UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LUIZA DE AZAMBUJA HAGEMANN

A Converter for Physically-Based Renderer Scenes

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Science

Advisor: Prof. Dr. Manuel Menezes de Oliveira Neto

Porto Alegre
March 2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Rui Vicente Oppermann
Vice-Reitora: Prof^{a}. Jane Fraga Tutikian
Pró-Reitor de Graduação: Prof. Wladimir Pinheiro do Nascimento
Diretora do Instituto de Informática: Prof^{a}. Carla Maria Dal Sasso Freitas
Coordenador do Curso de Ciência de Computação: Prof. Raul Fernando Weber
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

ABSTRACT

Placeholder.

Keywords: Physically-based rendering. scene conversion. PBRT. Mitsuba. LuxRender.

Um conversor de cenas para renderizadores fisicamente realísticos.

RESUMO

Placeholder.

Palavras-chave: Physically-based rendering. PBRT. Mitsuba. LuxRender..

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

PBR	Physically Based Rendering
PB	Physically-based

CONTENTS

1 INTRODUCTION

Ever since the development of modern Computer Graphics, one of the goals researchers aspired to was being able to synthesize images indistinguishable from real photographs. In order to produce physically accurate images, the process of image synthesis - also called **rendering** - simmulates the interaction of light with the representation of a three-dimensional scene. **Physically-Based Rendering** (PBR) is a complex process that requires thorough knowledge of optics, material properties, geometry and light propagation.

1.1 Physically Based Rendering (PBR)

Over the years, PBR became quite popular and was widely incorporated into the entertainment industries. From movies to videogames, from ads to interior design, PBR made it possible for artists to bring their creations and their vision one step closer to reality. Today, we can say that many - if not most - algorithms used in computer animation, geometric modeling and texturing require that their results be passed through some sort of rendering process.

As PBR popularity grew, a brand new market opened up for physically-based (PB) renderers. Following the creation of *PBRT* and the publishing of *"Physically Based Rendering: From Theory to Implementation"* (**??**), several other research-oriented renderers were created. Among them is *Mitsuba*, one of the renderers chosen for this research, which places strong emphasis on experimental rendering techniques.

Following the lead of Pixar's *Renderman* (**??**), many commercial and performance-oriented renderers appeared on the market. Focused on animation techniques and visual effects for movies, these renderers provide well-established, stable rendering techniques. These renderers, such as *LuxRender* (**??**) and *Octane* (**??**), are state-of-the-art renderers used by the animation and gaming industries.

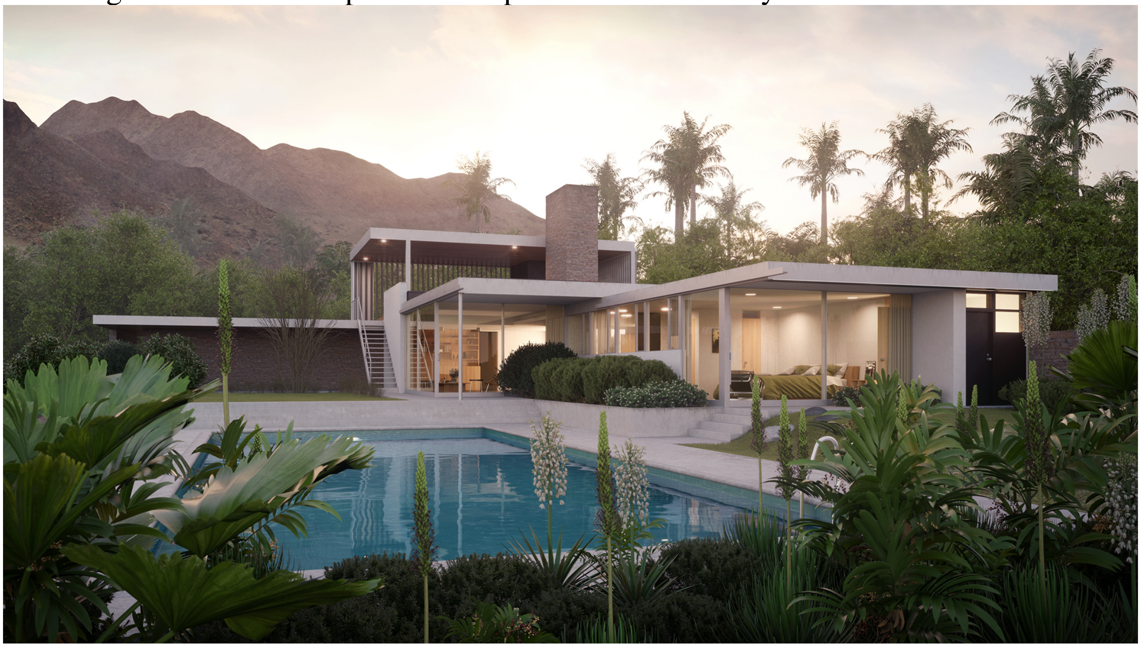
Even with different applications, the vast majority of modern PB renderers follows the same general guidelines for defining scene directives and world descriptions. Scene directives establish parameters such as which integration and sampling techniques the renderer must use, the view matrix and other camera properties. World descriptions state which objects compose the scene and which materials must be used to render them. This ensemble of descriptions is what, in PBR, is called a **scene**.

1.2 Rendering a Scene

<TO BE REWRITTEN> The process of rendering an image usually starts with a scene. But what defines a scene? In PBR, a scene is a collection of descriptions: of the scene's geometry (the objects), of the objects' materials and textures, of the light sources and of the rendering techniques the user would like to use to form the image. This description is then read and interpreted by the renderer, and the scene is then processed into an output image.

But even with the vast array of softwares at our disposal, creating scenes is still a complex process. For instance, scenes created for architecture or design-oriented softwares often compile hundreds of 3D models and dozens of customized materials and textures, as shown in Figure 1.1. Each material and texture has to be carefully defined, taking into account the target renderer's limitations and particularities. Crafting such scenes is an onerous, time-consuming effort: so much that

Figure 1.1: An example of a complex scene created by Laubwerk Plants Kits



<CONVERTING A SCENE>

2 RELATED WORKS

(https://github.com/RenderToolbox/RenderToolbox4/wiki/Overview)

3 THEORETICAL FOUNDATIONS

4 PROJECT

5 RESULT ANALYSIS

6 CONCLUSION