

Research Article

Performance Optimization of 3-DOF Application Scene Based on 360-Degree Panoramic Technology with Depth Information

Qinghua Wang ,¹ Linjuan Ma ,² Xiaotong Huang ,¹ Lan Lyu ,¹ and Fuquan Zhang ,^{3,4}

¹College of Mathematics and Data Science (Software College), Minjiang University, Fuzhou, China 350108

²School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China 100081

³College of Computer and Control Engineering, Minjiang University, Fuzhou, China 350108

⁴Digital Media Art, Key Laboratory of Sichuan Province, Sichuan Conservatory of Music, Chengdu, China 610021

Correspondence should be addressed to Fuquan Zhang; zfq@mju.edu.cn

Received 28 February 2022; Accepted 29 March 2022; Published 26 April 2022

Academic Editor: Chao-Yang Lee

Copyright © 2022 Qinghua Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

High-quality 3D scenes often show poor rendering effect and insufficient operational performance in low-end devices. Therefore, how to make better use of 360-degree panoramic technology and replace the traditional pure model scene by visual deception is of great significance to the picture quality improvement and performance optimization of low-end equipment. In this paper, a three-degree-of-freedom (3-DOF) application scenario for indoor simulation is used as an example. Under the premise of 360-degree panoramic technology, depth information is obtained through laser ranging, converted into a two-dimensional form of depth image with the help of spherical coordinates, and combined with a transparent information image, the depth image is compressed using DEFLATE compression algorithm, and finally be stored. By optimizing the data through bilinear sampling and with the help of geometric knowledge, the occluded parts are removed or translucent according to the actual situation when the graphics are rendered, and the occlusion relationship is correctly handled in order to achieve a real integration of the virtual environment with the 3D objects. Through multiple data, it is shown that the method possesses significant performance improvement while ensuring geometric realism.

1. Introduction

With the development of digital media technology, the 3D rendering effect has been continuously improved and applied to various fields [1–10]. However, the generated 3D images require heavy computation, which result in significant increases in performance requirement by operating equipment. In order to make high-quality 3D images available on low-end devices, we often use 360-degree panoramic technology instead of the pure modeling scheme (PM) [11–16]. Nevertheless, because of the noninteractivity of virtual background, it is difficult to achieve real integration with the internal 3D object, with a result that it is generally being used only in distant scenes, or comes alone. Therefore, to further replace the 3D scene and increase the sense of reality, it is necessary to solve the occlusion relationship between this two.

As early as the augmented reality was proposed, people have carried out relevant studies on occlusion consistency [17–21], in which the technology that can be directly applied to 3-DOF application scenarios is the occlusion scheme eliminated by rendering sequence based on environmental modeling (OEM) [22]. Due to the invariant nature of the virtual scene, this technique reconstructs the environment with a low-model mesh, with depth testing and depth writing enabled, we adjust the rendering order so that the environment model is rendered first and the later rendered 3D objects are excluded from the back part of the environment and the opacity of the environmental low-model is set to 0. At present, the main problem of this solution is that the accuracy of the low model is not high and the masking of small objects is poor, while if the model accuracy is increased, the performance requirements of the equipment are significantly higher, resulting in the inability to play an

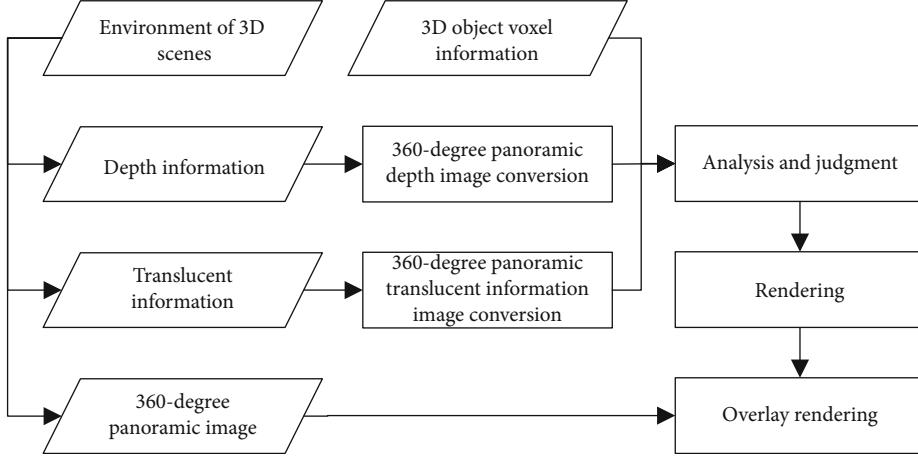


FIGURE 1: Flow chart of optimization method.

optimal role. Another common solution is a technique commonly used in 2D games which is implemented in 3D by layering the scene content into multiple 360-degree panoramas based on distance and proximity to achieve a sense of different levels of visualization. This approach is simpler, but the problem is also obvious: there is no transition effect when 3D objects span between layers.

In this paper, through the comparison and test of various solutions, we propose an occlusion scheme based on a 360-degree panoramic depth image (OPDI) for 3-DOF application scenarios of indoor simulation, based on the premise of optimizing performance and preserving spatial realism, using the depth information of 360-degree panoramic environment and eliminating the occluded parts of objects through graphical depth calculation, so that 3D objects can form a real visual sensation of occlusion with the environment in the only 360-degree panoramic image technology environment, which allows for the preservation of spatial realism while optimizing performance.

2. Method

This section specifically introduces the optimization scheme of 3-DOF application scenario using 360-degree panoramic technology with depth information in interior design simulation. The 360-degree panoramic image is acquired by the camera; furthermore, the 360-degree panoramic depth information is obtained by laser ranging and then saved in a 16-bit single-channel 360-degree panoramic grayscale image by spherical coordinate conversion and compression using DEFLATE compression algorithm and then based on the 360-degree panoramic depth image and scene translucent objects to draw a 360-degree panoramic transparency information image. Finally, when rendering the 3D object, the depth information and transparency information of the environment relative to the camera in each spatial voxel are retransformed to compare with the spatial voxel corresponding to the 3D object, so as to determine how to render and display. Figure 1 shows the specific process, where parallelograms represent the data, and rectangles represent the steps.

First of all, a 360-degree panoramic view of the required indoor environment needs to be collected according to the 360-degree panoramic camera, as shown in Figure 2.

In the same position of the camera, the laser ranging sensor rotates at high speed in the range of horizontal 360-degree and vertical 180 degrees, and the depth information of each angle is detected and collected in turn. The next step is to generate an associated 360-degree panoramic depth image. Unlike traditional 2D image methods, depth information will be built in a spherical coordinate system, which is more suitable to the properties of the depth data and then a lossless depth image is generated. Take the left-handed coordinate system as an example, the corresponding spherical coordinates are shown in Figure 3. Then, the information of each point on the spherical coordinate is mapped to the depth image, where the yaw angle $\theta(V_e)$ is the row coordinate of the depth image pixel, and the pitch angle $\omega(V_e)$ is the vertical coordinate of the depth image pixel. It is worth noting that for the convenience of the following calculation in this sphere coordinate, the yaw angle θ is calculated starting from +Z axis, and the pitch angle ω is calculated starting from -Y axis, while the distance value $distance(V_e)$ is converted to pixel value $P(P_{xi}, P_{yi})$. The mapping relationship is as follows:

$$\begin{aligned}
 P_{xi} &= \text{round}(\theta(V_e) \times \text{multiple}), \\
 P_{yi} &= \text{round}(\omega(V_e) \times \text{multiple}), \\
 P(P_{xi}, P_{yi}) &= \text{round}(distance(V_e)).
 \end{aligned} \tag{1}$$

In these formulas, the round function is used to round a number and multiple represents the quantization coefficient of the angle, which determines the resolution of the 360-degree panoramic depth image. Since $1/\text{multiple}$ is exactly the step size of the sensor, the obtained depth information density can be effectively controlled by adjusting the coefficient.

The generated depth images are saved in 16-bit single-channel PNG format grayscale images using DEFLATE compression derived from LZ77, and the data are



FIGURE 2: 360-degree panoramic image.

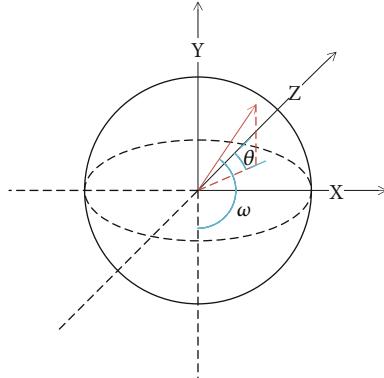


FIGURE 3: Ball coordinates.

compressed using DEFLATE compression algorithm after processing by the Lempel-Ziv (LZ77) [23] algorithm to ensure lossless compression while reducing spatial redundancy. The final result is shown in Figure 4.

After acquiring the 360-degree panoramic depth image, the next step is to perform the most important occlusion removal work in this paper, which will be done in the graphics rendering, where the model spatial coordinates of the 3D object vertices are obtained through the vertex shader and converted to world spatial coordinates and then further into camera spatial coordinates position_{camera}(x_{camera} , y_{camera} , z_{camera}).

$$\text{position}_{\text{camera}} = \text{position}_{\text{model}} \times \text{matrix}_{\text{mw}} \times \text{matrix}_{\text{wc}}, \quad (2)$$

where matrix_{mw} is the transformation matrix from the model space to the world space and matrix_{wc} is the transformation matrix from the world space to the camera space. After transformation, the results are transmitted to the patch shader, in which the vertex coordinate information is automatically linear interpolated, thus ensuring that the current camera space coordinates of each voxel of the object are obtained. Then, the distance value distance(V_o) is calculated according to the camera spatial coordinates of each voxel. The formula is as follows:

$$\text{distance}(V_o) = \sqrt{x_{\text{camera}}^2 + y_{\text{camera}}^2 + z_{\text{camera}}^2}. \quad (3)$$

The next step is to obtain information about the direction of each corresponding voxel in the 360-degree panoramic depth image. According to the spatial coordinates of each voxel, the yaw angle $\theta(V_o)$ and the pitch angle $\omega(V_o)$ relative to the center position are calculated according to the following formula:

$$\begin{aligned} \alpha &= \arctan \left(\frac{x_{\text{camera}}}{z_{\text{camera}}} \right) \times \frac{180}{\pi}, \\ \theta(V_o) &= \begin{cases} 180 - \text{abs}(\alpha), & (\text{IF } z_{\text{camera}} \leq 0 \text{ AND } x_{\text{camera}} > 0), \\ 180 + \text{abs}(\alpha), & (\text{ELSE IF } z_{\text{camera}} < 0 \text{ AND } x_{\text{camera}} \leq 0), \\ 360 - \text{abs}(\alpha), & (\text{ELSE IF } z_{\text{camera}} \geq 0 \text{ AND } x_{\text{camera}} < 0), \\ \text{abs}(\alpha), & (\text{ELSE}), \end{cases} \\ \omega(V_o) &= \arctan \left(\frac{y_{\text{camera}}}{\sqrt{x_{\text{camera}}^2 + z_{\text{camera}}^2}} \right) \times \frac{180}{\pi} + 90. \end{aligned} \quad (4)$$

It is worth noting that the value range of function arctan is $[-\pi/2, \pi/2]$; therefore, we also need to make interval judgment of angle to ensure that $\theta(V_o)$ is in $[0, 360]$ interval. Since we defined $\omega(V_o)$ is in $[0, 180]$ interval as to store data conveniently before, here, we also need to add 90 to it in order to comply with the previous setting of ω starting with $-Y$ axis.

Based on yaw angle $\theta(V_o)$ and pitch angle $\omega(V_o)$ obtained from 3D model voxels, mapping to the $[0, 1]$ interval which served as the UV value of the pixel point of the depth image, the corresponding distance $P(UV_{xi}, UV_{yi})$ is read out from _MainTex texture using the sampling function tex2d.

$$\begin{aligned} UV_{xi} &= \frac{\theta}{360}, \\ UV_{yi} &= \frac{\omega}{180}, \end{aligned} \quad (5)$$

$$P(UV_{xi}, UV_{yi}) = \text{tex2d}(_{\text{MainTex}}, UV(xi, yi))$$

Due to the interval of sampling steps, the final effect will produce a certain degree of sawtooth. Therefore, according to the step size and application requirements when



FIGURE 4: 360-degree panoramic depth image.



FIGURE 5: 360-degree panoramic transparent information image.

collecting the 360-degree panoramic depth map, the 360-degree panoramic depth image can be selectively bilinear sampling processing, so as to smooth the data. Finally, the distance value in the depth image is obtained according to the UV value of the corresponding voxel and compared with that voxel to determine whether the voxel needs to be rejected, where Max is the maximum value that can be stored in the depth image.

$$\text{IsDiscard}_{(1)} = \begin{cases} \text{true}, & (\text{IF } P(UV_{xi}, UV_{yi}) < \text{distance}(V_o) \text{ AND } \text{distance}(V_o) < \text{Max}), \\ \text{false}, & (\text{ELSE}). \end{cases} \quad (6)$$

$\text{IsDiscard}_{(1)}$ can now do most of the scene masking effects, but in reality, there are some cases where full masking is not desired, for example, glass and other transparent material environment objects which need to translucent processing rather than simply eliminate, so we also need a 360-degree panoramic image of transparent information. It is worth mentioning that through the drawing software, a copy is made on the depth image, and the part that needs to be transparent is filled with different grayscale colors according to the intensity of transparency, while white is completely opaque, and the resolution can be reduced appropriately according to the actual situation, so a 360-degree panoramic environmental transparency information image is obtained, as shown in Figure 5.

We can use it to calculate the opaque value Alpha of 3D objects to achieve the superposition effect with the 360-degree panoramic environment. The data acquisition method of 360-degree panoramic transparency information image is the same as acquiring depth information, where $A(UV_{xi}, UV_{yi})$ is the opacity value of the corresponding point on the 360-degree panoramic transparency information image, and when $A(UV_{xi}, UV_{yi}) = 1$, it means completely opaque, factor dynamically controls the adjustment of transparency intensity, function saturate limits the value range to $[0, 1]$ interval, and at the same time, we need to redefine the elimination condition $\text{IsDiscard}_{(2)}$ on the basis of $\text{IsDiscard}_{(1)}$.

$$\begin{aligned} \text{Alpha} &= \begin{cases} 1, & (\text{IF } \text{IsDiscard}_{(1)} = \text{true AND } A(UV_{xi}, UV_{yi}) \neq 1), \\ 1, & (\text{ELSE}), \end{cases} \\ \text{IsDiscard}_{(2)} &= \begin{cases} \text{true}, & (\text{IF } \text{IsDiscard}_{(1)} = \text{true AND } A(UV_{xi}, UV_{yi}) = 1), \\ \text{false}, & (\text{ELSE}). \end{cases} \end{aligned} \quad (7)$$

3. Experiment

In order to verify the superiority of the scheme in this paper, we conducted three groups of control experiments on the virtual scene in Unity Engine 2020.3.11f1c1: they are OPDI,

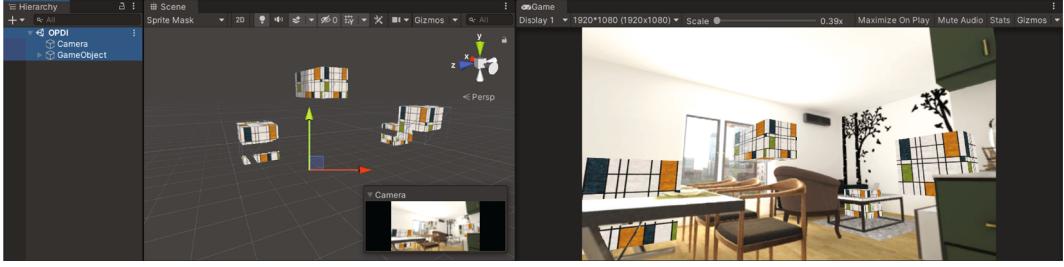


FIGURE 6: OPDI.

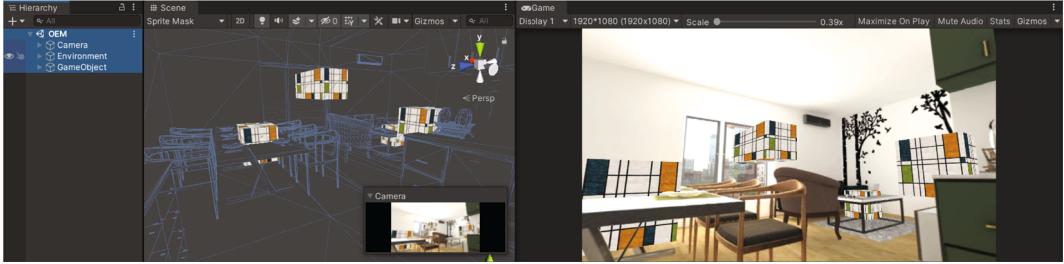


FIGURE 7: OEM.

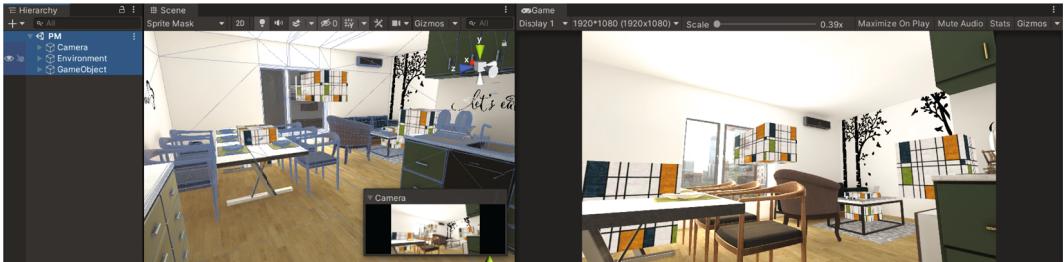


FIGURE 8: PM.



FIGURE 9: Translucent effect image of OPDI.

OEM, and PM. In the experiment, the Game panel was rendered with a resolution of 1920px * 1080px, and an indoor environment (model from Unity Asset Store) was selected as the scene; the scene of three scenarios was tested for the

corresponding masking effect with a rectangular cube as the 3D object.

In order to observe clearer, the display type of Scene Panel in Figures 6–8 is Sprite Mask. Through the Hierarchy

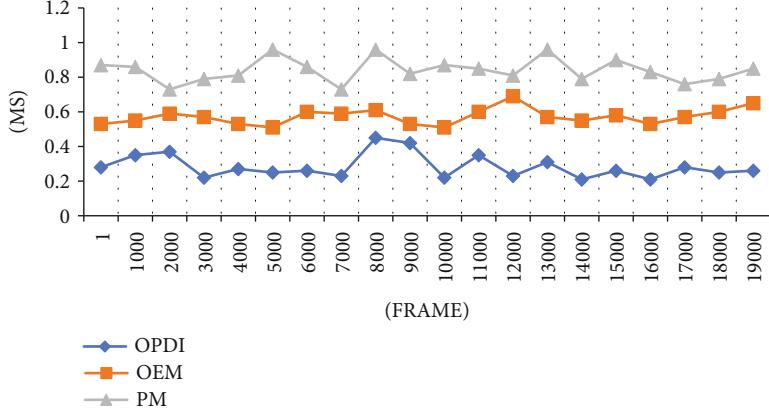


FIGURE 10: CPU usage (Rendering).

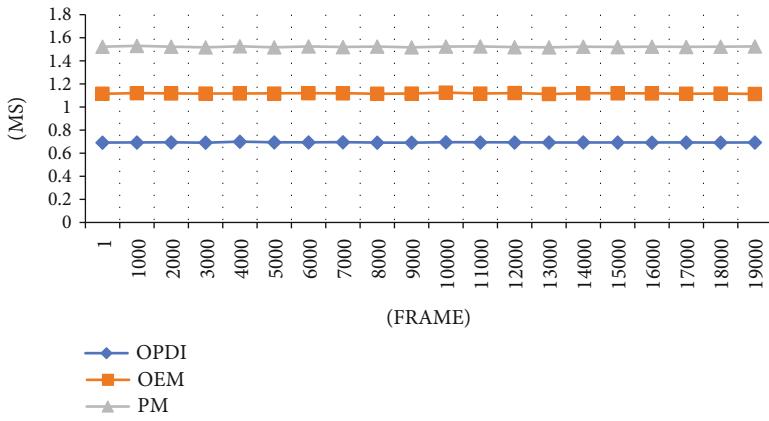


FIGURE 11: GPU usage (Camera.Render).

TABLE 1: Memory usage.

Experiment name	Memory usage
OPDI	1.92 GB
OEM	2.08GB
PM	4.61GB

TABLE 2: Experiment related environment.

OS	Windows 10 64-bit (DirectX 12)
CPU	Intel® Core™ i7-8750H CPU @ 2.20 GHz
GPU	Nvidia GeForce GTX 1060 (6 GB)
RAM	32 GB (DDR4 2666 MHz)

TABLE 3: Rendering.

Experiment name	Draw calls	Triangles	Vertices
OPDI	6	1.7 k	5.1 k
OEM	73	12.5 k	23.4 k
PM	79	56.0 k	50.7 k

Panel and the Scene Panel, we can clearly see the differences of these schemes in terms of model requirements, in which the OPDI does not use any environmental

model, while the model compression rate of OEM is about 75% of the model of PM. The Game Panel on the right side shows the occlusion effect under different schemes, and there is almost no difference in vision. The OPDI also reproduces the translucent effect almost realistically, as shown in Figure 9.

With the help of Unity Profiler Panel, the performance consumed by the three schemes is statistically analyzed. Figures 10 and 11 and Table 1 show the differences of CPU usage (Rendering), GPU usage (Camera.Render), and memory usage in different schemes during a period of running. The runtime configuration is shown in Table 2.

By comparison, it can be found that no matter in CPU usage (Rendering) or GPU usage (Camera. Render), OPDI takes the shortest processing time, and memory usage is also the least, followed by OEM, and the last is PM.

By analyzing the data in Table 3, triangles, vertices, and draw calls, all three of which are major factors affecting performance, where triangles and vertices represent the number of triangles and vertices to be drawn, and too much slice by slice and vertex by vertex computation will seriously affect the GPU usage. Draw calls, on the other hand, represent the number of times the CPU submits data to the GPU and then issues rendering commands to the GPU, so it will directly determine the CPU usage. Therefore, reducing the number of these three has a significant effect on improving

performance. The depth image-based occlusion scheme provided in this paper does exactly exploit this property while ensuring a sense of spatial realism as a way to greatly improve performance. Moreover, the scheme proposed in this paper has no significant change in the number of draw calls, triangles, vertices, and memory required by the scene in any environmental scenario; however, through the above data analysis, the relatively more complicated pure model-nature scenario scheme, however, can optimize the performance substantially.

4. Conclusion

In this paper, taking the 3-DOF application scenario of indoor simulation as an example, a relatively novel performance optimization scheme is proposed under the condition of ensuring the spatial realism. Using 360-degree panoramic image instead of real 3D environment, mapping the distance of 360-degree panoramic environment relative to the camera to 2D image with the help of spherical coordinate conversion, and bilinear sampling data with the help of data alignment features in the image so as to optimize it, and comparing and eliminating the distance of each voxel of 3D object relative to the camera with the corresponding information saved in the 360-degree panoramic depth image to form the corresponding occlusion effect. Similarly, with the help of a 360-degree panoramic transparent information image, the occlusion of glass and other materials has a translucent effect. Through the method proposed in this paper, both the picture quality and the spatial realism are ensured, and the performance optimization is further achieved. According to the experimental results, the proposed scheme is obviously superior to the current mainstream 3D scene optimization schemes in terms of visual effect and performance. At the same time, the scheme is not only suitable for 360-degree panoramic image but also for 360-degree panoramic video. Therefore, it will have a broader application scenario, for example, in order to solve the problem of insufficient performance and poor display effect of 360-degree panoramic games with nonmobile perspective using high-quality rendering on traditional mobile platforms, the 360-degree panoramic video can also be used to achieve some visual effects of scene view movement, which is of certain significance for the expansion of virtual reality technology.

Data Availability

The data we used is available and the performance optimization scheme proposed in this paper can be used in 3-DOF application scenarios. And part of them is available to you from the corresponding author upon request (zfq@mju.edu.cn).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the study of this work and publication of this paper.

Acknowledgments

Our work is supported by the Digital Media Art, Key Laboratory of Sichuan Province, Sichuan Conservatory of Music, Project No.: 21DMAKL01; supported by the first batch of industry-university cooperation collaborative education project funded by the Ministry of Education of the People's Republic of China, 2021, Project No.: 202101071001; supported by Minjiang University school-level scientific research project funding, Projects No.: MYK17021, MYK18033, and MYK21011; and supported by Minjiang University Introduced Talents Scientific Research Start-up Fund, Projects No.: MJY21030.

References

- [1] N. F. Aristin, B. Budijanto, D. Taryana, and I. N. Ruja, "3D map of dry land use based aerial image as learning media in era of education 4.0," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 7, p. 171, 2020.
- [2] O. López Chávez, L.-F. Rodríguez, and J. O. Gutierrez-Garcia, "A comparative case study of 2D, 3D and immersive-virtual-reality applications for healthcare education," *International Journal of Medical Informatics*, vol. 141, article 104226, 2020.
- [3] H. Siala, E. Kutsch, and S. Jagger, "Cultural influences moderating learners' adoption of serious 3D games for managerial learning," *Information Technology & People*, vol. 33, no. 2, pp. 424–455, 2020.
- [4] J. Kim, "Digital and postdigital 3D animation in the contemporary Chinese art scene: Miao Xiaochun and Lu Yang," *Journal of Chinese Cinemas*, vol. 11, no. 3, pp. 227–242, 2017.
- [5] F. Zhang, G. Ding, L. Ma, Y. Zhu, Z. Li, and L. Xu, "Research on stage creative scene model generation based on series key algorithms," in *International Conference on Smart Vehicular Technology, Transportation, Communication and Applications*, pp. 170–177, Mount Emei, China, 2018.
- [6] F. Zhang, G. Ding, Q. Lin, L. Xu, Z. Li, and L. Li, *Research of Simulation of Creative Stage Scene Based on the 3DGans Technology14*.
- [7] P. Derudas and Å. Berggren, "Expanding field-archaeology education: the integration of 3D technology into archaeological training," *Open Archaeology*, vol. 7, no. 1, pp. 556–573, 2021.
- [8] T. K. Coşkun and G. F. Deniz, "The contribution of 3D computer modeling education to twenty-first century skills: self-assessment of secondary school students," *International Journal of Technology and Design Education*, vol. 31, 2021.
- [9] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, p. 40, 2019.
- [10] A. Laksono and Aditya, "Utilizing a game engine for interactive 3D topographic data visualization," *ISPRS International Journal of Geo-Information*, vol. 8, no. 8, p. 361, 2019.
- [11] K. A. Ritter and T. L. Chambers, "Three-dimensional modeled environments versus 360 degree panoramas for mobile virtual reality training," *Virtual Reality*, vol. 25, 2021.
- [12] F. Chiariotti, "A survey on 360-degree video: coding, quality of experience and streaming," 2021, <https://arxiv.org/abs/2102.08192>.

- [13] S. Gilbert, W. Boonsuk, and J. W. Kelly, *Virtual Displays for 360-Degree Video*, Proceedings of SPIE - The International Society for Optical Engineering, Burlingame, California, USA, 2012.
- [14] R. Eiris, M. Gheisari, and B. Esmaeili, "PARS: using augmented 360-degree panoramas of reality for construction safety training," *International Journal of Environmental Research and Public Health*, vol. 15, no. 11, p. 2452, 2018.
- [15] R. Eiris, J. Wen, and M. Gheisari, "iVisit: digital interactive construction site visits using 360-degree panoramas and virtual humans," in *Construction Research Congress 2020*, pp. 1106–1116, American Society of Civil Engineers, Tempe, Arizona, 2020.
- [16] C. Snelson and Y.-C. Hsu, "Educational 360-degree videos in virtual reality: a scoping review of the emerging research," *TechTrends*, vol. 64, no. 3, pp. 404–412, 2020.
- [17] A. Holynski and J. Kopf, "Fast depth densification for occlusion-aware augmented reality," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–11, 2018.
- [18] H. Kim and K. Sohn, *Hierarchical Depth Estimation for Image Synthesis in Mixed Reality*, A. J. Woods, M. T. Bolas, J. O. Merritt, and S. A. Benton, Eds., Proceedings of SPIE - The International Society for Optical Engineering, Santa Clara, CA, 2003.
- [19] D. E. Breen, R. T. Whitaker, E. Rose, and M. Tuceryan, "Interactive occlusion and automatic object placement for augmented reality," *Computer Graphics Forum*, vol. 15, no. 3, pp. 11–22, 1996.
- [20] D. R. Walton and A. Steed, "Accurate real-time occlusion for mixed reality," in *23rd ACM Symposium on Virtual Reality Software and Technology*, pp. 1–10, ACM, Gothenburg Sweden, 2017.
- [21] A. K. Hebborn, N. Hohner, and S. Muller, "Occlusion matting: realistic occlusion handling for augmented reality applications," in *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 62–71, Nantes, 2017.
- [22] H. Kim and K. Sohn, "3D reconstruction from stereo images for interactions between real and virtual objects," *Signal Processing: Image Communication*, vol. 20, no. 1, pp. 61–75, 2005.
- [23] M. Sudan and D. Xiang, "A self-contained analysis of the Lempel-Ziv compression algorithm," 2019, <https://arxiv.org/abs/1910.00941>.