Concept for Using Eye Tracking in a Head-Mounted Display to Adapt Rendering to the User's Current Visual Field

Daniel Pohl*
Intel Corporation

Xucong Zhang†
Max Planck Institute for Informatics

Andreas Bulling‡
Max Planck Institute for Informatics

Oliver Grau§
Intel Corporation

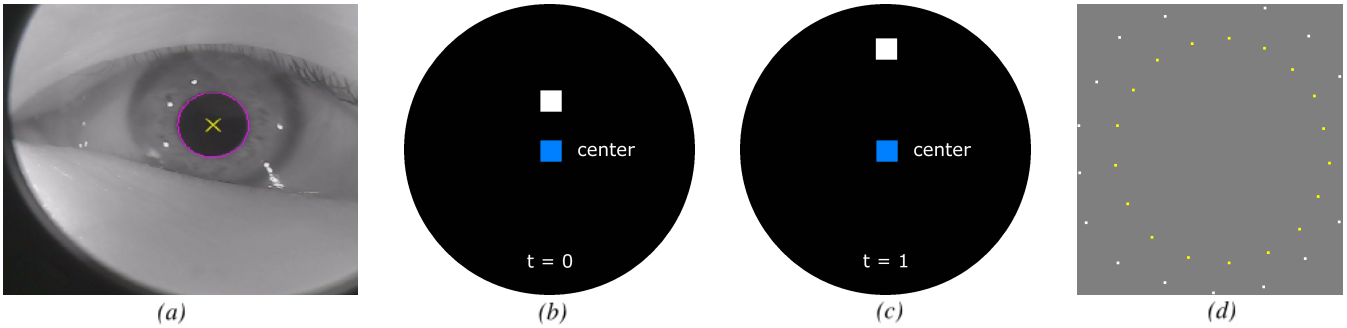

Figure 1: *(a): image from the eye tracker. (b) and (c): calibration routine at different time steps. (d) combined results of the visual field calibration based on the screen center. The white dots represent the maximal outer points of the visual field while the user is looking at the screen center. The yellow dots are the maximal outer points of the visual field when the user is directing his eye gaze at the corresponding yellow dot. When looking at one of these yellow dots, the user cannot see the area beyond it.*

Abstract

With increasing spatial and temporal resolution in head-mounted displays (HMDs), using eye trackers to adapt rendering to the user is getting important to handle the rendering workload. Besides using methods like foveated rendering, we propose to use the current visual field for rendering, depending on the eye gaze. We use two effects for performance optimizations. First, we noticed a lens defect in HMDs, where depending on the distance of the eye gaze to the center, certain parts of the screen towards the edges are not visible anymore. Second, if the user looks up, he cannot see the lower parts of the screen anymore. For the invisible areas, we propose to skip rendering and to reuse the pixels colors from the previous frame. We provide a calibration routine to measure these two effects. We apply the current visual field to a renderer and get up to 2× speed-ups.

Keywords: virtual reality, eye tracking, rendering, head-mounted display

Concepts: •**Computing methodologies** → **Rendering;**

1 Introduction

Virtual reality HMDs are becoming popular in the consumer space. To increase the immersion further, higher screen resolutions are needed. Even with expected progress in future GPUs, it is challenging to render in real-time at the desired 16K HMD retina resolution[1]. To achieve this, the HMD screen should not be treated as a regular 2D screen where each pixels is rendered at the same quality. Eye tracking in HMDs gives several hints of the user's perception which can be used to reduce the work load. In this work, we propose to use the current visual field, depending on the eye gaze, to skip rendering to certain areas of the screen.

2 Related Work

[Tong and Fisher 1984] built a flight simulator with a projection system on a dome. Depending on the head orientation of the user, they mixed together a low quality image for the whole dome with a high quality image at the gaze of the user. [Guenter et al. 2012] used an eye tracker in a setup with multiple monitors and adjusted the quality depending on the tracked eye gaze. Eye tracking in HMDs was demonstrated by [Duchowski et al. 2001]. Using cheap components, students added eye tracking to modern consumer HMDs [Stengel et al. 2015]. [Pohl et al. 2016] combined foveated rendering in HMDs with rendering optimizations towards lens astigmatism. Compared to their work, we account for the current visual field of the user, depending on the eye gaze.

3 Visual Field Specific Rendering

To find the visual field, we developed a calibration routine using the Pupil Labs [Kassner et al. 2014] eye tracker (example image in Figure 1 (a)) in the Oculus Rift DK2, consisting of two steps. First, the user always looks at the center point inside the HMD. Meanwhile, another blinking dot moves smoothly from the center towards the outer area of the screen (Figure 1 (b) and (c)). The user keeps looking at the center and will press a key once the moving dot is not visible anymore within the user's field of view while wearing the HMD. We repeat this procedure for different angles like on a clock, e.g. the first time the dot moved from the center straight up to the screen, in the second run, the dot moves from the center towards the position of one o'clock and so on. Based on the experience from our experiments, we suggest using between 12 and 20 of these tests for a reasonable sampling result of the user's visual field when looking at the center. In the second step of the calibration routine, the user always follows the moving point and presses a key once it is invisible. An exemplary result can be seen in Figure 1 (d), where the white dots are the stored positions of the moving dots when the user was looking in the center and was not able to see them anymore. The yellow dots are the positions of the dots the user followed with his eyes during the movement, until they were not visible anymore.

*e-mail: daniel.pohl@intel.com
†e-mail: xczhang@mpi-inf.mpg.de
‡e-mail: bulling@mpi-inf.mpg.de
§e-mail: oliver.grau@intel.com

VRST '16, November 02-04, 2016, Garching bei München, Germany
ISBN: 978-1-4503-4491-3/16/11
DOI: http://dx.doi.org/10.1145/2993369.2996300

[1] vrfocus.com/2015/03/abrash-vr-needs-hit-16k-match-retinal-res

From this image, we conclude that when the user is looking at the center, he can see more area than when directly looking into these outer areas, which is a lens defect in the Oculus Rift DK2 and other HMDs.

This leads to the first part for our rendering optimizations depending on the current visual field. We lay curves between the individual sampling points from Figure 1 (d), roughly approximating an ellipsoid. This is shown in Figure 2 (left). If the user looks at a point at the yellow ellipsoid, we know that at least the area beyond it cannot be seen anymore from that eye gaze. For an exemplary eye gaze in Figure 2 (right), we hatched that invisible area in the image, which is about 10%. We skip rendering that area and leave the pixels from the previous frame there to avoid larger illumination changes, which might be perceived through internal reflections inside the HMD. In more detail, in a ray traced renderer, no primary rays would be shot for these pixels. In a rasterization setup, these pixels would be stenciled out. The same can be applied if the eye gaze is even further beyond the yellow ellipsoid.

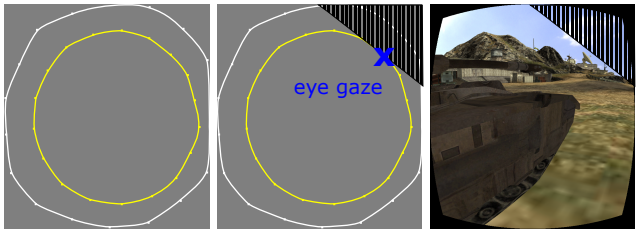

Figure 2: *Left: added curves through the results of the visual field calibration. Center: with an eye gaze at the marked blue cross, at least the hatched area will not be visible anymore. Right: the hatched area can be skipped for rendering.*

The second part of the rendering optimizations is to adjust the opposite side of the current eye gaze, e.g. if someone is looking up, he cannot see the full area on the screen below. To calibrate this, another routine is used where instead of the screen center, the starting point for the moving dots would be placed at one of the yellow points in Figure 3 (left) and repeated for all other yellow points. From the new center, we smoothly move again a blinking dot into various directions (Figure 3 center and right) and the user has to press a key, when it becomes invisible.

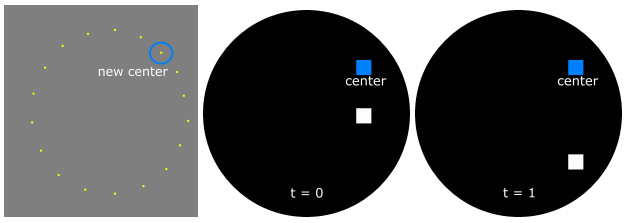

Figure 3: *Left: Selecting one of the previously determined yellow points as the new center for the second calibration step. Center and right: second calibration routine with the new center at different time steps.*

After the result of this calibration step (Figure 4 left), we lay in curves (Figure 4 center) across the data points. We use the inner area of that shape as the user's visual field for that specific eye gaze and can skip a large amount of screen area for rendering as shown in Figure 4 (right).

In our test case, 57% of the pixels are marked invisible with the selected sampling point. We apply that mask of the visual field to a self-written ray tracer, running on a Dual-CPU (Intel Xeon E5-2699 v3), 1920 × 1080 pixels workstation with the Oculus Rift DK2, rendering a game scene from Enemy Territory: Quake Wars. The performance of rendering the full image with 2× supersampling is 26 fps. When we use the mask, the performance increases to 55 fps,

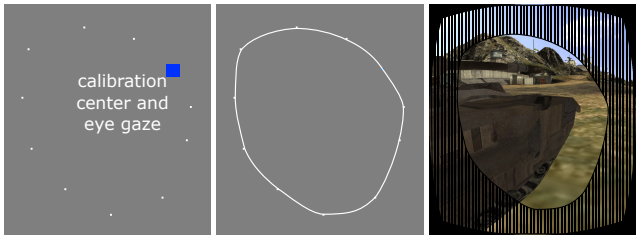

Figure 4: *Left: results of the calibration with the eye gaze in the blue marked upper right corner. Center: added curves through the data points. Right: the hatched area can be skipped for rendering.*

showing a close-to-linear scaling of performance with the number of pixels rendered.

The more sample points are taken as center and calibration routines are run with it, the more accurate model can be determined for all eye gazes for a certain user inside a specific HMD. One calibration procedure with one center and 20 directions for the moving dots takes about one to two minutes to complete. As this can be quite time-consuming with many sample points, we suggest for future work to provide a detailed user study and to develop a generic model for a specific HMD which works well for most users with an optional calibration for fine tuning. Our technique could also be combined with traditional foveated rendering and optimizations towards the lens astigmatism as described in Section 2.

4 Conclusion

We have shown how to calibrate the current visual field inside an HMD. Through occurring lens defects and limitations of the human visual system, certain areas for rendering can be skipped depending on the current visual field to enable faster frame rates.

Acknowledgements

This work was supported, in part, by the Cluster of Excellence on Multimodal Computing and Interaction (MMCI) at Saarland University. Further thanks to Timo Bolkart for his support.

References

DUCHOWSKI, A. T., MEDLIN, E., GRAMOPADHYE, A., MELLOY, B., AND NAIR, S. 2001. Binocular eye tracking in VR for visual inspection training. In *Proceedings of the ACM symposium on Virtual reality software and technology*.

GUENTER, B., FINCH, M., DRUCKER, S., TAN, D., AND SNYDER, J. 2012. Foveated 3D graphics. *ACM Transactions on Graphics (TOG) 31*, 6, 164.

KASSNER, M., PATERA, W., AND BULLING, A. 2014. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Adj. Proc. of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2014)*, 1151–1160.

POHL, D., ZHANG, X., AND BULLING, A. 2016. Combining eye tracking with optimizations for lens astigmatism in modern wide-angle HMDs. In *IEEE Virtual Reality (VR)*.

STENGEL, M., GROGORICK, S., EISEMANN, M., EISEMANN, E., AND MAGNOR, M. 2015. An affordable solution for binocular eye tracking and calibration in head-mounted displays. In *Proceedings of the 23rd ACM international conference on Multimedia*, 15–24.

TONG, H., AND FISHER, R. 1984. *Progress Report on an Eye-Slaved Area-of-Interest Visual Display*. Defense Technical Information Center.