Gaze-Contingent Rendering in Virtual Reality

Fang Zhu[1], Ping Lu[1], Ping Li[2], Bin Sheng[3], and Lijuan Mao[4]

[1] ZTE Corporarion, Nanjing, People's Republic of China
[2] The Hong Kong Polytechnic University, Hong Kong, People's Republic of China
[3] Shanghai Jiao Tong University, Shanghai, People's Republic of China
shengbin@sjtu.edu.cn
[4] Shanghai University of Sport, Shanghai, People's Republic of China
maolijuan@sus.edu.cn

Abstract. Virtual reality (VR) is a technology that relies on a computer graphics system and other external display and control interfaces, to create an immersive experience by generating an interactive three-dimensional environment on a computer. Currently, however, most virtual reality scenes are far behind the real world in naturalism. One of the limitations is the insufficient graphics computing performance of the computer. It is difficult for mainstream consumer GPUs to meet the requirements of high picture quality and high fluency at the same time when running VR scenes, resulting in a reduction in the game's visual experience and even human discomfort. In order to balance the quality and fluency of the picture, the areas within and outside the focus range of the user's sight can be rendered hierarchically, so as to efficiently use computing resources. In order to achieve this goal, the following article proposes a model that combines the saliency information of the virtual scene and the head motion information to predict the focus of the field of view in real time. The model can assign different rendering priorities to objects in the field of view according to the prediction results, and give different priorities, use different rendering algorithms to provide a flexible VR scene rendering optimization solution.

Keywords: VR · Viewpoint prediction · Foveated rendering · Picture saliency

1 Introduction

1.1 Significance

In recent years, with the development and progress of virtual reality (VR) technology and the iterative update of VR devices, more and more people choose to use VR games as one of the daily entertainment options. Compared with games displayed on ordinary screens, VR games provide users with a stronger sense of immersion and realism, which greatly improves the amusement of the game.

In addition, VR can also be used in many other fields, such as education and medical treatment. However, the actual expressiveness of VR technology still has a certain gap compared with traditional three-dimensional scenes, and it is far from the "realistic" rendering target.

The reason is that VR technology requires a larger display angle, a higher graphics resolution and a higher number of display frames than traditional flat three-dimensional display. Related research shows that in order to achieve the ideal display effect, the display angle must reach more than 150°, the graphic resolution calculated in real time should reach 4K or even higher, and in order to avoid the dizziness caused by the grainy picture, the display frames needs to reach more than 90 frames per second.

Currently, consumer-grade graphics processors on the market are difficult to meet these two requirements at the same time, which leads to the fact that most of the VR scenes currently put into application have simple structures, rough details, and poor sense of reality.

Compared with non-VR scenes, they have obvious disadvantages. In order to solve the above problems and give full play to the computing performance under the existing hardware to improve the VR display effect, the academic community has proposed a method for tracking the focus of the sight in real time and rendering the areas inside and outside the focus in a hierarchical way.

The basis of this method is that the focus of the visual field where the user's attention is concentrated at any time, as long as the focus area of the visual field is rendered with high precision and the lower precision processing is performed outside the area, it will make efficient use of computing resources, so as to balance the picture quality and fluency. The main difficulty in implementing this solution is the ac-curate prediction of the focus of the field of view (FoV).

The current high-precision prediction method is to use hardware devices, such as eye trackers, to track the eye movement trajectory and calculate the focus of sight in real time. However, this solution has strong hardware dependence, high implementation cost, and small application scope. Another solution is software-level line-of-sight focus prediction.

The specific method is to detect the saliency, depth information, color information, etc. of the scene in combination with the content of the scene, so as to determine the objects in the scene that may become the focus of attention. Based on the head movement information, the line of sight movement trajectory is calculated accordingly.

Based on extensive investigation of existing research results, this article develops a viewpoint prediction system based on head motion and scene information, as well as hierarchical rendering based on the prediction. The structure is simple and easy to use. The main contributions of this system are as follows:

1. The system is completely based on software implementation, and does not rely on eye tracker equipment to achieve better viewpoint prediction, and on this basis, the viewpoint rendering is realized, reducing the rendering overhead.
2. The system structure is streamlined and efficient, and can achieve higher rendering effects under general hardware conditions.

3. By analyzing the experimental results, this article proposes the future development direction of VR rendering.

1.2 Article Structure

The workflow of this article is shown in the Fig. 1 below. First, the motion information of the device sensors including speed, acceleration, etc. is collected through the program, and the saliency information is calculated from the scene screenshots.

Then, the obtained information is input into the Sgaze [6] model, and the predicted view-point position is output, then pass the viewpoint position to the foveated rendering program, and finally realize the foveated rendering of viewpoint tracking.

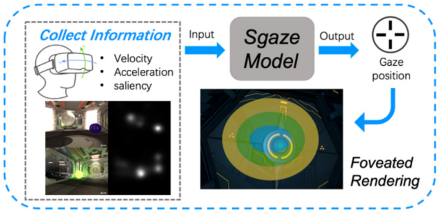

Fig. 1. System structure

2 Related Work

2.1 Image-Based Viewpoint Prediction

Image-based viewpoint prediction has been studied in the field of computer vision, and many saliency models have been proposed in the past thirty years. L. Itti et al. [7] proposed a traditional viewpoint model, which uses multi-scale image features to calculate a saliency map. Oliva et al. [15] noticed the importance of scene content and proposed a saliency model that takes scene content information into account. In short, most existing models use basic information such as color and brightness of the image [2], or specific scenes and objects in the image [10,11]. With the development of deep learning, many models based on CNNs have achieved good performance [5,12,24]. Most of the aforementioned models are applied to a single picture. In addition to these models, researchers also studied the saliency of stereoscopic images [5,9] and videos [18,20], both Sitzmann et al. [16] and Rai et al. [16] studied the saliency of 360° panoramic images. Xu et al. [22] established a model for viewpoint prediction in 360° immersive video. These models usually calculate a density map of the position of the viewpoint rather than directly predict the position of the line of sight in real

time. However, in VR applications, functions such as foveated rendering based on view-points and the interaction of eyeball rotation require real-time line-of-sight positions, and calculating positions from density maps is not efficient enough.

2.2 The Relationship Between Eye and Head Movements

The relationship between eye and head movements has been studied in recent years. Yarbus [23] found that when the gaze shifts, the eye and head movements are always coordinated and related to visual cognition. Nakashima and Shioiri [13] further explained that the difference in line-of-sight direction and head direction can interfere with visual processing, that is, humans have the highest visual cognition efficiency when the two directions are the same.

Einhauser et al. [3] discovered the coordination of eyes and heads when people freely observed natural scenes, and some work [1] revealed a delay between eye rotation and head rotation, the former is usually faster than the latter. Many studies have focused on the magnitude of head rotation and eyeball rotation, and have shown that they are closely related [4]. Stahl [19] found that when the eye rotation range is limited to a small range, the head will not rotate, and when the eye rotation range is large, the head rotation range is linearly related to the eye rotation range within a certain range. Nakashima et al. [14] used head rotation information to successfully improve the accuracy of saliency prediction.

2.3 Foveated Rendering

In the field of computer graphics, foveated rendering is a widely studied subject. Based on fixed-point grading or the line-of-sight position obtained by the eye tracker device, VR devices have also begun to implement grading rendering in practical applications.

3 Content and Methods

3.1 Hardware System

This article uses Oculus Rift as the experimental equipment, and the rendering of the scene and the running of the script are implemented through Unity. The system environment of the entire experiment is Windows10 1903, and the CPU and GPU of the platform are Intel® Core i7-9750H @ 2.6 GHz and NVIDIA GeForce 1660Ti 6 GB.

3.2 Viewpoint Prediction

According to the experience and intuition of daily life, there is a strong correlation between the movement of the eyeball and the head, that is to say, when the head turns in a certain direction, the eyeball also has a high probability of moving in the same direction.

It is reflected in the mathematical model that the position of the viewpoint has a certain linear correlation with the speed and acceleration of the head rotation. Many studies [1,21] have shown that there is a delay between head movement and eye movement.

Head movements tend to lag behind eye movements [21], and the magnitude of the delay is different in different speed regions. Therefore, the head motion information used in the prediction model should be the speed and acceleration ahead of the current certain time. In actual situations, the factors that affect human eye movements are complex and diverse, including the current scene, purpose, and delay. Combining various factors, the prediction formula given by the Sgaze model [6] is:

$$\begin{cases} x_{gaze} = a_x \cdot \omega_x(t + \Delta t_{x1}) + b_x \cdot \beta_x(t) + F_x(t + \Delta t_{x2}) + G_x(t) + H_x(t) \\ y_{gaze} = a_y \cdot \omega_y(t + \Delta t_{y1}) + F_y(t + \Delta t_{y2}) + G_y(t) + H_y(t) \end{cases} \tag{1}$$

Picture Saliency and Viewpoint Prediction: The saliency information of the scene picture also has a great influence on human viewpoint prediction; therefore, it is reasonable to introduce the saliency information of the picture image into viewpoint prediction. The model used in this article uses SAM-ResNet saliency predictor [2]. In practical applications, it takes too much resources to calculate the saliency of the image; therefore, it is impossible to achieve real-time prediction. We design to calculate the saliency value of the scene for every 250ms, and only the central part of the picture, to reduce the time spent on prediction. The saliency highlight image in collect information of Fig. 1 is an example of the saliency calculation of the scene in this article.

3.3 Foveated Rendering

MBFR. Mask Based Foveated Rendering (MBFR) is relatively easy to implement foveated rendering, its method is as follows:

1. According to the distance between the pixel and the viewpoint, the image is divided into two areas (or more)
2. During the rendering process: the higher priority area is rendered without any special processing; the lower area discards some pixels without rendering but only reconstruct by calculating the average value from the neighboring pixels.

4 Experiment Results and Analysis

4.1 Test Methods and Evaluation Standards

This article uses the Unity official scene "Corridor Lightning Example" for testing. The scene has more lighting effects. Most of the materials in the scene use

the Standard (Specular set-up) shader, which makes the scene have a lot of specular reflection light. There are moving balls and shadows in the scene. During the conversion of the field of view, the model will calculate the position of the viewpoint and use this as the basis to achieve foveated rendering.

When the foveated rendering is turned on and off separately, Unity's own profile performance analyzer will record the GPU time and CPU time required to observe the rendering of a frame and calculate the average. There are more detailed rendering steps in GPU time. Viewpoint prediction results and foveated rendering effects are evaluated by the subjective feelings of participating testers.

4.2 Viewpoint Prediction

Due to the lack of an eye tracker to obtain an accurate gaze position, the reference position of the line of sight is obtained by collecting subjective marks of the participants. Figure 2 shows the effect of line-of-sight prediction, where green dots indicate the line of sight of the participants and red dots indicate the position of the line of sight, and the accuracy of the prediction results is high.

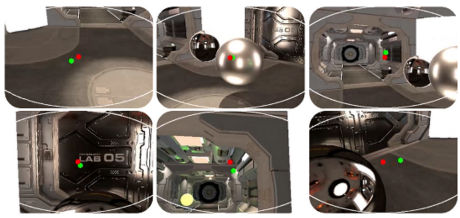

Fig. 2. Viewpoint prediction and ground truth

4.3 Rendering Effect

In the area where the scene is far away from the center of the viewpoint, we per-formed a relatively blurry rendering, but from the perspective of the tester. From the perspective of actual experience, there is no loss of perceived detail, and the expected effect is achieved from a subjective perspective.

4.4 Rendering Efficiency

Table 1 shows the rendering calculation time. It can be seen from the table that the overall rendering time per frame has decreased by about 14% after the foveated rendering is turned on, of which the rendering process time has dropped by about 26%, and the post-processing process has increased by about 4 times. Since the post-processing effect of the test scene is less, pixel reconstruction has become the main part of post-processing. As can be seen from the table, the reduction in rendering time for pixel discarding is greater than the time for pixel reconstruction, so the foveated rendering of this project successfully improves rendering efficiency.

Table 1. System load changes when Hierarchical Rendering On and Off

Foveated rendering	Total time (ms)	Drawing time (ms)	Image effect (ms)
On	1.276	0.893	0.198
Off	1.428	1.208	0.051

5 Conclusion

This article establishes a viewpoint prediction system based on head motion and scene information, implements foveated rendering based on this, afterwards, tests and analyzes this system. The analysis and comparison of viewpoint prediction accuracy, rendering effect and rendering efficiency prove that the system can predict the line-of-sight position more accurately without eye tracker and can improve rendering efficiency without reducing too much image quality and other advantages. Combining viewpoint prediction with foveated rendering are important development directions for VR rendering in the future.

On top of the current results, the following work can be done to further improve the system. Test the project's foveated rendering method in more complex scenarios and collect more data to analyze the actual performance of the method; use an eye tracker to obtain the true position of the line of sight, compare the predicted position of the project method with it, and quantify the error size; try more methods such as deep learning algorithms applied to viewpoint prediction, and compare with the current method, try to improve accuracy; try to use Other technologies such as Variable Rate Shading (VRS) implement foveated rendering and compare with current methods.

Acknowledgement. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFF0300903, in part by the National Natural Science Foundation of China under Grant 61872241 and Grant 61572316, and in part by the Science and Technology Commission of Shanghai Municipality under Grant 15490503200, Grant 18410750700, Grant 17411952600, and Grant 16DZ0501100.

References

1. Biguer, B., Jeannerod, M., Prablanc, C.: The coordination of eye, head, and arm movements during reaching at a single visual target. Exp. Brain Res. **46**(2), 301–304 (1982)
2. Borji, A., Sihite, D.N., Itti, L.: Probabilistic learning of task-specific visual attention. In: Computer Vision and Pattern Recognition, pp. 470—477 (2012)
3. Einhauser, W., et al.: Human eye-head co-ordination in natural exploration. Netw. Comput. Neural Syst. **18**(3), 267–297 (2007)
4. Fang, Y., Nakashima, R., Matsumiya, K., Kuriki, I., Shioiri, S.: Eye-head coordination for visual cognitive processing. PloS ONE **10**(3), e0121035 (2015)
5. Guo, F., Shen, J., Li, X.: Learning to detect stereo saliency. In: International Conference on Multimedia and Expo (ICME), pp. 1–6 (2014)

6. Hu, Z., Zhang, C., Li, S., Wang, G., Manocha, D.: SGaze: a data-driven eye-head coordination model for realtime gaze prediction. IEEE Trans. Visual Comput. Graphics **25**(5), 2002–2010 (2019)
7. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
8. Lu, P., Sheng, B., Luo, S., Jia, X., Wu, W.: Image-based non-photorealistic rendering for realtime virtual sculpting. Multimedia Tools Appl. **74**(21), 9697–9714 (2014). https://doi.org/10.1007/s11042-014-2146-4
9. Kamel, A., Sheng, B., Yang, P., Li, P., Shen, R., Feng, D.D.: Deep convolutional neural networks for human action recognition using depth maps and postures. IEEE Trans. Syst. Man Cybern. Syst. **49**(9), 1806–1819 (2019)
10. Karambakhsh, A., Kamel, A., Sheng, B., Li, P., Yang, P., Feng, D.D.: Deep gesture interaction for augmented anatomy learning. Int. J. Inf. Manage. **45**, 328–336 (2019)
11. Kümmerer, M., Wallis, T., Gatys, L., Bethge, M.: Understanding low-and high-level contributions to fixation prediction. In: 19th IEEE International Conference on Computer Vision (ICCV 2017), pp. 4799–4808 (2017)
12. Meng, X., et al.: A video information driven football recommendation system. Comput. Electr. Eng. **85**, 106699 (2020). https://doi.org/10.1016/j.compeleceng.2020.106699
13. Nakashima, R., Shioiri, S.: Why do we move our head to look at an object in our peripheral region? lateral viewing interferes with attentive search. PloS ONE **9**(3), e92284 (2014)
14. Nakashima, R., et al.: Saliency-based gaze prediction based on head direction. Vis. Res. **117**, 59–66 (2015)
15. Oliva, A., Torralba, A., Castelhano, M.S., Henderson, J.M.: Top-down control of visual attention in object detection. In: International Conference on Image Processing, pp. 253–256 (2003)
16. Rai, Y., Gutierrez, J., Callet, P.L.: Dataset of head and eye movements for 360 degree images. In: ACM SIGMM Conference on Multimedia Systems, pp. 205—210 (2017)
17. Sitzmann, V., et al.: Saliency in VR: how do people explore virtual environments? IEEE Trans. Visual Comput. Graphics **24**(4), 1633–1642 (2018)
18. Sheng, B., Li, P., Zhang, Y., Mao, L.: GreenSea: visual soccer analysis using broad learning system. IEEE Trans. Cybern., 1–15 (2020). https://doi.org/10.1109/TCYB.2020.2988792
19. Stahl, J.S.: Amplitude of human head movements associated with horizontal saccades. Exp. Brain Res. **126**(1), 41–54 (1999)
20. Wang, W., Shen, J., Shao, L.: Consistent video saliency using local gradient flow optimization and global refinement. IEEE Trans. Image Process. **24**(11), 4185–4196 (2015)
21. Whittington, D.A., Heppreymond, M.C., Flood, W.: Eye and head movements to auditory targets. Exp. Brain Res. **41**(3–4), 358–363 (1981)
22. Xu, Y., et al.: Gaze prediction in dynamic 360 immersive videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5333–5342 (2018)
23. Yarbus, A.: Eye Movements and Vision. New York (1967)
24. Zhang, P., Zheng, L., Jiang, Y., Mao, L., Li, Z., Sheng, B.: Tracking soccer players using spatio-temporal context learning under multiple views. Multimedia Tools Appl. **77**(15), 18935–18955 (2017). https://doi.org/10.1007/s11042-017-5316-3