An Edge-Assisted Video Computing Framework for Industrial IoT

Zeng Zeng[1(✉)], Yuze Jin[2,3], Weiwei Miao[1], Chuanjun Wang[1], Shihao Li[1], Peng Zhou[3], Hongli Zhou[3], and Meiya Dong[2]

[1] Information and Communication Branch, State Grid Jiangsu Electric Power Co., Ltd, Nanjing, China
zengking913@126.com
[2] Tsinghua Wuxi Research Institute of Applied Technologies, Beijing, China
[3] Digital China Information Service Company Ltd, Beijing, China

Abstract. With the rapid development of industrial demands, the Internet of Things triggers enormous interests by industry and academia. By employing IoT technologies, a large number of problems in the industry can be solved by intelligent sensing, wireless communication, and smart software analysis. However, in applying Industrial IoT to improve real-time and immerse user experiences, we found that compared to traditional application scenarios such as tourism, or daily experiences, industrial IoT applications face challenges in scalability, real-time reaction, and immerse user experiences. In this paper, we propose an edge-assisted framework that fits in industrial IoT to solve this fatal problem. We design a multi-pass algorithm that can successfully provide a real sense of immersion without changing the single frame image visual effect in terms of increasing rendering frame rate. From experimental evaluation, it shows that this edge-assisted rendering framework can apply to multiple scenarios in Industrial IoT systems.

Keywords: Industrial IoT · Edge computing · Rendering

1 Introduction

With the rapid development of ubiquitous sensing, wireless communication and data processing technologies, Internet of Things (IoT) has been widely used in various fields, such as environment monitoring, inventory control, and intelligent transportation [14, 15]. Among all these applications, Industrial IoT is projected to be of high potential in the future. For example, in 2015, China pioneered the concept of "Smart Manufacturing 2025", which aims to equip traditional manufacturing with the Internet of Things to drive a new industrial revolution. Furthermore, many international research groups and industrial application alliances have also been established. Within these organizations, global leaders in the manufacturing, telecommunications, networking, semiconductor, and computer industries collaborate to boost Industrial IoT innovations. According to Global Market Research Report [17], the market size of Industrial IoT is expected to reach 753.1 billion USD by the year 2023.

J. Liu et al. (Eds.): MobiCASE 2020, LNICST 341, pp. 53–67, 2020.
https://doi.org/10.1007/978-3-030-64214-3_4

In particular, Industrial IoT applications can monitor the real-time information of physical assets, processes, and systems by sensors. Thereby it can comprehensively monitor the life cycle of industrial production. Meanwhile, Industrial IoT systems employ modern data visualization technologies such as virtual reality (VR) and augmented reality (AR) to construct a "digital twin" [16] of the physical world. In these systems, real-time sensory data are integrated and visually represented on industrial assets. And workers can operate mechanical equipment, vehicles, and pipelines in a remote manner [3, 5].

However, in order to provide users with a real sense of immersion to perceive and manipulate objects in the virtual world, the design and implementation of Industrial IoT systems still face challenges. On the one hand, among the five sensory systems of humans, vision transmits most of the information to the human brain, capturing most of the user's attention, and the contribution of vision to brain stimulation accounts for about 70% [1]. On the other hand, when users perform remote operations in Industrial IoT scenarios, immersive operation methods will cause serious problems such as motion sickness. These problems are not severe in daily experience or under game scenarios. However, the operation of industrial scenarios will bring severe consequences. For example, 1-s operation delay caused by problems such as motion sickness is likely to cause equipment damage [4, 12].

Motion sickness is caused when the signal received by the human visual system and the stimulation felt by the vestibular system is inconsistent, causing the observer's brain to be confused. In the industrial virtual reality scene, the helmet covers the entire field of vision of the user. In order to deceive the user to the maximum, the field of vision felt by the visual system is no longer the size of a screen but approximates the real world. The entire visual field of the human eye, the picture will change in real-time as the user's head and body move. Therefore, two factors can lead to motion sickness. The first factor is that the Industrial IoT system cannot provide real motion output that matches the virtual reality scene. Moreover, the second factor is that the high-resolution rendering in the system brings the system delay and insufficient frame rate, which leads to a mismatch between visual and sensory motion.

In this paper, we propose an edge assisted rendering framework for Industrial IoT. First, we proposed an Edge-based framework, so that the rendering of video information can be completed on edge. Second, we proposed multi-pass and multi-resolution: MultiPass algorithm. Finally, we conducted experiments based on the Unity 3D engine and evaluated it in many scenarios. This framework is proved to perform well in such immerse industrial IoT applications.

2 Related Work

How to improve user experiences of the industrial information system has drawn research interest since the 1980s. Foveal rendering technology was once proposed in military applications [9, 11]. Kocian and Longridge et al. rendered the area of interest of the eye at high resolution and inserted it into the low-resolution area, but the paper did not accurately describe how to operate the rendering pipeline to reduce the resolution of the surrounding area. Geisler and Perry [13] proposed to use multi-resolution graphics

pyramids to generate pictures of different resolutions in real-time so that the shape and size of the foveal area changed. The system can produce high-quality images with low aliasing at high frame rates. Clarke [7] then proposed to simplify the resolution of geometric objects far away from the observer and determine the LOD (Levels of Detail, LOD) hierarchy of the object according to the projection area covered by the geometric object. When the area covered by the object is small, a lower resolution model of the object is used. Otherwise, a higher resolution model is used to draw complex scenes quickly. However, the LOD algorithm produces isotropic degraded geometric objects from different perspectives, which is not always satisfactory, especially when viewing large objects at close range. The above method is still in the initial exploration of foveal rendering for video, which has many shortcomings and is not suitable for the industrial IoT systems especially the cloud based systems [20].

Guenter et al. [8] multi-channel rendered three layers of different resolutions around the center of the line of sight. The angle size of the three layers gradually increased, but the sampling rate gradually decreased. The aliasing phenomenon is caused by reducing the resolution of the picture. The author combined multiple anti-aliasing techniques to eliminate them. The system can accelerate the rendering of desktop HD display 3D scenes 5 to 6 times faster, but the advantage of this method is that it is suitable for existing hardware and easy to implement under the software platform. However, this method does not take into account the visual distribution of human eyes in a virtual reality scene, and serious-time aliasing problems have occurred after implementation. Therefore, this solution does not fully use the GazeRender system. Vaidyanathan and Salvi [2] proposed a single-channel. They bounded memory order-independent transparency algorithm, which is mainly used for fast and high-quality rendering of scenes containing geometric primitives with transparency, but the paper does not attempt to eliminate the effects of foveal rendering. Perceptual aliasing problem, this solution is not consistent with the problem to be solved by the system in this paper. Vaidyanathan et al. [10] proposed a method of decoupling the pixel shading rate and the visibility sampling rate for rendering acceleration in 14 years. This method uses the sampling rate of Full HD in the rasterization visibility detection stage and down sampling only in the coloring stage, thereby improving the overall rendering speed while retaining the details of the object. This method significantly reduces the coloring consumption without the introduction of perceptual aliasing and blurring. However, this method currently has no hardware support and is still in the simulation stage, so it is not suitable for the implementation of this system. Patney et al. [6] explored the implementation of this method in virtual reality scenes in 16 years. In summary, none of the existing foveal rendering technologies are fully applicable to the multi-resolution rendering in today's industrial IoT systems [19].

3 System Design

The system architecture and workflow between modules are shown in Fig. 1.

The foveal rendering module is mainly composed of three parts. The first is the discretization of the sampling rate distribution. The target sampling rate of each point on the screen is determined based on the coordinates provided by the eye-tracking and

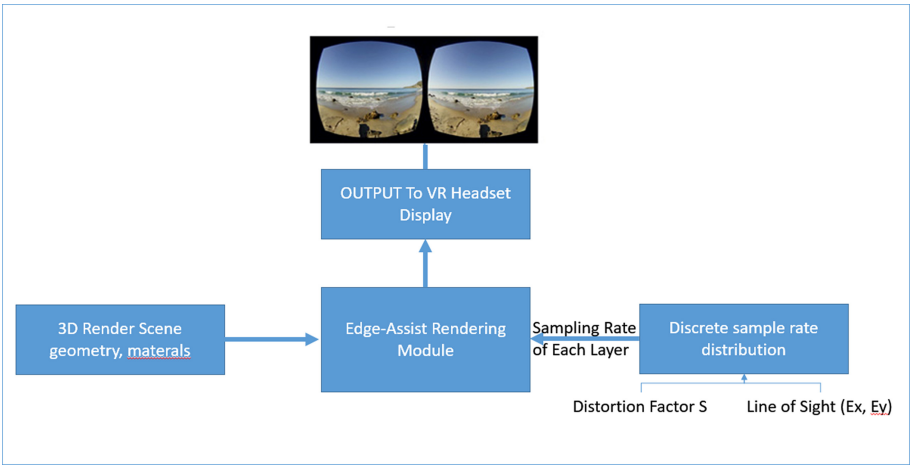


Fig. 1. System architecture

the distribution of human eye acuity. Partitioning and discretization to provide a basis for sampling rates for multi-resolution rendering. After obtaining the sampling rate distribution, the second sub-module of the foveal rendering multi-resolution rendering module performs multi-resolution rendering on the screen according to the sampling rate, and MultiPass implements multi-resolution rendering. The third sub-module is to call the rendering engine to execute a multi-resolution rendering solution and render and output the image to the screen of the virtual reality helmet according to the information of the 3D scene.

This paper mainly introduces the implementation and improvement scheme of the foveal rendering algorithm. The foveal rendering refers to the characteristic that the visual acuity of the human eye is gradually reduced in the visual edge area, and the resolution of the rendered image is gradually decreased from the focus area of the line of sight to the periphery to improve the overall rendering speed. The critical point is to achieve multi-resolution rendering on one screen. This chapter first introduces the implementation of MultiPass (Multi-Pass multi-resolution Rendering). This method can run directly on current GPU hardware without modifying the rendering pipeline. The method is simple and easy to implement. However, after implementing the effect analysis, this paper finds that MultiPass has severe spatial and temporal aliasing in the virtual reality scene due to the down-sampling of the edge layer. Then, for the aliasing problem, we have combined the MultiSampling Anti-Aliasing (MSAA) technology, Temporal Anti-Aliasing technology, and image contrast enhancement directly supported by the hardware into the MultiPass rendering algorithm. The aliasing is reduced to a level almost imperceptible to the human eye. The description of the foveal rendering technology based on the MultiPass algorithm includes the principles and implementation methods of the multi-pass multi-resolution rendering algorithm MultiPass.

3.1 Discretization of Sampling Rate Distribution

According to the idea of sample rate discretization, this paper adopts a three-layer sample rate discretization and nesting and superposition scheme to achieve the concave center rendering, as shown in Fig. 2. Since each layer is rendered with the same resolution and the sensitivity of the human eye is gradually reduced from the center to the periphery, the resolution of each layer must meet the highest resolution requirements of the human eye. People feel a significant change in resolution. The oblique line in Fig. 2 represents the distribution of the minimum resolving angle MAR of the human eye along with the eccentric angle. In this paper, two eccentric angles e_1 and e_2 are selected to divide the screen into three parts: the center layer, the transition layer, and the edge area. The three visual layers use sampling rates are R_0, R_1, R_2, and the corresponding minimum resolution angles are $\omega_0, \omega_1, \omega_2$.

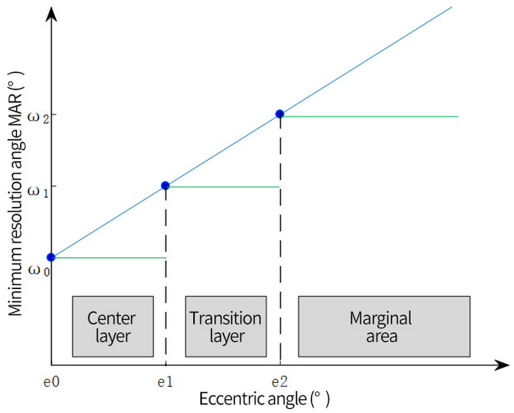


Fig. 2. Parameter selection when the number of eccentric layers is 3.

The selection of volume discretization angle and sampling rate need to consider both human visual effects and rendering performance acceleration. First of all, the visual effect of the human eye needs to be guaranteed. In the sight tracking model in Chap. 3, the concept of sight tracking error δ is proposed. After the experimental measurement, when δ is 1.5, the sight tracking model will ensure that the recognition point is 90% within error δ. In order to compensate for the eye-tracking error, when determining the eccentric angles e1 and e2 of the two layers at the center, it is necessary to increase the angle by δ from the theoretically derived angle. It is to ensure that the area within the e1 and e2 angles around the center of sight of the human eye can always be sampled according to ω0 and ω1 Rate rendering. Figure 3 shows a discretization model that compensates for the aforementioned tracking error. Assume that the eccentric angle of e1 is selected as the dividing line, Ex and Ey represent the visual center recognized by the human eye. After the minimum sampling rate ω_0 of the region is determined, due to the existence of human eye recognition error δ, ω0 will be used as the angular radius as Rendering sampling rate in the range of δ + e1.

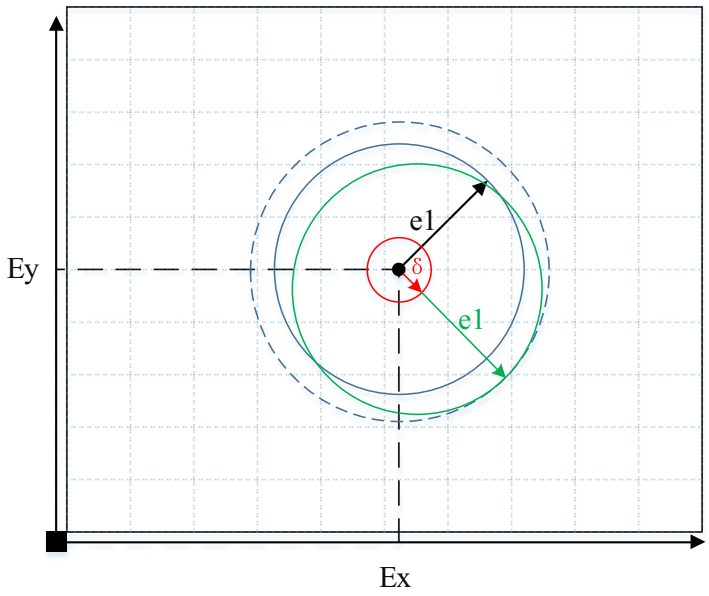


Fig. 3. Sampling rate discretization model to compensate for tracking error.

The relationship between the sampling rate and the minimum resolution angle is described below. Thus, the sampling rate R of each region is calculated according to the minimum resolution angle MAR at which the inner edge is located. In general, the sampling rate can be determined based on the distance D between the human eye and the screen, combined with the minimum resolution angle ω0. However, in a virtual reality helmet, the change in the distance between the screen and the human eye due to eyepiece refraction must also be considered.

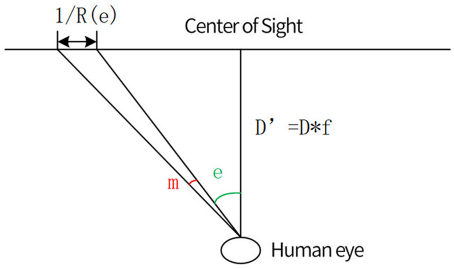


Fig. 4. Relationship between sampling rate and minimum definition view angle.

The sampling rate calculation model is shown in Fig. 4. It is assumed that the refraction of the eyepiece magnifies the screen distance by f times, e is the eccentric angle, ω is the minimum resolution angle MAR at e, and R (e) represents the sampling rate at e. D is the actual distance of the human eye from the screen, and D is the distance between

the human eye and the screen after being refracted by the eyepiece. The formula for calculating the sampling rate is derived,

$$\frac{1}{R(e)} = \frac{D'}{(\cos(e))^2} \sin\left(\frac{\omega}{2}\right) * 2 \tag{1}$$

Therefore, the minimum viewing angle MAR is small, so a small angle trigonometric function approximation formula can be used, and the formula 1 can be simplified to,

$$\frac{1}{R(e)} = \frac{D'\omega}{(\cos(e))^2} \tag{2}$$

In summary, if e1 and e2 are selected as the dividing line of the eccentric angle, the sampling rate calculation formula of the three regions is,

$$R_i = \frac{(\cos(e_i))^2}{D'\omega_i} \quad i = 0, 1, 2 \tag{3}$$

According to previous studies [18], in order to ensure the balance between rendering effect and rendering speed, the eccentric angles $e_1 = 5^{\circ}, e_2 = 30^{\circ}$ were selected. Because within a 5° viewing angle, the human eye recognizes objects with near full HD visual acuity. The "aliasing phenomenon" occurs in the area from 5° to 30°. That is, the human eye can see some high-frequency information, but cannot clearly see the texture and direction of high-frequency details. From 30° outward, it is the edge area of the human eye. In this visual area, only the low-frequency information of the picture can be seen. According to the research on the distribution of visual acuity of human eyes, the minimum resolution angle of human eyes within 5° is 1/48°. After the screen, the virtual reality helmet is refracted by the eyepiece. The distance D' from the screen to the human eye is about 25 cm. According to the formula 4–3, the minimum sampling rate of the visual center area is 275PPI. The above is a combination of the classic sampling rate discrete method and the virtual reality environment, which is called the MultiPass algorithm in this paper.

3.2 Implementation of Multi-layer Channel Overlay

This paper mainly introduces the implementation of MultiPass with multi-channel resolution rendering. MultiPass divides the screen into several concentric square layers according to the eccentric angle radius e from the center of the human eye. Each square area is rendered at a gradually lower sampling rate from the inside to the outside, and then multi-layered and nested to form a foveal rendering image.

This paper implements the MultiPass algorithm with three layers of overlay. Around the focus of the line of sight, MultiPass rendered three eccentric layers (the yellow border represents the rendering range of the center layer, the blue border represents the middle layer, and the green border represents the edge layer). The two eccentric layers in the middle cover only a part of the display area, and the edge layer covers the entire display area. The center layer is rendered at the highest sampling rate (consistent with the display resolution). The sampling rate of the two viewing areas outward from the center layer is obtained from the sampling rate calculation formula introduced above.

The three channels of MultiPass render the covered display area separately and then superimpose them into a high-definition image with blurred visual edges. But it is worth noting that the overlapping part of each layer is superimposed instead of hollow nested, which causes the overlapping area to be repeatedly rendered more than once, which brings a certain amount of performance waste. However, existing GPUs and rendering pipelines currently only support rendering for uniform sampling rates. Although the idea of hollow nesting has been proposed by researchers [6, 30], there is only a stage of software simulation and theoretical derivation.

MultiPass achieves a multi-sampling rate by simulating three virtual "screens." The resolution of each virtual screen is set to the minimum sampling rate of the region. In the Unity 3D game rendering engine, multiple cameras (Cameras) are nested with each other to implement the above MultiPass algorithm.

In the Unity3D game rendering engine, the Camera control provides users with a field of view to watch the 3D world. It has a renderer packaged inside it that can render 3D models within the field of view according to angle and position. For example, a virtual reality game usually developed based on Unity3D can display the virtual world from different angles and distances by controlling the position and posture of the Camera, and provide users with a picture.

In order to implement the three-layer MultiPass algorithm, three cameras (c1, c2, c3 from the inside to the outside) were used as a set of renderers in the experiment. Then control camera parameters to achieve different resolution levels of rendering (c1, c2, c3 in order from high-definition to low-definition rendering), the parameter list is shown in the following Table, (Table 1)

Table 1. MultiPass camera perimeter in unity 3D.

Camera no.	Rendering view (Eccentric angle diameter)	Rendering sample rate (ratio to full resolution)
C1	120	0.3
C2	60	0.6
C3	10	1

Then link these Cameras together and control their direction through different controls. The outermost c3 represents the field of view of the helmet and is controlled by the helmet's gyroscope. The center of the two cameras is the same, and the coordinate system is the relative coordinate system of c1. Controlled by the line-of-sight tracking module, that is, the center of the camera coincides with the center of the line of sight. When the human eye rotates, the two cameras in the middle also rotate accordingly so that the visual center area is rendered in c1 HD, the transition area is rendered in c2, and the edge area The MultiPass algorithm rendered by c3 low-resolution. In Untity 3D, the algorithm is described as follows (Table 2):

Table 2. Algorithm description

```
Input: The position of focus updated by gaze tracking
1:  FocusPointPosUpdate(Position){
2:    SetFocusPoint(new Vector2(Position.x, Position.y));
3:}
4:  //Setting Camara Parameter according to the position of the focus
    5:  SetFocusPoint(newFocusPoint, Camera cam){
    6:  Left = newFocusPoint.x-cam.width/2
    7:  Right = newFocusPoint.y+cam.width /2
    8:  Bottom = newFocusPoint.y- cam.height/2
    9:  Up = newFocusPoint.y+ cam.height/2
    10:  // Set the position of the four vertices of Camara, the distance between the
closest rendering and the furthest rendering
    11:  cam.projectionMatrix = PerspectiveOffCenter(left, right, bottom, top,
    12:  cam.nearClipPlane, cam.farClipPlane)
13:  }
```

3.3 MultiPass Algorithm Analysis

After implementing the MultiPass algorithm in a layered and nested manner, this section analyzes its performance and verifies the effect, and lays the foundation for the improvement of MultiPass in the following. According to the principle and implementation of the above MultiPass algorithm, it is known that the three layers are superimposed on each other. The central layer area is rendered three times by three resolutions, and the middle layer is rendered two times by two resolutions. Wasted rendering performance. Therefore, we need to choose the proper demarcation angle to minimize this kind of performance waste. The following section analyzes the selection of MultiPass eccentric layer parameters.

The visual acuity model MAR slope is defined as m, the horizontal resolution of the virtual reality helmet D*, the field of view width W*, the aspect ratio is α*, and the viewer-to-screen distance V*. The eccentric layer is represented as L_i, when $i = 1$ represents the center layer, when $i = n$ represents the outermost layer, and the corresponding eccentric angle is e_i.

The eccentric layer sampling factor S_i ,when $S_i \geq 1$, it means that the pixel size of this layer is a multiple of the pixel size of the display unit. When $S_i = 1$, it means that the sampling rate of the center layer is consistent with the sampling rate of the display. The outermost layer is a self-sampling of the entire screen image and maintains the aspect ratio of the display. e* indicates the maximum field of view of a single eye, ω* indicates the minimum resolution angle supported by the virtual reality helmet, and is calculated by the display configuration parameters as the following Equation. ω_0 indicates the minimum resolution angle that can be distinguished by the human eye, usually $\omega_0 \leq \omega^*$.

$$\omega^* = tan^{-1}\left(\frac{2W^*}{V^*D^*}\right) \tag{4}$$

The oblique line segment indicates a MAR line with a slope of m. The horizontal line segment represents the selected eccentric layer parameters. The eccentric angle gradually increases from left to right. The eccentric layers are the center layer, the middle layer, and the edge layer. e* represents the maximum field of view of a single eye, ω* represents the minimum resolution angle supported by the virtual reality headset, and ω_0 represents the minimum resolution angle that can be resolved by the human eye. The eccentric layer parameter uses a piecewise constant to approximate the MAR linear model (shown in purple horizontal line segments in above Fig. 2. It is a conservative parameter selection method because the parameters are always below the target MAR line. The increasing eccentric angle e_i determines the corresponding sampling factor S_i, that is,

$$\begin{cases} s_{i+1} = \frac{\omega_i}{\omega^*} = \frac{me_i+\omega_0}{\omega^*}, 1 \le i \le n-1 \\ s_1 = 1 \end{cases} \tag{5}$$

The horizontal diameter D_i of each layer in pixels is calculated from the eccentric angle e_i and the sampling factor S_i, that is,

$$D_i = \begin{cases} 2\frac{D^*}{s_i} tan(e_i) \frac{V^*}{W^*}, 1 \le i < n \\ \frac{D^*}{s_i}, i = n \end{cases} \tag{6}$$

The number of pixels rendered per layer is given by

$$P_i = \begin{cases} (D_i)^2, 1 \le i < n \\ \alpha^*(D_i)^2, i = n \end{cases} \tag{7}$$

Finally, minimizing the total pixels gives

$$min(P) = min\left(\sum_{i=1}^{n} w_i P_i\right) \tag{8}$$

Since the range of e* is constant and known, we discretize the range of e* and search for the P value at each eccentricity angle $0 < e_1 < e_2 < \cdots < e_{n-1} < e^*$ to find the minimized result.

From the above analysis, it can be known that the MultiPass algorithm can determine the size and sampling rate of each eccentric layer based on simple mathematical formulas and optimization equations, so as to ensure that rendering effects are minimized due to the overlapping of rendering channels waste.

4 Experimental Evaluation and Verification

We conduct an experimental evaluation of a context-aware virtual reality rendering acceleration system.

4.1 Experimental Design

The multi-pass and multi-resolution rendering algorithm MultiPass is implemented based on existing hardware and rendering pipelines, so it can directly compare the rendering frame rate with full-pixel rendering that does not use foveal rendering, thereby representing the improvement in algorithm speed of foveal rendering. Through the comparison of shaded pixels and rendering effects, the improvement over MultiPass is explained. In order to test the above algorithm, the virtual reality scenes selected in the experiment include three as shown in Fig. 5: (a) a game scene with more complex textures and models, (b) a classroom scene with a simple model, and (c) a scene containing two The museum scene of the layer definition model (LowPoly and HighPoly) is used to test the effect of the MultiPass algorithm.

（a） （Gaming Scene） （b） （ClassRoom Scene） (c) （Museum Scene）

Fig. 5. Three virtual reality scenarios used in the experiment

4.2 Foveal Rendering Module Evaluation

When analyzing the performance of foveal rendering, we first compare the rendering sampling rate and sampling points of foveal and full-pixel rendering and then compare the frame rates of MultiPass and full-pixel rendering. Figure 6 shows the comparison of the sampling rate between foveal rendering and full pixel rendering. In the foveal rendering algorithm, we developed the foveal rendering based on the distribution of human visual acuity (black and green oblique lines). Sampling rate distribution scheme (shown in red). The central sampling rate of this experiment is between visible acuity and separable acuity. Experiments have proved that this compromised sampling rate scheme and anti-aliasing can make users unable to perceive edge down-sampling. As can be seen from the figure, compared with full pixel rendering (blue line), the sampling rate of the foveal rendering in the visual transition area and edge area is much lower than that of the full pixel rendering. In general, the sampling rate shown in the figure is used for the foveal rendering. The sampling point in the visual transition area (5°–30°) is 20–0% less than the original. The visual edge area can be rendered about 80 less than the original image. −90% of sampling points because the number of sampling points is proportional to the square of the sampling rate.

Due to the reduction in rendering sampling points, the rendering rate will be greatly improved. Even if there is a performance waste phenomenon caused by layer overlay in the MulitPass algorithm, the visual edge area occupies more than 60% of the area of the map, so the rendering speed improvement is still considerable. Figure 6 shows the

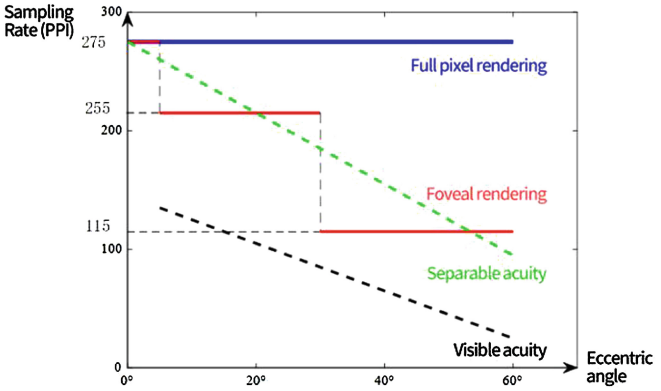


Fig. 6. Comparison of sampling rates between MultiPass and full rendering.

comparison between the full resolution and the foveal rendering algorithm MultiPass in the rendering frame rate in three scenarios. It can be seen that foveal rendering has brought significant performance improvements in various different scenarios. The rendering frame rate has been increased by 2–3 times, all exceeding the preset target value of 120 frames/second.

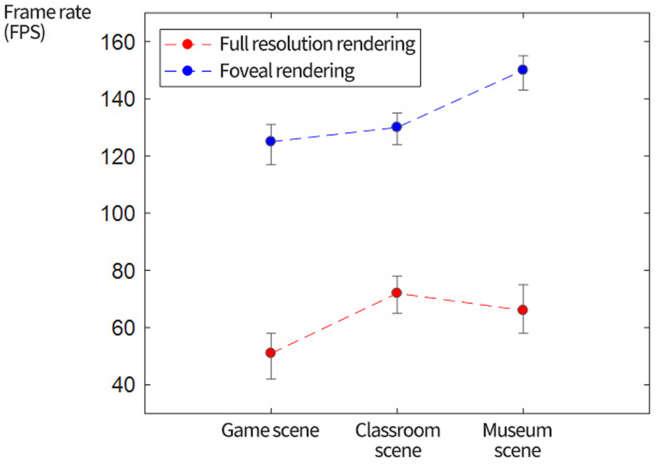


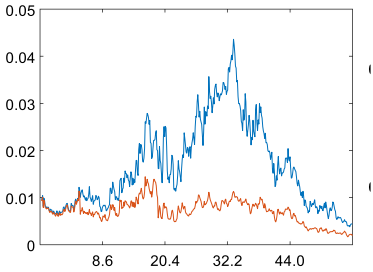
Fig. 7. Comparison of the frame rate of the three virtual reality scenes with MultiPass foveal rendering and full rendering.

In order to help further analyze the performance of the proposed post-processing foveal rendering method in the system, this paper uses wavelet transform to analyze the performance of images at different frequencies and different spatial positions. In this experiment, a two-dimensional Haar wavelet transform is used to analyze the image at different decomposition levels (Image Decomposition Level). The wavelet coefficients

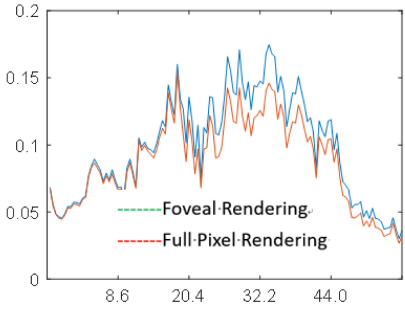
of different regions of the image are combined with the eccentricity to analyze the relationship between frequency changes and the angle of sight.

The mother wavelet ψ (t) of the Haar wavelet can be described as

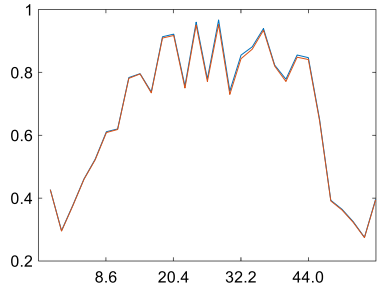
$$\psi(t) = \begin{cases} 1\ 0 \le t < \frac{1}{2} \\ -1\ \frac{1}{2} \le t < 1 \\ 0\ \text{Other} \end{cases} \tag{9}$$



(a) Eccentricity under HF Wavelet Transform



(b) Eccentricity under IF Wavelet Transform



(c) Eccentricity under LF Wavelet Transform

Fig. 8. Wavelet transform analysis of foveal rendering and full pixel rendering

The zoom function ϕ (t) can be described as:

$$\phi(t) = \begin{cases} 1\ 0 \le t < 1 \\ 0\ Other \end{cases} \tag{10}$$

The Haar wavelet coefficients are calculated in the changing eccentricity. As the eccentricity changes, the image information is gradually discarded, and the wavelet coefficient (wavelet energy) also decreases. The larger wavelet coefficients highlight the important feature information that is retained. Figure 7 shows the comparison between foveal rendering and full pixel rendering on high frequency, intermediate frequency and low frequency wavelet transforms. We choose 1 × 1, 3 × 3, and 5 × 5 granularities as high-frequency, intermediate-frequency, and low-frequency wavelets, respectively. The abscissa of the image represents the angle away from the visual center, and the ordinate represents the wavelet energy. The higher the wavelet energy, the more image textures representing this frequency, and the richer the information. In the high-frequency wavelet transform in Fig. 8, as the eccentricity angle increases, the wavelet energy is significantly reduced compared to full-pixel rendering, indicating that high-frequency details are lost in the visual edge region. The foveal rendering of the mid-frequency and low-frequency wavelet energy has less loss than the full-pixel rendering, which indicates that the foveal rendering retains the mid- and low-frequency information of the image at the visual edge.

5 Conclusion

In this paper, we studied a significant framework to improve user experiences in industrial IoT systems. We design a new foveal rendering technology to accelerate the rendering rate to avoid motion sickness in industry.

Given the characteristics of industrial IoT equipment: high refresh rate requirements, large field of vision, and eyepiece distortion, we have conducted a full investigation, demonstration, and improvement of existing related technologies to adapt to the use scenario of virtual reality equipment [18]. We proposed an edge-assist rendering framework to solve the problem. We implemented a multi-pass and multi-resolution rendering algorithm MultiPass, which was based on the Unity3D development engine. In this paper we evaluated its performance and rendering effect. From experimental evaluation, our framework can increase the rendering rate by 2–3 times without affecting the user's visual experience. These results indicate that this research is an effective method to solve the existing problem in terms of user experiences improvement in industrial IoT systems.

Acknowledgement. The paper is supported by the Science and Technology Project of State Grid Corporation of China: "Research on Key Technologies of Edge Intelligent Computing for Smart IoT System" (Grant No. 5210ED209Q3U), NSF China Key Project (Grant No. 61632013), National Key Research and Development Project (Grant No. 2018YFB2200900).

References

1. Lang, D.J.: For virtual reality creators, motion sickness a real issue (2016). http://phys.org/news/2016-03-virtual-reality-creators-motion-sickness.html
2. Salvi, M., Vaidyanathan, K.: Multi-layer alpha blending. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pp. 151–158 (2014)
3. Meiya, D., Jumin, Z., Biaokai, Z., Zhaobin, L.: CLOAK: Visible Touching and Invisible Protecting in Cloud Based IOT System, CBD2018 (2018)
4. Bohil, C.J., Alicea, B., Biocca, F.A.: Virtual reality in neuroscience research and therapy. Nat. Rev. Neurosci. **12**(12), 752–762 (2011)
5. Meiya, D., Jumin, Z., Biaokai, Z., Zhaobin, L.: "CHAMELEON"- hides privacy in cloud IoT system by LSB and CSE. Concurr. Comput.: Pract. Exp. **31**(245)
6. Patney, A., Salvi, M., Kim, J., et al.: Towards foveated rendering for gaze-tracked virtual reality. ACM Trans. Graph. (TOG) **35**(6), 179 (2016)
7. Clark, J.H.: Hierarchical geometric models for visible surface algorithms. Commun. ACM **19**(10), 547–554 (1976)
8. Guenter, B., Finch, M., Drucker, S., et al.: Foveated 3D graphics. ACM Trans. Graph. (TOG) **31**(6), 164 (2012)
9. Kocian D. Visual world subsystem. Super Cockpit Industry Days: Super Cockpit/Virtual Crew Systems. pp. 97–103 (1987)
10. Wang, J.G, Sung, E., Venkateswarlu, R.: Eye gaze estimation from a single image of one eye. In: IEEE International Conference on Computer Vision, p. 136. IEEE Computer Society (2003)
11. Longridge, T.: Design of an eye slaved area of interest system for the simulator complexity testbed. Area of Interest/Field-of-View Research Using ASPT (1989)
12. Tan, K.H., Kriegman, D.J., Ahuja, N.: Appearance-based eye gaze estimation. Applications of Computer Vision, pp. 191–195 (2002)
13. Geisler, W.S., Perry, J.S.: Real-time foveated multi-resolution system for low-bandwidth video communication. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 3299, pp. 294–305 (1998)
14. Xufei, M., Xin, M., Yuan, H., Xiang-Yang, L., Yunhao, L.: CitySee: Urban CO2 monitoring with sensors. In: 2012 Proceedings IEEE INFOCOM (2012)
15. Qingping, C., Hairong, Y., Chuan, Z., Zhibo, P., Li, D.X.: A reconfigurable smart sensor interface for industrial WSN in IoT environment. IEEE Trans. Indust. Inform. **10**(2), 1417–1425 (2014)
16. He, Y., Guo, J., Zheng, X.: From surveillance to digital twin: challenges and recent advances of signal processing for industrial IoT. IEEE Signal Process. Mag. **35**(5), 120–129 (2018)
17. More, A: Market Share (2019). https://www.marketwatch.com/press-release/industrial-IoT-market-2019—globally-market-size-analysis-share-research-business-growth-and-forecast-to-2023-market-reports-world-2019-05-03
18. Mao, X., Miao, X., He, Y., Li, X.Y., Liu, Y.: CitySee: Urban CO2 monitoring with sensors. In: 2012 Proceedings IEEE INFOCOM, pp. 1611–1619. IEEE (2012)
19. Liu, K., Ma, Q., Gong, W., Miao, X., Liu, Y.: Self-diagnosis for detecting system failures in large-scale wireless sensor networks. IEEE Trans. Wirel. Commun. **13**(10), 5535–5545 (2014)
20. Chen, Z., Zhao, Y., Miao, X., Chen, Y., Wang, Q.: Rapid provisioning of cloud infrastructure leveraging peer-to-peer networks. In: 2009 29th IEEE International Conference on Distributed Computing Systems Workshops, pp. 324–329. IEEE (2009)