Ray Tracing in Computer Graphics

Zheng Yu
Monash University, Clayton, Victoria, 3168, Australia

Abstract. The research is about the development of light and shadow in video games on the Internet, and also to compare the light and shadow effects displayed on video game screens using different techniques. Then, the types and history of ray tracing technology are explained, its working principle is studied, and its pros and cons are analyzed. After the research, this paper finally determined that a viable ray tracing technique in the current society is hybrid ray tracing, which is a combination of forward and backward ray tracing. Because forward ray tracing methods are too inefficient while backward ray tracing methods cannot calculate and render the light entering in front of the lens. By studying 3 main hybrid ray tracing, their advantages, disadvantages and suitable application scenarios are analyzed and determined in detail. Finally, the development and improvement of the technology in recent years are discussed. Currently, analyses of a particular ray tracing technology do exist in the market. However, there are still few articles that explore all available ray tracing technologies, taking the reader through the history and current state of ray tracing development and giving developers a more visual comparison to help them choose the most appropriate technology for the scenarios they want to develop.

Keywords: Light and Shadow in Video Games; Rasterization; Ray Tracing; Main Hybrid Ray Tracing; Development and Improvement.

1. Introduction

In computer graphics, developers often use a ray tracing technique that simulates light transmission to generate various rendering algorithms for digital images in order to produce more realistic pictures. In the film industry, Disney [1] skillfully used ray tracing to simulate optical effects such as reflection, refraction and motion blur to make its films realistic (Walt Disney Animation Studios). Moreover, ray tracers were already capable of handling effects such as shadows and transparency quickly and accurately in 2003, Shirley and Morley [2] suggested that it was becoming increasingly popular and they predicted that people would soon be using the technology to power video games. With hardware support for real-time ray tracing technology and the opening up of graphics APIs, people made this happen in 2018. Shortly afterwards, the first video game [3] to achieve light and shadow effects entirely with ray tracing technology was released in August 2019.

2. Ray Tracing in Video Game

Before ray tracing was used in video games, rasterization techniques were widely used to simulate light and shadow. Generally speaking, rasterization [4] is the process of converting a vector map described by a mathematical formula into a bitmap made up of pixels. Specifically, rasterization used in games [5] is a large set of triangles described in mathematical coordinates for game scenes, and based on parameters such as the position of the player's camera and the viewpoint, the triangles are calculated to cover those pixel points in the screen. Yet, with the increasing number of graphics cards that support ray tracing, developers are increasingly using ray tracing instead of rasterization because ray tracing has two very distinct advantages.

Firstly, ray tracing is much more efficient. The lighting effects produced by rasterization are very deliberate. In fact, the technique implements a scene of a character looking in a mirror, rendering the same model and duplicating the scene with two cameras. This halves the computer performance due to the requirement to render the same pair of model and scene. Moreover, if there are multiple reflective mirrors at different angles in the scene, the computation will be multiplied by several times, which is incredible for any game. Therefore, mirrors usually only appear in small scenes or on-thefly animations in games that do not use ray tracing, such as in Max Payne 3, where the protagonist cuts his own hair in the bathroom (Figure 1). However, using of ray tracing technology for this effect is much more economical, because the resources it requires are fixed and do not increase as the scene grows. As long as it is set at the right range and precision, the amount of computing consumed by all reflections is well controlled.

Figure 1. The main character cuts off his own hair in Max Payne 3

Moreover, ray tracing technology can calculate the intensity and angle of reflections dynamically. It works by calculating the reflectance based on the difference between the light and dark sides of the transparent glass and the angle of the player's camera. This is not possible with rasterization, so even if game manufacturers were willing to sacrifice half the performance to make a large amount of glass in the game scene reflective, the lack of these two dynamic calculations would still make the reflections look very stiff. In contrast, “Control” for Microsoft Windows [3], released by Remedy Entertainment in August 2019, became the first game to have reflections of scenes and characters visible at all times throughout the game with the help of ray tracing. Comparing the scenes in Figure 2 without ray tracing to those in Figure 3 with ray tracing, it is clear that the reflections in the ground and glass are much more realistic.

Figure 2. Reflections without ray tracing in “Control”

Figure 3. Reflections with ray tracing in "Control"

Subsequently, the types of ray tracing and the history of its development will be discussed.

3. Research

The types of ray tracing [6] can be classified as forward ray tracing, backward ray tracing and hybrid ray tracing (Lu et al.). Forward ray tracing follows light particles emitted at the light source, backward ray tracing follows light particles emitted at the eye, and hybrid ray tracing follows both light particles emitted at the light source and the eye. Since both forward and backward ray tracing have significant drawbacks, researchers invented hybrid ray tracing to tracing lights. The hybrid ray tracing has 3 main approaches, which will be described and analyzed in detail later.

First of all, forward ray tracing [6] is light-oriented and referred to as light ray tracing and photon tracing. The tracing follows light particles from the light source to the object (Figure 4), so its most significant advantage is to determine the color of each object most accurately. However, tracing each light from the light source means that many rays that do not contribute to the final image seen by the eye will be wasted. Therefore, it will spend more resources on unnecessary rays, causing it very inefficient.

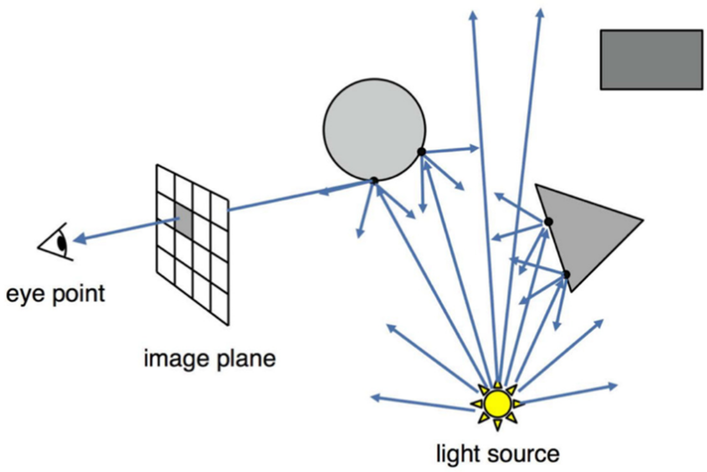

Figure 4. Forward ray tracing

In contrast, backward ray tracing [6] is eye-oriented, which traces the eye ray generated at the eye and then enters the scene through the view plane. Additionally, Figure 5 shows its schematic representation. Moreover, the eye ray shows the first object the ray encounters and can find out the specific coloring and shading of the point on the view plane, thus offering more accurate pixels for the thing to make ray tracing more efficient. This is because it does not waste computational power on rays that are not seen, all the rays are contributed to the final image. However, there is a downside to its feature of counting only the light entering the eye. For a piece of a lens with a focal point of light, this tracing will not be able to calculate and render the light before it enters the lens because those rays do not enter the eye but have an impact on the scene, thus causing the scene to appear distorted.

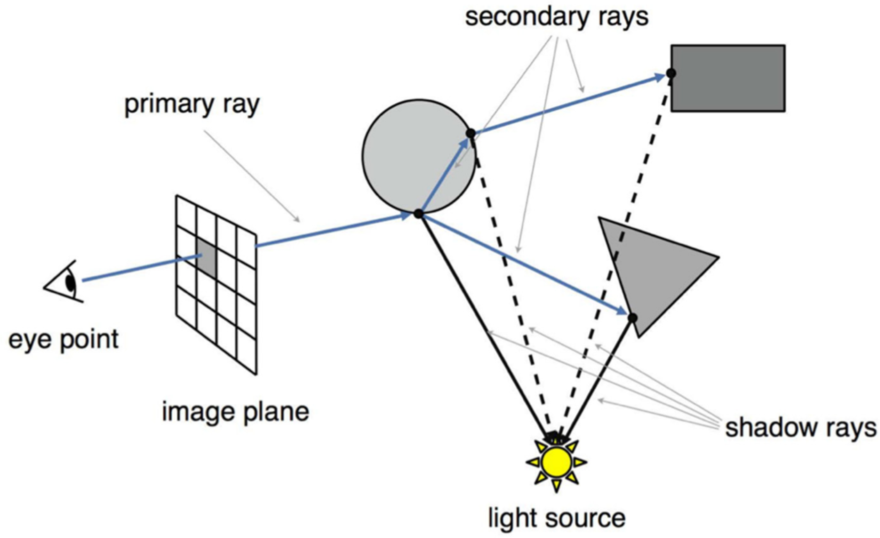

Figure 5. Backward ray tracing

To mitigate the impact of the shortcomings of forward and backward ray tracing, researchers invented 3 hybrid ray tracing sequentially. These methods perform only part of the forward ray tracing and record the data. Then, backward ray tracing is continued. The structure of both backward raytracing and forward raytracing calculations are considered to determine the final coloring of the scene.

In 1980, Whitted's method [7] solved the technical difficulty that ray tracing could not simulate complex reflections and refractions, which is depicted in Figure 6. The algorithm can simulate complex reflections and refractions using backward tracking, and determines the type of surface when the eye ray touches the first visible surface:

- If the surface of the object is opaque and the reflection is a diffuse property (e.g., a table), then directly calculates the intersection color of the contact surface.
- If the object is opaque but specularly reflective (e.g., a mirror), only the reflected rays are projected, and the color of the intersection point of the contact surface is calculated.
- If the object is transparent (e.g., a glass bottle), a transmitted ray and a reflected ray are projected, and the color of the surface is calculated at the point of intersection of their contact surfaces, respectively.
- The background colour will be returned if the projected light does not intersect any object.

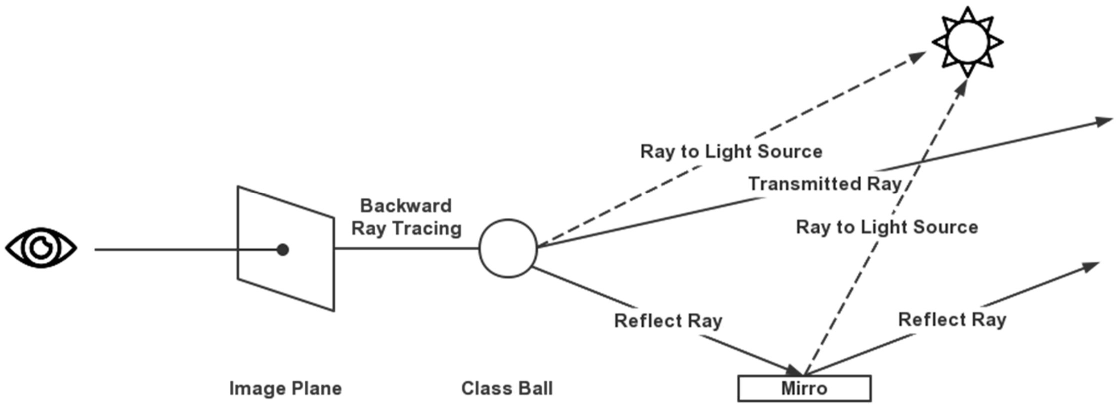

Figure 6. Schematic representation of Whitted's concept

Although this method enables reflection and refraction, making the scene light look better, which is shown in Figure 7, it can only simulate perfect specular reflection or refraction. In reality, most reflections and refractions are imperfect, so they cannot realistically manufacture all lights.

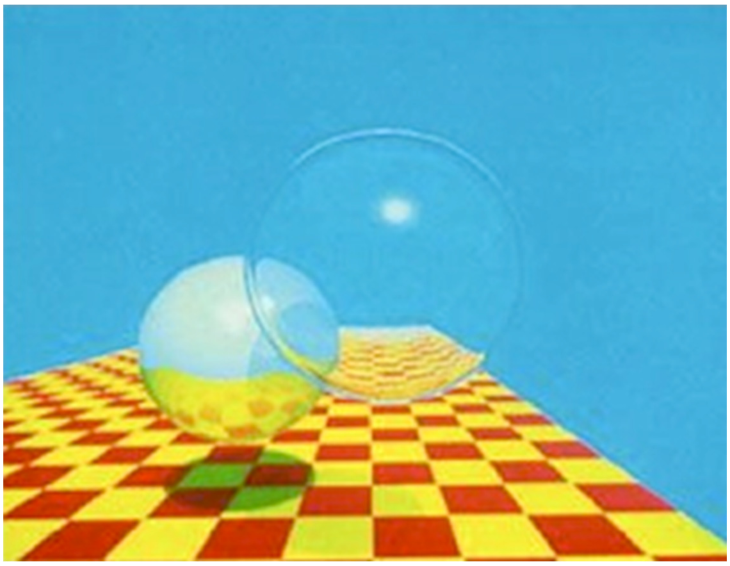

Figure 7. Image produced by Turner Whitted with his algorithm in 1980

Heckbert [8] proposed another type of hybrid ray tracing in 1990. He stated that the final image could be rendered more accurately by storing global illumination information on the surface. Its schematic representation can be seen in Figure 8. In this algorithm, each pixel would have a "ray" tree for storing global illumination information of the pixel intensity. When the eye ray touches the first surface, it extends to other characters and light sources. The shader then traverses this tree to determine the pixel's light intensity. In addition to the effects simulated by traditional shaders, this shader also can accurately simulate realistic reflections, shadows, and refractions. Moreover, the algorithm has anti-aliasing capabilities to display curved and polygonal surfaces.

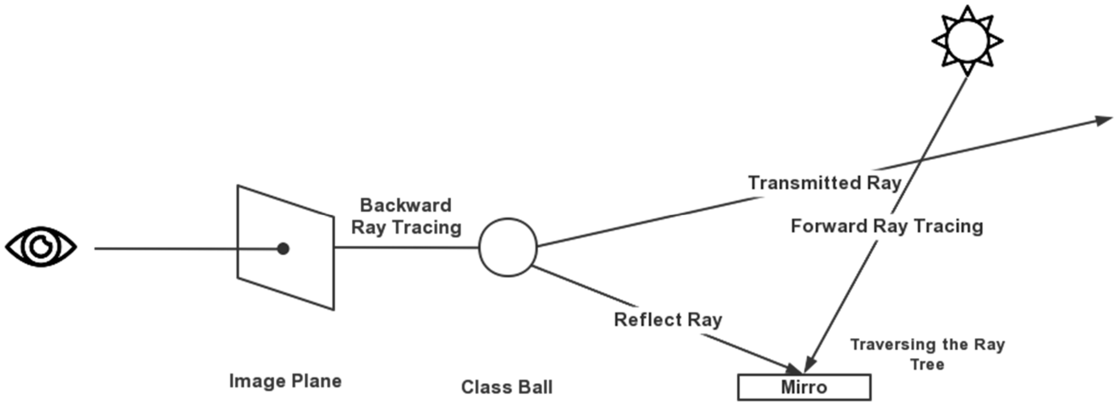

Figure 8. Schematic representation of Heckbert's concept

Heckbert achieved a more extensive range of non-perfect refraction and reflection through this method in 1980, making the scene more realistic, which can be seen in Figure 9. However, this method requires storing global illumination information at each pixel, requiring more computing power for the computer.

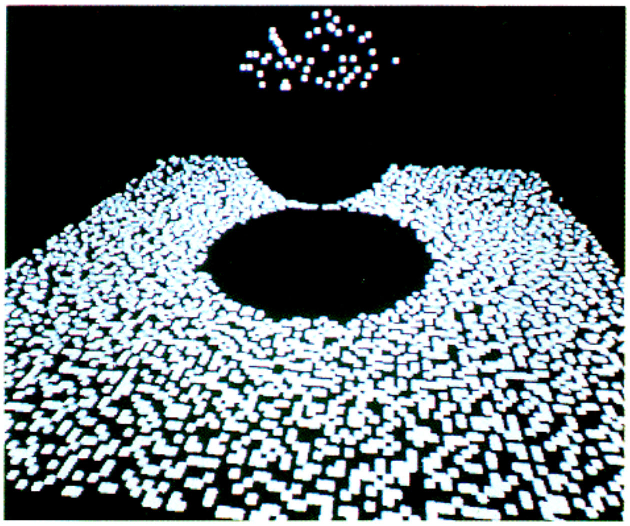

Figure 9. Image produced by Paul S. Heckbert with his algorithm in 1980

Last but not least, Veach [9] invented Bidirectional Path Tracing in 1995, a mixture of backward ray tracing, forward ray tracing and connecting lines. The principle is that when the light from the eye and the light source hit at the same time, after several bounces, determine the visibility of the eye path and light path on the vertex, then connect the 2 points so that many paths can be generated quickly (Figure 10).

However, connecting only two points would make it difficult to guarantee the quality of the image. To ease this problem, Veach also proposed Multiple Importance Sampling to combine the results of various sampling methods with appropriate weights to achieve noise reduction (Figure 11). However, bidirectional path tracing can be seriously biased when dealing with caustics on the water or transparent glass sphere because the method does not connect two vertices on a specular or refractive surface.

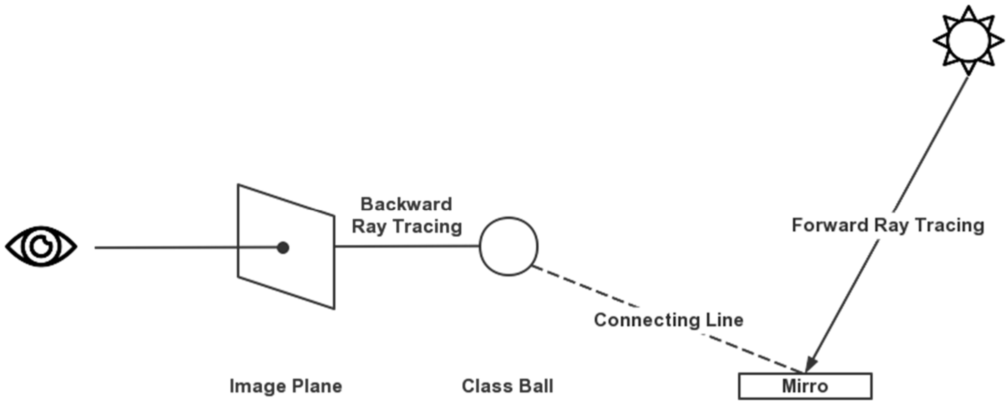

Figure 10. Schematic representation of Veach's concept

Figure 11. Image produced by Eric Veach with his algorithm in 1995

Based on the theory of wave packet dynamics, Richardson, Bonoli and Wright [10] invented a field reconstruction algorithm in 2009 to address the issues that Bidirectional Path Tracing does not handle caustics correctly. After applying the algorithm to a cold plasma slab model, they obtained experimental results that were consistent with the analytical analysis. Twelve years later, Gerasimos and Konstantinos [11] proposed a hybrid approach using ray tracing and rasterization to enable the coverage of water caustic to a wide view distance in real time. In the same year, a major breakthrough was made in rendering transparent objects in real time with caustics. Xin and Risong [12] used a local ray-tracing method that tracks the light in the direction of the light source to find the incident point to make the caustics on transparent objects displayed correctly. After applying a denoiser to suppress noise, they obtained improved quality of the rendered image at a playable frame rate.

4. Conclusion

As ray tracing technology in Computer Graphics continues to mature, ray tracing will play an increasingly important role in movies, animation, and even video games. In 3 types of ray tracing, people would not use forward or backward ray tracing alone to chase light because the former is inefficient, while the latter cannot simulate reflection and refraction, thus hybrid ray tracing that combines the advantages of the above methods is popular currently. Whitted first invented hybrid ray tracing to achieve complex reflection and refraction. Still, it could only handle perfect reflections and refractions, and it could handle mirrors and glass in more scenes very well. Later, Heckbert invented a method to handle non-perfect reflections and refractions by letting each pixel store global illumination information. However, this method did not handle noise particularly well. Next, Veach's bi-directional path tracing handles noise well, but it also has a problem of not being able to handle caustics. By 2009, Richardson et al. had correctly handled caustics with a field reconstruction algorithm. Twelve years later, caustics on large areas of water and Caustics on transparent objects could also be simulated in real time.

In short, ray tracing technology is developing very fast and is gradually entering the entertainment of the masses. In the future, it can be applied to video games and even metaverse, which will bring an additional dimension of perception to the player to make the virtual world more realistic. For example, in a shooting game, players will be able to judge the arrival of an enemy based on the reflection.

References

[1] Walt Disney Animation Studios. “Disney’s Practical Guide to Path Tracing.” Youtube, 10 Aug. 2006, www.youtube.com/watch?v=frLwRLS_ZR0&t=288s.

[2] Shirley, Peter, and Keith Morley. Realistic Ray Tracing. CRC Press LLC, 1 July 2003.

[3] Walton, Jarred. “Control Is the Best, Most Complex Implementation of Ray Tracing in a Game so Far.” PC Gamer, 30 Aug. 2019, www.pcgamer.com/control-is-the-best-most-complex-implementation-of-ray-tracing-in-a-game-so-far/.

[4] “Vector vs. Raster Images: Choosing the Right Format.” Pavilion Dinfos Online Learning, 18 Dec. 2020, pavilion.dinfos.edu/Article/Article/2223089/vector-vs-raster-images-choosing-the-right-format/#: ~: text =Vector%20graphics%20are%20digital%20art.

[5] Munkberg, Jacob, et al. “Per-Vertex Defocus Blur for Stochastic Rasterization.” Computer Graphics Forum, vol. 31, no. 4, June 2012, pp. 1385–1389, 10.1111/j.1467-8659.2012.03133. x.

[6] Lu, Charity, et al. “1997 Sophomore College Ray Tracing Site.” Cs.stanford.edu, 1997, cs.stanford.edu/ people/ eroberts/courses/soco/projects/1997-98/ray-tracing/.

[7] Whitted, Turner. “An Improved Illumination Model for Shaded Display.” Communications of the ACM, vol. 23, no. 6, June 1980, pp. 343–349, 10.1145/358876.358882.

[8] Heckbert, Paul S. “Adaptive Radiosity Textures for Bidirectional Ray Tracing.” ACM SIGGRAPH Computer Graphics, vol. 24, no. 4, Aug. 1990, pp. 145–154, 10.1145/97880.97895.

[9] Eric, Veach, and Guibas Leonidas. “Bidirectional Estimators for Light Transport.” Photorealistic Rendering Techniques, edited by Georgios Sakas et al., Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 145–167.

[10] Richardson, A S, et al. “Reconstructing Lower Hybrid Fields from Ray Tracing Data.” AIP Conference Proceedings, vol. 1187, no. 1, 2009, pp. 629–632.

[11] Kougianos, Gerasimos, and Konstantinos Moustakas. “Large-Scale Ray Traced Water Caustics in Real-Time Using Cascaded Caustic Maps.” Computers & Graphics, vol. 98, Aug. 2021, pp. 255–267, 10.1016/ j. cag. 2021.06.008.

[12] Wang, Xin, and Risong Zhang. “Rendering Transparent Objects with Caustics Using Real-Time Ray Tracing.” Computers & Graphics, vol. 96, May 2021, pp. 36–47, 10.1016/j.cag.2021.03.003.