



Visualization of Real-World 3D Reconstructed Objects with Real-Time Ray Tracing on Ampere Architecture Graphic Processing Unit

Nur Hikmah¹, Moch Fachri², Risanto Darmawan³

^{1,2,3}Informatics, Faculty of Engineering, Krisnawipayana University, Jl. Raya Jatiwaringin, Jawa Barat 13077, Indonesia

E-mail: ¹nurhikmah@unkris.ac.id, ²moch.fachri@unkris.ac.id, ³risantodarmawan@unkris.ac.id

ARTICLE INFO

ABSTRACT

Article history:

Received: September 12, 2021
Revised: October 16, 2021
Accepted: November 30, 2021

Keywords:

Ray tracing;
3D Visualization;
Graphic processing unit;

3D models from the reconstruction of real-world objects have photorealistic details. This realistic detail in the visualization needs to be supported by a visualization environment that is able to represent realistic real-world conditions. However, the environmental light information on the reconstructed 3D object was lost in the reconstruction process. This is because the light obtained from scanned real-world objects for reconstruction can reduce the accuracy of the reconstruction results. This inaccuracy is caused by alterations in the light reflected from the captured object, causing the texture of the object difficult to reconstruct. In the visualization of the scanned object, the light that appears on the object is reformed virtually through ray tracing. This research utilizes the RT core technology on the Ampere graphics processor architecture introduced by Nvidia. With this technology, the ray tracing process can be carried out in real-time, enabling realistic and dynamic visualization. In this study, an application that is able to perform 3D visualization will be made by utilizing real-time ray tracing with an RT core on the graphics processing unit of the Ampere architecture. This application is able to blend virtual environments and 3D models that are reconstructed from the results of scanning real-world objects. We will demonstrate that this application can later become a 3D visualization solution for models from the reconstruction of real-world objects.

Copyright © 2021 Jurnal Mantik.
All rights reserved.

1. Introduction

Ray tracing is a set of algorithms that allow users to achieve photorealistic graphics[1], algorithms that allow users to apply next-generation lighting effects that look as realistic as possible: scattered shadows, background darkening, reflections, refractions, and more[1]. There have been several hardware acceleration implementations for ray tracing in the past, but none are widely available in terms of implementations accessible in consumer-grade graphics accelerator devices[2]. This is until the solution of real-time ray tracing technology was introduced by Nvidia Corporation [3]. The RT cores introduced in the Turing architecture by Nvidia Corporation are capable of accelerating the ray tracing process which allows the use of ray tracing on consumer-grade graphics processors [4].

The emergence of this technology opens up many opportunities, especially in the field of computer graphics, especially in the visualization of 3D objects resulting from the reconstruction of real objects. Light plays an important role in object visualization [5]. However, in 3D objects reconstructed from real-world objects, the environmental light information captured in the retrieval of real-world object information can reduce the accuracy of the reconstruction results[6][7]. So that in the visualization, the light that appears on the object is regenerated virtually through ray tracing. This process requires high computational performance so it is difficult to do in real-time and ubiquitous [8]. This results in the lighting of the virtual environment in which the 3D object is visualized are not dynamic. Lighting is done statically using illumination that is



"baked" to objects in the environment [10]. So that changes in light due to perspective or dynamic interactions with objects are not visualized.

With the advent of ray tracing technology in the Turing architecture, interactive games are able to utilize this ray tracing in-game visuals [9]. The artificial 3D objects that are visualized in the game can be rendered with dynamic lighting in real-time. This research will utilize ray tracing technology to visualize the results of the 3D reconstruction of real-world objects with global and dynamic environmental lighting. It is hoped that the results of this study will produce prototypes and design pipelines for the visualization of 3D objects obtained from the reconstruction of scanned real-world objects.

Ray tracing is a technique that utilizes ray casting to recursively collect light contributions from reflective and refracted objects [11]. Ray casting itself is the process of finding the nearest object, or sometimes an arbitrary random object, along the ray [11]. As an example in Figure 1, three rays travel from the camera to the scene. The top, green light directly on the box. The middle purple ray hits the mirror and reflects off to pick up the back of the box. At the bottom, the blue light hit the glass ball, spawning reflection, and refraction rays. The refracted ray in turn produces two more daughter rays, with the ray traveling through the glass producing two more rays. This example is a simple multipoint ray cast, wherein actual application, there are many light points that are cast against the visual perspective of the user's camera.

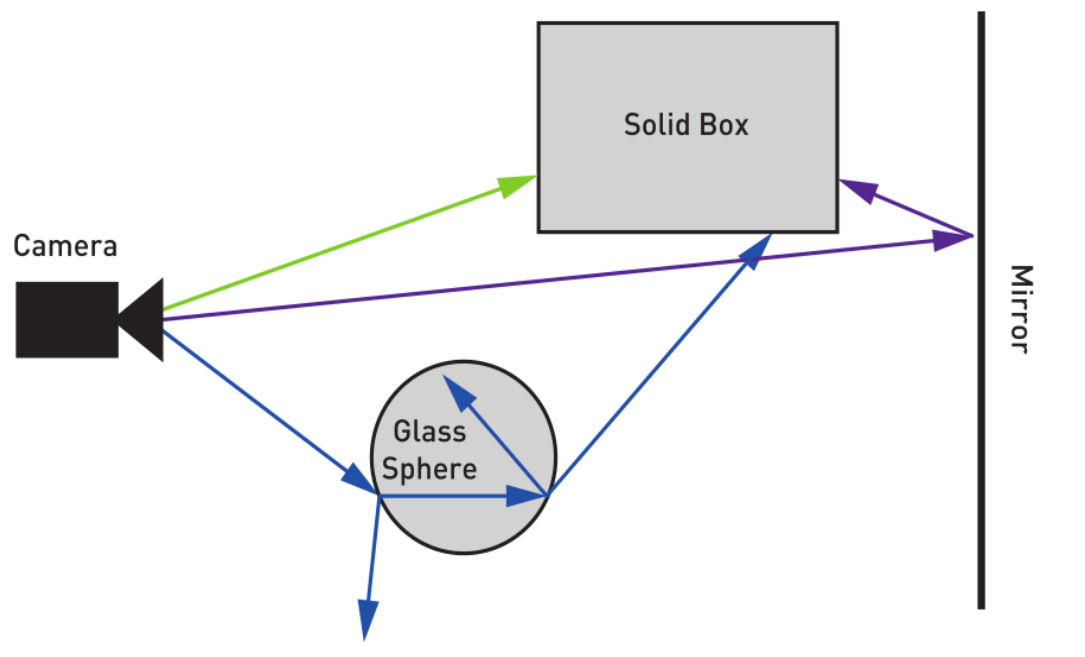


Fig. 1 Ray tracing illustration[11]

In the real world, many sources of light emit, which reach the eye in various ways, including refraction and reflection. Glossy surfaces reflect light in many directions, not just along the direction of reflection. Diffuse or matte surfaces scatter light in a wider spread as illustrated in figure 2. In tracing the path of the cast from the beam, it is necessary to take into account the scattering behavior of the light, the outward direction, and the material to help determine the importance of the various ingress directions to the surface shadow.

This ray tracing process when carried out in real-time requires large computational power [12]. This process in computing requires a professional-class graphics processor such as the Titan series from Nvidia [13]. But lately, hardware technology, especially in graphics processors, has improved rapidly, so that we are in an era where real-time ray tracing can be done with consumer-grade graphics processors [4]. Nvidia introduced RT cores technology in their GPU (Graphics Processing Unit) architecture, the Turing architecture. This technology enables real-time ray tracing on consumer-grade graphics processors. Moreover, in the latest architecture from Nvidia which was announced in September 2020, namely the Ampere architecture, the latest GPU from Nvidia has higher performance at a price half times cheaper than

the previous generation[14]. This shows the trend of real-time ray tracing technology which is increasingly affordable, where hardware access is widely accessible to consumers.

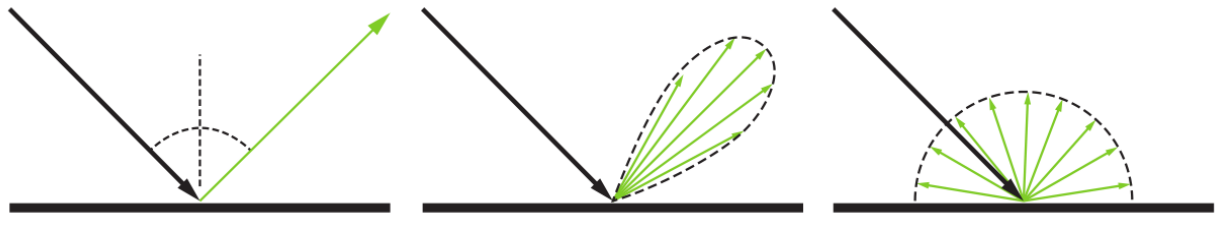


Fig. 2. Mirror reflection, glossy and diffused rays. Left: incoming light is reflected in one direction from the mirror surface. Center: a polished surface, such as brass, which reflects light near the direction of reflection and gives it a glossy appearance. Right: diffuse or matte material, such as plaster, and incoming light is scattered in all directions.[11]

Seeing the potential ubiquity of this technology, the applications offered with real-time ray tracing are also very broad. One study showed the performance of this GPU device from Nvidia for linear algebra computing showed an increase of up to 1.8 times [15]. Another study using ray tracing from Nvidia in radio wave propagation shows that only the ray tracing application from the NVIDIA OptiX GPU is able to provide convergent results [16]. Examples of other applications are in the field of satellite communication [17], distribution of the electromagnetic spectrum [18], aero optics [19], and many others [20][21][22].

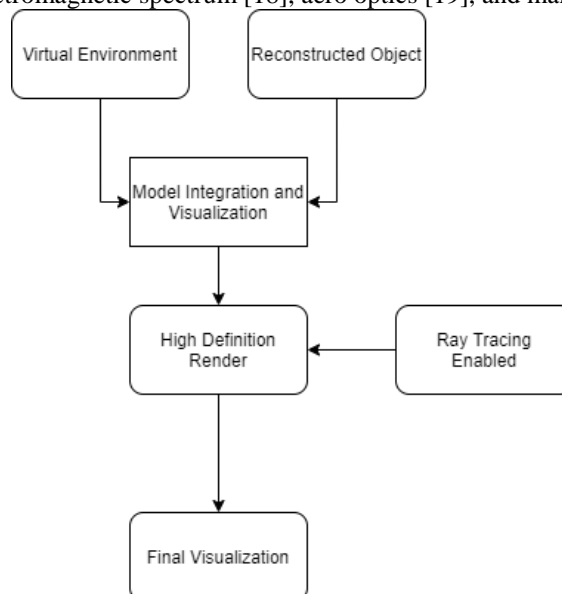


Fig. 3. Pipeline of 3D visualization with ray tracing enabled

For the use of real-time ray tracing in 3D visualization, it has been widely implemented in electronic games [9]. Real-time ray tracing is able to provide a real perception of a visualized virtual object [23]. With the potential that exists in GPU capabilities from Nvidia, it allows artist directors to change the technique of visualizing artwork in virtual 3D models. Techniques that previously used pre-rendered animation became live dynamic interactions by the user manipulating shapes and viewing them from new angles as they were displayed in real-time[23]. This interactive visualization has been designed since the era of Nvidia's Turing architecture GPU, the Nvidia GeForce RTX 2080 Ti consumer-grade GPU[23].

2. Method

The implementation of this research is in accordance with the flow shown in Figure 3. For the visualization of this 3D model, an application that is able to display 3D objects is needed in real-time. This application is planned to be made using the Unity3D game engine. This game engine was chosen because it supports real-time ray tracing capabilities available on Nvidia GeForce GPU devices[24]. This application is

designed with dynamic interaction capabilities in its visualization so that real-time ray tracing capabilities will be deployed to the maximum.

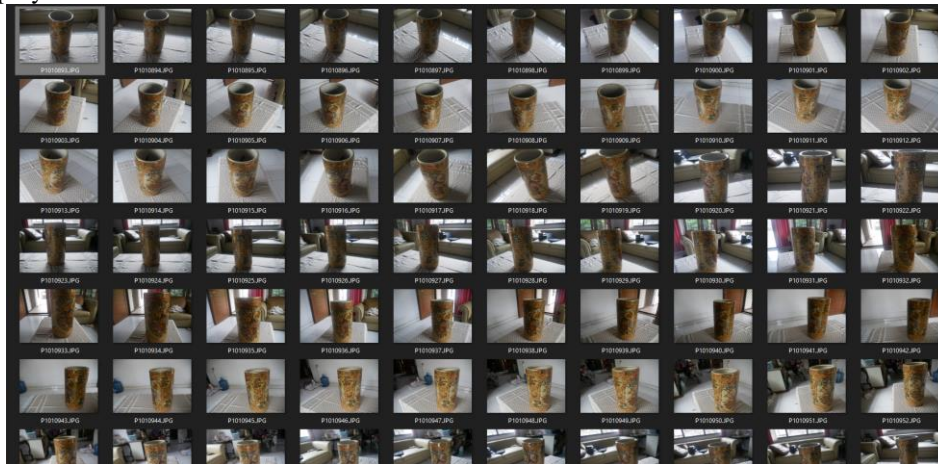


Fig 5. Digital image collection of the object of interest that are used for 3D reconstruction[25]



Fig. 5. 3D scanned result of the object of interest for visualization with ray tracing enabled [25]

In addition to making applications, it is also necessary to create a virtual environment that will be displayed in the visualization. This virtual environment is created with refractive and reflective virtual materials to enhance the dynamic light effect of real-time ray tracing. Visualization applications and virtual environments that will be displayed in visualizations are made with real-time ray tracing that utilizes RT cores from the Nvidia GeForce GPU with the latest architecture, the Ampere architecture [14]. Therefore this application requires a graphics card device that has the GPU to run.

The main model that becomes the center of visualization will be placed in the artificial environment. This main model itself is available from research data that has been done previously [25]. This model is the result of reconstruction from scanning real-world objects shown in figure 4. The reconstructed model in figure 5 does not have light information, later artificial light from real-time ray tracing carried out in a virtual application environment will be able to affect the visualization of the objects. Other studies have demonstrated the effect of real-time ray tracing on artificial objects[23]. Later, by using reconstructed objects from the real world, the effects of real-time ray tracing on the photorealistic level of the 3D objects can be analyzed. This visualization application will be engineered with Unity game engine, while the virtual environment is created by a 3D artist. After the two processes are completed, as shown in Figure 3, the integration of all components will be carried out, including applications, virtual environments, and 3D models resulting from the reconstruction.

3. Result and Analysis

Model integration and visualization of a virtual environment and 3D scanned object shown in figure 6.a. As shown in that figure, the 3D environment, as well as the 3D scanned object appear to blend since it's artificially enlightened with a uniform light without ray tracing process enabled. The jar object which is taken from a real-world object 3D scan is appeared very bright. This is caused due to the nature of the scanned model as it is taken from the real world with neutral lighting. As mentioned before, the information regarding surface lighting of the 3D scanned object is lost. The visualization with the recreation of light data in virtual environments is shown in figure 6.b. With ray tracing enabled at high definition render process on Unity3D game engine, environmental light is recreated thus affecting the visualization result. As shown in figure 6.b the 3D scanned object appears more natural since the light reflected from object texture is in unison with reflection from environmental light toward the view projection of the virtual camera. The puddle at the environment below the 3D scanned object is also able to project mirror-like reflection of the 3D scanned object.

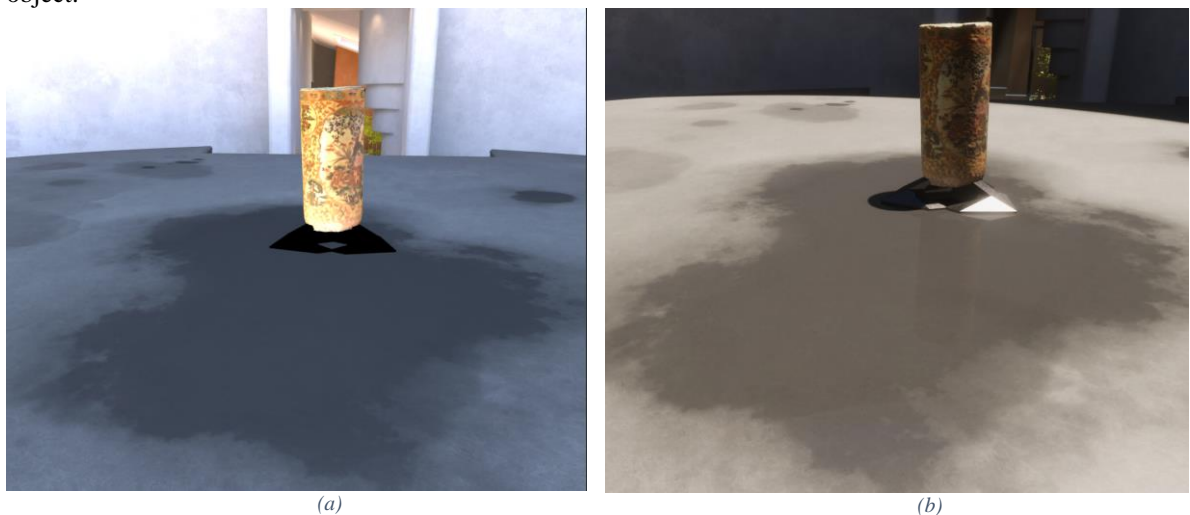


Fig. 6. 3D scanned object visualization in virtual environment (a) without ray tracing enabled, (b) with ray tracing enabled.

The force stand object that reinforces the 3D scanned object is made of the metallic refractive object, at figure 6.a that object appears to be black due to the nonexistence of simulated environmental light, thus no light refracted. Figure 6.b shows that the force stand object is able to refractively appear as a metallic object with ray tracing enabled. Figure 7 shows another perspective of the environment and 3D scanned object. With simulated light source appearing from the left relative at 3D scanned object position, the object texture at figure 7.b also appear brighter at the source of light exposure, while at the opposite side of the light source, the object texture appears darker, also the shadow can be seen opposite from the light source. This was achieved with simulated light from ray tracing. Whereas figure 7.a show the environment and 3D scanned object without a ray-traced light source.

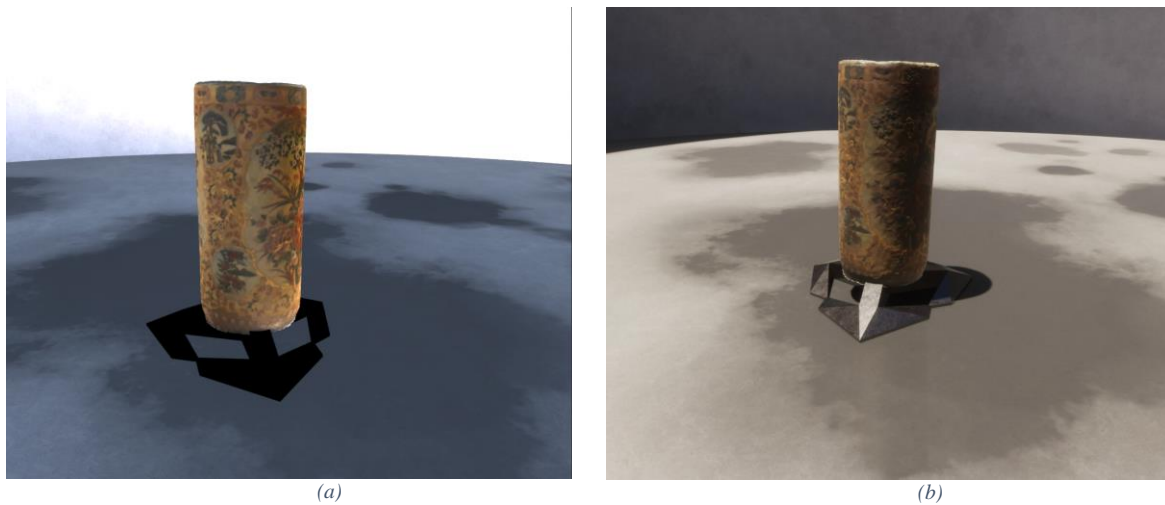


Fig. 7. Another 3D scanned object visualization in virtual environment (a) without ray tracing enabled, (b) with ray tracing enabled.

4. Conclusion

Our result demonstrates the capability of ray tracing technology based on Nvidia Ampere Architecture GPU on consumer level graphic cards. The ray tracing handled by the GPU is able to visualize 3D models reconstructed from real-world objects with artificially generated light data. Our method can be applied in creating light data either fully artificial or reconstructed based on real-world data. Since it can be done using consumer level graphic cards, the future ubiquity of this technology will emerge as the graphic processing technology progresses and become more accessible.

5. References

- [1] A.A. Bychenkov, and N.N. Perepechko, "How ray tracing affected Nvidia" Belarusian National Technical University. 2020.
- [2] V. V. Sanzharov, V. A. Frolov, and V. A. Galaktionov, "Survey of Nvidia RTX Technology," *Program Comput Soft*, vol. 46, no. 4, pp. 297–304, Jul. 2020, doi: 10.1134/S0361768820030068.
- [3] J. Burgess, "RTX on—The NVIDIA Turing GPU," *IEEE Micro*, vol. 40, no. 2, pp. 36–44, Mar. 2020, doi: 10.1109/MM.2020.2971677.
- [4] B. Санжаров et al., "Examination of the Nvidia RTX," in *GraphiCon'2019 Proceedings. Volume 2*, Nov. 2019, pp. 7–12, doi: 10.30987/graphicon-2019-2-7-12.
- [5] P. Wisessing, K. Zibrek, D.W. Cunningham, J. Dingliana, and R. McDonnell, "Enlighten Me: Importance of Brightness and Shadow for Character Emotion and Appeal". *ACM Trans. Graph.* 39, 3, Article 19 (June 2020), 12 pages, DOI:https://doi.org/10.1145/3383195
- [6] C. A. Jones and E. Church, "Photogrammetry is for everyone: Structure-from-motion software user experiences in archaeology," *Journal of Archaeological Science: Reports*, vol. 30, p. 102261, Apr. 2020, doi: 10.1016/j.jasrep.2020.102261.
- [7] S. Voisin, D. L. Page, S. Fougou, F. Truchetet, and M. A. Abidi, "Color influence on accuracy of 3D scanners based on structured light," San Jose, CA, Feb. 2006, p. 607009, doi: 10.1117/12.643448.
- [8] A. Keller et al., "Ray tracing is the future and ever will be...," in *ACM SIGGRAPH 2013 Courses on - SIGGRAPH '13*, Anaheim, California, 2013, pp. 1–7, doi: 10.1145/2504435.2504444.
- [9] A. Keller, T. Viitanen, C. Barré-Brisebois, C. Schied, and M. McGuire, "Are we done with ray tracing?," in *ACM SIGGRAPH 2019 Courses*, Los Angeles California, Jul. 2019, pp. 1–381, doi: 10.1145/3305366.3329896.
- [10] C. Luksch, M. Wimmer, and M. Schwärzler, "Incrementally baked global illumination," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, Montreal Quebec Canada, May 2019, pp. 1–10, doi: 10.1145/3306131.3317015.
- [11] E. Haines and T. Akenine-Möller, Eds., *Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs*. Berkeley, CA: Apress, 2019.
- [12] Biagio Cosenza, Carsten Dachsbacher, and Ugo Erra, "GPU Cost Estimation for Load Balancing in Parallel Ray Tracing," in *Proceedings of the International Conference on Computer Graphics Theory and Applications and*

- International Conference on Information Visualization Theory and Applications, Barcelona, Spain, 2013, pp. 139–151, doi: 10.5220/0004283401390151.
- [13] Y. Deng, Y. Ni, Z. Li, S. Mu, and W. Zhang, “Toward Real-Time Ray Tracing: A Survey on Hardware Acceleration and Microarchitecture Techniques,” *ACM Comput. Surv.*, vol. 50, no. 4, pp. 1–41, Nov. 2017, doi: 10.1145/3104067.
- [14] M. Wuebbeling, “Giant Step into the Future: NVIDIA CEO Unveils GeForce RTX 30 Series GPUs | The Official NVIDIA Blog.” <https://blogs.nvidia.com/blog/2020/09/01/nvidia-ceo-geforce-rtx-30-series-gpus/> (Diakses 15 Okt, 2020).
- [15] Y. M. Tsai, T. Cojean, and H. Anzt, “Evaluating the Performance of NVIDIA’s A100 Ampere GPU for Sparse Linear Algebra Computations,” arXiv:2008.08478 [cs], Aug. 2020, Accessed: Oct. 16, 2020. [Online]. Available: <http://arxiv.org/abs/2008.08478>.
- [16] Y. Liu, D. Shi, A. Li and P. Zeng, "Radio Wave Propagation Prediction Based on the NVIDIA OptiX GPU Ray Tracing Engine," 2019 IEEE 6th International Symposium on Electromagnetic Compatibility (ISEMC), Nanjing, China, 2019, pp. 1-3, doi: 10.1109/ISEMC48616.2019.8986068.
- [17] P. Chen and Y. He, “Analysis and Simulation of Multipath Effect in High Precision Satellite Communication Based on Ray-Tracing Approach,” in 2019 IEEE 2nd International Conference on Electronics and Communication Engineering (ICECE), Xi’an, China, Dec. 2019, pp. 185–189, doi: 10.1109/ICECE48499.2019.9058560.
- [18] Z. Chen and Y. Zhang, “Offshore Electromagnetic Spectrum Distribution Prediction Algorithm Based on Ray Tracing Method,” *Antennas Wirel. Propag. Lett.*, vol. 18, no. 10, pp. 2061–2065, Oct. 2019, doi: 10.1109/LAWP.2019.2937414.
- [19] L. Xu, D. Xue, and X. Lv, “Influence of ray-tracing step on aero-optics imaging deviation computation,” in 2017 IEEE International Conference on Imaging Systems and Techniques (IST), Beijing, Oct. 2017, pp. 1–5, doi: 10.1109/IST.2017.8261482.
- [20] J. Ulseth, Z. Zhu, Y. Sun, and S. Pang, “Accelerated X-Ray Diffraction (Tensor) Tomography Simulation Using OptiX GPU Ray-Tracing Engine,” *IEEE Trans. Nucl. Sci.*, vol. 66, no. 12, pp. 2347–2354, Dec. 2019, doi: 10.1109/TNS.2019.2948796.
- [21] C. H. Teh, B. K. Chung, and E. H. Lim, “Multilayer Wall Correction Factors for Indoor Ray-Tracing Radio Propagation Modeling,” *IEEE Trans. Antennas Propagation.*, vol. 68, no. 1, pp. 604–608, Jan. 2020, doi: 10.1109/TAP.2019.2943397.
- [22] R. Shimogawa, K. Izumi, and T. Tsujimura, “Optimum Design of Optical System for Free Space Optics Device by Ray Tracing,” in 2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Hiroshima, Japan, Sep. 2019, pp. 857–862, doi: 10.23919/SICE.2019.8859914.
- [23] A. Lomas, “Enhancing Perception of Complex Sculptural Forms Using Interactive Real-time Ray Tracing,” presented at the Proceedings of EVA London 2020, Jul. 2020, doi: 10.14236/ewic/EVA2020.37.
- [24] U. Technologies, “Unity real-time ray tracing for high fidelity reflections & global lighting | Graphics & Rendering | Unity.” <https://unity.com/ray-tracing> (Diakses 15 Okt, 2020).
- [25] M. Fachri and N. Hikmah, “Digital Preservation of Real World Objects Using Photogrametry from Digital Image,” *Jurnal Mantik*, vol. 4, no. 3, p. 6, 2020.