UNIVERSITÀ DEGLI STUDI FIRENZE

Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

Tesi di Laurea

EFFETTI DI RIFLESSIONE NELLA COMPUTER GRAFICA FISICAMENTE ACCURATA

REFLECTION EFFECTS IN PHYSICALLY BASED COMPUTER GRAPHICS

LEONARDO ZETTI

Relatore: *Stefano Berretti*

Correlatore: *Michele Ginolfi*

Anno Accademico 2023-2024

CONTENTS

List of Figures 3
1 Introduction 7
 1.1 Computer Graphics 7
 1.2 Rendering and PBR 9
 1.3 Historical Notes 10
 1.4 Thesis Outline 12
2 Physics of Light 15
3 Ray Tracing 17
4 BRDF Models 19
5 A Ray Tracer Implementation 21
6 Results 23
7 Conclusions and Future work 25
Bibliography 27

LIST OF FIGURES

Fig. 1	The *Stanford Bunny Model*, drawn on the left as a *triangle mesh* (random color for each triangle). Model courtesy of the Stanford Computer Graphics Laboratory, rendered with Blender	8
Fig. 2	Stylized render from Blender Studio's *Project Gold*. ©Blender Foundation	10
Fig. 3	One of the renders in Whitted's 1980 paper. ©ACM	11
Fig. 4	*Monsters University*, ©Disney/Pixar 2013	13

"Insert citation"
— Insert citation's author

1

INTRODUCTION

The aim of this thesis is to discuss a set of methods used in 3D computer software for the generation of photo-realistic images.

These techniques can be grouped together under the umbrella term *Physically-Based Rendering*. There's a lot that goes into the creation of physically-based images, and our discussion will only scratch the surface of the problem. More specifically, we'll only be concerned with capturing realistic *reflection* of light on a 3D material, as there's a lot to be said on this topic alone.

The final part of the thesis will present a C++/OpenGL implementation of the Disney reflection model, presented by Brent Burley in his 2012 talk *Physically Based Shading at Disney* (Burley 2012). The talk was part of the 2012 SIGGRAPH course *Physically Based Shading in Film and Game Production*.

1.1 COMPUTER GRAPHICS

Computers have the potential to be powerful tools for artistic expression. One way to create art with a computer is through *computer graphics*. *Computer graphics* can be defined as

> "The science and art of communicating visually via a computer's display and its interaction devices."
> (Hughes et al. 2014)

Advancements in this field have made computer-generated imagery a pervasive component of our day-to-day lives, ranging from special effects in blockbuster movies to the *Graphical User Interfaces* (GUIs) on our phones.

Thus, the world of computer graphics is a vast one. We should keep in mind that the problems and solutions presented here refer to the context of "geometry-based 3D graphics", as we'll call them. Our graphics will be "geometry-based" in the sense that we'll first describe the objects we want to draw on the screen with *geometric models*[1] (lines, polygons, polygonal meshes, ...), to then *sample* them for visualization. We can imagine our process to be as follows.

1. Create a *geometric model* of the object,
2. Describe the *material* it is made of,
3. Place the object in a *virtual scene* with *light sources*,
4. Use a *virtual camera* to "take a picture" of the object (this is the so-called *sampling* phase).

Our graphics will be 3D, meaning that the geometrical models, other than width and height, will also have *depth*. Our virtual camera will then communicate that depth to the viewer through the use of *perspective*.

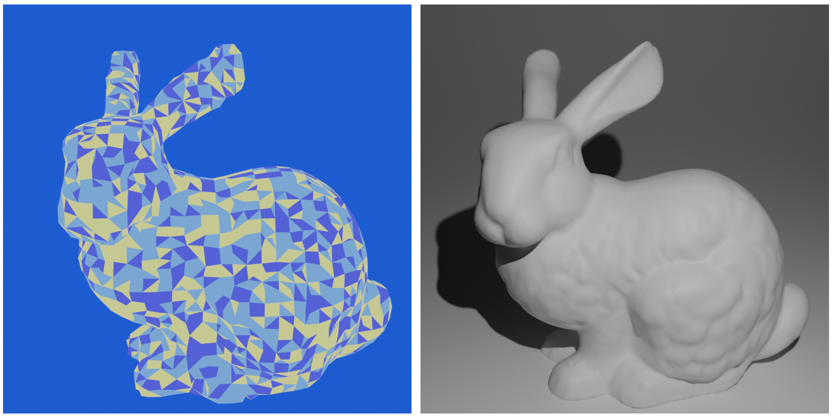

Fig. 1: The *Stanford Bunny Model*, drawn on the left as a *triangle mesh* (random color for each triangle). Model courtesy of the Stanford Computer Graphics Laboratory, rendered with Blender

To restrict ourselves even further, we'll keep our focus mainly on steps 2. and 4. of the just mentioned process. Our aim will be to derive a single *mathematical model* than can describe as many materials as feasible, in

1 In the field of computer graphics, the word "model" is used to refer both to *geometric models* and *mathematical models*. A *geometric model* describes an object we want to "take a picture" of (for example, the *Stanford Bunny*). A *mathematical model* describes some physical or computational process that we use to take that picture (for example, the equations we use to tell how much light is reflected by the bunny's surface).

relation to how they *reflect light*. We'll then use this model to record reflection effects with our virtual camera.

The final images will be generated so:

1. Light gets emitted from a light source,
2. The light hits a surface and is reflected in various amounts and directions, depending on the *material* of the surface,
3. Light that's reflected towards the virtual camera gets recorded in the final image.

We'll now give a more rigorous definition of the virtual "picture-taking" step that we're interested in. We'll refer to it as *rendering*.

1.2 RENDERING AND PBR

In the book *Physically Based Rendering, From Theory to Implementation* - an important resource for this thesis - *rendering* (or, as it has been historically referred to, *image synthesis*) is defined as

> "The process of converting a description of a three-dimensional scene into an image."
> (Pharr, Jakob, and Humphreys 2023)

Note that a *3D scene* is comprised not only of 3D objects, but also light sources and a camera. Generally speaking, the final image of this process can be generated using any rules we choose. This flexibility enables a wide range of different rendering techniques, each of which will communicate a different *message* to the viewer.

This is our final goal in graphics; to *communicate* with viewers through *meaningful images*. As Andrew Glassner put it:

> "The field of image synthesis, also called rendering, is a field of transformation: it turns the rules of geometry and physics into pictures that mean something to people."
> (Glassner 1995)

If our aim is to create art - as in the case of *animated movies*[2] - rendering can be an important tool for creative expression. The end result of

2 In 3D animated films, each of the frames shown on screen was produced by some advanced rendering software. Achieving high levels of realism demands intense computation; applications used in movie production spend *hours* computing a *single frame*.

a rendering process will influence the emotional responses elicited in spectators. This interplay of different fields of study - math, physics, art, computer science, psychology, to name a few - is what makes rendering, and computer graphics in general, fascinating disciplines that are worth exploring.

Fig. 2: Stylized render from Blender Studio's *Project Gold*. ©Blender Foundation

So, in theory, there are no rules on how to *render*[3] a picture. However, what we are often interested in is making our images *physically plausible*, that is, *realistic*. This is why various rendering applications - such as *RenderMan*, used to make Pixar movies - are, we say, written to be *physically-based*.

Physically-Based Rendering (or *PBR* for short) is a collection of rendering techniques based on physical models of real-world light and materials.

Our objective in this thesis will be physical plausibility, so we'll stick with PBR. As an alternative approach one could, for example, render *stylized* images to favour the expressiveness of their art over its realism.

1.3 HISTORICAL NOTES

In this paragraph, we'll go through a short summary of some of the key steps that have been taken in the research progress of physically based rendering. We will also see how PBR has been gradually adopted in film production.

3 The word "render" can be used to express two different concepts. When used as a *verb*, it is the process followed by the computer to produce our image. When used as a *noun*, it refers to the final image produced.

Physically based rendering research started to catch on only after the 1980s. An important seminal work was Whitted's 1980 paper, *An Improved Illumination Model for Shaded Display* (Whitted 1980), which has since served as a foundation for modern *ray-traced* computer graphics which feature *global illumination*.

PBR generally makes substantial use of *ray tracing*, and for this reason Chapter 3 will contain a description of the main ideas behind this technique. For now, let's just say that ray tracing involves - very surprisingly - tracing rays of light in a 3D scene, to determine the colors of objects. As for *global illumination*, we'll give a brief introduction to this topic as well in Chapter 3.

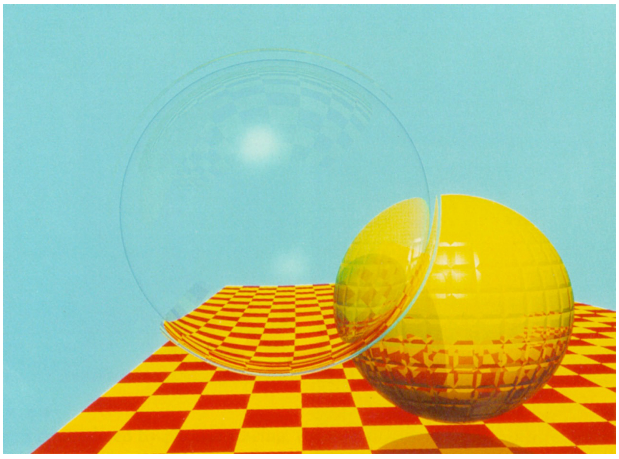

Fig. 3: One of the renders from Whitted's 1980 paper. ©ACM

We should mention that an important early contribution to PBR techniques was also given by Cook and Torrance, when they proposed a new reflection model based on *microfacet theory* (Cook and Torrance 1982). The techniques for realistic reflections described in Chapter 4 will be based on said theory.

Many other historic steps were necessary in order to make today's PBR techniques a reality. One of these was the work of Kajiya, who in 1986 introduced the *rendering equation* (Kajiya 1986). This equation describes in a concise and elegant way the problem solved by any realistic rendering application. More on the rendering equation will be said in Chapter 3.

Kajiya also introduced *path tracing*, which we can consider as an advanced form of *ray tracing* that properly takes into account *global illumination* effects, through the use of *Monte Carlo integration*.

Again, we'll mention more about this topic in Chapter 3. For now, let's just say that, to do proper realistic rendering, we need to approximate the values of many definite integrals. *Monte Carlo integration* is a technique, based on random numbers, for doing just that.

Monte Carlo integration really transformed the field, allowing the creation of images unlike any before. Many important contributions were made to Monte Carlo-based efforts during the years, one of the most important ones being Veach's 1997 PhD thesis. Veach advanced key theoretical foundations of Monte Carlo rendering, and also developed new algorithms that improved its efficiency, like *bidirectional path tracing* (Pharr, Jakob, and Humphreys 2023).

It took some time to incorporate physically based rendering techniques in film production, due to its computational costs. An early example of a movie made with Monte Carlo global illumination was the short film *Bunny* (1998), by Blue Sky Studios. Its visual look was substantially different from other films and shorts of the past. Before then, renderers were mostly based on *rasterization*, a rendering technique that's faster than ray tracing, but, in general, less realistic. *Reyes* is an important example of a *rasterization-based* architecture that had been used to generate photo-realistic images (Pharr, Jakob, and Humphreys 2023).

After *Bunny*, another watershed moment came in 2001, when Marcos Fajardo presented at the SIGGRAPH conference an early version of his *Arnold* renderer. *Arnold* was able to render scenes with complex geometry, textures, and global illumination in just tens of minutes (Pharr, Jakob, and Humphreys 2023). With the help of Sony Pictures Imageworks, *Arnold* was developed into a production-capable rendering system. It is now available as a commercial product.

In the early 2000s, Pixar's *RenderMan* renderer started to support hybrid rasterization and ray-tracing algorithms. It also introduced a number of innovative algorithms for computing global illumination solutions in complex scenes (Pharr, Jakob, and Humphreys 2023).

RenderMan was recently rewritten to be a physically based ray tracer. It was first employed in its new form for the production of the 2013 movie *Monsters University* (Hery and Villemin 2013).

1.4 THESIS OUTLINE

The rest of the thesis will be organized as follows.

Chapter 2 will be dedicated to the properties and behaviour of light, in a physical sense. Establishing a solid theoretical basis, borrowed from physics, is fundamental for PBR.

Chapter 3 will describe the algorithm at the base of our PBR rendering techniques; that is, *ray tracing*.

Chapter 4 is where we'll get into the main topic. We'll analyze some of the most popular theoretical models behind light reflection in state-of-the-art rendering software.

In Chapter 5 an implementation of the previously discussed techniques will be presented, in the form of a program called *BoxOfSunlight*. The focus here will be on the architectural choices made during development that make said program stand out.

Chapter 6 will be dedicated to the results achieved by *BoxOfSunlight*. We'll see how even just by changing the way light get reflected off a surface, it's possible to simulate the looks of a wide range of materials.

Finally, Chapter 7 will be dedicated to some of the shortcomings of *BoxOfSunlight*, and will contain ideas on how it could be extended into a more complete *physically-based rendering engine*.

Fig. 4: Monsters University, ©Disney/Pixar 2013

2

PHYSICS OF LIGHT

3

RAY TRACING

4

BRDF MODELS

5

A RAY TRACER IMPLEMENTATION

6

RESULTS

7

CONCLUSIONS AND FUTURE WORK

BIBLIOGRAPHY

Burley, Brent (2012). *Physically-based shading at Disney*. URL: https://blog.selfshadow.com/publications/s2012-shading-course/burley/s2012_pbs_disney_brdf_notes_v3.pdf. (accessed: 22.02.2025).

Cook, R. L. and K. E. Torrance (Jan. 1982). “A Reflectance Model for Computer Graphics”. In: *ACM Trans. Graph.* 1.1, pp. 7–24. ISSN: 0730-0301. DOI: 10.1145/357290.357293. URL: https://doi.org/10.1145/357290.357293.

Glassner, Andrew S. (1995). *Principles of Digital Image Synthesis*. San Francisco: Morgan Kaufmann.

Hery, Christophe and Ryusuke Villemin (2013). *Physically Based Lighting at Pixar*. URL: https://graphics.pixar.com/library/PhysicallyBasedLighting/paper.pdf. (accessed: 23.02.2025).

Hughes, John F. et al. (2014). *Computer graphics: principles and practice*. 3rd edition. Upper Saddle River, NJ: Addison-Wesley.

Kajiya, James T. (Aug. 1986). “The rendering equation”. In: *SIGGRAPH Comput. Graph.* 20.4, pp. 143–150. ISSN: 0097-8930. DOI: 10.1145/15886.15902. URL: https://doi.org/10.1145/15886.15902.

Pharr, Matt, Wenzel Jakob, and Greg Humphreys (2023). *Physically based rendering: From theory to implementation*. 4th edition. Cambridge, MA: MIT Press.

Whitted, Turner (June 1980). “An improved illumination model for shaded display”. In: *Commun. ACM* 23.6, pp. 343–349. ISSN: 0001-0782. DOI: 10.1145/358876.358882. URL: https://doi.org/10.1145/358876.358882.