



Design and Modeling of Chinese Classical Lanterns Based on Different Processes

Min Ji

School of Animation and Digital Arts, Communication University of Shanxi, Taiyuan, China
201010013433@stu.swmu.edu.cn

Abstract. In recent times, Three-dimensional(3D) technology has gained significant popularity across various industries including gaming, movies, medicine, and architecture, primarily due to advancements in computer graphics technology. However, the requirements for modeling tools and functionalities vary across industries. Therefore, it is crucial to select the most suitable software that aligns with the specific needs and professionalism of each industry. This research paper focuses on exploring the work processes that effectively capture the essence of Chinese classical lantern aesthetics. The research begins by utilizing OpenSCAD to construct a highly detailed model and subsequently combines it with the capabilities of Three.js to generate interactive rendering results. Additionally, the study incorporates the versatile functionalities of Blender to create intricate models, stunning renderings, and captivating animations, presenting an alternative method of showcasing the aesthetics. Based on a survey conducted among random viewers, the model rendered using the Blender Cycles engine was found to be highly effective.

Keywords: 3D model, OpenSCAD, lantern, graphics rendering.

1 Introduction

3D modeling is now an indispensable tool for different applications. For example, the game-making industry creates models that use different characters and props to interact with them; cultural heritage restoration can be achieved by using 3D samples from digital photography; and digital 3D models also have a positive impact on education and research of treating diseases and medicine [1-3].

In the field of computer graphics today, software for modeling can be divided into two different types, parametric modelers and free-form grid modelers. The former uses parameters and constraints to define the shape and geometric characteristics of the model. The user can change the shape of the model by adjusting the value of the parameter without having to edit every detail manually; the latter defines the model by creating and manipulating the grid. Users are free to add, move, rotate, and scale grid nodes to accurately shape the model [4]. This paper is dedicated to strengthening the protection and inheritance of traditional Chinese lantern culture and exploring the potential of different types of 3D modeling software for cultural transmission. By comparing Blender, OpenSCAD and Three.js, the widely used 3D modeling

software, is expected to offer more possibilities for digital presentation of traditional cultures and promote the integration of cultural conservation and technological innovation. We will study the modeling process, texture mapping process, animation effect and final rendering result in detail, and show the comparison results with actual cases.

2 The Design and Implement of the lantern by OpenSCAD

Our work is about the creation of traditional Chinese lanterns. To refine the details of the project and make it look like a better presentation, we compared the effects of two different software production streams and added some additional content, including wind simulation and flame simulation. The model itself is made with OpenSCAD, rendering using two different software, Three.js and Blender.

The lantern comprises three crucial components, including a frame, lampshade, and decoration. Through the dissection of a complicated structure into several fundamental geometric shapes and applying boolean operations (e.g. union, difference, intersection) [5], the lantern's authentic appearance can be accurately portrayed. The procedure for modeling this project is elaborated in detail below.

2.1 The lampshade design of the lantern

The lampshade of the lantern consists of two basic cubes (Fig.1). By adjusting their dimensional parameters, the central Boolean algebra ensures that the lantern body is in the center and aligned for the subsequent building of other components.

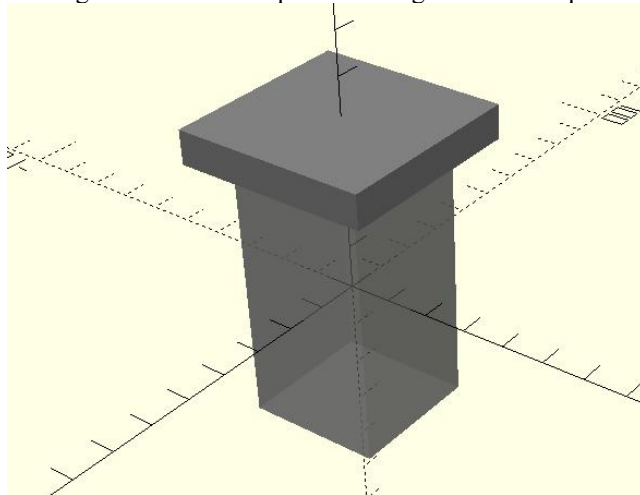


Fig. 1. The lampshade model at the center of coordinates (Original)

2.2 The outer frame design of the lantern

To create the frame structure outside the lampshade, the program uses differentials. By removing some of the smaller cubes from the white cubes (Fig.2), the program creates a frame that closely resembles the structure of the real lantern.

The design of the window grilles includes 16 corners, inspired by the symmetrical and regular appearance of Chinese lanterns. To achieve this, the first step is to create a list of all the different shapes of individual grills in a specific area, which are then combined using a process of union. After creating one corner model, the next step is to systematically place this model in each corner of the outer frame using the mirror.

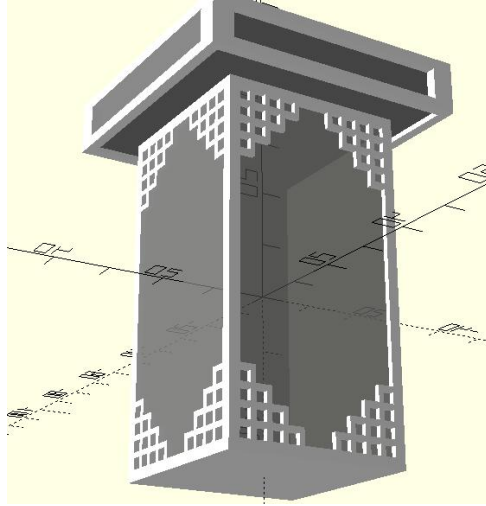


Fig. 2. The outer frame model (Original)

2.3 The decoration design of the lantern

To make the model look richer and more realistic, we chose to add more detail to the top, bottom, and perimeter of the lantern. Similar to the above, most decorations frequently use the programming capabilities of differentials and mirrors. In addition, Using the for loop function, the coordinate position of each cylinder is controlled so that it can be generated one by one at the same spacing in a fixed range of circles. Through this way, the model will be closer to the shape of the real tassels (Fig.3).

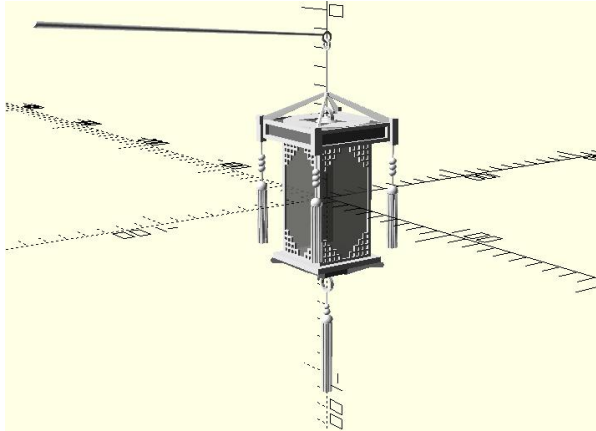


Fig. 3. The decoration model (Original)

When working with OpenSCAD, we utilize basic geometry to efficiently design our projects through parameterized modeling. This method employs boolean operators and geometric primitives to produce precise modeling outcomes. However, it may not support certain advanced modeling operations and intricate geometry, which can restrict the user's capacity to create complex models.

3 Rendering in Three.js

3.1 Platform and environment

We enhanced the appearance of our lantern by applying textures using Three.js. First, color-hex color values are written to render the color of the lantern's surface using MeshPhongMaterial. Secondly, by inputting specific codes, we were able to incorporate a repeating checkerboard texture onto the plane that will receive the shadow (Fig.4). This was made possible by utilizing Three.js's capabilities for integrating textures into objects. Phong material simulates mirror reflection on the surface of an object, thus displaying bright highlights on a smooth surface. It can be widely used in real-time rendering and requires relatively low computational resources.

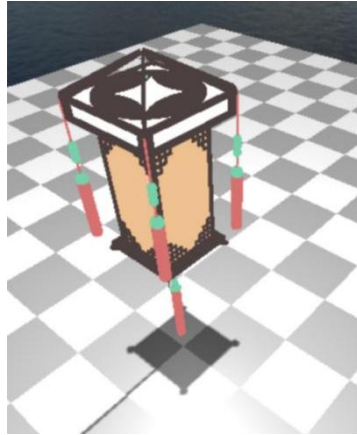


Fig. 4. Lantern texture and checkerboard grid texture's effect (Original)

In order to enhance the realism of a 3D scene, it is recommended to maintain camera stability while improving the visual effects. This can be achieved by utilizing skybox rendering technology. The process involves carefully selecting a suitable background map and loading a texture with six panoramic images, each representing a different side of the skybox (as shown in Fig. 5). Create a cube map loader instance using `CubetextureLoader` in `Three.js` and upload the six cube maps above using the `load` method, and finally apply them to the scene [6]. Using this way, the images are created as a skybox grid to ensure that they appear behind all other objects. The skybox is then linked to the camera to ensure that it remains synchronized as it moves or rotates, creating a natural and immersive environmental background.

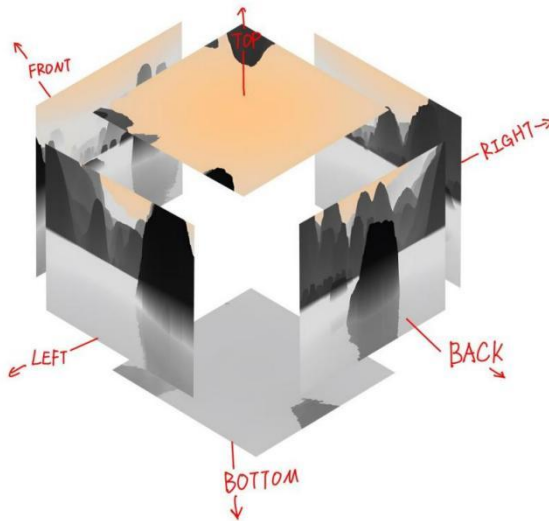


Fig. 5. Background [7]

3.2 Normal Map

In order to further enhance the authenticity and richness of the lampshade model, we have employed the use of normal maps to imitate concave and convex details. This particular technique provides a performance boost by incorporating more intricate details without the need to increase the model's complexity or number of vertices. Furthermore, it allows for the flexible modification of different maps to achieve a wide range of effects.

Since the addition of this content, the previously flat surface without any details has now transformed into a slightly wrinkled paper-like texture, which better resembles the rough texture of an actual lantern surface (Fig.6 and Fig.7).

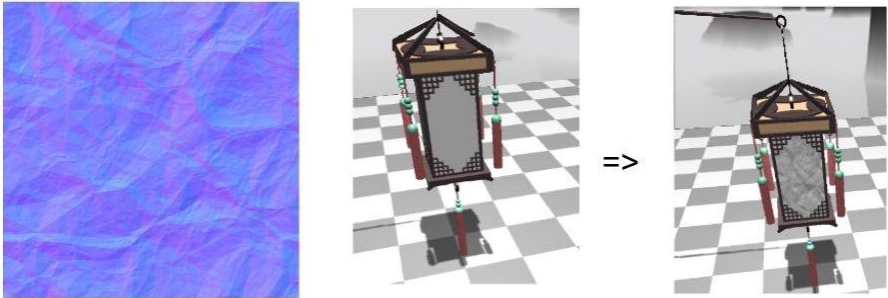


Fig. 6. Effect comparison using normal mapping (Original)

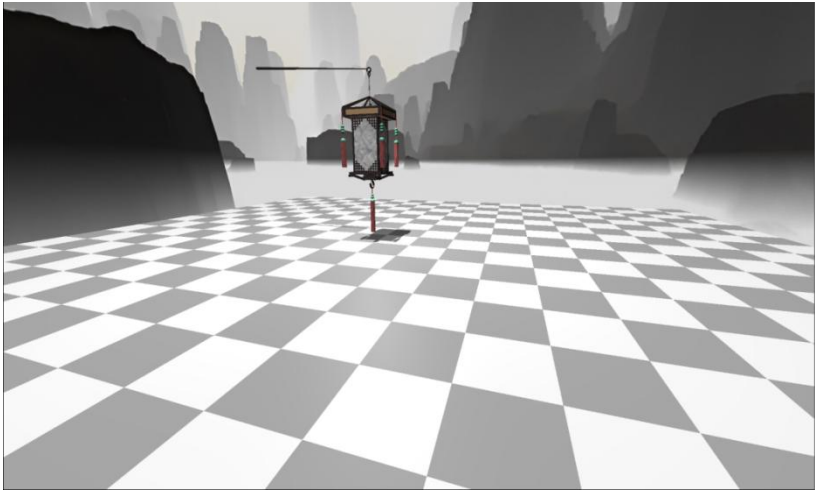


Fig. 7. Result of Texturing and Shading (Original)

4 Rendering in Blender

To verify the effectiveness of rendering results using different methods in different software, models in OpenSCAD are classified by material and imported into Blender

as Stereolithography(STL) for rendering. STL is a common 3D model file format that is used to describe the surface geometry of a three-dimensional object. It decomposes the object into many small triangular pieces and records the vertex coordinates and normal direction information of each piece [8].

Blender is a free and open-source 3D graphics and image software that provides solutions from modeling, materials, and rendering [9]. In order to better perform the functions of the software, we changed the details of some models and added a surface subdivision modifier to improve the resolution of geometric details.

Instead of making Phong material directly for the model in Three.js, we first expand the surface of the model into a plane in Blender, a process called UV unwrapping, the letters “U” and “V” denote the axes of the 2D texture. Because the phong material lacks modeling of the material's microstructure, such as fine concave and convexity, it may not be realistic in these situations. For better handling of rough surfaces and microscopic details. On the contrary, some advanced texture effects, such as normal mapping, bump mapping, displacement mapping, can be achieved by the precise texture coordinate mapping provided by UV unwrapping. This process ensures that each point on the model's surface corresponds with a position on the plane (Fig.8). Once the map is ready, we apply it to the model's surface, ensuring that it matches the UV coordinates of the model for accurate display [10].

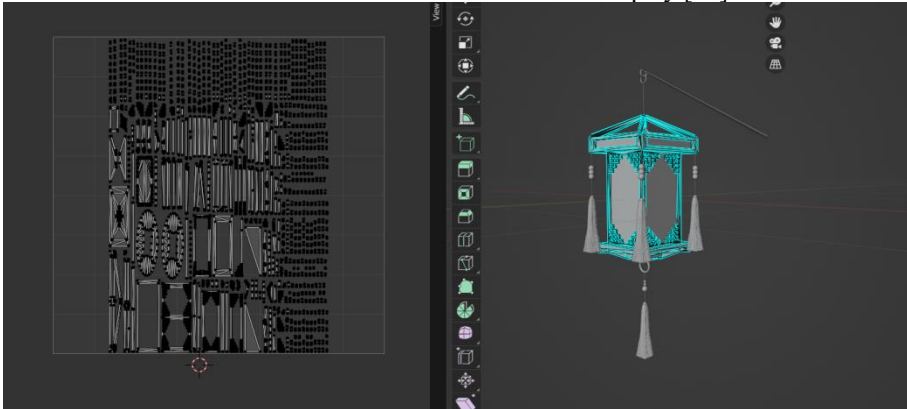


Fig. 8. Use Blender to open the model by material (right) and UV mapping of partial models (left) (Original)

4.1 Material mapping

This paper adopts two methods to make material mapping for different parts of the model. For flat and patterned lampshades, use picture materials directly for different properties, as shown in Fig 9. The lampshade consists of four posters. The color map (left 1) is used to define the basic color of the model; ambient occlusion (left 2) is used to simulate the masking effect of areas where light cannot enter; normal mapping (left 3) is used to add detail and bumpiness to the model; the roughness map (left 4) is used to define the smoothness of the model surface.

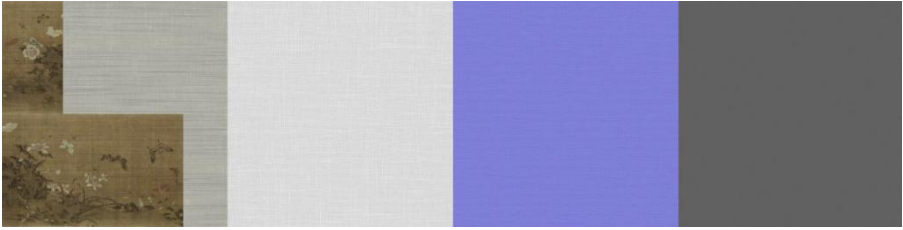


Fig. 9. Four pictures of texture of the lampshade [11]

Using Blender's material editor adjusts the effect of each node, resulting in a more realistic lampshade model. By using high-resolution maps, people can quickly add high-quality results to the model while saving time and effort in creating textures manually (Fig.10).



Fig. 10. The lampshade texture (Original)

As for dynamic and reusable materials such as lantern frames, agate decorations, metal rings, and tassels, these areas are mapped with programmed textures. More flexible results are achieved by superimposing the programmed texture nodes built into Blender and adjusting their parameters (Fig.11). They are not limited to map size, can meet the real-time customization of high-freedom requirements.



Fig. 11. Materials created using programmed maps (Original)

Take the metal rings used repeatedly in lanterns for example (Fig.12). Initially, the color ramp node is utilized to establish the fundamental color for the metal ring. Subsequently, the mapping node facilitates the adjustment of properties such as noise texture, Voronoi texture translation, rotation, and scaling. Concurrently, overlay nodes come into play, merging these diverse textures to generate novel effects and thereby determining the roughness of the model. Ultimately, the Ambient Occlusion node is harnessed to introduce darker hues into the vicinity of the two metal rings, effectively simulating diminished lighting due to object proximity. By combining those different techniques, an intricately detailed metal material can be attained.

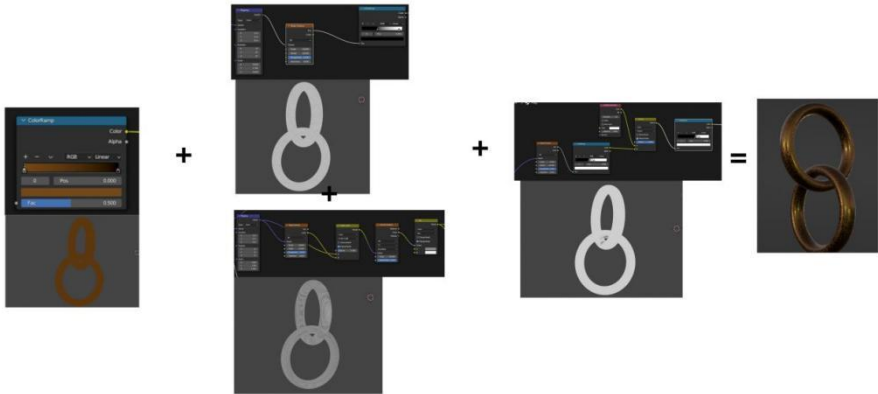


Fig. 12. Metal ring material created by using programmed maps (Original)

4.2 Animation

In order to simulate the candle wobble in real environment, animation is added to realize it. Because candles are placed in a translucent lampshade, this paper does not use a relatively more complex particle system to create a flame. Using the Displace modifier, a native tool in Blender that allows users to manually enhance the intricate details of their models. This feature empowers creators to manipulate vertices and create more complex surfaces, adding depth and realism to their designs [12]. By adjusting the parameters, the irregularity of the flame details can be achieved. At the same time, an empty object is used as the coordinate to influence the surface of the distorted texture model. Entering #frame on the Z-axis in the transformation attribute of an empty object, which is a special symbol representing the current number of frames that can be used to create continuous animation effects. Therefore, the candle can rely on empty objects to automatically move the flame without manually changing the details of each frame (Fig.13).

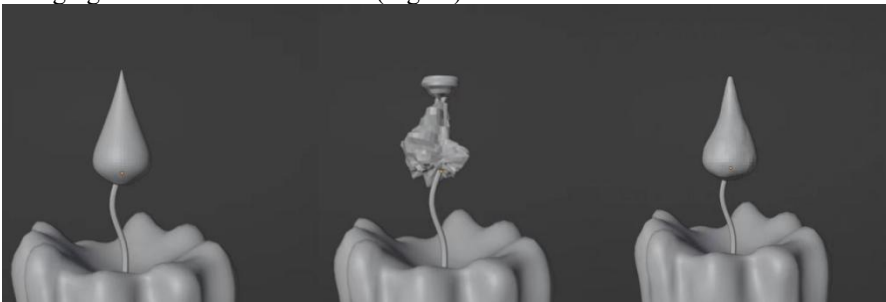


Fig. 13. Flame model (first left) Add Displace modifier (second left) Adjust appropriate parameters (third left) (Original)

4.3 Illumination

Candlelight in the lantern is also done through the Material Editor in Blender. To simulate the color of the light and the transparency of the fire's bottom, use Gradient Texture's black-and-white gradients and use Color Ramp to enhance contrast. Finally, the node is connected to Alpha channel in Bidirectional Scattering Distribution Function Shader, the default shader for materials in Blender, and the black part of color ramp node is transformed into transparent effect (Fig.14).

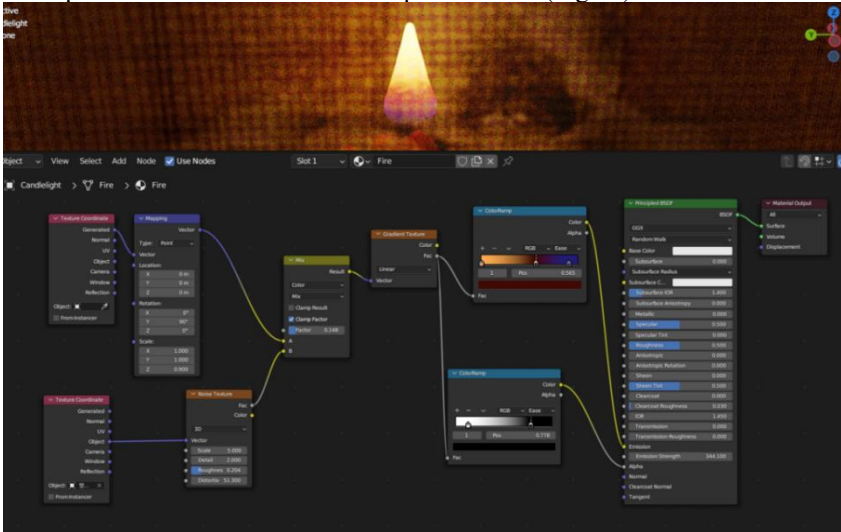


Fig. 14. Candlelight material nodes and effects (Original)

Candlelight alone does not simulate a real environment, so adding more extra light sources can provide a more balanced lighting effect throughout the scene. It also directs the audience's attention and highlights the details of the model (Fig.15).



Fig. 15. The result of light improvement (before improvement left, after improvement right) (Original)

5 Result and Discussion

We have created lantern scenarios on our website using Three.js, which can be accessed on different browsers. Visitors can interact with the lanterns by using the mouse-drag or allowing the lights to auto-rotate. For Blender, we have rendered images and videos for viewers to enjoy. In order to test whether the lanterns constructed in two different ways can convey Chinese lantern aesthetics to the audience, Table 1 and Table 2 were designed and 30 people were invited to evaluate the lantern aesthetics. These include four metrics: appearance, model details, rendering effects, and experience. The evaluation is categorized into four grades: excellent, good, general, and poor.

Table 1. evaluation results for OpenSCAD and Three.js process

Grade	Appearance	Model Details	Rendering Effects	Experience
Excellent	2	5	3	12
Good	11	14	9	10
General	14	10	12	6
Poor	3	1	6	2

Table 2. evaluation results for Blender process

Grade	Appearance	Model Details	Rendering Effects	Experience
Excellent	14	7	16	2
Good	10	16	7	6
General	4	4	6	17
Poor	2	3	1	5

After conducting a thorough analysis utilizing OpenSCAD and Three.JS, we discovered that lantern models created through Three.JS exhibited higher user experience scores, but lower ratings in terms of appearance and rendering compared to those constructed via Blender. An interesting finding was that the higher the Blender ratings for appearance and rendering, the lower the user experience ratings. However, regarding the model's details, there was minimal differentiation between the two methods, with the majority of respondents providing a rating of Good. This difference may be due to the fact that OpenSCAD may be more suitable for engineering models. Meanwhile, Three.JS web development not only allows users to observe models, but also interact more with viewers; On the other hand, Blender is better suited for creating artistic and visual models, focusing on appearance and rendering.

6. Conclusion

The purpose of this article is to exhibit 3D representations of traditional Chinese lanterns. OpenSCAD was utilized to construct models, while Three.js was used to

generate renders. Additionally, a similar set of processes is done again using Blender. Finally, we conducted a comparison between the two different processes to evaluate the outcomes. Our findings revealed that Blender produced superior renders, albeit with slower processing speeds. In contrast, Three.js provided better interactivity for model presentations. However, individuals without programming or web development experience may require additional time to learn how to create desired interactive effects using Three.js.

In the future development, along with technological progress, the 3D graphic display will have more innovations and breakthroughs. Parameterized modeling software can benefit from the addition of more modeling operations to facilitate the construction of intricate models. Rendering software can also focus on optimizing rendering time and improving efficiency. Through this article, it is also hoped that the relevant fields can be used for modeling and rendering to explore more widely.

References

1. Kopel, M., & Hajas, T. (2018). Implementing AI for non-player characters in 3D video games. In *Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018, Proceedings, Part I 10* (pp. 610-619). Springer International Publishing.
2. Scopigno, R., Callieri, M., Cignoni, P., Corsini, M., Dellepiane, M., Ponchio, F., & Ranzuglia, G. (2011). 3D models for cultural heritage: Beyond plain visualization. *Computer*, 44(7), 48-55.
3. Erolin, C. (2019). Interactive 3D digital models for anatomy and medical education. *Biomedical Visualisation: Volume 2*, 1-16.
4. Machado, F., Malpica, N., & Borromeo, S. (2019). Parametric CAD modeling for open source scientific hardware: Comparing OpenSCAD and FreeCAD Python scripts. *Plos one*, 14(12), e0225795.
5. Siu, A. F., Kim, S., Miele, J. A., & Follmer, S. (2019, October). shapeCAD: An accessible 3D modelling workflow for the blind and visually-impaired via 2.5 D shape displays. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 342-354).
6. Bergman, L. (2020). Creating an Appealing 3D-visualization Tool for Baseboards in the Web Browser.
7. <https://github.com/Nesh108/babylon-voxel-skybox/tree/master>
8. Nilsiam, Y., & Pearce, J. M. (2017). Free and open source 3-D model customizer for websites to democratize design with OpenSCAD. *Designs*, 1(1), 5.
9. Dovramadjiev, T. (2015). Modern accessible application of the system blender in 3d design practice. *International scientific on-line journal " Science & Technologies". Publishing House " Union of Scientists-Stara Zagora"*, ISSN, 1314-4111.
10. Verhoeven, G. J. (2017). Computer graphics meets image fusion: The power of texture baking to simultaneously visualise 3D surface features and colour. *Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 295-302.
11. Yanyan, Grass with Insect and Butterflies, Southern Song.
12. Gamarra, R. S., Moraes, C. A. D. C., da Rosa, E. L. S., Seelaus, R., da Silva, J. V. L., & Dib, L. (2020). Multiresolution and displacement: modifier attribution to enhance realistic 3D photogrammetry for models of the face. *Biomed J Sci Technol Res*, 29(5), 22812-7.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

