

# Physically Based Rendering

The art of the science of Light

$$L_o(p, \omega_0) = L_e(p, \omega_0) + \int_{S^2} f(p, \omega_0, \omega_i) L_i(p, \omega_i) |\cos \theta_i| d\omega_i$$

**Sohaib Alam**

A thesis presented for an Extended Project Qualification

St Bartholomew's School  
United Kingdom  
2024-03-10

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	The Ray-tracing Algorithm . . . . .	2
1.3	History and Current State of the Field . . . . .	2
1.3.1	Relevant Literature . . . . .	3

# 1 Introduction

Since the inception of computing technology, there has been an concerted effort to replicate observable phenomena from the natural world as a means of solving problems that once couldn't be solved without computers.

For example, Craig Reynolds in his seminal paper [1] is an excellent case study, of such an example. Reynolds defined three very simple rules pertaining to the behaviours of birds in a flock. These rules are:

- Alignment: the birds will steer towards the average heading of their peers.
- Cohesion: the birds will steer towards the center of the mass of its peers.
- Separation the birds will steer away from colliding with their peers.

These simple laws, though primitive, produce a "realistic" approximation of how birds behave in actuality, so much so, that the US Army uses this algorithm, for their UAV-UGV programs [2].

In essence, computer simulation provides a "good enough" approximation for the theories encompassing the real world, crafted by mathematicians and physicists alike.

## 1.1 Motivation

Rendering is the process of producing an image from a description of a 3D scene. This daunting task can be easily understood by asking: Given a set configuration of a room, what would a camera see, in a set location within the room. If left pondering, one can easily come to a reasonable algorithm, befit the style of rendering they want. For example, a very simple approach could be to check if a light ray enters the camera from light sources (if any) within the room. If so, then the camera could record a colour based on some product of the colours the light ray hit before it entered the camera.

However, I intend to implement the "Physically Based" rendering algorithm. As the name suggests, this algorithm tries to stay true to the physics of the world, imitating its behaviour as matter and light mesh together. It differentiates different materials dependent on their reaction with light incident upon them, as well as how light itself reacts through mediums not necessarily vacuums, like fog.

I intend to implement this algorithm not only because it is a excellent opportunity to mix the three subjects I do for A-Level together (Physics, Maths, Computing), but also as an challenging extension in the field, in which I will be considering a future in. TODO: SUCCESS CRITERIA

## 1.2 The Ray-tracing Algorithm

TODO

## 1.3 History and Current State of the Field

Unsurprisingly, Physically Based Rendering (PBR), is a relatively nascent field (only being studied since the 70s), and as the field has advanced to cleverer solution to increasingly difficult problems. For example, in the 70s, the biggest problem to solve was the lack of memory available to computers (1 MB at its rare), where physical accuracy was not biggest focus, but to speed up how long an image took to render [3]. This was achieved by taking a subset of the entire scene representation into memory, however this caused problems with global illumination algorithms, as this (the name might suggest) required a larger (or the entire) scene to be loaded into memory. So, with speed, creators of such graphics decided that geometric accuracy was more important then realism of light. Because of this, many people though at the time that realism was not as important as artis-

tic merit, where the computer graphics was used as an aid, rather than a tooling which is essential from the ground up.

The best example of this is the rise of Computer Generated Imagery (CGI) within the film industry, where some of the biggest blockbuster releases, leveraged the power of computers to render backgrounds and add complex elements to a scene, or augment a character's physical appearance (e.g. *Terminator 2*) [4].

When computers did become advanced enough to render scenes in full, many film studios adopted the physically based rendering pipeline into their film-making process. A good case study is Pixar, who have been propellers of this field into the mainstream. Their *Renderman* renderer has used the photorealistic algorithm, and have won several awards for their films. They also contribute to the SIGGRAPH community, which is another essential source in this field.

However, as Jim Blinn states: "as technology advances, rendering time remains constant". This observes that as technology advances, people's demand for "better" and "more realistic" images increases, rather than being content with current standards, and rendering those scenes faster [3].

### 1.3.1 Relevant Literature

- Physically Based Rendering: From Theory to Implementation 3rd edition [3] is perhaps the best book on this topic. It was written by ex-NVidia and ex-Google employees, who specialise in image rendering. It is the only consolidatory source in this field and provides excellent resources for further research (which proved useful for certain topics within my EPQ), as well as provide exercises to get a better understanding of the topic.
- Glassner's introduction to the ray-

tracing algorithm provides an excellent insight into the basics of the algorithm. Though from the 80s, it still is a excellent foundation for more complex and modern algorithms [5]

- Highner's book on numerical algorithms, isn't related to ray-tracing or PBR, but provides an excellent bank of algorithms and "tricks" to solving complex problems, such as the Monte-Carlo Integration algorithm [6].
- Glassner's book on "Graphics Gems" is also a very useful resource and similar Highner's book, provides insightful knowledge into algorithms referenced in Physically based Rendering 3rd ed [7].
- Woop, Benthin and Wald's paper on ray-triangle intersections is a essential resource, as provides a novel algorithm in solving the problem of ray-triangle intersections. Without it, it would be near impossible to render complex objects which aren't quadrics [8].
- Shirley's 3-part series on ray-tracing are an excellent method, from which I dipped my feet into the world of PBR. While the premise of the series might be considered by most as simple, it still is an important resource on solving problems simply rather than robustly [9].
- Cohen and Kappauf's paper on colour theory was also useful when making the camera of the ray-tracer, it broke down the complicated Commission Internationale de l'Éclairage (CIE) standards on colour [10].

## References

- [1] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’87, New York, NY, USA: Association for Computing Machinery, Aug. 1, 1987, pp. 25–34, ISBN: 978-0-89791-227-3. DOI: [10 . 1145/37401 . 37406](https://doi.org/10.1145/37401.37406). [Online]. Available: <https://dl.acm.org/doi/10.1145/37401.37406> (visited on 03/11/2024).
- [2] M. Saska, V. Vonásek, T. Krajník, and L. Přeučil, “Coordination and navigation of heterogeneous MAV–UGV formations localized by a ‘hawk-eye’-like approach under a model predictive control scheme,” *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1393–1412, Sep. 1, 2014, ISSN: 0278-3649. DOI: [10 . 1177 / 0278364914530482](https://doi.org/10.1177/0278364914530482). [Online]. Available: <https://doi.org/10.1177/0278364914530482> (visited on 03/13/2024).
- [3] M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation (3rd Ed.)* 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Nov. 2016, 1266 pp., ISBN: 978-0-12-800645-0.
- [4] C. Bramesco, “Terminator 2 at 30: Groundbreaking sequel that led to CGI laziness,” *The GuardianFilm*, Jul. 3, 2021, ISSN: 0261-3077. [Online]. Available: <https://www.theguardian.com/film/2021/jul/03/terminator-2-at-30-groundbreaking-sequel-that-led-to-cgi-laziness> (visited on 03/14/2024).
- [5] A. S. Glassner, *An Introduction to Ray Tracing*. Morgan Kaufmann, Jan. 28, 1989, 364 pp., ISBN: 978-0-12-286160-4. Google Books: [YPb1YyLqBM4C](https://books.google.com/books?id=YPb1YyLqBM4C).
- [6] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2002, 680 pp., ISBN: 978-0-89871-521-7.
- [7] A. P. (Firm) and E. E. I. (Firm), Eds., *The AP Professional Graphics CD-ROM*. Chestnut Hill, Mass.: AP Professional, 1995, 1 p., ISBN: 978-0-12-059756-7.
- [8] S. Woop, C. Benthin, and I. Wald, “Watertight Ray/Triangle Intersection,” vol. 2, no. 1, 2013.
- [9] Peter Shirley, Trevor David Black, Steve Hollasch. “Ray Tracing in One Weekend.” (), [Online]. Available: <https://raytracing.github.io/books/RayTracingInOneWeekend.html#citingthisbook/basicdata> (visited on 08/20/2023).
- [10] J. B. Cohen and W. E. Kappauf, “Color Mixture and Fundamental Metamers: Theory, Algebra, Geometry, Application,” *The American Journal of Psychology*, vol. 98, no. 2, pp. 171–259, 1985, ISSN: 0002-9556. DOI: [10 . 2307 / 1422442](https://www.jstor.org/stable/1422442). JSTOR: [1422442](https://www.jstor.org/stable/1422442). [Online]. Available: <https://www.jstor.org/stable/1422442> (visited on 01/18/2024).