Perceptual Model Optimized Efficient Foveated Rendering

Zipeng ZHENG
School of Computers
Guangdong University of Technology

Zhuo YANG
School of Computers
Guangdong University of Technology
dr.yangzhuo@qq.com

Yinwei ZHAN
School of Computers
Guangdong University of Technology

Yuqing LI
School of Computers
Guangdong University of Technology

Wenxin YU
School of Computer Science and Technology
Southwest University of Science and Technology

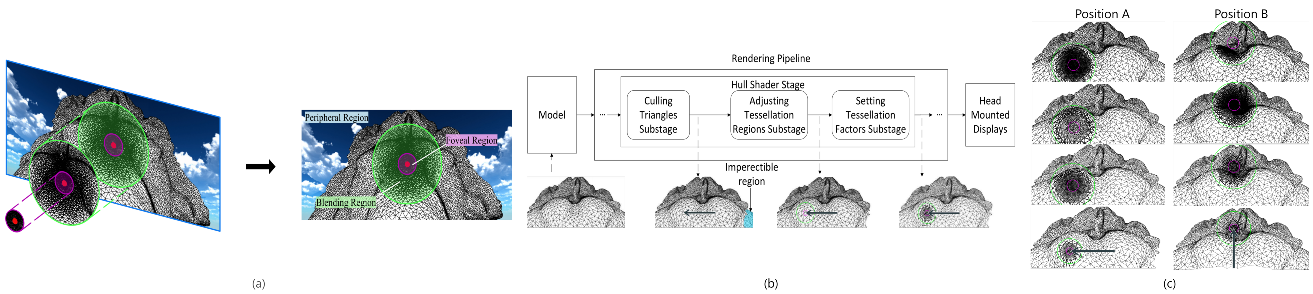

Figure 1: (a) Foveated rendering.(b) Block diagram of the proposed foveated rendering method. The black arrows represent the movement path of the gaze point.(c) Rendering results of different methods. The results of Guenter et al. 's method and Swafford et al. 's method are shown in the first two rows. The last two rows show the results of the proposed method, with two gaze behaviors of fixation and smooth pursuit.

ABSTRACT

Higher resolution, wider FOV and increasing frame rate of HMD are demanding more VR computing resources. Foveated rendering is a key solution to these challenges. This paper introduces a perceptual model optimized foveated rendering. Tessellation levels and culling areas are adaptively adjusted based on visual sensitivity. We improve rendering performance while satisfying visual perception.

CCS CONCEPTS

• **Computing methodologies → Rendering**; *Virtual reality*;

KEYWORDS

Computer graphics, Virtual Reality, Rendering, Perceptual model

ACM Reference Format:
Zipeng ZHENG, Zhuo YANG, Yinwei ZHAN, Yuqing LI, and Wenxin YU. 2018. Perceptual Model Optimized Efficient Foveated Rendering. In *VRST 2018: 24th ACM Symposium on Virtual Reality Software and Technology (VRST '18), November 28-December 1, 2018, Tokyo, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3281505.3281588

ACM ISBN 978-1-4503-6086-9/18/11.
https://doi.org/10.1145/3281505.3281588

1 INTRODUCTION

Foveated rendering is an important research topic in VR. There are two kinds of approaches: image-based rendering and model-based rendering. Watson et al.[4] pointed out that model-based rendering can be more effective in reducing rendering time, as it reduces resolution by directly manipulating model geometry rather than applying a full-screen convolution filter.

Guenter et al.[1] blended three layers of different resolutions to provide a high-quality foveated rendering result(Figure 1(a)). They proposed that tessellation levels in different regions could be adjusted by the distance between the camera and the object's center. Swafford et al.[3] proposed to statically set tessellation levels in different regions. But all the previous methods did not consider visual perceptual size, which lead to waste of rendering resources and perceivable loss in quality.

In this study, we establish a perceptual model to calculate visual sensitivity in different regions, then remove the imperceptible triangles and tessellate the model based on visual sensitivity to improve rendering performance while satisfying human visual perception system.

2 PROPOSED METHOD

Now we introduce a perceptual model based surface tessellation algorithm for foveated rendering.

2.1 Perceptual Model

Vision science shows that a 35-fold reduction in spatial sensitivity exists from the fovea out to the extremities of human vision. In addition, velocity across retina reduces the range and upper limit of frequencies that eye can perceive. We call measure of stimulus as spatial frequency defined as units of cycles per degree (c/deg). As for computer graphics system, we remove features smaller than the 60 c/deg threshold, and the user shouldn't notice any change.

We propose a perceptual model based on Reddy's visual acuity model[2], which deduces visual sensitivity $H(v, r)$ with given a velocity v pixels/s and peripheral extent r pixels. It's a measure of the smallest detail that an observer can resolve.

$$G(v) = \begin{cases} 60.0, & v \leqslant \rho d_u \tan 0.825 \\ 57.69 - 27.78 \lg(v), & \rho d_u \tan 118.3 \geqslant v > \\ & \rho d_u \tan 0.825 \\ 0.1, & v > \rho d_u \tan 118.3 \end{cases}$$

$$M(r) = \begin{cases} 1.0, & r \leqslant \rho d_u \tan 5.79 \\ 7.49/(0.3 \arctan \frac{r}{\rho d_u} + 1)^2, & r > \rho d_u \tan 5.79 \end{cases} \tag{1}$$

$$H(v, r) = G(v)M(r)$$

where ρ is the pixel density of the display(pixels/mm), d_u is the user's distance from screen(mm).

By using the perceptual model, we can determine whether a polygon is perceptible to the user. If the computed extent of the polygon is smaller than the smallest size of stimulus that should be perceptible to the user, we can assume the polygon is imperceptible and does not need to be rendered.

2.2 Improved Tessellation Method

In order to apply the perceptual model to the tessellation stage, we introduce three additional steps, as shown in Figure 1(b).

2.2.1 Culling Triangles. According to the perceptual model, we can calculate the visual sensitivity of the mesh. When the visual sensitivity is lower than human's feeling, the corresponding part becomes the imperceptible region. Removing the triangles in the imperceptible region can reduce the number of vertices to be processed in the tessellation stage.

2.2.2 Adjusting Tessellation Regions. Foveated rendering result is presented by blending three layers of different resolutions. These three regions correspond to three visual sensitivity thresholds respectively. By calculating the visual sensitivity and comparing it with these thresholds, the positions and ranges of the three regions can be determined. This dynamic adjustment method enables the tessellation result to match different gaze behaviors.

2.2.3 Setting Tessellation Factors. After adjusting tessellation regions, we use the perceptual model to determine the appropriate edge tessellation factor f for each edge of a triangle

$$f = \begin{cases} \frac{l_s}{l_{mp}}, & l_s > l_{mp} \\ 1.0, & l_s \leqslant l_{mp} \end{cases} \tag{2}$$

where l_s is the edge length in screen space, l_{mp} is the user's minimum perceptible length. This avoids subdividing the triangles that already appear small in screen space, and the size of tessellated triangle can match the minimum perceptible size too.

3 RESULTS AND DISCUSSION

Three methods are evaluated. Figure 1(c) shows results under different methods and gaze behaviors. In method one[1], the tessellation level of blending region is higher than that of foveal region in Position B. Method two[3] only applies to fixed size triangles. Tessellation results of much larger or smaller triangles do not match the visual perceptual size. Method three in this study can be adaptively applied to different gaze behaviours. With the perceptual model, it subdivides triangles according to user's perceptible size.

Table 1: The number of output vertices and average rendering time under different methods and gaze behaviors

Method	Gaze Behavior	Position	VSInput	DSOutput	Duration (μs)
One	Fixation	A	237279	284619	2585.056
		B	237279	265221	2545.344
Two	Fixation	A	237279	350163	3466.184
		B	237279	616035	6757.072
Three	Fixation	A	237279	111528	2107.008
		B	237279	167448	2388.84
Three	Smooth Pursuit	A	237279	105531	2056.408
		B	237279	117723	2323.153

As shown in Table 1, the number of vertices output and rendering time by method three are reduced by 49% and 12% when compared with method one, are reduced by 71% and 56% when compared with method two. As the gaze point moves, they can be further reduced by 20% and 2% when compared with fixation gaze behavior.

4 CONCLUSION

This paper proposed a perceptual model optimized efficient foveated rendering. Culling area and tessellation levels are dynamically adjusted. Comparing to existing methods, user visual region is divided according to visual sensitivity. Experimental results show better efficiency of proposed method.

ACKNOWLEDGMENTS

This research is supported by Science and Technology Planning Project of Guangdong Province with grant no. 2017B010110007, 2014B040401012 and 2017B010110015, Natural Science Foundation of Guangdong Province with grant no. 2018A030313802, Scientific Research Fund of Sichuan Provincial Science and Technology Department with grant no. 2018GZ0517.

REFERENCES

[1] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D Graphics. *ACM Trans. Graph.* 31, 6, Article 164 (Nov. 2012), 10 pages.
[2] M. Reddy. 2001. Perceptually optimized 3D graphics. *IEEE Computer Graphics and Applications* 21, 5 (Sep 2001), 68–75.
[3] Nicholas T. Swafford, José A. Iglesias-Guitian, Charalampos Koniaris, Bochang Moon, Darren Cosker, and Kenny Mitchell. 2016. User, Metric, and Computational Evaluation of Foveated Rendering Methods. In *Proceedings of the ACM Symposium on Applied Perception (SAP '16)*. ACM, New York, NY, USA, 7–14.
[4] Benjamin Watson, Neff Walker, and Larry F. Hodges. 2004. Supra-threshold Control of Peripheral LOD. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 750–759.