Central-Eye: Gaze Tracking Research on Visual Rendering Method in Industrial Virtual Reality Scene

Meiya Dong
Taiyuan University of Technology
P.R.China
dongmeiya@163.com

Jumin Zhao*
Taiyuan University of Technology
Jinzhong, Shanxi, P.R.China
zhaojumin@tyut.edu.cn

Dianqi Wang
Jinan Urban and Rural Water Authority
P.R.China
esa.eten@163.com

Xin Ding
Jinan Heating Group Ltd.
P.R.China
77816491@qq.com

Zhaobin Liu
Suzhou Vocational University
P.R.China
zbliusz@126.com

Biaokai Zhu
Shanxi Police College
Taiyuan, P.R.China
hongtaozhuty@gmail.com

ABSTRACT

With development of computing technology, virtual reality devices are widely used in Internet-enabled industrial platforms. The problem of virtual reality stuns caused by the increase in visual frequency and real-world scenarios in industrial applications is still remaining the core problem. We proposes a new method called Central-Eye. It adopts a new context-aware solution for eye tracking to obtain the focus of the human eye on the screen, and then based on the distribution of human visual acuity, down-sampling the rendering in the visual edge area to increase the rendering speed to solve the user experience problem. Aiming at the characteristics of industrial virtual reality equipment with large field of view, high frame rate requirements, and eyepiece distortion, we propose a new gaze tracking method combining pattern recognition with local linear fitting. From experimental evaluation, we finally conclude the research that Central-Eye method can virtually improve the user experiences.

KEYWORDS

Rendering Acceleration, Industrial Vision, Gaze Tracking, Feature Extraction, Central-Eye

ACM Reference Format:
Meiya Dong, Jumin Zhao, Dianqi Wang, Xin Ding, Zhaobin Liu, and Biaokai Zhu. 2020. Central-Eye: Gaze Tracking Research on Visual Rendering Method in Industrial Virtual Reality Scene. In *ACM Turing Celebration Conference - China (ACM TURC 2020) (ACM TURC 2020), May 22–24, 2020, Hefei, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3321408.3322865

*This is the corresponding author

ACM TURC 2020, May 22–24, 2020, Hefei, China

ACM ISBN 978-1-4503-7158-2/19/05.
https://doi.org/10.1145/3321408.3322865

1 INTRODUCTION

With the continuous development of displays and smart phones, a new generation of lightweight, wearable, and highly practical industry virtual reality devices will become the next generation of human-computer interaction computing platforms after PCs, CRFID and smart phones [11]. At present, virtual reality (VR) has become a widely accepted field in academia and industry. Virtual reality technology will also gradually change people's production and lifestyle.

Commonly we define virtual reality as it uses a computer as a platform to create realistic sensory stimuli through digital devices, allowing users to have an immersive experience. Users can also use natural interaction methods such as head rotation, gestures and even moving in a virtual space to interact with the virtual world[1][2][3]. This brand-new experience and interaction method has broad application prospects in industry.

However, there is still a very important problem in virtual reality devices so far. The user experience of virtual reality is not as good as expected [6]. According to research, a large number of users quickly change from pleasant immersion to extreme symptoms such as disgust, discomfort, disorientation, and nausea. This phenomenon is called Virtual Reality Sickness.

To solve this problem, we focus on designing and producing a virtual reality device with gaze tracking technology. According to the characteristics of the limited space of virtual reality equipment, severe light pollution, and the need to ensure the safety of the user's eyes, a novel algorithm was proposed to track the sight points of interest in real time. The pattern recognition method is used to avoid the eyelid occlusion problem that cannot be solved by traditional human eye recognition based on spatial geometry, and then the local linear fitting method is used to achieve high-precision focus determination on small-scale training samples.

The Central-Eye system performance and actual display effects of the module of gaze tracking was measured, proving that the system proposed in this study can be used.

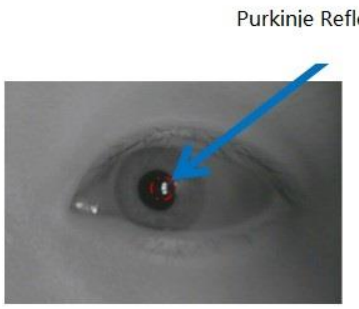

Figure 1: Purkinje image diagram

2 RELATED WORK

In this section, we will introduce current research status of gaze tracking technologies at home and abroad.

The research of human eye gaze tracking has been attracting much attention in the field of Human Computer Interaction. In the past few decades, gaze tracking technology has been a big challenge for researchers and different companies in the fields of IT, medical equipment or multimedia business equipment. The gaze tracking system is a device based on tracking human eye movements, and is used to accurately assess the position where the user is watching and the gaze time. It involves pupil detection algorithms, image processing, data filtering, and recording eye movements through fixed points, fixed times, and glances. With the advancement of technology, researchers and companies have tried a variety of hardware and software methods to choose the appropriate eye movement acquisition equipment and software algorithms according to different application needs. The related work of gaze tracking is briefly introduced below.

Gaze tracking based on Purkinje images: Purkinje images come from the reflection of light sources on different surfaces of the eyeball. In the gaze tracking system, an infrared camera and an infrared light source are usually used as equipment. The infrared light source illuminates the eye to generate a Purkinje image (see Figure 1). The small white bright spot next to the pupil is the Purkinje reflection point). The Purkinje image is derived by compensating the refraction of the corneal surface. The center of the corneal curvature is the direction of the line of sight. The reflection point is used as a reference point for line-of-sight estimation. When the eye or head moves, the vector difference between the pupil and the reflection point remains constant [5]. The researchers found that adding multiple light sources to the system gave better results than using a single point light source.

The difficulty of this method is how to capture a sufficiently high-resolution image in a limited field of view. In order to ensure the stable reflection of Purkinje images by the eyeball, this method usually requires multiple high-intensity infrared light sources, which leads to an increase in the amount of equipment, which is relatively bulky, and is often applied to the gaze tracking of non-head-mounted devices in real-life scenarios. In order to ensure the stable reflection of Purkinje images by the eyeball, this method usually requires multiple high-intensity infrared light sources, and in order not to restrict the user's head movement, multiple cameras are usually required. However, the space of the virtual reality helmet is limited, it is not suitable to use multiple devices, and there is a danger of burning human eyes with multiple LED infrared light sources, so this solution is not suitable for the GazeRender system. Eye tracking based on human eye geometric model: This method approximates the eye structure to a simple geometric model, as shown in Figure 2 [8]. The eyeball is a sphere with a radius R. The contour of the iris is a circle with a radius r. The direction of the line of sight can be determined by the visual axis of the eye, which passes through the center of the eyeball and the center of the iris. When the user looks around, the eyeball rotates around the center point, and the direction of the line of sight changes accordingly. For example, Nishino and Narya [7] use an ellipse model with fixed eccentricity and size to approximate the cornea. Wang et al. [10] used the normal direction of the iris to approximate the ellipse to determine the gaze direction. Zhang et al. [8] used the iris ellipse to project the eyeball in 3D space to obtain the direction of sight. All of the above methods also require a personal calibration.

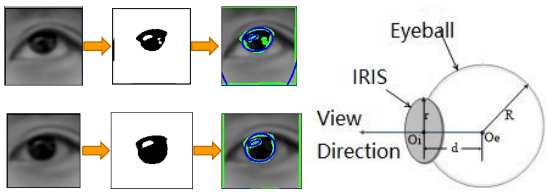

Figure 2: Schematic diagram of the human eye geometric modeling solution and geometric model.

This method is usually used to identify local features of the eye that do not change much in illumination and viewing angle [4], and performance problems occur when the ambient light is unstable or the human eye features are difficult to extract. Therefore, this solution has poor anti-interference ability, and is affected by factors such as changes in ambient light, occlusion of eyelids, confusion of iris and eyelash colors, and rotation of the eyeball. Therefore, this solution is not applicable to GazeRender system. Gaze tracking based on template matching: Usually based on the overall picture of the eye as a training sample, the image data is mapped to screen coordinates to estimate the direction of the gaze. For example, Tan et al. [9] used the local linear characteristics of the apparent manifold of the eye and collected 252 training samples for interpolation. Sugano et al. [8] used saliency maps to generate the probability distribution of gaze points. From user eye images watching video clips, a training data set consisting of saliency maps and human eye images at a fixed position was obtained, although the user's initiative Collect training samples, but the accuracy is only 6°. The method based on template matching is to directly map on the image content. Usually, it is not necessary to calibrate the camera

and geometric data, but collecting a large number of training samples virtually increases the user's burden. In summary, none of the existing gaze tracking technologies are completely applicable to the gaze tracking in the virtual reality device of this paper.

3 CENTRAL-EYE GAZE TRACKING SYSTEM RESEARCH

The main functions and inputs and outputs of each module in the Central-Eye system architecture are shown in the figure. The hardware basis of the gaze tracking module is a 120HZ infrared camera, which is integrated in the virtual reality helmet to track the eyes. This module uses pattern recognition to track the eyes. The workflow mainly includes human eye detection, pupil feature point extraction, model training and real-time tracking. Among them, human eye detection occurs every time the device is worn to calibrate the position of the human eye. The training of the model occurs when the user wears the device for the first time. As long as the training is performed once, real-time gaze tracking can be performed later. This module outputs the coordinates (Ex, Ey) of the focus position of the human eye on the screen at a refresh rate of 120HZ (As shown in Fig.3).

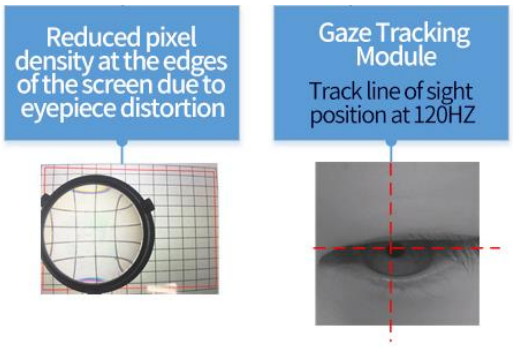

Figure 3: Central-Eye Gaze Tracking.

This section will analyze the characteristics of the human eye recognition technology integrated in the virtual reality helmet, and propose a hardware integration scheme for gaze tracking in the virtual reality helmet. The gaze tracking of the virtual reality helmet has the following characteristics:

(1) The space in a virtual reality helmet is limited. The solution using multiple cameras will occupy limited space, and multiple LED infrared light sources pose a risk of burning to the human eye. Therefore, a multi-camera multi-LED point light source solution cannot be adopted.

(2) The virtual reality helmet can only use infrared light sources. The scheme for identifying the iris depends on visible light, which is not feasible under infrared light sources.

(3) The field of view of virtual reality is about 110 degrees, and severe eyelid occlusion, inability to distinguish between the iris and the corner of the eye will inevitably occur at large angles.

(4) In order to avoid dizziness, the refresh rate of the screen is required to be above 120HZ, and the refresh rate of line-of-sight tracking must be greater than or equal to the refresh rate of the screen to avoid delay. Therefore, the refresh rate of the camera is required to be above 120HZ.

Based on the basic principles of gaze tracking technology and the use scenarios of virtual reality helmets, to accurately perform eye tracking while meeting the above requirements, we propose a hardware solution as shown in Fig.4:

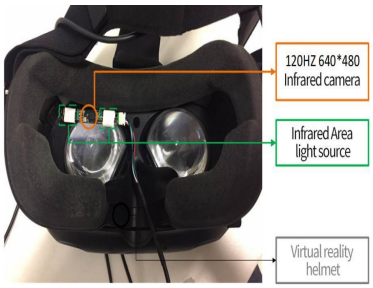

Figure 4: Hardware integration scheme.

Based on the above hardware scheme, the human eye tracking technology mainly includes three parts: feature extraction, model training, and real-time tracking.

3.1 Feature extraction

The steps of feature extraction in the image acquired by the infrared camera is introduced in this section. Gaze tracking algorithm proposed in this paper uses the position of the pupil in the human eye as a feature. Therefore, feature extraction is divided into two steps. First, human eyes are detected in the picture, and the area where the human eyes are located is accurately extracted. Then, in the captured human eye pictures, through a series of image processing methods, the relative position of the pupil in the human eye is extracted. This process includes following flow Fig.5:

3.2 Human eye detection

In this research, the object detection algorithm based on Haar features in OpenCV was used for human eye recognition, and a pre-trained model was directly used [3]. However, the human eyes under the infrared camera have different color expressions from the human eyes in this training model. The specific manifestation is that the gray levels of the iris and eyelids in the pictures taken by the infrared camera are very light. In order to be applicable to this pre-trained human eye detection model, we will enhance the contrast of infrared photos during the human eye detection stage to improve the recognition accuracy in this model.

After measurement, the recognition speed of the above-mentioned human eye detection method is 30ms per frame, and the accuracy is 90%. In order to ensure that the subsequent algorithms have higher accuracy and stability, after the user wears the device, the system will ask the user to

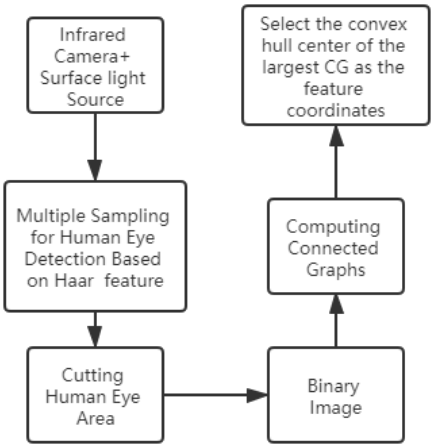

Figure 5: Feature extraction Flowchart of Human Eyes.

stare at the center point of the screen to ensure a stable eye posture, and then perform eye detection for 10 consecutive frames. Take the result with error within 10 pixels as the correct result set, and then take the average of the above result range as the final range for human eye recognition.

After the above calibration, the range error of the human eye detection is controlled within 3 pixels, which is sufficient to ensure the accuracy required for subsequent feature extraction.

3.3 Binary image

Through the steps of human eye detection, the precise range of the human eye is obtained. Then each frame of the image is taken out of the human eye. The captured human eye picture contains iris, pupil, eyelashes, eyelids and other structures. The goal of feature extraction is to obtain features related to the sight, remove irrelevant details, and use it for model training. As mentioned earlier, the feature we extracted is the relative position of the pupil in the orbit.

3.4 Model training and real-time tracking

We build gaze tracking model based on the nearest neighbor pattern recognition method. This section mainly introduces the training method and tracking algorithm of the model. The characteristic of this model is that as long as the limited points on the screen are trained, continuous gaze tracking can be performed. We have extracted the relative position of the human eye pupil in the orbit as the feature coordinates, and the idea we adopted is to directly establish a mapping from the feature coordinates to the coordinates on the screen. After fixing the relative position of the human eye and the screen, we first detect the position of the human eye, so that subsequent recognition of feature points with low computational complexity is performed at each frame. Then we ask the test user to track the anchor point trajectory in the screen as shown in Figure 6, and record the feature coordinates corresponding to each anchor point.

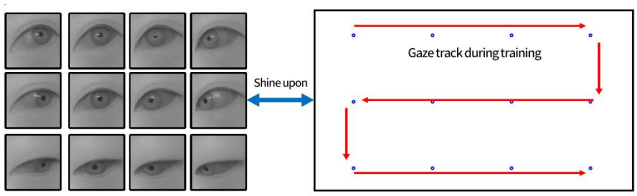

Figure 6: Schematic diagram of human eye feature mapping screen model training.

Then we save the feature coordinates corresponding to each anchor point in the form of a matrix file as a model. When performing real-time tracking of the human eye, the program will read the model from the file. This model can be used in all real-time eye tracking after the user. Only initial eye position detection and center point calibration are required.

The steps of the real-time tracking program are shown in Figure 7. First, human eye detection and center verification are performed to determine the position and range of the human eye. Later, this rectangular frame will be used to intercept the human eye. Then the real-time tracking program reads the mapping matrix of feature coordinates to screen points, and combines it with the position of the human eye to form a real-time tracking model. However, since the model training can only map a limited number of anchor points, we have the following determined mapping functions:

$$f(Ex_i, Ey_i) = (Sx_i, Sy_i),\ i = 0, 1, 2, \cdots n \qquad (1)$$

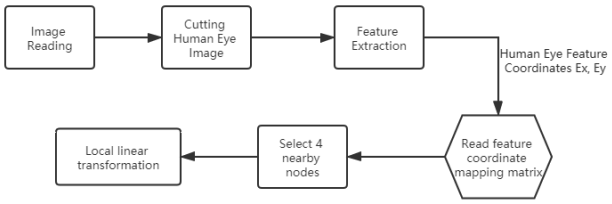

Figure 7: Real-time human eye tracking flowchart

The following mainly introduces the continuous mapping of human eyes in order to achieve continuous eye tracking, using nearest neighbors and local transformation properties. Before that, we can easily think of a simple nearest neighbor pattern matching algorithm. After calculating the feature coordinates (Ex, Ey) in step 3, we directly find the nearest point (Ex_i,Ey_i) in the training model to get the mapping. The result is (Sx_i,Sy_i), however, this will cause the identified points to be discrete, and the eye tracking will jump.

Therefore, in order to solve this situation, we propose a local linear model. Although the distribution of the overall screen anchor points and the distribution of model feature

coordinates do not correspond linearly, locally, we can think of the feature points and anchor points as linear correspondence of. Therefore, we can first calculate the relative distance between the feature coordinates (Ex, Ey) and the nearest four points, and then map them to the screen in proportion to calculate the screen coordinates corresponding to any one of the feature coordinates. Suppose the four nearest neighbors are:

$$(Ex\_0, Ey\_0\ ), (Ex\_1, Ey\_1\ ), (Ex\_2, Ey\_2\ ), (Ex\_3, Ey\_3\ ) \tag{2}$$

The corresponding four anchor points are,

$$(Sx_0, Sy_0), (Sx_1, Sy_1), (Sx_2, Sy_2), (Sx_3, Sy_3) \tag{3}$$

Then, the screen coordinates corresponding to the feature coordinates (Ex, Ey) are,

$$\begin{aligned} Sx = \tfrac{1}{2}\{&(Sx_0 + \tfrac{|Ex-Ex_0|}{Ex_0-Ex_1} \times |Sx_0 - Sx_1|) \\ &+(Sx_3 + \tfrac{|Ex-Ex_3|}{Ex_3-Ex_2} \times |Sx_3 - Sx_2|)\} \end{aligned} \tag{4}$$

$$\begin{aligned} Sy = \tfrac{1}{2}\{&(Sy_0 + \tfrac{|Ey-Ey_0|}{Ey_0-Ey_3} \times |Sy_0 - Sy_3|) \\ &+(Sy_1 + \tfrac{|Ey-Ey_1|}{Ey_1-Ey_2} \times |Sy_1 - Sy_2|)\} \end{aligned} \tag{5}$$

The screen coordinates (Sx, Sy) corresponding to the feature coordinates (Ex, Ey) are calculated through the above formula, and this is used as the current human eye's area of interest for the picture, which is the input of the foveal rendering module for future study.

4 EXPERIMENTAL EVALUATION

In this section we will introduce the whole designed experiment. First, it introduces the software and hardware platforms implemented by the system, and then defines the evaluation criteria for each module based on the virtual reality acceleration scenario, and formulates an experimental scheme. In this section, we conducted specific experimental evaluations and analyses.

4.1 System implementation

In terms of hardware, we use the combination of the vision tracking system and virtual reality helmet HTCVive described above, using a desktop computer with a high-performance graphics card as the computing platform. The hardware configuration in the system is shown in following Table. On the software development platform, we choose Unity 3D 5.5 as the development engine of the virtual reality program, use DirectX 11 as the graphics rendering engine, and use OpenCV 3.0 as the graphics processing library in sight tracking.

Based on the software and hardware platforms described above, we have implemented a gaze tracking module supporting to form a complete context-aware virtual reality rendering acceleration system. As shown in the Tab.1 is a schematic diagram of the experimental device, assembled according to the hardware design scheme. After wearing the device, the user will see the virtual reality scene in the helmet, and the infrared camera is illuminated by the infrared surface light source. Shoot the human eye to complete the gaze tracking(As shown in Fig. 8).

Table 1: System hardware configuration list

Name	Major Configuration
PC	CPU: Intel i7-6700 4.00GHZ Storage:16GB Nvidia GTX970 Windows 10.
HTCVive	Screen Definition: 2160*1200 View Angle: 110*110 Frequency: 850nm.
Infrared Camera	Definition: 640*480 Frame:120HZ

Figure 8: Schematic diagram of the experimental setup.

4.2 Experimental design

his section mainly introduces the evaluation criteria and corresponding experimental schemes of each module. The overall goal of the Central-Eye system is to reduce the visual edge sampling rate of virtual reality rendering and increase the frame rate of the picture, so that virtual reality motion sickness can be avoided. Then the evaluation of this system should start from the above design goals, mainly around the following two points: first, the visual effect should be as close to full sampling as possible, so that the user does not perceive the down-sampling of the concave rendering on the visual edge; In terms of module performance, it can match the frame rate requirement of 120HZ for virtual reality. So based on the above goals, we formulated the following experimental goals and schemes for the module:

The main evaluation criteria of this module are the accuracy and refresh rate of tracking. Gaze tracking accuracy largely determines the effect of foveal rendering. The accuracy is determined by measuring the deviation between the eye-tracking landing point and the actual landing point. Because the human eye rotates on a spherical surface and the screen is a flat surface, we use the difference in viewing angle as a measure between the tracking landing point and the time focus. This is in line with the model of using fovea to determine the sampling rate in foveal rendering.

In addition to the average value, another criterion for evaluating the accuracy of gaze tracking is stability, which

refers to the concentration of the accuracy distribution of eye tracking, which is even more important than the average value of accuracy, because accuracy can be properly centered. The concave rendering algorithm adds fault tolerance interval to compensate. And if the tracking accuracy variance is relatively large, there is a large chance that the user feels that the rendered high-definition area is out of his line of sight, so we also need to measure the distribution of accuracy. The third evaluation index of eye tracking is the refresh rate. According to the related research of virtual reality, only the frame rate of 120HZ can meet the requirement of the screen refresh rate of human head rotation, so in order to track the human eye in time in each frame Position, the refresh rate of human eye tracking must also meet the 120HZ requirement. In the experiment, we will measure the speed of human eye tracking, and record its average and jitter interval. Like the accuracy of gaze tracking, speed will also be measured from both the average and distribution.

4.3 Gaze Tracking Module Evaluation

The error of the line-of-sight tracking represents the angle difference between the calculated focus f' and the actual line-of-sight focus f. The experimental model is shown in Figure 9:

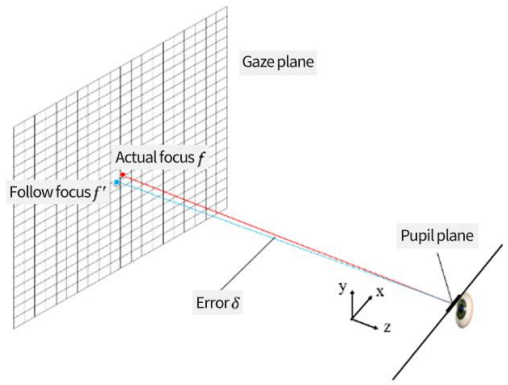

Figure 9: Measuring gaze tracking accuracy.

According to the above experimental model, we performed user measurement on the gaze tracking with a sample size of 10 people, and took the measurement results of one of the experiments for monomer analysis.

Figure 10 shows the result image of a single sample of the gaze tracking experiment, from which we can make a preliminary assessment of the results and observe the characteristics of the spatial distribution. The blue circle in the figure represents the target point that we calibrate to the user. After the user looks at the target point, click the button, and print the focus position identified by the current gaze tracking program on the picture, that is, the Red Cross mark. It can be seen that the red mark is basically close to the blue target point, which indicates that the gaze tracking model of the present invention is more accurate. Judging from the distribution of accuracy on the screen, as the closer to the edge of the screen, the error of eye tracking gradually increases, the main reason is that at this time the eyeball rotation angle increases, the pupil center and the point on the screen are no longer close to linear This will lead to an increase in the distortion of the linear model we use, which is not as accurate as the recognition in the center area of the screen. To solve this problem, during the model training phase, we can appropriately increase the density of the visual edge region sampling. Below we perform a quantitative analysis of the overall experimental results of the eye-tracking speed. In the experiment, the variable that can be controlled is the density of sampling points. Obviously, the higher the sampling density, the more accurate the identification. However, too many sampling points will be very troublesome for users in actual products, so we hope to achieve the required tracking error with the least sampling points.

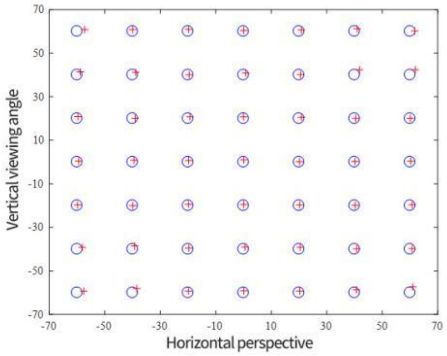

Figure 10: Single sample result of gaze tracking.

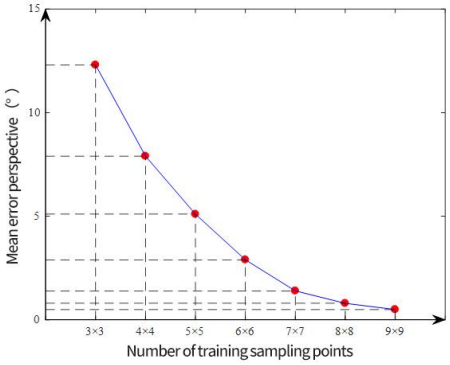

Figure 11: Relationship between average tracking error and training sampling point density.

Figure 11 shows the relationship between tracking error and training sampling point density in all experimental samples. In this study, it is considered that an error with an average of about 1° is an acceptable range, so the 8 * 8 sampling point density can meet the needs of the accuracy of line-of-sight tracking. In addition to the average value of the

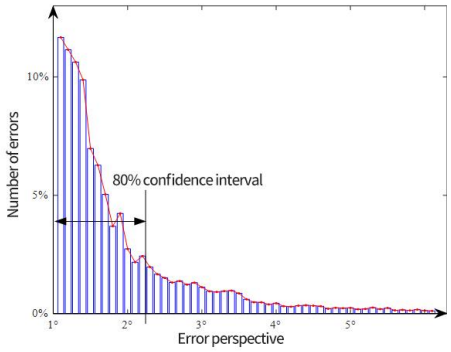

Figure 12: Error distribution.

error, we also pay more attention to the distribution of the error. Compensation can be performed during discretization. So in we select the 80% confidence interval and use the right boundary value of the interval as the compensation value. It can ensure that 80% of the cases are within the error range. In this way, it is almost impossible for the consecutive frames to exceed the error range. Users are unaware of the instability caused by recognition errors(as shown in Fig.12).

5 CONCLUSION

This paper studies gaze-tracking context perception to accelerate the rendering rate in virtual reality, thereby avoiding motion sickness in industry. Gaze tracking shows its mature research results. However, combining and applying them to the emerging field of virtual reality faces many unknown challenges. This paper, as a relatively advanced exploration research in the field of virtual reality applications, first designs and innovates theoretically, and develops and experiments based on existing virtual reality hardware systems and software platforms. This paper proposed gaze tracking technology based on pattern recognition. Existing gaze tracking technology often estimates the direction of the line of sight from the spatial geometric model. This method cannot solve the situation of eyelid occlusion. Among them, the more accurate technical solutions often use auxiliary infrared point light sources, which will cause harm to the human eye. This study considers the use of a more robust pattern recognition method to track the eyes under the harmless infrared surface light source. This research implements the virtual helmet based on industrial applications of the system, and experimentally tests each module through experiments and user research. The results show that the eye-tracking module designed in this study can track human eyes without harming the user. Through further compensation of the sampling rate distribution, the user cannot detect the slight tracking error. The above results show that the system designed and implemented in this research is a practical solution for accelerating virtual reality rendering, and the design has innovations for industrial virtual reality scenes.

REFERENCES

[1] Meiya Dong, Jumin Zhao, Dengao Li, Biaokai Zhu, and Zhaobin Liu. Cloak: visible touching and invisible protecting: cloud privacy protection based on lsb and chaotic approach. In *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*, pages 225–229. IEEE, 2018.

[2] Meiya Dong, Jumin Zhao, Biaokai Zhu, and Zhaobin Liu. Chameleon: Hides privacy in cloud iot system by lsb and cse. *Concurrency and Computation: Practice and Experience*, 31(24):e5505, 2019.

[3] Zhenge Guo, Zhaobin Liu, Jizhong Zhao, Hui He, and Meiya Dong. Towards secure device pairing via vibration detection. In *International Conference on Cloud Computing and Security*, pages 177–186. Springer, 2018.

[4] Giancarlo Iannizzotto and Francesco La Rosa. Competitive combination of multiple eye detection and tracking techniques. *IEEE Transactions on Industrial Electronics*, 58(8):3151–3159, 2010.

[5] Eui Chul Lee, You Jin Ko, and Kang Ryoung Park. Fake iris detection method using purkinje images based on gaze position. *Optical Engineering*, 47(6):067204, 2008.

[6] Justin Munafo, Meg Diedrick, and Thomas A Stoffregen. The virtual reality head-mounted display oculus rift induces motion sickness and is sexist in its effects. *Experimental brain research*, 235(3):889–901, 2017.

[7] Ko Nishino and Shree K Nayar. Eyes for relighting. *ACM Transactions on Graphics (TOG)*, 23(3):704–711, 2004.

[8] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Calibration-free gaze sensing using saliency maps. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2667–2674. IEEE, 2010.

[9] Kar-Han Tan, David J Kriegman, and Narendra Ahuja. Appearance-based eye gaze estimation. In *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.*, pages 191–195. IEEE, 2002.

[10] Ronda Venkateswarlu et al. Eye gaze estimation from a single image of one eye. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 136–143. IEEE, 2003.

[11] Biaokai Zhu, Jumin Zhao, Dengao Li, Hong Wang, Ruiqin Bai, Yanxia Li, and Hao Wu. Cloud access control authentication system using dynamic accelerometers data. *Concurrency and Computation: Practice and Experience*, 30(20):e4474, 2018.