR reduces the computational requirements to graphics hardware. Depending on the utilized algorithm, one either renders in a low-resolution buffer, skip fragments, or use another technique which reduces the number of pixels and triangles to compute. Thus, it facilitates high-quality VR on mobile devices and other systems with limited graphics performance.

Our research focuses on the user-aspect of FR, on two points in particular. First foveated rendering is not free of visual artifacts. These artifacts are noticeable [1]. Previous research reported that users see antialiasing in their peripheral vision [2], experience a sense of tunnel vision [3] or notice a screen-door effect [4]; note that these artifacts highly depend on the implementation of FR. We are interested in learning whether or not these artifacts distract the user when performing tasks in a virtual environment (VE) or if the user does not mind the artifacts. Related to the first question, we are interested to see if any artifacts or eye tracking problems reduce the user's performance when learning procedural tasks in a virtual environment. Therefore, we conducted a user study that asks the user to perform a training task. The task incorporates a virtual engine room. Volunteers were asked to operate buttons, levers, and other instruments. The paper reports about the study and the results. The remainder of this paper is structured as follows: The next section introduces the essential FR background and different implementations. Section 3 explains the VE and our FR implementation. Section 4 presents the results. The paper closes with a conclusion and an outlook.

2 Foveated Rendering Background

FR exploits the fact that the human perception is focused onto a small central area which aligns with fovea, the point of highest density of light-receptive cells on the retina. This central area covers, depending on the source, between 3°–5° of the visual field. People are mostly aware of the objects that we see in this central area. The remaining peripheral area is covered with a lower density of light receptive cells. Consequently, its recognizable visual fidelity is limited. FR exploits this and renders a scene with high quality only for the focused area. The remainder of the scene is generated with lower fidelity.

Different FR algorithms have already been introduced. In general, one can distinguish two approaches (Fig. 2): Layer-based vs. region-based foveated rendering [10]. Layer-based approaches render multiple images. Using two layers, for instance, one low-resolution layer covers the entire scene view, and a high-resolution layer shows high-quality content. The number of layers and the size of each layer can vary. User studies are essential to identify the right parameters. Region-based foveated rendering split the scene into a fixed number of regions. The inner region is rendered with the highest quality, the outer region with the lowest quality. All intermediate regions are rendered with interpolated quality. Since the field is currently very active, variations of those two approaches were also already introduced. The following paragraphs introduce some of these algorithms.

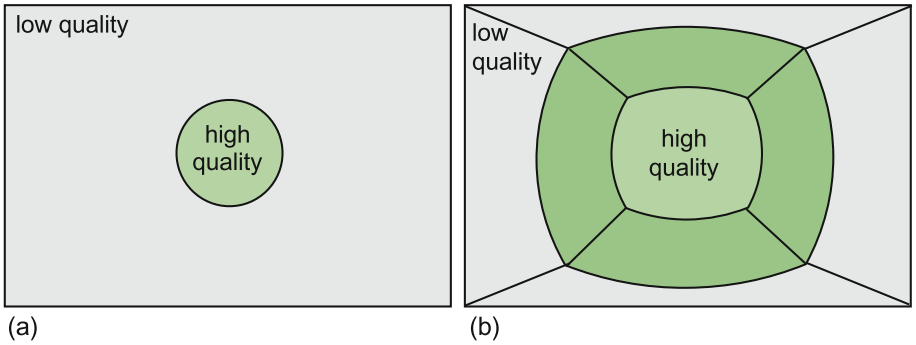

Fig. 2. (a) Layer-based approach - the different regions are rendered as individual layers and then stacked together. (b) Region-based approach - The display is split into different regions, each region is rendered with different quality.

Guenter et al. [6] introduce a layer-based approach combining three layers with three different resolutions. The inner layer is rendered with the highest quality, the outer layer - the remaining image - with the lowest quality. The second layer covers near peripheral vision. It provides an average quality.

Meng et al. [5] suggest a kernel-based foveated rendering using two pass rendering approach. The first render pass generates the low-quality images, and the second render pass augments a circular kernel area with the full image resolutions.

Patney et al. (Patney, 2016), (Patney, 2018) introduce a perceptually-based foveated rendering algorithms. The author report that the human peripheral vision is best at seeing things like color, contrast, edges, and motion in the peripheral area. Thus, the authors focus on these aspects in the peripheral area and ignore high-fidelity details. Additionally, the authors also suppress anti-aliasing artifacts. They report that the employed techniques reduce the number of artifacts so that a user does not notice any of them anymore.

Aldehayyat et al. [7] propose a foveated rendered, that first blurs the entire image and then refocuses the fovea area with a high-quality renderer.

All these techniques require eye tracking. Besides these, some authors introduced hardware solutions and techniques that work without eye tracking.

For instance, Tan et al. [4] suggest a hardware solution. They developed an HMD with two displays embedded. The first display renders the full-resolution image for the fovea area, the second one the low-quality image. An optical combiner merges both images. The authors report that this technique reduces visual artifacts successfully.

Additionally, the area is currently highly industry-driven. Head-mounted display (HMD) manufacturers and others already provide software solutions for their headsets. For instance, Oculus supports two foveated rendering techniques, both work without eye tracking: Fixed Foveated Rendering (FFR) [8] allows one to render the center portion of the display with high resolution and the edges with lower resolution. The second technique, Mask-based foveated rendering (MBFR) [9] drops every other pixel on the complete rendering. However, the pixels are reconstructed in a subsequent render pass. This approach gains the same performance win by ignoring pixels, although without the need for eye tracking and regions.

Although previous studies already demonstrate that artifacts are noticeable, depending on the utilized technique, the studies did not assess to what extent the user mind the artifacts and whether or not they affect his/her performance.

3 VR Test Environment

This section describes our VE, including the test scene, the implemented foveated renderer filter, the utilized headset along with the eye tracking system. Figure 3 shows the VE. The content of the VR scene resembles part of an engine room of size 20 × 30 ft, with a row of control panels and several valves and levers to operate. Figure 3(b–c) depicts a part of the control panel in detail. It incorporates three active buttons and two dials; active means they can be operated. All operational parts are within 10 ft walking distance, along one corridor of the room. The scene was prepared so that the user can reach all relevant locations by walking. Although the content of the scene appears to be old and in bad condition, we selected this content since it exhibits plenty of high-fidelity details. The application conveyed instructions via a small board in front of the user. Its position is fixed relative to the user's position, at his/her lower left side. Thus, the user has to look down and to the left to see the instructions. The entire VE was prepared with Unity.

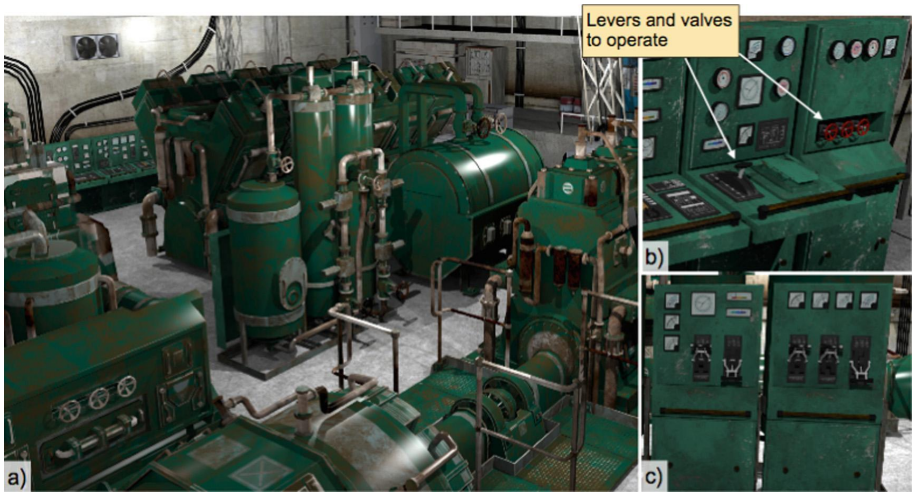

Fig. 3. (a) The VE resembles part of an engine room. It comes with plenty of high-fidelity details and contains several buttons and lever to be operated. (b) A part of the control panel.

We implemented two different foveated rendering filter: a kernel-based foveated rendering (KFR, [5]) filter and a layer-based rendering (LBR) filter with three layers [6]. The first one renders a low-resolution image (720 × 800) depicting the entire scene. We also used a black-white pattern to ignore every other fragment. This image is upscaled and augmented with a high-fidelity rendering for the fixation area. The area is round with a radius of 300 pixels. Thus, the area covers approximately 20% of the display (1440 × 1600). The second one works similar to the first filter with the difference that three layers are used. First, the entire image is rendered with low-quality (720 × 800). Two kernels at the user's fixation with gradually decreasing quality augment the first one. Here, we first used a 300-pixel radius for the second layer and a 150-pixel radius for the first layer. We used the same size to be able to compare both techniques.

Also, all these parameters were found empirically. We internally optimized those parameters with tests within the research group until everybody was more or less satisfied with the visual result. However, it was impossible to find a parameter set that satisfied everybody completely. All filters are implemented with Cg.

Our foveated rendering implementation relies on eye tracking. We use an HTC Vive Pro HMD and a Pupils Labs eye tracking add-on for this purpose (https://pupil-labs.com). The eye tracking system needs to be mounted into the HTC Vive. It samples the eye position 120 times/s. Pupil Labs provides open source Python scripts to operate the device. Additionally, a script for Unity is available to directly process the data.

Since every user has different eyes, every user needs to calibrate the eye tracking system. Pupil Labs provides a tool for this purpose, which we adopted. It asks the user to follow a dot that jumps along the periphery of a circle. The entire calibration procedure is straightforward, requires 20-s attention, and is part of a Unity application.

Our VE uses hand tracking for user interaction. The user sees white 3d hand models. The hand models are articulated and match the user's hand postures. The application uses one set of hand models for all users. Each hand changes its color to red if an interaction or an interaction attempt is detected. This feature provides user feedback so that he or she knows that the interaction attempt is detected. We use a Leap Motion sensor for hand tracking, attached to the Vive Pro display (Fig. 4).

Fig. 4. (a) The Pupils Labs eye tracking. (b–c) It tracks both eyes with a sampling rate of 120 samples second using infrared lights (Color figure online)

4 Study

The goal of this research is to determine whether or not foveated rendering affects the user in a virtual environment. Previous studies reported that the user could notice artifacts. However, noticing artifacts and being affected, e.g., exhibiting a lower task performance, are two different aspects. We conducted a user study to determine whether or not the user notice a difference. Three modes were compared: No FR (Mode A), KFR (Mode B), and LBR (Mode C). We followed a between-subject design and asked each volunteer to repeat the tasks four times. Each user was subjected to two times to Mode A and two times either to Mode B or to Mode C. In total, 24 volunteers (m = 17, w = 7, age 21–33) perform the task. The next sections describe the procedure, explain the results, and discuss the outcome (Fig. 5).

4.1 Procedure and Measurement

Upon arrival, each user was asked to complete a pre-questionnaire pertaining to the user demographics, as well as to sign a consent form. The experimenter then briefly explained the task (follow instructions). Each user could study a printout showing the instruction panel. Additionally, the experimenter introduced the safety measures (white grid when approaching the tracking limit, curbs on the floor) to prevent the user from running into walls. Next, the user puts on the HMD (no users were allowed to wear glasses since the eye tracking device does not work with glasses) and the experimenter

started the VR environment. Initially, the user calibrated the eye tracking device and verified that it is functional by directing a dot onto three targets using his/her eyes. The experimented observed this task on a screen and manually restarted the calibration if the user were not able to hit the targets or proceed to the next step. After completing the calibration task, the VR scene became active. Next, the user verified that the Leap Motion hand tracking system worked as expected. He or she were asked to observe his/her hand models and to verify that the models match the user's hand posture. Additionally, the user had to push one button and to grab a screwdriver to verify functionality. This step concluded the initialization.

Subsequently, the engine room appeared, and the first task started. Each task came with 21 interactions (operations). The user had to walk six times to different locations within a 10 × 10 ft area. Each user was asked to repeat the task four times. Each trial slightly changed parameters (e.g., set the lever to position A for trial 1 and to position B for trial B), so that no two trials were utterly equal. This measure should mitigate learning effects. The VR application automatically selected the modes (Mode A, B, C). All trials were administered in a random order.

Between trials, the experimenter asked the user to complete a NASA TLX and a Simulator Sickness Questionnaire (SSQ). The user needed to answer the questions verbally since we intended to keep the user in the virtual environment. The experimenter read the questions aloud and noted the volunteer's answer. The answer options were displayed in the VE. We did not expect severe simulation sickness problems since the entire scene does not contain any effects that elicit simulation sickness. However, previous studies reported that a user might experience a tunnel view, which can cause simulation sickness. Also, we do not believe that the scene and task ask too much from the user. Thus, we use the NASA TLX questionnaire to verify that frustration does not affect the user.

After the user performed the four trials, we ask each user to complete a post-assessment questionnaire. Its questions (Table 1) directly ask the user whether he or she noticed the difference between trials or artifacts. We used a 5-point Likert scale to obtain the answers. Additionally, it verifies that the application was usable and that the user did not experience any significant interaction problems. The experiment concluded after the user completed this questionnaire.

Additionally, the VR application automatically registered each user interaction, the time of interaction, and the user's head position (moving path). Also, the entire time was recorded starting when displaying the first task and ending when the user performed the last task. The application also detected user mistakes, e.g., wrong lever position, the wrong button pushed, and recorded them. All data were logged in an ASCII file, one per user.

4.2 Results

We analyzed the time the users required per mode as well as the mistakes the users made; Fig. 5 illustrates the results. Here, Mode AB means that this user experience Mode A and Mode B, every two times. Mode AC is the group that experienced Mode A and Mode C, also every two times. We performed an ANOVA for the group subject do mode A and B ($F_{1,11} = 0.13$, $p < .05$), A and C ($F_{1,11} = 0.73$, $p < .05$),

and between Mode B and Mode C ($F1,11 = 0.85$, $p < .05$). However, the results do not indicate any significant differences. Figure 5(b) illustrates the mistakes the users made. The results vary between Mode A, 1.25 and 0.66 mistakes, Mode B, 3.33 mistakes, and Mode C, 2.67 mistakes, with a total number of 21 steps per trial; thus, 21 possible mistakes. We conducted an ANOVA, again for the group subjected to Mode A and B ($F1,11 = 4.78$, $p < .05$), Mode A and C ($F1,11 = 9.54$, $p < .05$). Both results show a statistically significant difference. We also analyzed the difference between Mode B and C ($F1,11 = 0.53$, $p < .05$), which does not yield any significant differences.

The NASA TLX results show no significant findings. The reported results for mental demands, physical demands, etc. are mostly low. The effort was reported to be low in general. Some users became frustrated as a result of eye tracking limitation and the extended test duration, which affected the results. Also, the procedure was monotonous, which also increased frustration; both aspects are discussed in the next sections.

Additionally, no user became severely sick in the virtual environment. Although the user spent almost 40 min in the virtual environment, the majority of users did not report discomfort, fatigue, or headaches. The only exceptions were eye strains and blurred vision. The first one reached moderate severity, and the second one affected some users severely. However, further investigation showed that this might be a result of foveated rendering. If eye tracking worked inadequately, the users just eye focused on a low-quality part of the rendering. A more significant number of users reported dry eyes. However, we do not attribute this to foveated rendering.

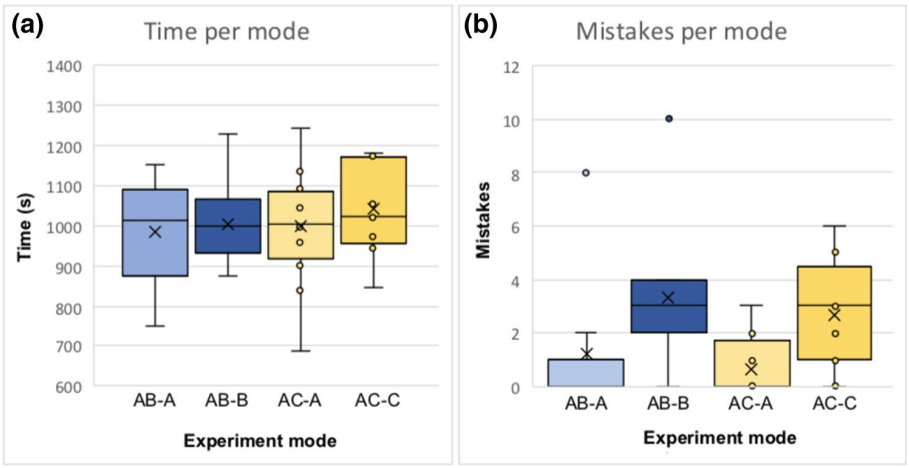

Fig. 5. (a) Time per mode, (b) Errors per mode

Table 1 summarized the results of the post-assessment questionnaire yielded per mode. The questionnaire asked ten questions aiming to render quality, interaction, and usability in general. We conducted a one-way ANOVA. The results are insignificant except for question Q1, Q4, and Q7. However, these results are most likely due to a technical limitation of the eye tracking hardware.

4.3 Discussion

The results of this experiment do not reveal essential significant difference pertaining to time. However, users that work with foveated rendering make more mistakes, although there are some reservations to this result since was most likely caused by the extended test period and user frustration.

First, we noticed a training effect between the first and the second trial as mentioned. We expected to find a training effect although the tasks slightly changed. The users remembered the general location of equipment and panels. They became familiar with the VE which allowed them to move faster. This training effect remained unnoticeable in subsequent trials.

Although the mistakes users made with FR and without FR are statistically significant, some user reported increasing frustration. The users' answer too Q5 and Q1 allow us to assume that they had problems reading the text. The eye tracking device requires thorough calibration; it did not work as expected for some of the user's, and we noticed this too late since these users also passed the initial test. Adjusting and calibrating the eye tracking device per user was challenging. Self-tests revealed that one has to carefully adjust the interpupil distance very accurately first before calibrating the eye tracking device. The calibration tool is not completely visible otherwise.

Unfortunately, we noticed an increasing frustration among some volunteers. Some users reported that they just wanted to finish the test and to be quick after trial number two or three. We attribute the increased number of mistakes to this trait.

Table 1. Questionnaire results, mean (std. deviation)

No.	Question	Mode A–B	Mode A–C
1	The scene was always clearly visible and not blurred	1.67 (0.77)	3.58 (0.79)
2	The display was free of aliasing and other artifacts	1.41 (0.51)	1.41 (0.66)
3	The field-of-view was appropriate for this scenario	1.67 (0.77)	1.91 (0.90)
4	The rendering quality of all four trials was equal	1.75 (0.86)	3.67 (0.88)
5	I was able to read the text in the virtual environment	2 (0.60)	3.92 (1.08)
6	I was able to identify all buttons, levers, and switches in the virtual environment	1.16 (0.38)	2.08 (1.16)
7	The eye tracking device and its functionality was invisible to me	2.83 (1.69)	2 (0.85)
8	All instructions were clear and easy to understand	1.5 (0.90)	2.17 (0.93)
9	I could easily interact with virtual parts using my hands	1.58 (0.90)	1.91 (0.90)
10	I would recommend this virtual reality training application to my friends	1.83 (1.11)	1,75 (0.86)

Furthermore, a more significant number of people noticed FR and its effect on the scene. Although the results are not significant, users reported that they notice some differences between the scenes, without being able to not further detailing them. However, users do not mind the differences. Also, the results let us believe that only the direct comparison of different renderers facilitates this outcome. Users who do not know about FR would not notice any difference.

4.4 Conclusion and Outlook

This research aimed to understand the impact of foveated rendering on users, especially on task performance. Foveated rendering can cause rendering artifacts, and previous studies reported that users notice them. They can distract users. Thus, we were interested to see if they have an effect on the users' performance in a training task, or if users do not mind them. User study results indicate that a significant number of users notice the different renderer since they had an opportunity to compare them, although they cannot describe the effect in detail. Moreover, users do not mind the differences. Thus, we also did not notice any significant differences when analyzing the training task results. Thus, we conclude that FR does not affect the user if correctly implemented. Nonetheless, eye tracking and display limitations can result in a suboptimal outcome, which could distract the user.

We are in the process of conducting a second user study with a focus on procedural task training with VR. The goal is to optimize the training task. This study gives us another opportunity to examine FR user impact.

References

1. Swafford, N.T., Iglesias-Guitian, J.A., Koniaris, C., Moon, B., Cosker, D., Mitchell, K.: User, metric, and computational evaluation of foveated rendering methods. In: Proceedings of the ACM Symposium on Applied Perception (SAP 2016) (2016)
2. Patney, A., et al.: Perceptually-based foveated virtual reality. In: SIGGRAPH Emerging Technologies (2016)
3. Patney, A., et al.: Towards foveated rendering for gaze-tracked virtual reality. ACM Trans. Graph. **35**, 179:1–179:12 (2016)
4. Tan, G., et al.: Foveated imaging for near-eye displays. Opt. Express **26**(19), 25076–25085 (2018)
5. Meng, X., Du, R., Zwicker, M., Varshney, A.: Kernel foveated rendering. PACMCGIT **1**, 5 (2018)
6. Guenter, B.K., Finch, M., Drucker, S.M., Tan, D.S., Snyder, J.: Foveated 3D graphics. ACM Trans. Graph. **31**, 164:1–164:10 (2012)
7. Aldehayyat, Y.: Foveated image refocusing in VR applications (2018)
8. Oculus Inc., Fixed foveated rendering. https://developer.oculus.com/documentation/unreal/latest/concepts/unreal-ffr. Accessed 11 Feb 2019
9. Oculus Inc., Masked-based foveated rendering. https://developer.oculus.com/documentation/unreal/latest/concepts/unreal-mbfr. Accessed 11 Feb 2019
10. Bastani, B., Turner, E., Vieri, C., Jiang, H., Funt, B., Balram, N.: Foveated pipeline for AR/VR head-mounted displays. Inf. Display **33**, 14–35 (2017)