E-ISSN 2586-9027
Homepage : *https://tci-thaijo.org/index.php/SciTechAsia* Science & Technology Asia
Vol.30 No.4 October - December 2025 Page: [323-332]
Original research article

Innovative Techniques for 3D Model Optimization in the Metaverse

Aphinun Rungsoongnern[1], Suwich Tirakoat[1,*], Manasawee Kaenampornpan[2]

[1]*Department of Creative Media, Faculty of Informatics, Mahasarakham University, Maha Sarakham 44150, Thailand*
[2]*Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand*

Received 30 March 2025; Received in revised form 6 August 2025
Accepted 13 August 2025; Available online 17 December 2025

ABSTRACT

3D models are essential components in creating a realistic metaverse and providing a good user experience. However, the large file sizes of 3D models often lead to challenges in storage, transmission, and real-time rendering. Therefore, reducing the size of 3D model files during the production process is a key solution to these problems. This research aims to (1) explore methods to optimize 3D models and propose add-ons for 3D modeling software, and (2) compare the effectiveness of these optimization methods. The researchers selected three optimization techniques to evaluate their performance based on key indicators including file size (MB), total vertices, unique materials, batches/draw calls, materials count, and memory usage (KB). The analysis revealed that the Create UV Map Technique combined with Texture Atlas by Merging Material Types into a Single Unit was the most effective method. This technique was then implemented to develop a Blender add-on. The add-on's performance was tested and compared against models created without it using comparative factors such as file size (KB), mesh vertices, unique materials, batches/draw calls, materials count, and materials memory usage (KB). The comparison showed that models using the add-on had significantly better performance with statistical significance at the 0.01 level.

Keywords: 3D models; Metaverse; Real-time rendering; Texture atlas; UV mapping

1. Introduction

At present, the metaverse development is growing, creating a digital world that is interconnected and filled with virtual environments in which people can work, play, and learn. These virtual worlds are

*Corresponding author: suwich.t@msu.ac.th

doi: 10.14456/scitechasia.2025.87

only possible because of 3D models that help to create realistic environments and objects [1]. Hence there is a great demand for 3D models that are to be used in the metaverse applications [2]. However, models that are created by beginner level software users for metaverse applications are not optimized for real-time rendering. This can lead to performance issues especially when trying to render large scenes that are made of many objects.

The process of 3D model creation is fundamentally similar to game model creation which includes shape creation, material selection, shader settings and texture application [3]. However, this leads to the use of more textures and materials in 3D models that are to be used in the metaverse, resulting in larger file sizes. This can cause problems in storage, transmission and rendering, especially for users with limited resource enabled devices [4].

In order to optimize 3D models for the metaverse, there are several methods that can be used which is very important as it will help to improve the performance and user experience in the metaverse as the demand for it increases. To meet these challenges and the growing need for optimized 3D models [5], the researchers developed improved texture and material management of 3D models and developed addons for Blender software for metaverse applications. The purpose is to decrease file sizes so that metaverse rendering is faster, through real-time rendering. Also, the objectives are to optimize processes, minimize production steps, and control the graphic styles of 3D models generated by novice modelers [6].

2. Methodology

2.1 Research design

The data collection process involved a comprehensive review of existing literature and experimental results related to texture and material management for 3D models in metaverse applications. Studies by Rungsoongnern and Chaiyasit [7] demonstrated that the use of UV Mapping techniques combined with texture painting allows detailed texture application directly onto 3D models, enhancing realism especially for complex topologies. Conversely, research by Rantakangas [8] supports the use of Texture Atlas, which consolidates multiple textures and colors into a single image, reducing file size and draw calls, thereby improving rendering performance.

Additionally, these approaches were experimentally evaluated by applying them to 3D building models in Unity Game Engine, where performance metrics such as file size, mesh vertices, unique materials, batches (draw calls), materials count, and memory usage were measured using the Unity Profiler tool [9]. The synthesis of these findings indicates that combining UV Mapping with Texture Atlas via material classification or merging material types into a single unit significantly enhances texture and material management efficiency. While some studies highlight advantages such as improved rendering speed and file size reduction, others note challenges including increased complexity in texture atlas creation or time-consuming texture painting processes. Overall, the consensus favors the combined use of these techniques to optimize 3D model rendering for metaverse environments. In this research, the optimization of the texture and material management was divided into three approaches, each based on the specific advantages of the techniques under study:

2.1.1 Using the create UV map technique combined with texture painting

This method entails generating UV maps for 3D models and then texturing by painting directly onto the images that are to be applied on the 3D model. The process starts with generation of the UV map for the 3D model, then comes the design and detailing of the texture through painting. Finally, the textured 3D model is exported and can be used in the metaverse.

The use of texture painting gives the user the ability to paint details directly on the 3D model, giving a good level of control when it comes to complex designs. Because it enables the application of several textures to a single 3D model, it is particularly helpful for models with complex topologies.

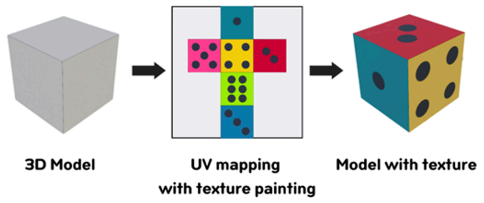

Fig. 1. Result of 3D model after applying the UV mapping with texture painting.

In common practice, texture sizes for 3D models typically range from 512×512 pixels to 2048×2048 pixels, depending on the level of detail needed. When higher-resolution details are required—such as for close-up views or final renders—larger texture sizes (e.g., 4096x4096 pixels or greater) are necessary. Using larger textures improves the visual quality by providing finer detail and reducing pixelation during close inspection. However, this also results in significantly larger texture file sizes, which directly increase the overall 3D model file size. In the context of the metaverse, larger texture files demand more memory and processing resources during rendering, potentially impacting performance and causing slower load times or frame rate drops, especially on devices with limited capabilities. Therefore, balancing texture size and detail is critical to maintain an optimal viewing experience while ensuring efficient real-time rendering.

However, there are some disadvantages of texture painting. The process is more time-consuming than conventional texture mapping because painting and using complicated functions of the 3D program is required.

2.1.2 Create UV map technique combined with texture atlas by material classification

In this approach, a UV map of a 3D model is created and multiple textures are combined into a single texture atlas arranged in a grid based on material classification. The researchers used a monochromatic texture pattern with minimal detail to keep the process simple. The texture atlas serves as a color palette, allowing users to select colors for their 3D models. Texture properties are assigned by selecting the faces of the 3D model, performing UV mapping, and applying colors directly onto the faces.

This approach reduces both the storage file size and the overall file size of the 3D model by consolidating multiple textures into a single texture atlas based on material classification. By minimizing redundancy in texture data, this method enhances rendering efficiency and reduces the number of texture files that need to be managed during real-time rendering.

Furthermore, this technique simplifies texture management because changes to texture size can be made more easily without modifying individual UV maps. As a result, the method effectively decreases file size and resource consumption, leading to faster rendering times, as demonstrated in Section 3. Some other advantages of using a texture atlas include improved performance due to fewer draw calls during rendering, which is crucial for real-time applications such as metaverse environments.

However, creating and editing texture atlases can be more complex compared to conventional UV mapping techniques. Despite this, the method remains user-friendly and suitable for beginners due to its simplified color selection and material classification process.

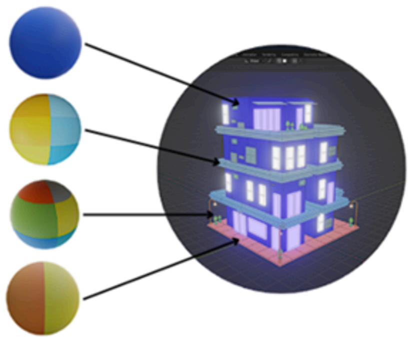
Fig. 2. Applying the Create UV map technique in association with Texture Atlas by distinguishing material types.

2.1.3 Create UV map technique combined with texture atlas by merging material types into a single unit

This method is quite similar to Method 2 but the difference between the two methods is the organization of the material configurations. In this approach, the different types of materials are combined into one file and this produces a single material setup that can be used on the entire model. This method is more efficient than Method 2 because combining different material types into one reduces file size and reduces the number of draw calls significantly. However, due to the fact that creating the texture atlas is quite complex, this process may be quite time-consuming and difficult to edit. However, it is user-friendly and ideal for beginners as a result of these characteristics.

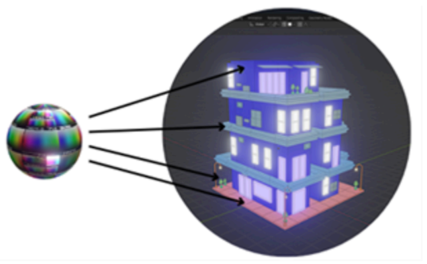
Fig. 3. Using the Create UV map technique in conjunction with Texture Atlas by combining material types.

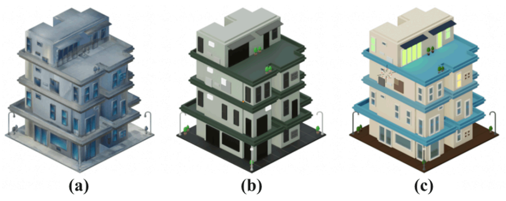

Fig. 4. The model which has been optimized in terms of texture and material is shown divided into three approaches.

In this study, 3D models were created and imported into the Unity Game Engine. These 3D models were created and then the three methods of optimizing the texture and material management that were previously discussed were applied to them. The Unity Game Engine's Profiler was used to collect rendering tests and performance data by rendering 10,000 building models. The analysis was made on performance indicators such as file size, mesh vertices, unique materials, batches/draw calls, material count, and memory usage for materials. The Unity Profiler was used as the main tool for data collection [10]. The results are presented in Section 3.1.

2.2 Development of blender add-on

Based on the performance comparison shown in Table 1, the ”Create UV Map Technique Combined with Texture Atlas by Merging Material Types into a Single Unit was selected as the most effective method for optimizing texture and material management in 3D models. This method demonstrated the lowest values across all key performance indicators, including file size, total vertices, unique materials, draw calls, material count, and memory usage, making it ideal for real-time rendering in metaverse applications.

The chosen method was implemented as a Blender add-on to facilitate efficient 3D model optimization. Blender was selected because it is a widely-used, open-source 3D modeling software that supports extensibility through Python scripting, making it accessible to both beginners and professionals. Furthermore, Blender’s compatibility with various game engines and real-time rendering workflows makes it an appropriate platform for metaverse content creation.

The process of designing and developing consists of those steps.

1) The researchers used a Texture Atlas to assign colors to help the user pick colors for their 3D models. Two color set sizes were developed: 64×64 pixels for materials that need a large number of colors and 32×32 pixels for materials that need fewer colors. The textures were created with power-of-two dimensions (for example, 128×128, 256×256) because GPUs are designed to work better with such images. This optimization enhances the processing and memory storage, which means that data can be managed and retrieved from the storage device easily and quickly [11]. The color sets were then separated into different categories and then all of them were placed in one image in power of two for use in defining material properties of 3D models in game engines.

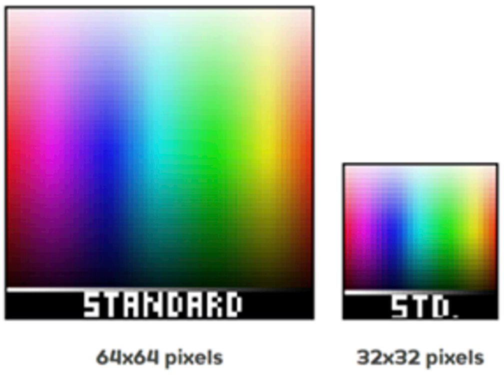

Fig. 5. Two sizes of Color palettes for users to select colors for their 3D artwork.

2) The material properties were assigned by the researchers using an image-based approach in the Unity game engine. This method used RGBA (Red, Green, Blue, and Alpha) numerical values that used the intensity of the color channels as variables to determine the intensity of the material’s properties (see Fig. 6.).

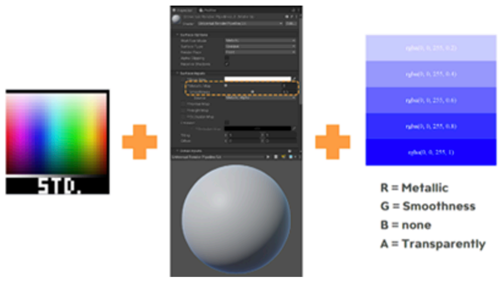

Fig. 6. Material properties are defined using RGBA values.

The red channel is the Metallic map (metallic properties). The green channel is the Smoothness (the level of smoothness).

The blue channel has no value.

The Alpha channel is the Transparency (the level of transparency).

When the RGBA values are combined, different material patterns can be produced to accurately define material properties. For example, using an image with a deep red color will give a high metallic property of the material and using an image with a light green color will give a high level of smoothness in the texture (see Fig. 7.).

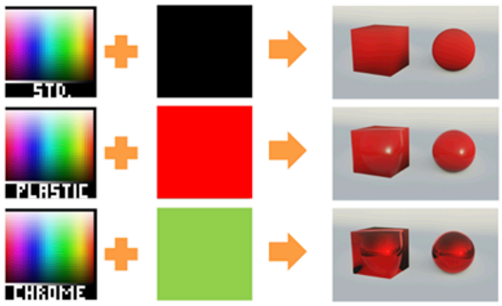

Fig. 7. Palette colors and RGBA values are used to define material properties.

3) Apply the Palette color and RGBA values to generate a texture image that can be applied on top of 3D models (see Fig. 8.).

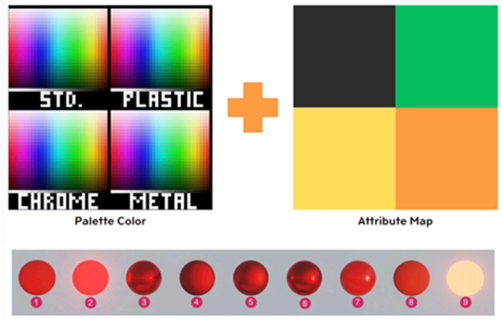

Fig. 8. Image texture map for application to 3D model.

The data collected from the study was used in developing texture maps to be used on 3D models, and the materials were sorted into seven groups: Matte, Metal, Gloss, Chrome, Semi-gloss, Plastic, and Glow (see Fig. 9.).

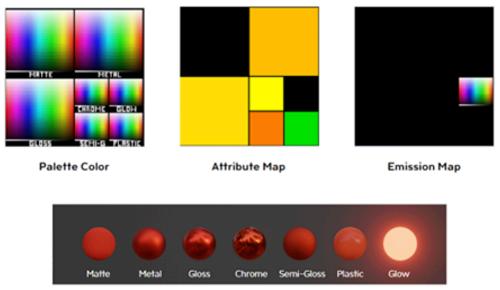

Fig. 9. Textures made for 3D models and divided into 7 material types.

4) Develop a set of materials for Unity and Blender programs. Next, code a Blender add-on in Python using the built-in Text Editor in the Blender software. The installation file of the Blender add-on will then be prepared and distributed to the sample subjects for installation and experimental use.

2.3 Data collection

A total of 44 undergraduate students were selected as participants for this study. These students were enrolled in courses related to 3D modeling and had foundational knowledge and practical experience in creating and developing 3D models. The selection of 44 participants was based on the class size available in these relevant courses.

Participants were asked to work with provided 3D building model files and apply textures and materials to these models. Initially, they performed this task without using any optimization add-ons to establish a baseline. Subsequently, they repeated the same texturing and material assignment tasks using the developed Blender add-on.

This approach allowed the collection of paired data sets (with and without the add-on) for each participant, enabling direct comparison of performance indicators such as file size, mesh vertices, the number of unique materials, batches (draw calls), count of materials, memory-use of materials, and count of objects that can be rendered in the Unity game engine at a target frame rate of 60.

2.4 Data analysis

The data from the tests were analyzed to evaluate the performance of created add-on. Statistical methods like mean, standard deviation, and t-test were used to analyze the performance related to two groups of model files. The first group uses the add-on and the second group does not use it. The outcome related to texture is presented in Section 3.2.

3. Results

3.1 Results of the comparison of optimization methods for texture and material management in 3D models for metaverse applications

The data obtained from rendering tests of 10,000 building models in Unity were analyzed to evaluate performance metrics. The findings were summarized to determine the best method for real-time processing in the context of the metaverse. The results are presented in Table 1, as follows.

From Table 1, it was revealed that the Create UV Map Technique Combined with Texture Atlas by Merging Material Types into a Single Unit method had the lowest mean values across all the performance indicators used for comparison. Such indicators are file size, total vertices, unique materials, batches/draw calls, materials count and memory use. On the other hand, other methods like Create UV Map Technique Combined with Texture Atlas by Material Classification and Using the Create UV Map Technique Combined with Texture Painting had higher means in the succeeding ranks. The analysis results show that the Create UV Map Technique Combined with Texture Atlas by Merging Material Types into a Single Unit method is the best in the management of textures and materials in a 3D model. The lowest values in all the indicators show that this method is suitable for use in the metaverse as it reduces the processing load and thus improves the rendering time.

3.2 Evaluation of the performance of the add-on developed with the create UV map technique combined with texture atlas by merging material types into a single unit

The performance was compared between the sample group who used the add-on and the group who did not use the add-on. The results are in Table 2.

From Table 2, it is evident that the variables, including file size, mesh vertices, unique materials, batches/draw calls, materials count, and materials memory use, exhibited statistically significant differences at the.01 level between the sample group using the addon and the group not using the addon. The values of these variables were significantly lower in the group using the addon, indicating that the use of the addon effectively reduces computer resource usage for real-time processing.

4. Conclusions and Discussion

This research aimed at identifying and applying different approaches to create a tool that can assist beginner users in the creation and development of 3D models for the metaverse. The findings show that various methods influence the file size and performance of the produced 3D models. It is important to select the appropriate technique as this will improve the rendering of 3D models in the metaverse, thus enhancing

Table 1. Comparison Table for Rendering Tests and Performance Data.

Performance Indicators	Using the Create UV Map Technique Combined with Texture Painting (mean ± S.D.)	Create UV Map Technique Combined with Texture Atlas by Material Classification (mean ± S.D.)	Create UV Map Technique Combined with Texture Atlas by Merging Material Types into a Single Unit (mean ± S.D.)
File size (MB)	1205.75 ± 907.74	1076.55 ± 823.31	1044.93 ± 782.32
Total Vertices	1350.25 ± 950.50	1225.15 ± 800.10	1180.80 ± 740.55
Unique Materials	52.10 ± 39.00	40.70 ± 30.50	30.20 ± 20.80
Batches/Draw Calls	150.32 ± 112.83	130.45 ± 90.25	110.10 ± 70.40
Materials Count	200.60 ± 150.42	180.80 ± 120.67	160.45 ± 110.23
Memory Usage (KB)	3040.45 ± 2100.32	2800.20 ± 2000.20	2500.35 ± 1800.10

Table 2. Evaluation of the performance of the add-on developed with the create UV map technique combined with texture atlas by merging material types into a single unit.

Comparative factors	Without add-ons		With add-ons		t	P-value
	(mean)	(S.D.)	(mean)	(S.D.)		
File size (KB)	142.45	14.69	122.50	9.67	14.47**	.000
Mesh vertices	16,424.95	443.60	14,165.32	319	25.85**	.000
Unique materials	18.82	7.91	1.05	.21	14.90**	.000
Batches/Draw Calls	41.64	15.83	6.50	2.73	14.34**	.000
Materials count	164.23	7.53	146.00	.22	15.92**	.000
Materials memory use (KB)	425.99	53.30	288.52	1.19	17.01**	.000

** Statistical significance level at .01

the user experience in virtual worlds.

The selection of the textures and materials is an important factor in the construction of realistic 3D models. The consequence of employing the same set of textures and materials is that a number of problems may be encountered including the size of the file which has implications for the cost of storage, the speed of data transfer and the quality of the render. These models also require more memory and computing power to run, which in the long run affects the overall performance. This finding is in agreement with the findings of Doungu-tha [4] and Kang and Park [5].

The researchers engaged in a study to determine the best ways of managing textures and materials in 3D models. Through the analysis of the performance test results, the traditional methods and new approaches were compared and the method of Create UV Map with Texture Atlas was adopted. This technique was found to be the most effective and easiest to use in the management of textures in 3D models. It assists entry level metaverse developers in generating the required assets for the metaverse business. This finding is in agreement with the study by Hristov & Kinaneva [3] who explained building a structural map of a 3D model in 2D and then using this map to design patterns. Such an approach is in concurrence with the work of Rungsoongnern & Chaiyasit [7] who explained the use of Create UV Map in conjunction with pattern creation to generate several texture files. However, these methods need some level of artwork to create the patterns, which may complicate the process.

To address these challenges, the researchers adopted the Texture Atlas technique by specifying the color sets that are to be used in the various graphical applications. The glossiness and the roughness values were integrated into the red (R) and green (G) color channels to generate different material types from a single texture. This approach is in line with the study by Rantakangas [8] which pointed out that combining the use of several texture types into one file is more efficient. Game engines only request materials once, thus optimizing the use of resources. Moreover, the performance of game engine add-ons developed using this method showed that there was a reduction in the consumption of processing resources, which in turn increased the rendering speed. This finding is in agreement with previous studies by Lohikoski-Håkansson & Rudén [9] and Koulaxidis & Xinogalos [10].

This paper proposed a way of creating tools that can help beginners to easily develop 3D models for the metaverse. The focus was made on the technique of UV map creation combined with texture atlases, where the material types were combined into a single entity in order to enhance the rendering of 3D models.

There are some limitations in this research, however. These include the fact that the study was conducted with a small number of participants and that only certain types of 3D models are supported. However, there is a possibility of improvement in the future. The study also recommends the continuation of the study to identify and enhance the methods in order to support a more extensive range of 3D model types as well as to integrate new features with the help of AI technologies to improve the available tools and workflows for creative activities in the metaverse. For example, features like color pickers that are easy to use and functions that suggest coordinates of colors that go well together can be created.

The findings of this study can lead to the creation of new tools and methods that would help beginners to produce 3D models better and faster and, therefore, contribute to the development of 3D design for the metaverse.

Acknowledgements

This research project was financially supported by Mahasarakham University.

References

[1] Techavimol P, Walsh J. Perceived benefits gained from online game playing among university students in Bangkok. Sci Tech Asia. 2011;16(2):54–65.

[2] Jarutkamolpong S, Nankhantee A, Mikhama K. Development of a metaverse model to enhance Renu Nakhon Phu-Thai cultural tourism in Nakhon Phanom Province, Thailand. Humanit Arts Soc Sci Stud. 2024;24(3):662–72.

[3] Hristov G, Kinaneva D. A workflow for developing game assets for video games. In: Proceedings of the 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA); 2021; Ankara, Turkey. p. 1–5.

[4] Doungu-tha C. Low-cost 3D scanner for creating virtual game scenes: a case study of developing virtual Kamphaeng Phet Historical Park scene. DEC J. 2022;1(2):11–34.

[5] Kang KK, Park CJ. Lightweight method with controllable appearance details for 3D reconstructed building models. In: 2022 13th International Conference on Information and Communication Technology Convergence (ICTC); 2022 Oct 17–19; Jeju, Korea. IEEE; 2022. p. 2034–6.

[6] Thongsaen Y, Thianmongkol R. Storing ancient Buddha images in the form of 3D virtual objects with the photogrammetry technique for conservation. Surin Rajabhat Univ J Ind Technol. 2023;8(1):1–16.

[7] Rungsoongnern P, Chaiyasit S. The designs and development of 3D fantasy model for study the composition in the old town of Nakhon Ratchasima. J Fine Arts Chiang Mai Univ. 2022;13(1):144–68.

[8] Rantakangas P. Comparing medium poly workflow with traditional workflow [Bachelor's thesis]. Tampere (Finland): Tampere University of Applied Sciences (TAMK); 2020.

[9] Lohikoski-Håkansson L, Rudén E. Optimization of 3D game models: a qualitative research study in Unreal Development Kit [Bachelor's thesis]. Södertörn (Sweden): Södertörn University; 2013.

[10] Koulaxidis G, Xinogalos S. Improving mobile game performance with basic optimization techniques in Unity. Modelling. 2022;3(2):201–23.

[11] Shreiner D, Sellers G, Kessenich J, Licea-Kane B. OpenGL programming guide: the official guide to learning OpenGL, versions 4.5 with SPIR-V. 9th ed. Addison-Wesley Professional; 2016.