International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

# Minimalistic Ray Tracing Engine for Experimentations

**Benjamin Douglas J.**

Department of Computer Science, Rajalakshmi Institute of Technology
*benjamindouglas.j.cse.2108[at]ritchennai.edu.in*

**Abstract:** ***Ray tracing helps to produce photo realistic images in real time. It is based on how light interacts with the different objects within the scene. There are lots of various physical effects that can be achieved using the ray tracing algorithm. This paper describes the various concepts and modules involved in a ray tracing program having minimalistic and easy codes that a user can tinker around with it to do experimentation and learn the fundamentals of ray tracing procedure. This helps people who need to understand the working principles and the various effects that can be achieved using ray tracing method.***

**Keywords:** ray-tracing, photo-realism, rendering, engine, light phenomena

## 1. Introduction

A ray tracing algorithm is a technique for modelling light transpose for producing photo realistic images/renders. It traces a beam of light from a viewport towards the objects present in the screen. If the object is opaque, it casts a shadow ray only. In case of reflective or refractive object, it can cast a reflective ray, a refractive ray and a shadow ray. The shadow ray is always traced towards the light source, and if it's blocked by another object, it casts shadow, otherwise it casts a colour. This provides how an object looks in a real-life situation. Ray tracing gained huge popularity after beginning its implementation in video games that provides real time outputs. This process of tracing rays from viewport is called as backward tracing. Whereas in forward ray tracing, rays are traced from light sources, which provides true to nature outputs.

## 2. Literature Survey

In July of 2022, a literature survey was conducted in the following worldwide online bibliographic databases: (a) IEEE Explorer, (b) SpringerLink, (c) ACM Portal and (d) Science Direct. ("Ray Tracing") AND ("Computer Graphics") AND ("Render Engine") were the search terms used. The articles were chosen based on the following inclusion criteria:

- Relevant research works proposing techniques for simple ray tracing engine
- Discuss the methods, development and evaluation of the simple ray tracing engine
- Published in the journal

Further filtering took place by excluding the articles based on the following exclusion criteria:

- Paper that are written in a language other than English
- Paper to which full access was not granted
- Papers that lacked a through explanation and assessment

## 3. Problem Definition

The problem faced while studying the field of ray tracing is that there is no a simple engine with easy and readable cone a new learner can understand while also having the basic functionalities of a ray tracing algorithm. This paper tends to solve this problem by providing a simple pipeline for building a minimalistic ray tracing engine with basic concepts implemented and that can be tinkered around to see how it affects the scene.

## 4. Methodology and Overall Concept

The engine consists of three components. They are;

- Objects
- Rendering
- GUI

These three modules work together to provide the desired output. Each model has its own functionality and each of them depends on one another. There will be predefined object and properties present within the engine, which can be tweaked or modified entirely for one's purpose.

## 5. Modules of the Engine

All The three modules must need to work synchronously to obtain the required result. This helps to understand the importance of each module.

### A. Object Module

This module consists of the predefined objects that are present in the scene, namely a box, a plane, a solid and a sphere. These are best suited for showing hoe the ray tracing algorithm works because each object has its own speciality. In the case of box, it contains flat surfaces and also sharp edges where tracing the shadow ray is difficult. For plane it shows the basic workings of reflection and refraction. For sphere finding the rays that bounces from the surface shows how good the output of ray traced image when compared to rasterization and also its disadvantage, that is the computational cost and time taken for tracing the rays. It

also defines the materialistic properties of the objects. They are;

- Diffuse colour.
- Specular colour.
- Reflectance, that varies between 0 and 1.
- Specular reflectance, that has value 0 for diffuse and value 1 for mirror.
- Transmissivity, that varies from 0 and 1.
- Refraction Index that has value 1 for air and value 2.4 for diamond.
- Roughness, that varies from 0 and 10.

These characteristics of the object help to determine how the ray interacts with object in the scene.

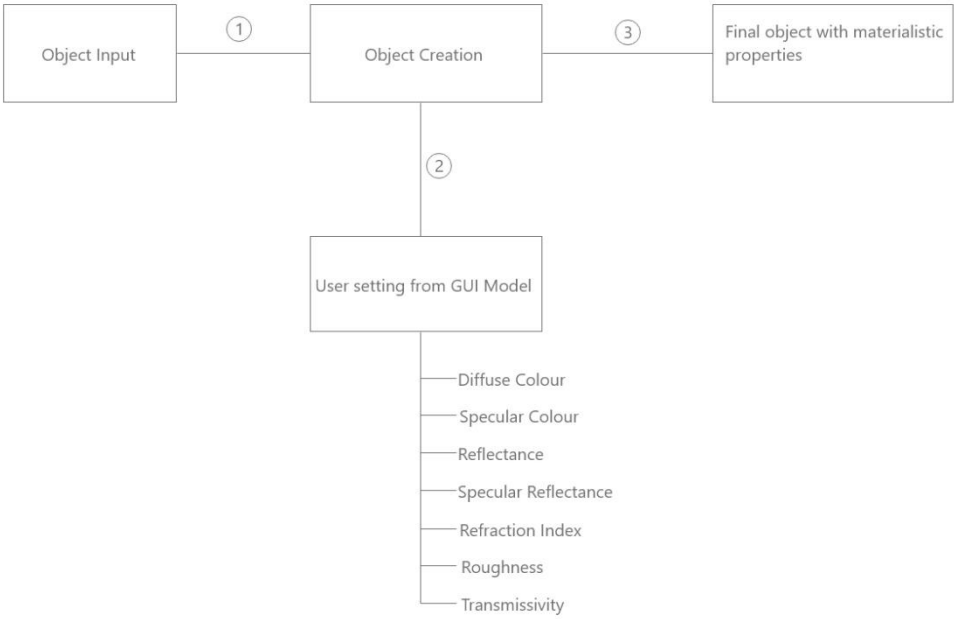


**Figure 1.1:** Object module and its various components

### B. Rendering Module

The Rendering module is built purely using C++ language and libraries. It uses Monte Carlo ray tracing algorithm to simulate many light phenomena. They are

- Colour Bleeding
- Soft Shadows
- Reflection
- Refraction

It also makes use of photon mapping algorithm to stimulate caustic effects. Parallelization is achieved using openMP library.

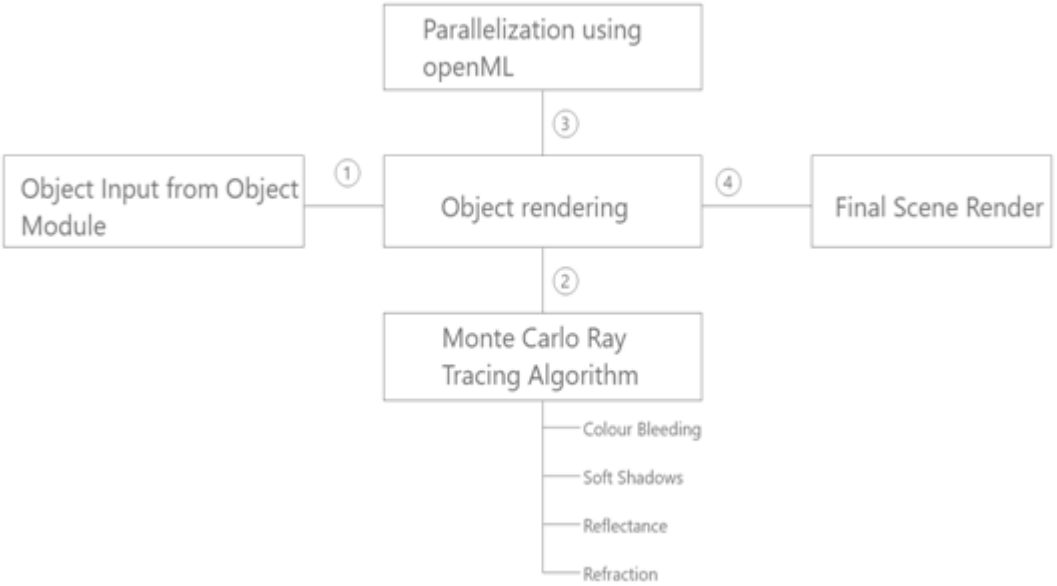


**Figure 1.2:** Rendering module and its various components

### C. GUI Module

The GUI Module consist of four components, namely

- Progress Dialog
- Setting Panel
- Viewport

The Progress Dialog show the total progress of the render finished out of 100 percentage. This helps the user to see how long a render takes based on the given custom inputs. The Setting panels help to control the scene setting namely reflectance, transmissivity, refraction index and roughness. The viewport is used to show the output of the final render.

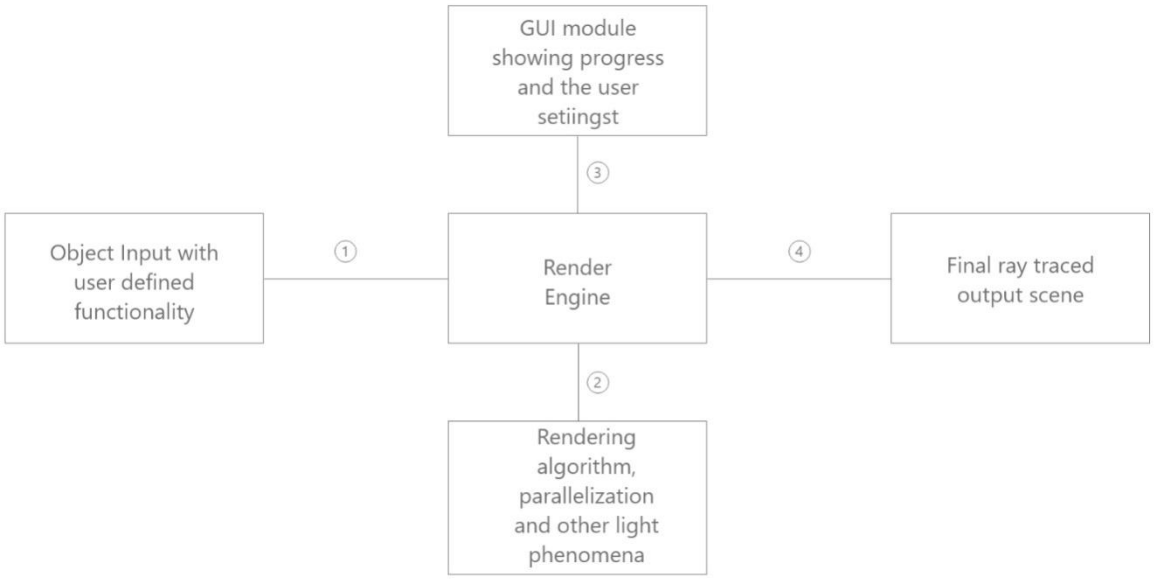


**Figure 1.3:** Overall pipeline of the ray tracing engine for easy understanding.

## 6. Results

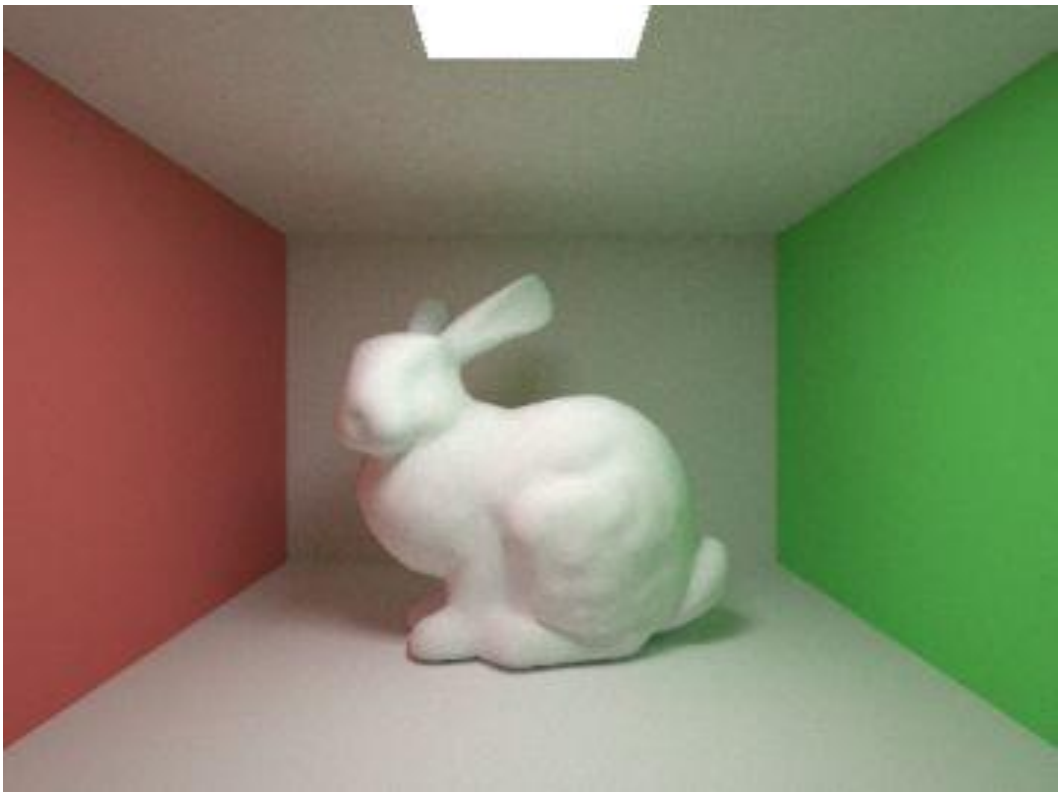

**Figure 1.4:** Final Render with an opaque object in the scene

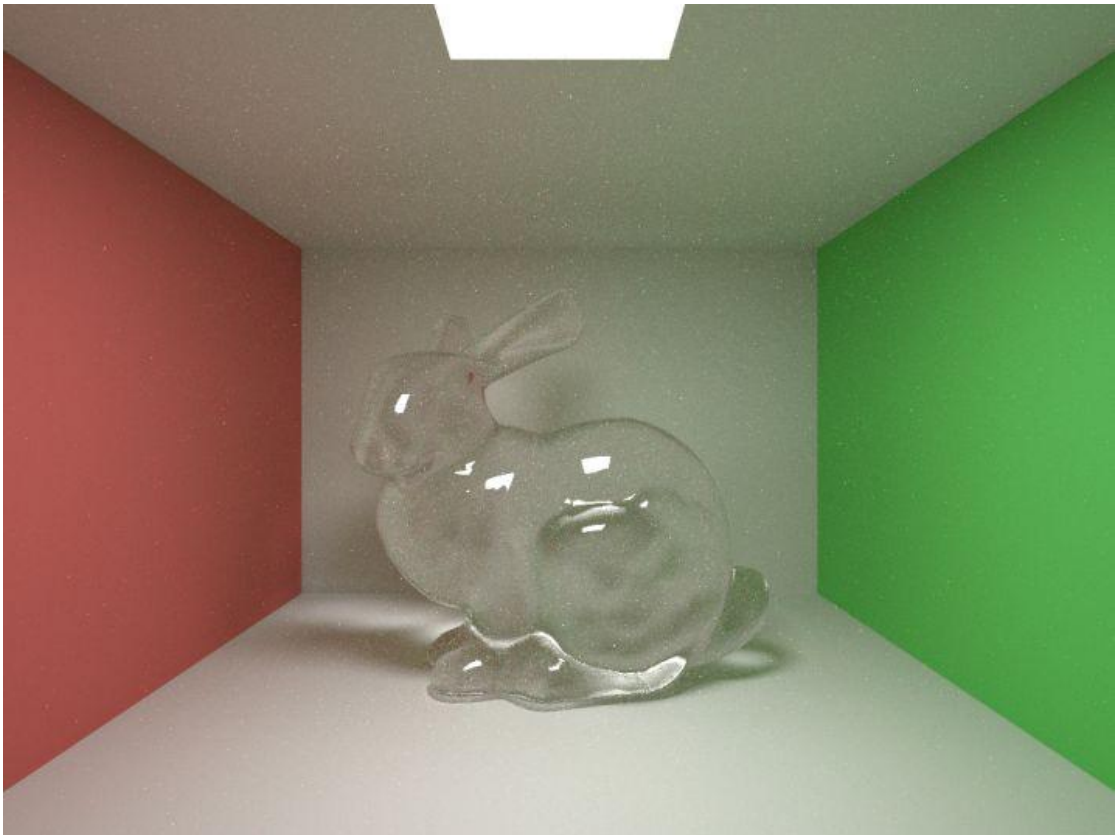

**Figure 1.5:** Final Render with a transparent object in the scene

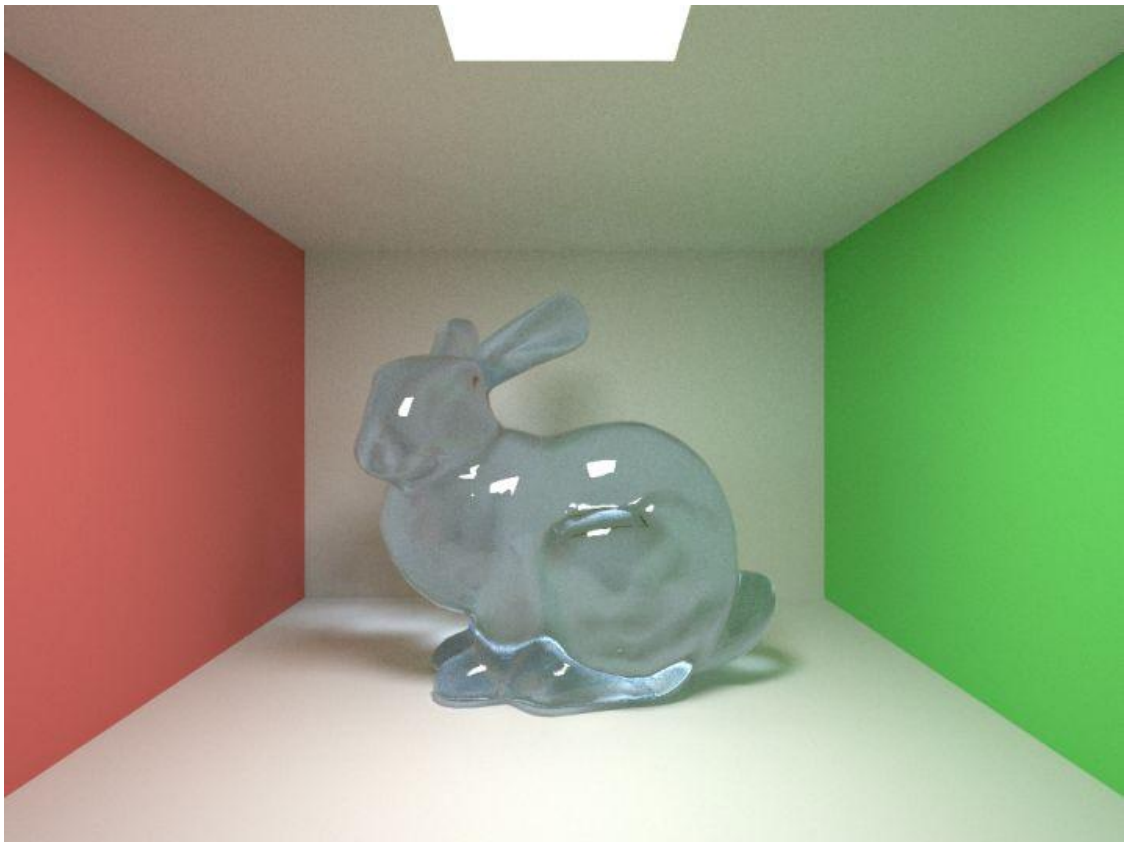

**Figure 1.6:** Final Render with a translucent object in the scene.

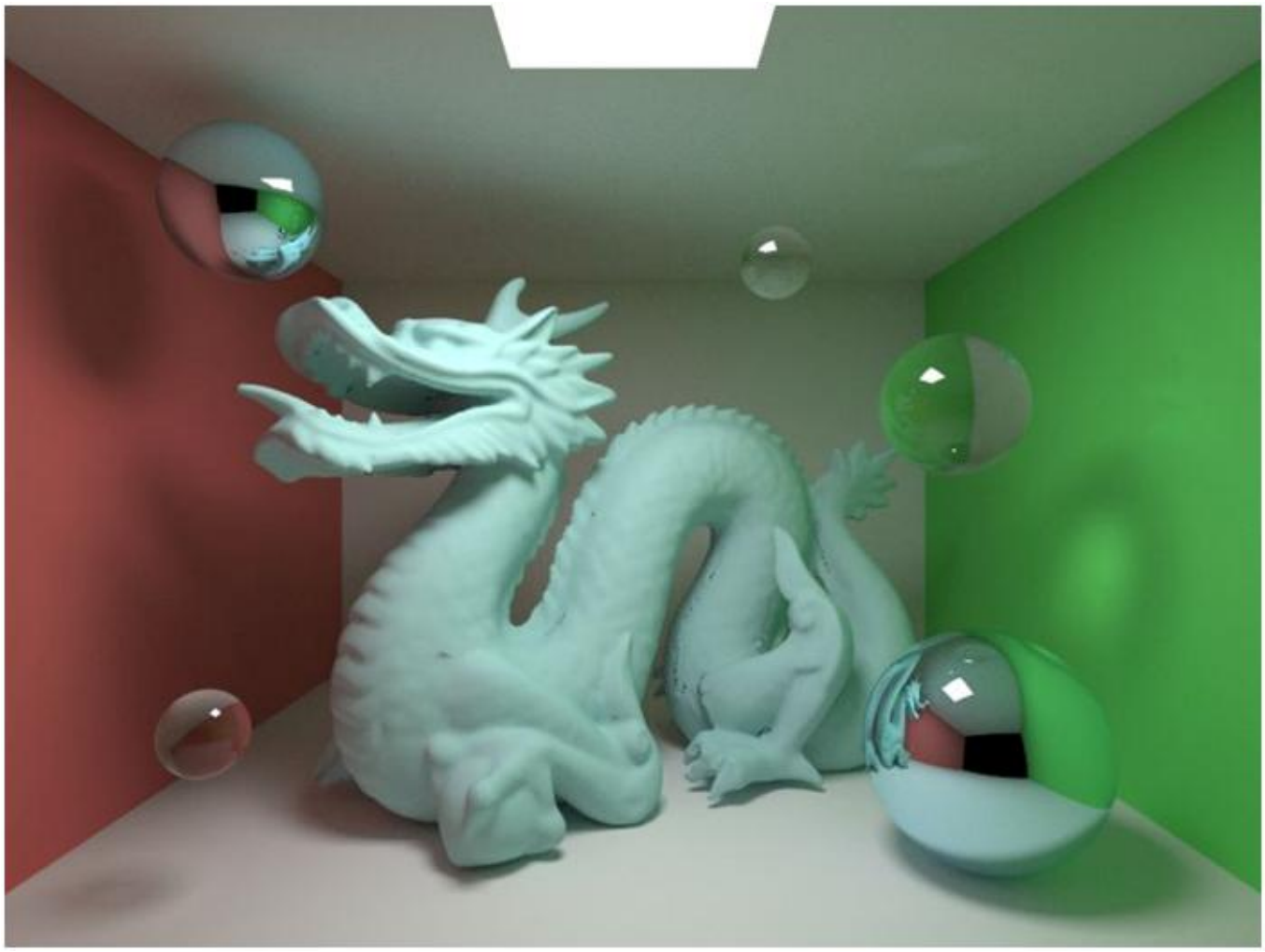

**Figure 1.7:** Final render with an opaque and multiple reflective objects to show how well the ray tracing algorithm can provide photorealism.

## 7. Future Works

In the futures the following additional components will be added to give user more control over the final render scenes. They are;

- Implementing a depth of field technique that gives cinematic feel to the render
- Optimization techniques that can reduce the render time drastically like reduce cash misses, simplify code and precomputing.
- Ad mor refraction amount that helps to increase the photorealism of the render.
- Improving parallelization by making use of better ray tracing API like CUDA and Vulkan.

## 8. Conclusion

This paper helps the user to create and experiment on a basic ray tracing engine with much needed functionalities. This also helps one to understand the concepts of computer graphics involved in ray tracing and intrigue the user to study upon mor on this domain. While the concepts of ray tracing are changing often, the basic concepts remain the same. This this engine last longer in terms of basic concepts learning of this domain.

## References

[1] C.F. Chang, K. -W. Chen and C. -C. Chuang, "Performance comparison of rasterization-based graphics pipeline and ray tracing on GPU shaders," 2015 IEEE International Conference on Digital Signal Processing (DSP), 2015, pp. 120-123, doi: 10.1109/ICDSP.2015.7251842.

[2] D. Bilibashi, E. M. Vitucci and V. Degli-Esposti, "Dynamic Ray Tracing: A 3D Formulation," 2020 International Symposium on Antennas and Propagation (ISAP), 2021, pp. 279-280, doi: 10.23919/ISAP47053.2021.9391318..

[3] J. Burgess, "RTX on—The NVIDIA Turing GPU," in IEEE Micro, vol. 40, no. 2, pp. 36-44, 1 March-April 2020, doi: 10.1109/MM.2020.2971677..

[4] J. Gambrych, "Influence of optimization techniques on software performance for subsequent generations of CUDA architecture," 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2021, pp. 1002-1009, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00140.

[5] S. Panghal, D. A. Bilung, N. Gupta and G. Kumar, "Enhancing Graphic Performance Curve using Ray Tracing," 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, pp. 55-59, doi: 10.1109/CICN49253.2020.9242622.

[6] Vasiou, E., Shkurko, K., Mallett, I. et al. A detailed study of ray tracing performance: render time and energy cost. Vis Comput 34, 875–885 (2018). https://doi.org/10.1007/s00371-018-1532-8