International Journal of Information Engineering and Science

E-ISSN: 3048-1953
P-ISSN: 3048-1902

Research Article

Comparative Investigation of Activity Rendering Utilizing Eevee, Cycles, and Radeon ProRender Procedures in Blender Applications

Ira Zulfa [1]*, Richasanty Septima [2], Iryana Rezeki [3], Rayuwati [4]

[1-4] Universitas Gajah Putih, Indonesia
* Corresponding Author: e-mail: ira.zulfaa@gmail.com

Abstract: The rapid development of multimedia technology has significantly advanced 3D animation techniques, enabling the production of high-quality visual content across industries such as film, gaming, architecture, and product visualization. Rendering, as the final stage of the 3D production pipeline, plays a crucial role in determining both visual realism and production efficiency. This study compares the performance of three rendering engines—Eevee, Cycles, and Radeon ProRender—by evaluating rendering speed, visual quality, and memory efficiency in Blender. The objective is to provide practical insights for designers and digital content creators in selecting the most suitable rendering engine based on project requirements. In this research, three identical 3D scenes were rendered using each of the three rendering engines under controlled experimental conditions. The comparison was conducted based on several parameters, including rendering time, output file size, shadow accuracy, lighting effects, and overall visual realism. Quantitative measurements were used to evaluate render speed and memory consumption, while qualitative analysis assessed differences in shadow detail, global illumination behavior, reflection accuracy, and material realism. The results indicate that Eevee outperforms the other engines in terms of rendering speed, making it highly suitable for real-time applications and projects requiring fast previews. Cycles produces the highest level of visual realism due to its physically based path-tracing algorithm, although it requires longer rendering time and higher computational resources. Meanwhile, Radeon ProRender demonstrates competitive performance, particularly in shadow quality and lighting effects, offering a balanced alternative between realism and efficiency. Based on the findings, Blender remains a flexible and effective platform. The choice of rendering engine should depend on whether speed, graphic quality, or memory optimization is prioritized.

Keywords: 3D Rendering; Blender, Eevee; Cycles; Radeon ProRender

Received: December 28, 2024
Revised: January 12, 2025
Accepted: January 26, 2025
Published: January 28, 2025
Curr. Ver.: January 28, 2025

1. Introduction

Interactive media technology, particularly three-dimensional (3D) design and animation, has become a major focus in the advancement of information technology. 3D animation is widely applied in fields such as film production, game development, architecture visualization, simulation, and digital marketing (Ganga Rama Koteswara Rao et al., 2021). One of the most critical stages in the 3D production pipeline is rendering, which converts 3D models into realistic 2D images or animations. Render quality and render speed are two fundamental factors that determine project efficiency and final visual output. High-quality rendering often requires substantial computational time and hardware resources, which may slow down production workflows and increase operational costs. Conversely, low-quality rendering may reduce realism and negatively affect the overall project value.

Currently, Blender provides two primary rendering engines, Eevee and Cycles. Eevee is a real-time rendering engine optimized for speed and interactive previews, while Cycles is a physically based path-tracing engine designed to produce highly realistic images. However,

both engines are limited to the Blender ecosystem. In contrast, Radeon ProRender offers broader compatibility, as it can be integrated with various popular 3D software platforms (S. Dudek et al., 2022). Each rendering engine presents different strengths and weaknesses in terms of visual realism, rendering performance, memory consumption, hardware optimization, licensing flexibility, and suitability for specific production timelines.

Rendering performance and visual realism are strongly influenced by the rendering algorithm and hardware acceleration used (Zhang et al., 2021). Real-time engines such as rasterization-based renderers provide faster output but may sacrifice physical accuracy (Kim & Lee, 2022). In contrast, physically based path-tracing engines generate more realistic global illumination and shadow behavior at the expense of longer computation time (Patel & Singh, 2023). GPU optimization has also been shown to significantly improve rendering efficiency in complex 3D scenes (Almeida et al., 2022). Memory consumption remains a critical factor in high-resolution rendering workflows (Huang et al., 2024). Comparative studies further emphasize the importance of selecting rendering engines based on project-specific requirements such as speed, realism, and hardware constraints (Rossi & Conti, 2020).

Despite the availability of multiple rendering engines, selecting the most appropriate engine remains challenging for designers and studios. Differences in speed, memory efficiency, shadow accuracy, lighting realism, and material representation can significantly influence production outcomes. Therefore, the research problem addressed in this study is how Eevee, Cycles, and Radeon ProRender compare in terms of rendering speed, memory efficiency, and visual realism under similar testing conditions.

2. Literature Review

2.1. Graphic

Rendering is a process that uses a computer program to create realistic or non-realistic images that show two or three dimensions. The result of this rendering process is called render. Hasil Render can be either a raster graphic or an advanced graphic. The rendering process is usually carried out by converting the output of an image, video, music, text, or other object into a single format. Twelve Rendering is typically used to create complex visuals, test systems, films, animations, video diversions, and architectural designs. Techniques and features used in rendering for the project being worked on. (Alfain et. al. 2022)

2.2 Animation

Animation itself comes from the Latin word "anima," which means "jiwa," "hidup," and "semangat." On the other hand, it refers to people, objects, or other things that are depicted in 2D or 3D images. Consequently, the character's animation. As a result of the collection of images being displayed regularly and alternately, it can be interpreted as a picture that depicts all objects that are part of everyday life. The objects in a picture can be written words, object shapes, words, or special effects. According to Ibiz Fernandes, animasi is a process of drawing and returning to a series of statistical illustrations to obtain an illusion. Based on the aforementioned article, animation is a form of entertainment. That is, the business of pointing out something that cannot move on its own. (Olessia Barkovska et al. 2024).

2.3 Blender

Blender is a comprehensive 3D graphic design and content creation software widely used to produce animated films, visual effects (VFX), 3D visualizations, interactive 3D applications, architectural renderings, and video games. As an integrated 3D creation suite, Blender supports the entire production pipeline, including modeling, sculpting, texturing, rigging, animation, simulation, rendering, compositing, and video editing. This makes Blender not only a modeling tool but also a complete digital content production environment suitable for both individual creators and professional studios.

Blender is generally recognized as a free and open-source 3D creation package, which means its source code is openly available and can be modified or extended by the developer community. The open-source nature of Blender encourages continuous innovation, rapid feature development, and strong community support. Unlike many commercial 3D software packages that require expensive licenses, Blender provides full professional functionality without subscription costs, making it highly accessible for students, researchers, freelancers, and small creative industries.

Another major advantage of Blender is its cross-platform compatibility. The software can run efficiently on various operating systems, including Windows, macOS, and Linux, allowing users to work across different hardware environments without compatibility issues (Dominik Schraml et al., 2024). In addition, Blender supports GPU acceleration and multiple rendering engines such as Eevee and Cycles, enabling users to balance real-time performance and physically accurate rendering. Due to its flexibility, extensibility through add-ons, and strong global user community, Blender has become one of the most widely adopted 3D software tools in both academic and professional contexts.

2.4 The Eevee

Eevee is an acronym for Extra Easy Virtual Environment Engine, also known as Mesin Lingkungan Virtual Ekstra Mudah in Indonesian. Eevee is a rendering machine. real-time that is created using OpenGL and focuses on speed and interaction. Eevee may be used with 3D Viewport in a very subtle way and produces high-quality final renders. Material Eevee is made using a shader node that is identical to Cycles, making it simple to render any existing agenda. Unlike Cycles, Eevee does not have a raytrace renderer. Instead of calculating every ray of light, Eevee simulates how light interacts with objects and materials using various algorithms. On the other hand, Eevee is designed to follow Physically Based Rendering (PBR) principles. (Md Saikat Hosen et al. 2019).

2.5 The Frames

Cycles is a ray-tracing Blender-based render production tool designed to produce realistic and high-quality images. Cycles may simulate a scene in a very accurate manner that allows for the emergence of a visually striking effect, realistic rendering, and materials similar to those seen in Viewport 3D. Cycles is designed to provide a high level of realism and can be used with either a GPU or a CPU. Using a technique known as path tracing, a global illumination rendering method, it produces highly realistic images with accurate lighting. Cycles can generate lighting effects such as depth of field, caustics, reflections, and soft shadows, among others. (Aviv Fitria Yulia et al. 2024).

2.6 ProRender for Radeon

AMD Radeon™ ProRender is a physically based rendering (PBR) engine built on AMD's advanced graphics architecture, designed to maximize both GPU and CPU performance for high-quality visual production. It enables creative professionals to generate photorealistic images by simulating accurate light behavior, reflections, refractions, and material properties. By leveraging modern hardware acceleration and optimized rendering algorithms, Radeon™ ProRender is capable of producing realistic visual outputs efficiently, making it suitable for professional visualization workflows (Ika Asti Astuti et al., 2022).

As a Blender plug-in, Radeon™ ProRender is designed to be fast, flexible, and user-friendly. Its integration within Blender allows users to seamlessly switch between rendering engines while maintaining compatibility with existing 3D assets and scenes. One of its key advantages is cross-platform capability and support for multiple hardware configurations, allowing it to operate on various GPUs and CPUs without being limited to a specific vendor ecosystem. This flexibility makes it appealing for studios and independent creators who work with diverse hardware setups.

Radeon™ ProRender can be applied across a wide range of industries, including product design visualization, architectural rendering, visual effects production, animation, and industrial prototyping. Its physically accurate rendering system enhances realism in lighting, shadows, global illumination, and material textures, which are critical for professional-quality outputs. Because of its balance between performance optimization and visual fidelity, Radeon™ ProRender serves as a competitive alternative to other rendering engines available within the 3D production environment.

2.7 Modeling

3D modeling is the process of creating three-dimensional digital objects using specialized software techniques commonly referred to as 3D modeling tools. The objects generated through this process possess three spatial dimensions—length, width, and height—allowing them to visually and structurally resemble real-world objects. Unlike two-dimensional representations, 3D models provide depth and volume, enabling users to observe objects from multiple angles and perspectives. This capability makes 3D modeling essential for producing realistic visualizations and simulations in digital environments.

The 3D modeling process typically involves constructing geometric shapes such as vertices, edges, and faces that form polygonal meshes. More advanced techniques include sculpting, parametric modeling, and procedural modeling, which allow designers to create highly detailed and complex structures. Materials, textures, and lighting parameters are later applied to enhance realism before the rendering stage. Because of these capabilities, 3D modeling has become a fundamental component in modern digital production pipelines.

Today, 3D modeling plays a crucial role across various creative and industrial sectors, including video game development, animated film production, product design, architecture, engineering visualization, virtual reality (VR), and augmented reality (AR). Its ability to simulate real-world objects and environments improves design accuracy, reduces prototyping costs, and enhances visual communication between stakeholders. Furthermore, advancements in rendering and real-time graphics technology continue to expand the applications of 3D modeling in immersive media and interactive systems (Zhang et al., 2021).

2.8 Texture

Texturing is the process of applying a specific image to an object's surface so that it appears more realistic. Another way to put it is the process of identifying the object's surface characteristics. One of the characteristics is kilauan, coloring, and so on. Generally speaking, texturing can also be referred to as wrapping to the object's surface or painting, even if there is a process that alters the object's geometry. However, texture has a different art from texturing. One way to describe texture is as an actual warning image from a material that helps to explain or reassure, utilizing a process known as rasterization. Texture can be described as an actual color image of a material that helps to explain or refine. (Eka Sahputra et al. 2022)

3. Method

In this research, the author employed an experimental and analytical approach supported by quantitative research methods. The quantitative method focuses on collecting numerical data and analyzing it using statistical techniques to obtain objective and measurable results. Through this approach, variables such as rendering time, memory usage, and visual performance indicators can be systematically measured and compared under controlled experimental conditions. Quantitative analysis enables researchers to test performance differences, identify patterns, and draw statistically valid conclusions based on empirical data. By integrating experimental design with statistical evaluation, this study ensures that the comparison between rendering engines is conducted in a structured, replicable, and data-driven manner (Creswell & Creswell, 2023).

a) **Eksperimen Method(Shichen Liu et al. 2020)**
   The experimental method is carried out using a 3D renderer that has distinct characteristics. Teknik Eevee, Cycles, and Radeon ProRender are the three distinct techniques that will be used in each demonstration.
b) **Analysis of Methods(Wenzheng Chen et al. 2019)**
   Information regarding rendering time and memory usage was collected using the Summary feature available in the Blender application. Subsequently, data analysis was conducted using SPSS version 26.0 and descriptive statistical analysis. To obtain information about the quality of the rendering results, questionnaires were distributed to respondents, and the responses were analyzed using a Likert scale.

4. Results and Discussion

4.1 Rendering Result

This is the result of rendering using the Eevee, Cycles, and Radeon ProRender techniques :

a) Visualization Results of the 3D Homestay Model

The first 3D model created is a visual representation of a homestay.

1) Eevee Homestay engineering rendering

After the render process is completed, the model homestay designer first uses the Eevee technique.

Figure 1. Result of Eevee rendering of the Homestay

As can be seen from the above illustration in Figure 1, the model that was rendered using the Eevee technique produced the illustration as shown.

2) Cycles Homestay engineering rendering

After rendering is complete, the model homestay is then rendered using the Cycles technique.

Figure 2. Result of Cycle Homestay Rendering

As can be seen from the above illustration in Figure 2, the model that was rendered using the Cycles technique produced the illustration that is shown.

3) Radeon ProRender Homestay engineering rendering

After rendering is complete, Model Homestay is rendered using the Radeon ProRender technique.

Figure 3. Redeon ProRender Homestay Render Result

As can be seen from the above illustration in Figure 3, the model that was rendered using the Radeon technique produced the illustration that is shown.

b) Visualization Results of the 3D Camping Model

The two 3D models that are rendered are visual representations of a camping area.

1) Eevee Camping engineering rendering

After rendering is completed, the first time the model is rendered is using the Eevee technique.

Figure 4. Eevee render result of Camping

As can be seen from the above illustration in Figure 4, the model that was rendered using the Eevee technique produced the illustration as shown.

2) Cycles Camping engineering rendering

After the render process is finished, the model camping is then rendered using the Cycles technique.

Figure 5. Results of Cycles render Camping

As can be seen from the above illustration in Figure 5, the model that was rendered using the Cycles technique produced the image that is shown.

3) Rendering Technique Radeon ProRender Camping
After rendering is complete, the model Camping is rendered using the Radeon ProRender technique.

Figure 6. Result of Radeon ProRender Camping rendering

As can be seen from the above illustration in Figure 6, the model that was rendered using the Radeon technique produced the illustration that is shown.

c) Visualization Results of the 3D Cafe Model
The rendered 3D model is a visual representation of a cafe building.

1) Eevee Cafe engineering rendering
Model Cafe was first rendered using the Eevee technique after rendering was completed.

Figure 7. Result of Eevee rendering Café

As can be seen from the above illustration in Figure 7, the model that was rendered using the Eevee technique produced the illustration as shown.

2) Cycles Cafe engineering rendering
After rendering is complete, the model Cafe is then rendered using the Cycles technique.

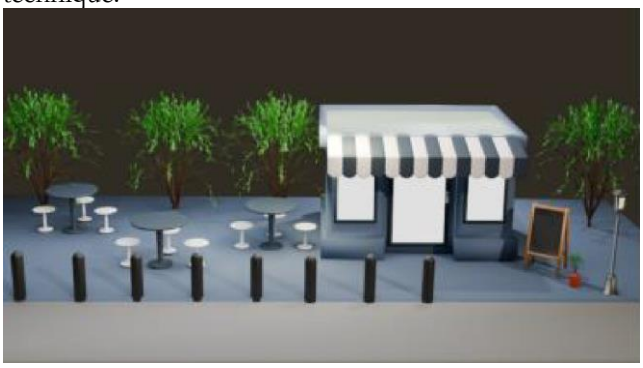

Figure 8. Cycles render the results of the Cafe

As can be seen from the above illustration, Figure 8, the model that was rendered using the Cycles technique produced the illustration that is shown.

3) Radeon ProRender Cafe Rendering
After rendering is completed, the final model of the cafe is created using the Radeon ProRender technique.

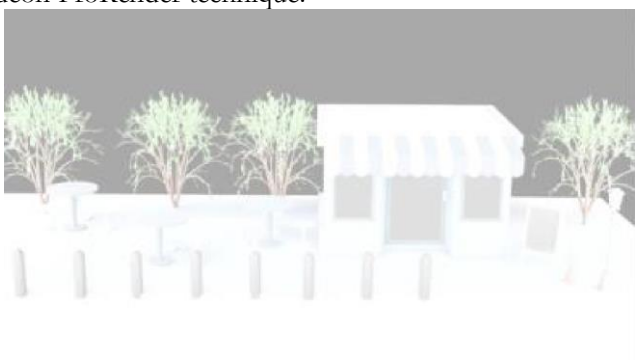

Figure 9. Result of Radeon ProRender Cafe rendering

As can be seen from the above illustration in Figure 9, the model that was rendered using the Radeon technique produced the illustration that is shown.

4.2 Research Result

Based on the results of the analysis and the testing that were completed, the following research findings were obtained :

a. Rendering Speed
The Eevee render technique shows the highest efficiency with a rata-rata render time of 12,67 second, but it's far faster than Radeon ProRender (1194,33 second) and Cycles (465,67 second). Because of this, Eevee is seen as a good choice for projects that require high speed.

b. File Size of the Rendered Output
The use of storage for rendering files from the three techniques is not significantly different. Render file sizes are roughly 2.52 MB for Eevee, 2.66 MB for Cycles, and 2.74 MB for Radeon.

c. Rendering Quality Results
- **1)** Eevee: Based on respondent questionnaires, the results of using the Eevee technique showed a mean score of 79.17% (Good). Eevee is suitable for projects that require quick rendering times with good visual quality.
- **2)** Cycles Technique: Based on the respondents' questionnaire results, the average score was 87.50% (Very Good). Cycles is considered a suitable option for projects that prioritize high visual quality, although it requires a longer rendering time.
- **3)** Radeon ProRender: Based on the respondents' questionnaire results, this technique obtained an average score of 46.67% (Fair). Compared to Eevee and Cycles, Radeon ProRender produces visual results that are relatively less satisfactory, particularly in terms of shadow rendering and lighting effects.

d. Color Accuracy, Light Effects, and Shadow Quality :
- **1)** Shadow Clarity: Compared to Radeon ProRender, Eevee, and Cycles produce shadows that are more accurate and well-defined.
- **2)** Light Effects: Radeon ProRender produces minimal lighting effects, whereas Cycles provides the most realistic lighting. Eevee also generates realistic lighting effects, although not as detailed as Cycles.
- **3)** Cycles and Eevee provide good color accuracy, whereas Radeon ProRender's color accuracy frequently deviates from the original.

Depending on the specific needs of a project in Table 1, choose between Eevee, Cycles, and Radeon ProRender. Eevee is the best option for quick prototyping or projects that emphasize speed over detail. On the other hand, even if the rendering time is longer, Cycles remains a popular choice for projects that require images with high quality. Radeon ProRender functions as a useful alternative for people who want to balance work and quality,

but it may also be necessary to adjust their work schedule to meet their needs. Understanding this difference might help the artist make an appropriate decision for their project's needs.

Table 1. Summary of Comparison

Feature	Eevee	Cycle	Radeon ProRender
Rendering Speed	Fast	Slow	Moderate
Image Quality	Lower	High	Moderate
Usability	User-friendly	More complex	Requires adjustment
Best Use Cases	Game assets, interactive	High-fidelity visuals	Interactive viewport

6. Conclusion

Based on the results of the research and analysis that have been completed, the following goals can be achieved:

a. Eevee is a rendering technique that is efficient in terms of process time, making it a suitable solution to address issues that users of Blender sometimes encounter when rendering 3D, particularly when they encounter issues related to lengthy rendering. By using Eevee, users can speed up the rendering process without significantly lowering visual quality.

b. Visual details produced by the rendering technique Cycles produce the best results, even though they require more time to render. Conversely, the Eevee technique offers a faster rendering time while maintaining high visual quality, making it a more efficient method of time management. However, in terms of storage efficiency, the render size of the results produced by the three techniques does not indicate any significant differences; the three techniques have a relatively consistent performance.

c. Compared to Radeon ProRender, Eevee, and Cycles produce images that are more accurate and visually clearer. In terms of rendering effects, Cycles delivers the most realistic results, while Eevee provides moderately realistic output. In contrast, Radeon ProRender produces relatively minimal rendering effects. Furthermore, Cycles and Eevee demonstrate good color accuracy, whereas Radeon ProRender's color output frequently deviates from the original settings.

Based on the results of the research that was completed, the following are some points that the author can make:

a. Development can continue the study by examining several rendering techniques, whether complimentary or berbayar, such as V-Ray, RenderMan, and Arnold, to assess the features and accuracy of each technique.

b. Further research can be conducted using a computer with a wide range of specifications and a higher level of precision to investigate the effects of specific hardware on render results. Specifically, using a CPU (processor) with:

- The best option is AMD Ryzen 9 5900X or Intel Core i9-12900K.
- The AMD Ryzen 7 5800X or Intel Core i7-12700K is the main specification.
- Entry-level specifications: AMD Ryzen 5 5600X or Intel Core i5-12600K
- The best option is 64 GB DDR4/DDR5.
- Menengah option: 32 GB DDR4
- Entry-level option: 16 GB DDR4
- Utilizing Capacity in this way:
- The best option is NVMe.SSD 1 TB or more (WD Dark SN850, Samsung 970 EVO, additionally)
- SSD SATA 1 TB (Samsung 860 EVO, Significant MX500) is the storage device.
- Entry-level Pilihan: SSD SATA 512 GB 3. Subsequent research can be carried out by creating a 3D display with a higher level of complexity, whether it be in the form of an animated or humorous movie.

Author Contributions: Conceptualization: I.Z. and R.S.; Methodology: I.Z. and I.R.; Software: I.Z.; Validation: R.S. and I.R.; Formal analysis: I.Z. and R.; Investigation: I.Z.; Resources: R.S.; Data curation: I.R.; Writing—original draft preparation: I.Z.; Writing—review and editing: R.S., I.R., and R.; Visualization: I.Z.; Supervision: R.S.; Project administration: I.R.; Funding acquisition: R.

Funding: "This research received no external funding."

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request. The experimental data consist of rendering time measurements, memory usage records, output file sizes, and visual comparison results generated during controlled testing of Eevee, Cycles, and Radeon ProRender within the Blender environment.

Acknowledgments: The authors would like to express their sincere appreciation to the Multimedia Laboratory of Universitas Gajah Putih for providing technical facilities and hardware support used during the rendering experiments. We also extend our gratitude to colleagues and laboratory assistants who provided valuable technical guidance, testing support, and constructive feedback throughout the research process. Their assistance greatly contributed to the completion of the experimental analysis and data validation.

The authors also acknowledge the open-source community of Blender and AMD Radeon™ ProRender developers for continuously improving rendering technologies and providing accessible documentation that supported this study. AI tools were used in a limited capacity to assist with language refinement and grammatical editing of the manuscript. The authors confirm that all research design, experimental procedures, data analysis, and interpretation of results were conducted independently by the research team.

Conflicts of Interest: "The authors declare no conflict of interest."

References

Almeida, R., Barbosa, J., & Silva, F. (2022). GPU acceleration strategies for real-time rendering in complex 3D environments. Journal of Real-Time Image Processing, 19(4), 843–856. https://doi.org/10.1007/s11554-021-01156-3

Astuti, I., Purwanto, I., Hidayat, T., Satria, D., H., & Purnama, R. (2022). Comparison of time, size and quality of 3D object rendering using render engine Eevee and Cycles in Blender. 2022 5th International Conference of Computer and Informatics Engineering (IC2IE), 54–59. https://doi.org/10.1109/IC2IE56416.2022.9970186

Author, C., Alfain, B., Bhakti, N., Nurfaizal, Y., Anwar, T., Digital, B., & Informasi, S. (2022). Analisis komparasi teknik rendering Blender Render dan Cycles Render pada video animasi 3D tentang alat pencernaan manusia.

Barkovska, O., Bilichenko, O., Uvarov, H., & Makushenko, T. (2024). Improved rendering method of skeletal animation on control points base. Computer Systems and Information Technologies. https://doi.org/10.31891/csit-2024-1-9

Carneiro, T., Da Nóbrega, R., Nepomuceno, T., Bian, G., De Albuquerque, V., & Filho, P. (2018). Performance analysis of Google Colaboratory as a tool for accelerating deep learning applications. IEEE Access, 6, 61677–61685. https://doi.org/10.1109/ACCESS.2018.2874767

Chen, W., Gao, J., Ling, H., Smith, E., Lehtinen, J., Jacobson, A., & Fidler, S. (2019). Learning to predict 3D objects with an interpolation-based differentiable renderer, 9605–9616.

Creswell, J. W., & Creswell, J. D. (2023). Research design: Qualitative, quantitative, and mixed methods approaches (6th ed.). SAGE Publications.

Dudek, S., & Dziedzic, K. (2022). Comparative analysis of the Cycles and Eevee graphics engines on the example of rendering 3D models of archaeological artifacts. Journal of Computer Sciences Institute. https://doi.org/10.35784/jcsi.2971

Hosen, M., Ahmmed, S., & Dekkati, S. (2019). Mastering 3D modeling in Blender: From novice to pro. ABC Research Alert. https://doi.org/10.18034/ra.v7i3.654

Huang, T., Wang, Y., & Li, Q. (2024). Memory-efficient rendering techniques for high-resolution 3D visualization. Computers & Graphics, 118, 34–45. https://doi.org/10.1016/j.cag.2023.10.012

Kim, S., & Lee, J. (2022). Performance comparison of rasterization and ray tracing in modern 3D engines. IEEE Access, 10, 55781–55792. https://doi.org/10.1109/ACCESS.2022.3178904

Liu, S., Li, T., Chen, W., & Li, H. (2020). A general differentiable mesh renderer for image-based 3D reasoning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44, 50–62. https://doi.org/10.1109/tpami.2020.3007759

Patel, R., & Singh, A. (2023). Physically based rendering approaches for photorealistic image synthesis. Multimedia Tools and Applications, 82(9), 13215–13234. https://doi.org/10.1007/s11042-022-13564-9

Rao, G., Sgar, V., Bikku, T., Prasad, C., & Cherukuri, N. (2021). Comparing 3D rendering engines in Blender. 2021 2nd International Conference on Smart Electronics and Commuznication (ICOSEC), 489–495. https://doi.org/10.1109/ICOSEC51865.2021.9591800

Rossi, M., & Conti, F. (2020). Evaluating rendering engines for architectural visualization workflows. Visualization in Engineering, 8(1), 1–12. https://doi.org/10.1186/s40327-020-00095-2

Sahputra, E., & Sucahyo, M. (2022). Analysis of Eevee engine rendering engineering in making 3D animation videos Mukomuko Hospital. Jurnal Komputer, Informasi dan Teknologi (JKOMITEK). https://doi.org/10.53697/jkomitek.v2i2.845

Schraml, D., & Notni, G. (2024). Synthetic training data in AI-driven quality inspection: The significance of camera, lighting, and noise parameters. Sensors (Basel, Switzerland), 24. https://doi.org/10.3390/s24020649

Yulia, A., Z., Bintoro, P., Andini, D., & Triloka, J. (2024). Perbandingan kinerja rendering EEVEE dan Siklus di Blender 3.5 dalam konteks visual interaktif untuk animasi 3D. Jurnal Penelitian Pendidikan IPA. https://doi.org/10.29303/jppipa.v10i7.5910

Zhang, Y., Li, X., Wang, H., & Chen, L. (2021). Advances in 3D modeling and visualization techniques for digital media applications. Computers & Graphics, 95, 75–89. https://doi.org/10.1016/j.cag.2021.01.015