Hindawi
Mobile Information Systems
Volume 2022, Article ID 2148742, 6 pages
https://doi.org/10.1155/2022/2148742

Research Article

Virtual Reality Software and Data Processing Algorithms Packaged Online for Videos

Li Zeng and Keke Guo

Hunan Mass Media Vocational and Technical College, Changsha, Hunan 410100, China

Correspondence should be addressed to Li Zeng; 33115225@njau.edu.cn

Received 6 May 2022; Revised 10 June 2022; Accepted 18 June 2022; Published 4 July 2022

Academic Editor: Muhammad Muzammal

Aiming at the problem of virtual reality and data processing algorithm of online video packaging, one transmission scheme uses TILES in HEVC to block the video and then applies MP4Box to pack the video and generate a DASH video stream. A method is proposed to process the same panoramic video with different quality. By designing a new index to measure the complexity of the coding tree unit, this method predicts the depth of the coding tree unit by using the complexity index and spatial correlation of the video, skipping unnecessary traversal range, and realizing fast division of coding units. Experimental results show that compared with the latest HM16.20 reference model, the proposed algorithm can reduce the coding time by 37.25%, the BD-rate only increases by 0.74%, and the video image quality is almost not lost.

1. Introduction

At present, the main form of a virtual reality video is a 360-degree video, which can meet the experience of 360 degrees in the horizontal direction and 180 degrees in the vertical direction from any perspective switch. In order to encode a spherical 360-degree video content, it is necessary to map the spherical mode to two-dimensional plane mode first [1]. The Joint Video Experts Team (JVET) has proposed a variety of projection formats, of which equirectangular projection (ERP) is the most widely used. In the projection format, the polar regions of the 360-degree video in virtual reality will cause excessive stretching, adding a large amount of redundant data, leading to a significant increase in coding time. The research on reducing intraframe coding time is divided into two aspects: intraframe mode decision and coding unit (CU). In CU partition, complexity evaluation or depth prediction is used to terminate the partition in advance to reduce the unnecessary rATE-distortion optimization process [2]. There are obvious differences between panoramic video creation and ordinary 2D video creation in expression and narrative mode. However, as a large number of creators continue to explore and try, a lot of high-quality panoramic video content has been produced. They enter their creations into competitions or share them with other VR creators on a small scale, but even the winning entries are almost unknown to the general public. As content producers, they want their work to be experienced by the public, and they are also eager to put into the market to get feedback as a guide for subsequent creation [3]. Panoramic video, due to its own characteristics, covers 360 degree $*$ 180 degree view information, supports users to change the view direction for experience, and includes video, audio, subtitles, interactive, and other types of data. The transmission process requires a great amount of bandwidth. The bandwidth requirement for a high-quality VR experience is about 5 Gbps. The simultaneous head motion and field-of-view latency (MTP) should be less than 20 ms. The requirements of network bandwidth and delay seriously limit the panoramic video-on-demand service and pose a challenge to the transmission of the panoramic video. Traditional streaming media transmission scheme is difficult to meet the real-time transmission requirements of panoramic video-on-demand service [4]. As the variety of virtual reality 60degree video increases, the scene becomes more and more complex. It is difficult to edit video frames according to latitude to adapt to

various kinds of video sequences. This paper analyzes the characteristics of 360degree virtual reality video in ERP projection format, designs a new index to measure the complexity of CTU, categorizes video frames at the CTU level, predicts the depth of CTU by using video spatial correlation, and realizes CU rapid division by skipping unnecessary depth traversal [5].

Kumar et al. proposed a 360-degree virtual reality video optimization algorithm based on CU depth range prediction and fast mode decision. By analyzing the video characteristics under ERP format, the video frame was divided into two poles and equatorial regions, and different reference blocks and judgment conditions were set according to the distortion degree, so as to ensure the quality and reduce the coding time [6]. Sun et al. proposed an algorithm based on CU complexity and CART decision tree to judge whether the current CU is divided and skip unnecessary traversal range. The research objects of the above optimization algorithms are mainly traditional video sequences. However, the projected 360-degree video of virtual reality shows characteristics significantly different from traditional video sequences, which need to be optimized according to these characteristics [7]. By analyzing the texture characteristics of the polar and equatorial regions, we can judge whether to terminate the CU partition in advance from the texture complexity and texture direction, so as to reduce the number of unnecessary traversals [8]. The basic idea of the data processing algorithm is to search the region with the most dense sample in the feature space through repeated iteration, which is called the modal of the sample. The principle of the data processing algorithm is simple, and the iterative efficiency is high, but the size of the search area in the iterative process has a great impact on the accuracy and efficiency of the algorithm.

In this paper, functional and nonfunctional tests of the platform were designed and implemented, and the test results met the expected goals. Besides, users were invited and organized to conduct subjective experimental tests by watching videos. The experimental results show that panoramic content, as a new media, is well-liked by users, and the application of block transmission and DASH strategy to achieve a better playback experience verifies the feasibility and effectiveness of the panoramic content playback platform [9].

2. Algorithm Idea

Virtual reality 360-degree video images projected by ERP format often have a large number of flat areas. This is because, on the one hand, virtual reality 360-degree videos often have a large number of background areas such as sky, ground, and water, which are generally relatively flat. On the other hand, in the ERP projection process of the spherical video, the region near the upper and lower poles requires a large amount of data interpolation, which leads to a large amount of redundant data [10]. Therefore, in the process of coding 360-degree virtual reality video, many algorithms divide the video frame into the equatorial region and polar region according to latitude and then optimize the CU division. It should be pointed out that, with the increase of 360-degree virtual reality video types and increasingly complex scenes, it is difficult to use fixed latitude to divide video frames to adapt to various types of video sequences. For example, there are still a large number of complex textures in high latitude regions of some videos, while a large number of flat regions exist in low latitude regions [11]. In this case, there are obvious limitations to partitioning video frames only based on latitude. Therefore, we need to find a more effective method of regional division. If we can find a low complexity measurement index of region flatness and use this index to judge whether a region is flat, so as to optimize the CU partition, the adaptability of the algorithm will be effectively improved. Therefore, this paper designed a new index to measure the regional complexity, classified the video frames at the CTU level, and proposed a fast CU partition algorithm based on CTU complexity and spatial correlation to speed up the coding process. The algorithm consists of three parts, including CTU complexity description based on gradient, CTU classification, and CTU depth prediction [12].

2.1. CTU Complexity Description. Commonly used complexity description methods include gray level co-occurrence matrix and Sobel operator. There are too many parameters of gray co-occurrence matrix, and the calculation complexity is large. Sobel operator can accurately judge texture direction, but it is difficult to describe the complexity of the image. Since there is a strong correlation between pixels of each frame of the video image, the difference between the current pixel and adjacent pixels (upper-left pixel, upper pixel, and left pixel) can be used to roughly express the image texture, that is, the image gradient is [13]

$$\mathrm{Grad}(x, y) = |\mathrm{d}x(i, j)| + |\mathrm{d}x(i, j)|, \tag{1}$$

$$\mathrm{d}x(i, j) = p(i, j) - p(i, j-1), \tag{2}$$

$$\mathrm{d}x(i, j) = p(i, j) - p(i-1, j). \tag{3}$$

Gradient can reflect the direction and speed of pixel change, reflect the fluctuation range of pixels in the block, and highlight the pixel jump. Thus, the texture complexity of a block can be represented by a CTU gradient. Formula (1) is designed to calculate the complexity of CTU for the special points of 360-degree virtual reality video in ERP projection format [14].

$$T = \sum_{j=2}^{H}\sum_{i=2}^{W}\left(\eta\left|p_{ij} - p_{i,j-1}\right| + (1-\eta)\left|p_{ij} - p_{i-1,j}\right|\right), \tag{4}$$

where W is the width of CTU, H is the height of CTU, $p_{i,j}$ is the current pixel, $p_{i,j-1}$ is the top pixel, and $p_{i-1,j}$ is the left pixel; $\eta = h/\widehat{H}$ is the ordinate of each CTU's upper left corner, the two ends of the image are defined as 0, and the middle of the image is defined as 1000. $\widehat{H}$ is the height of the image. In the ERP projection format, images of different dimensions have different stretching degrees, Therefore, the closer CTU is to the poles, the smaller the η become and the

smaller the reference to the upper pixels. The larger 1- η's value is, the greater the reference to the left pixel [15–18].

2.2. CTU Classification. Statistical methods are used to determine the number of CTU classifications and thresholds. A total of 334,770 experimental data were collected, and data distribution statistics are shown in Figure 1. Polynomial functions were used to fit the data [12]. As can be seen from Figure 1, a large number of CTUs have low texture complexity, and the number of CTUs gradually decreases with the increase of texture complexity. The determination process of classification quantity and classification threshold is as follows: firstly, the x-coordinate of the fitting curve is selected at equal intervals, and the corresponding y-coordinate value is taken as the initial classification threshold value. After that, all the video sequences are tested repeatedly. Under the premise that the performance of the control algorithm is basically unchanged, the classification intervals are merged gradually, and finally, the optimal CTU classification number and threshold are obtained [19].

2.3. CTU Depth Prediction. It is easy to calculate that the CTU's predicted depth is 0,1 when the average depth of CTU is between 0 and 1. When the CTU's average depth is 1.25 to 1.75, the CTU's predicted depth is 1 and 2. When the CTU's mean depth is 1.3125 to 2.5 (which does not include depths 1.5 and 1.75), CTU's predicted depths are 1, 2, and 3. When CTU's average depth is greater than 2.0625, the CTU's predicted depth is 2 and 3.2, the optimized depth traversal range. The initial depth traversal scope does not take into account video characteristics and spatial correlation. In this paper, based on the initial depth traversal range, considering the characteristics of ERP projection video, and combined with the complexity index of CTU, the depth prediction interval of each category of CTU was optimized [20].

The CTU of category 5 is less complex, the CTU of this class is flat, and the depth traversal range is only 0 and 1. The CTU of category 1 is relatively complex, and the depth traversal scope only includes 2 and 3. CTU of categories 2, 3, and 4 is complex. In this paper, the depth of adjacent blocks is used to predict the depth of the current block. If the average depth of the CTU on the left is D_1, the average depth of the CTU on the top is D_u, the average depth of the CTU on the top left is D_{u1}, and the CTU complexity category is L_{CTU}, the average depth of the current CTU is defined as follows:

$$D_{\mathrm{CTU}} = \frac{\left(D_1 + D_u + D_{u1}\right)}{3} + \frac{1}{L_{\mathrm{CTU}}}. \tag{5}$$

For example, if the current CTU is classified as category 3 and the average depth of the left, upper, and upper-left CTU are 2.185, 2.4375, and 2.9375, respectively, then the average depth of the current CTU is 2.8542. The final depth traversal range is shown in Table 1 after the threshold value is adjusted under the condition that the guaranteed rate distortion cost is basically unchanged.

2.4. Analysis of Client Function Requirements

2.4.1. Personal Module. First of all, 3D models are needed to display facial expressions of the client, which requires an engine that can support 3D models and a game engine fits this need perfectly. The most commonly used game engines are Ogre, Unity, Unreal, etc. After investigation, Ogre has a good production effect and strong openness, but it is not good enough to support the client-server mode. It is good at making PC games, and the tools in Ogre are not complete and convenient enough to use because it requires a lot of energy. The multiplatform support for Unreal was not good enough, so Unity3D was considered. Personal module includes user login and registration, basic information editing and filling in, the content and the creator of the collection of thumbs up, and other operations. After login, users can like the content they are interested in, download the content, create the collection, and add the content they are interested in to the collection. And, they can go to the personal module at "my" page to view the collection of content and download the content of the creator of his choice. If the user does not log in to the account, the functions related to the account such as liking and collecting cannot be used, and it will jump to the login and registration module to guide the user to complete the account registration or login. The downloaded content can be viewed offline. Authenticated user creators can upload content, delete content, etc.

2.4.2. Content Discovery Module. The content discovery module includes the home page and the search page. The home page is the page displayed after the application of the content recommendation strategy. Users come to the panoramic video content platform with the expectation of discovering and experiencing the content. In order to facilitate the user to browse and view the content, the client uses one line to display multiple individual contents on the home page and flip the page to display the contents of the previous page and the next page. Each individual piece of content consists of a thumbnail of the video, a description of the content, and the author of the content. For paid content, they show the price of the content.

3D models can be created using professional modeling software. At present, there are many excellent modeling software in the market, such as Maya and 3DMax. The common feature of these software is to use simple geometric models, such as cubes and balls, to build complex models through translation, stretching, and rotation. 3D modeling is an important part of the performance of virtual games. 3D models can also be generated from digital signals scanned by 3D scanners or modeled from pictures or videos. The face model used in this topic is a 3D model made by traditional modeling software.

2.5. Algorithm Process. (1) Get current CTU pixel values and calculate the texture complexity as equation (4). (2) Determine the CTU category according to the texture complexity classification interval. (3) Deep prediction is determined according to the CTU category: if category is 1,

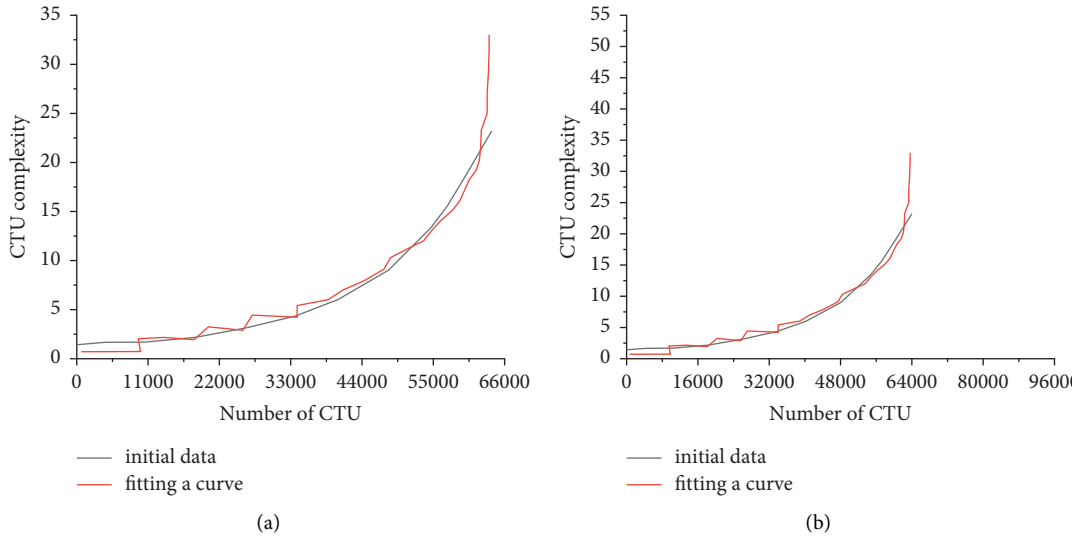

Figure 1: CTU complexity data distribution: (a) 4K video sequence acquisition data and (b) 6K video sequence acquisition data.

only depths 2 and 3 are traversed; if category is 5, only depths 0 and 1 are traversed; if category is 2, 3, and 4, then the average depth of the corresponding reference CTU is obtained. Substitution formula (5) calculates the average depth of the current CTU and then obtains the predicted depth according to the corresponding average depth threshold interval. (4) CU is divided according to the predicted depth.

Table 1: Predicted depth and CTU mean depth.

Prediction depth	CTU average depth
0, 1	[0, 1.25]
1, 2	(1.25, 1.4375]
1, 2, 3	(1.4375, 2]
2, 3	(2, 3.5]

3. Experimental Analysis

The above algorithm was integrated into HM16.20 and 360lib-4.0 to test the performance of the algorithm. The hardware parameters of the experimental platform are as follows: Intel(R) Core(TM) I7-7700 CPU@3.60 GHz CPU, 8.0 GB memory. The main encoding parameters of the experiment are as follows: all Intra Main10 (AI-Main10) encoding mode, the number of encoding frames is 100, and the initial QP is 22, 27, 32, and 37. In order to evaluate the comprehensive coding performance of the algorithm, the BD – rate calculation method provided by JVET is used to measure the relationship between the bit rate and the image quality. If ΔBD – rate is negative, the overall coding performance is improved. In addition, the WS – PSNR index defined by WMSE is also used to evaluate the image quality, as follows:

$$\text{WS} - \text{PSNR} = 10\log\left(\frac{\text{MAX}^2}{\text{WMSE}}\right), \tag{6}$$

$$\text{WMSE} = \sum_{i=0}^{\text{width}} \sum_{j=0}^{\text{height}-1} \left[y(i,j) - y'(i,j)\right]^2, \tag{7}$$

$$W(i,j) = \frac{w(i,j)}{\sum_{i=0}^{\text{width}-1}\sum_{j=0}^{\text{height}-1} w(i,j)}, \tag{8}$$

where MAX is the maximum value of the image pixel, $y(i,j)$ and $y'(i,j)$ represent the original pixel and reconstructed pixel, respectively, and $w(i,j)$ is the weighting scaling factor of the normalized sphere. The calculation of ΔWS – PSNR formula is as follows:

$$\Delta\text{WS} - \text{PSNR} = \text{WS} - \text{PSNR}_{\text{proposed}}. \tag{9}$$

The percentage of time saved compared to the reference algorithm is represented by ΔTime, using the following formula:

$$\Delta\text{Time} = \frac{T_{\text{proposed}} - T_{HM16.20}}{T_{HM16.20}} \times 100\%. \tag{10}$$

In this paper, 16 standard test sequences recommended by GoPro, InterDigital, Nokia, and Letin VR were used to test the algorithm. For the accuracy of the quality assessment, the sequence was converted to low resolution ERP projection format video before coding. For 8K and 6K video, the encoding size is 4096×2048 pixels, and for 4K video, the encoding size is 3328×1664 pixels.

Compared with the standard algorithm, the proposed algorithm reduces the coding time by 37.25%, reduces the WS – PSNR by 0.10 dB, and increases the BD – rate by 0.74% on average. Among them, sequence Balboa, sequence Broadway, sequence Skateboark-trick, and sequence *Train_le* save more time. This is because the low-complexity areas in these videos make up the majority of the entire region, while the higher complexity areas are extremely complex. A large number of CTUs in these video sequences can be divided directly according to the set depth range,

skipping unnecessary traversal and thus saving a lot of time. ChairliftRide and KiteFlite sequences save less time, an important reason is that there are many CTU with an average depth of 1.4375~2 in these video sequences, and traversing depths 1, 2, and 3 consume more time. Figure 2 shows the comparison of RD curves for different sequences. As can be seen from the figure, the RD curve of the proposed algorithm is basically coincident with HM16.20 at both high and low bit rates, indicating that the algorithm reduces the coding time and has almost no loss of video quality. This is because the algorithm in this paper fully considers the characteristics of 360-degree virtual reality video in ERP projection format and takes CTU as the basic unit for classification processing. It provides a quick decision scheme for CTU of categories 1 and 5 and sets up a depth prediction scheme based on spatial correlation for CTU of categories 2, 3, and 4. Compared with the method that divides the video image directly into the poles and the equator, the method presented in this paper is more applicable and more accurate in predicting the depth of CTU.

According to the subjective experimental results, the 720P resolution content was consistently evaluated by users, with blurred images and serious distortion. Under the condition of excellent network environment, the higher the code rate of video, the higher the perceived quality of video. However, when playing the 4K video, the full view of the transmission scheme has a significant increase in the number of times. Despite the improved picture quality, the user experience was affected by the lag. So, the score is about the same as the lower resolution. In the case of high-quality video transmission scenes in the range of FOV, the number of times of lag is significantly reduced, and the picture quality perceived by users is about the same as 2K sharpness, which is scored more than 4 points. In the case of a good network environment, the 2K resolution viewing experience is very poor, which is only about 1 point. The perceived quality of the video stream provided by the adaptive transmission scheme using DASH can reach around 3 points, also higher than other resolution cases. Through the comparative analysis of subjective and objective scores of the above experiments, it can be concluded that the DASH whole-view delivery scheme can provide users with a higher quality of user experience.

This paper discusses the research background and significance of data processing algorithm, analyzes the research status, summarizes the key problems involved in motion tracking and the existing common algorithms, and summarizes the problems in the tracking algorithm. The motion tracking algorithm MS, based on the mean shift, and the continuous adaptive mean shift algorithm CAMS are implemented. The mean shift is a common iterative algorithm for probability density estimation, with simple principle and high iteration efficiency. In motion tracking, the template matching problem can be transformed into the process of the mean shift convergence by constructing proper kernel function. The tracking algorithm has good real-time performance and can solve the partial occlusion problem to a certain extent.

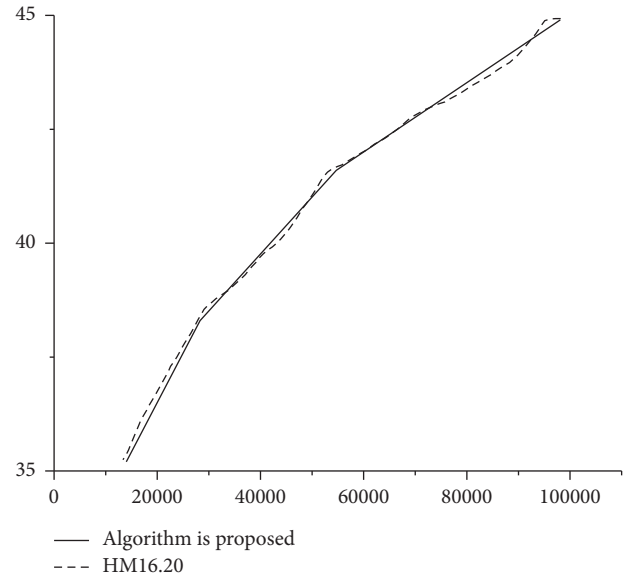

Figure 2: Comparison of the RD curve of the proposed algorithm and HM16.20.

4. Conclusion

The research and design of the panoramic content playing platform based on the adaptive strategy of streaming media has completed requirement analysis, outline design, detailed design, coding implementation, and testing. The subjective test is designed, and the evaluation of the broadcast experience is obtained through the subjective test. In order to reduce the coding time and coding complexity of 360-degree virtual reality video, experimental results show that compared with the standard algorithm, the proposed algorithm reduces the coding time by 37.25% and the BD-rate increases by 0.74% on average, with almost no loss of video quality.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the education sciences project in 2021 of the Hunan Province's 14th Five-Year Plan (project no. ND214462).

References

[1] Z. Liu, C. Xu, M. Zhang, and W. Yue, "Fast intra prediction and cu partition algorithm for virtual reality 360 degree video coding," *IEICE-Transactions on Info and Systems*, vol. E102.D, no. 3, pp. 666–669, 2019.

[2] Z. Liu, K. Yang, X. Fu, M. Zhang, and F. Mao, "Adaptive qp offset selection algorithm for virtual reality 360-degree video

based on ctu complexity," *Multimedia Tools and Applications*, vol. 80, no. 8, pp. 1–17, 2021.

[3] Z. Wang and Y. Zhu, "Video key frame monitoring algorithm and virtual reality display based on motion vector," *IEEE Access*, vol. 99, p. 1, 2020.

[4] X. Feng, Z. Bao, and S. Wei, "Liveobj: object semantics-based viewport prediction for live mobile virtual reality streaming," *IEEE Transactions on Visualization and Computer Graphics*, vol. 99, p. 1, 2021.

[5] P. Hou and Y. Zhang, "Dynamic image sampling and swimming motion image recognition in immersive virtual reality," *Microprocessors and Microsystems*, vol. 82, no. 3, Article ID 103760, 2020.

[6] P. Kumar, W. Hu, and Y. Yang, "Virtual reality based 3d video games and speech-lip synchronization superseding algebraic code excited linear prediction," *International Journal of Parallel Programming*, vol. 4, no. 12, pp. 303–321, 2017.

[7] W. Sun, J. Yu, Y. Kang, S. Kadry, and Y. Nam, "Virtual reality-based visual interaction: a framework for classification of ethnic clothing totem patterns," *IEEE Access*, vol. 12, no. 99, p. 1, 2021.

[8] G. Qin and G. Qin, "Virtual reality video image classification based on texture features," *Complexity*, vol. 2021, no. 2, pp. 1–11, Article ID 5562136, 2021.

[9] L. Xu, "Fast modelling algorithm for realistic three-dimensional human face for film and television animation," *Complexity*, vol. 2021, no. 2, pp. 1–10, Article ID 3346136, 2021.

[10] Y. Jin, M. Chen, T. Goodall, A. Patney, and A. C. Bovik, "Subjective and objective quality assessment of 2d and 3d foveated video compression in virtual reality," *IEEE Transactions on Image Processing*, vol. 99, p. 1, 2021.

[11] Y. Wang, J. Yue, Y. Dong, and Z. Hu, "Review on kernel based target tracking for autonomous driving," *Journal of Information Processing*, vol. 24, no. 1, pp. 49–63, 2016.

[12] S. R. Rasmussen, L. Konge, P. T. Mikkelsen, M. S. Sørensen, and S. A. W. Andersen, "Notes from the field," *Evaluation & the Health Professions*, vol. 39, no. 1, pp. 114–120, 2016.

[13] B. Yun, "Design and reconstruction of visual art based on virtual reality," *Security and Communication Networks*, vol. 2021, no. 8, pp. 1–9, Article ID 1014017, 2021.

[14] M. Zikky, M. J. Arifin, K. Fathoni, and A. Z. Arifin, "Performance analysis of specification computer and mobile with implementation tawaf virtual reality using a∗ algorithm and rvo system," *EMITTER International Journal of Engineering Technology*, vol. 7, no. 1, pp. 55–70, 2019.

[15] B. Gan, C. Zhang, Y. Chen, and Y. C. Chen, "Research on role modeling and behavior control of virtual reality animation interactive system in internet of things," *Journal of Real-Time Image Processing*, vol. 1, pp. 1–15, 2020.

[16] C. Liu, Q. I. Yue, and W. Ding, "The data-reusing mcc-based algorithm and its performance analysis," *Chinese Journal of Electronics*, vol. 25, no. 4, pp. 719–725, 2016.

[17] X. Liu, "Three-dimensional visualized urban landscape planning and design based on virtual reality technology," *IEEE Access*, vol. 99, p. 1, 2020.

[18] J. Deng and Y. Xie, "Performance characterization of illumination algorithms for reconfigurable graphics processor," *The Journal of China Universities of Posts and Telecommunications*, vol. 26, no. 5, pp. 64–75, 2019.

[19] V. Hohmann, R. Paluch, M. Krueger, M. Meis, and G. Grimm, "The virtual reality lab: realization and application of virtual sound environments," *Ear and Hearing*, vol. 41, no. Supplement 1, pp. 31S–38S, 2020.

[20] D. Jeong, S. Yoo, and Y. Jang, "Motion sickness measurement and analysis in virtual reality using deep neural networks algorithm," *Journal of the Korea Computer Graphics Society*, vol. 25, no. 1, pp. 23–32, 2019.