UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS no. 1375

Physically based rendering

Jure Ratković

Zagreb, April 2017

CONTENTS

1. Introduction **1**

2. Physics of light **2**

2.1. Light and material interaction 2

2.2. Light scattering 3

2.2.1. Scattering on a planar boundary 3

2.2.2. Reflection 5

2.2.3. Refraction 6

3. The mathematical model **8**

3.1. Reflectance equation 8

3.2. The BRDF 9

3.3. Cook-Torrance specular BRDF 9

3.3.1. Microfacet theory 9

3.3.2. Fresnel reflectance 10

3.3.3. Normal distribution term 12

3.3.4. Geometric term 13

3.4. Lambert's diffuse BRDF 14

4. Implementing PBR **16**

4.1. Gamma and high dynamic range 16

4.1.1. Gamma correction 16

4.1.2. High dynamic range 17

4.2. Solving the reflectance equation 17

4.2.1. Punctual light sources 17

4.2.2. Indirect lighting 19

4.3. Comparison to Blinn-Phong model 21

5. Conclusion **23**

Bibliography 24

1. Introduction

Physically based rendering, or PBR, is a set of shading models that attempt to achieve a higher level of realism and rendering quality through accurately modeling light and material interaction. Earlier shading models such as Phong and Blinn-Phong were ad hoc, meaning that they were conceived by observing objects under various lighting conditions and then inventing shading equations that would produce similar results. Contrary to this, in PBR the shading equations are derived from laws of light interaction. Physical concepts such as diffusion, reflection and conservation of energy are studied under more rigorous mathematical framework in order to produce physically correct shading equations.

2. Physics of light

This chapter will give a qualitative description of the physics underlying physically based shading models.

2.1. Light and material interaction

Light is an electromagnetic transverse wave, and as such its interaction with different materials is defined by electromagnetic properties of that material. The physical property that defines the interaction is the refractive index. This index is a complex number whose real part determines the speed of light in a given material, and the complex part determines how much light is absorbed as it propagates through the material.

The simplest form of light material interaction is light passing through homogeneous medium. Such media have a uniform index of refraction throughout their volume. If the complex part of the index is low, the media will appear transparent, and if it is high the media will absorb some wavelengths of light, giving the transparent media its color (Figure 2.1). This type of light material interaction is called *absorption*.

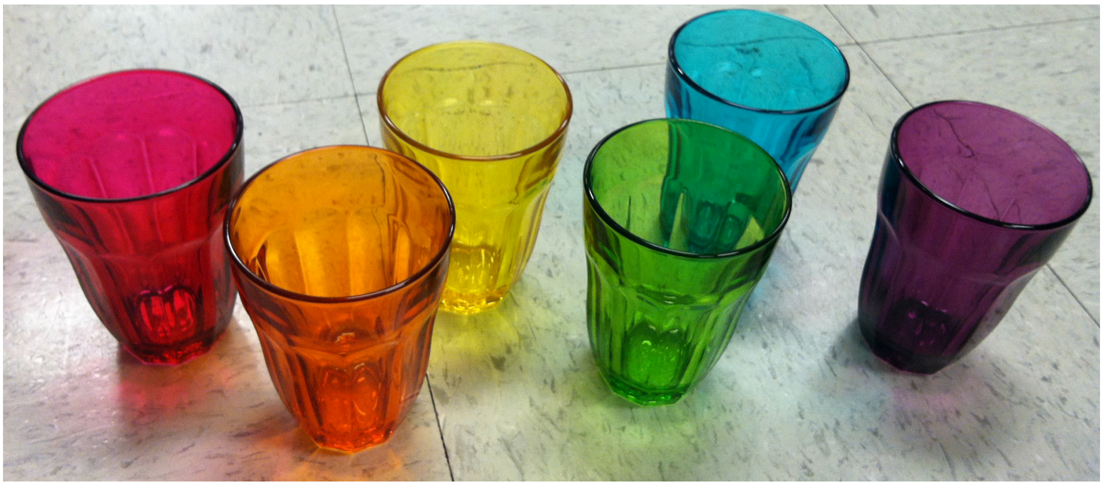

Figure 2.1: Light absorption in transparent media

If the media are heterogeneous the index of refraction changes abruptly over small distances[1]. In this case light hitting the media *scatters* over all possible outgoing directions. The distribution of the scattered light is often not uniform, but has spikes in certain directions, depending on surface properties. It is worth noting that all media scatter light to some degree over sufficiently large distances (Figure 2.2).

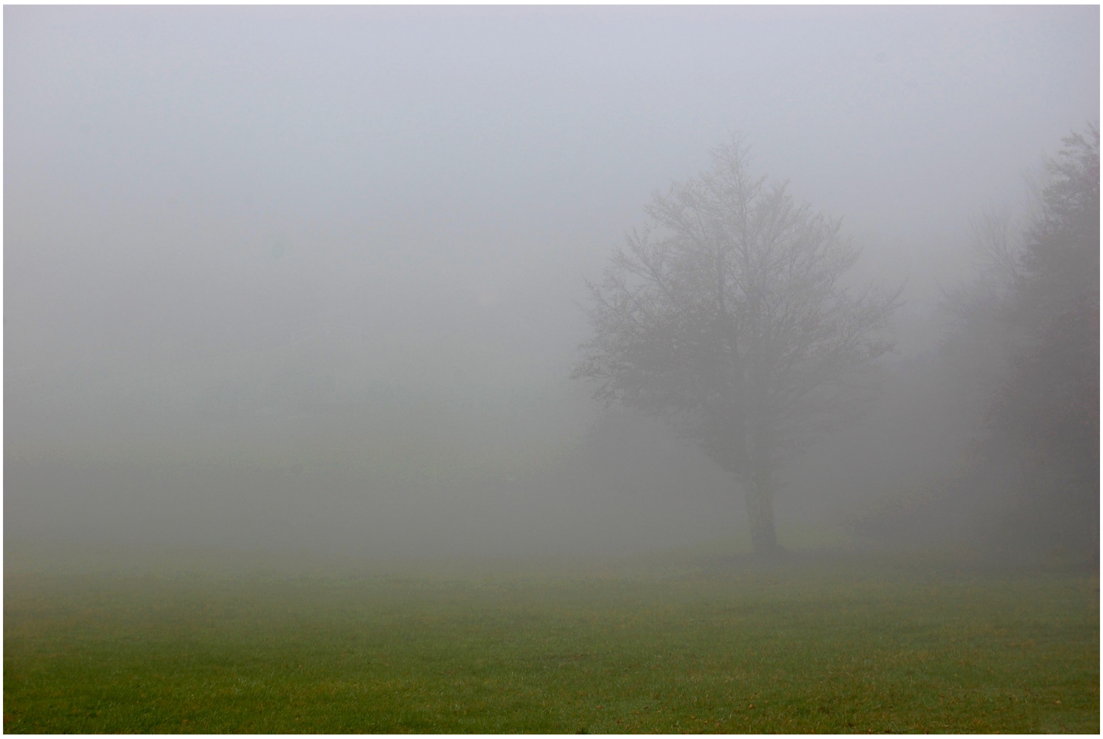

Figure 2.2: Air also scatters light

There is a third type of light and material interaction; *emission*. Emissive materials convert other forms of energy into light, like a tungsten light bulb. Emissive materials are not particularly relevant in shading, as light sources are often mathematically modeled.

2.2. Light scattering

2.2.1. Scattering on a planar boundary

Physically based shading models mostly do not consider light emission and absorption, but light scattering. In most cases the behavior of scattering light is very complex and it cannot be analytically solved. Fortunately, for shading purposes the only important

[1] Relative to the wavelength of light.

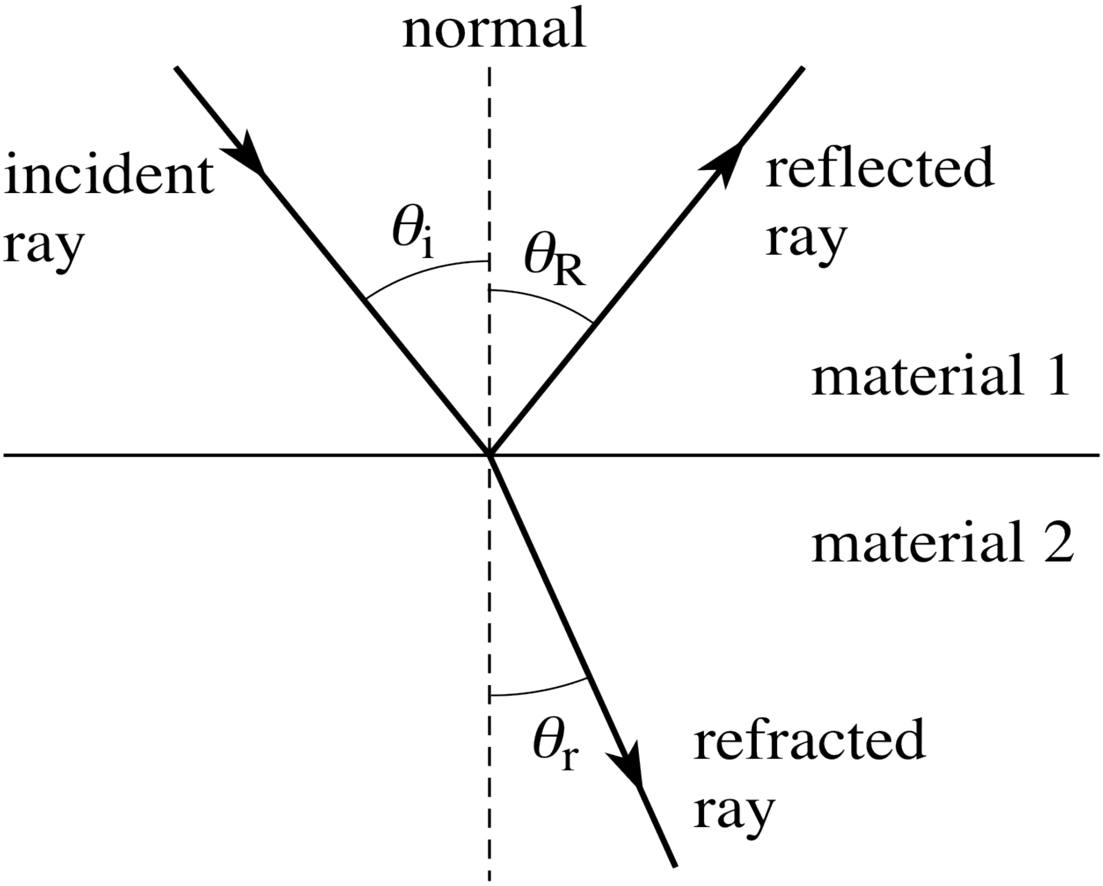

Figure 2.3: Light scattering at a planar boundary

case is scattering at an infinite planar boundary. This is because even the microscopic bumps on a material's surface can be treated as infinite and planar relative to the wavelength of light. In this case, the light does not scatter over all possible directions, but each ray splits into exactly two: the reflected and the refracted ray (Figure 2.3). Reflection angle θ_R is always equal to the angle of incidence θ_i. Refraction angle θ_r is defined by Snell's law:

$$\sin\theta_r = \frac{n_1 \sin\theta_i}{n_2}$$

where n_1 and n_2 are the refractive indices of material 1 and material 2, respectively. The energy between these rays is conserved; the total energy of reflected and refracted ray is equal to the energy of incident ray.[2] The energy ratio between the refracted and reflected rays is defined by the *Fresnel equations*, which will be discussed in later chapters.

[2] Some energy is lost to heat or other types of radiation, but this is relevant only in global illumination models, where light rays bounce multiple times.

2.2.2. Reflection

In the real world, not many surfaces are polished to mirror like accuracy. Most of them have microscopic "bumps", which are much smaller than a pixel, but much larger than the wavelength of light. These kinds of surfaces are modeled as a large collection of microscopic, optically flat surfaces. Each of these tiny surfaces reflect light at a different angle. (Figure 2.4).

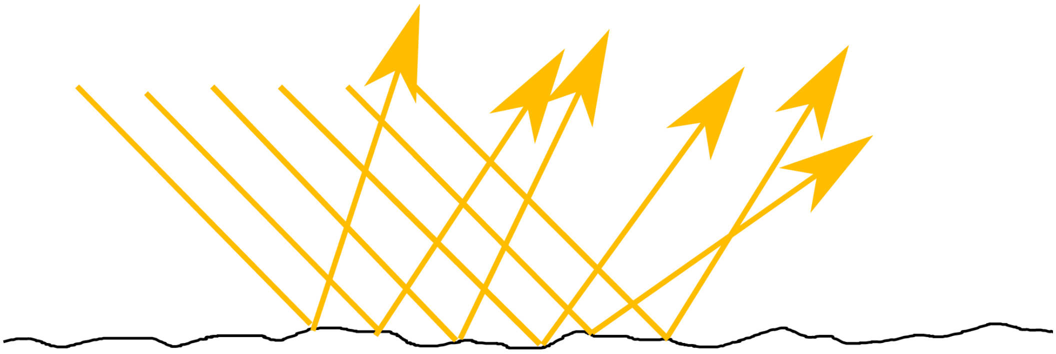

Figure 2.4: Light reflection on a rough surface

In PBR models, this surface property is parametrized by *roughness*. If a surface is optically flat, the value of this parameter is 0, whereas the value of 1 signals maximum roughness[3]. It is worth noting that the surfaces that are rougher in this sense appear equally smooth to the human eye or touch. For example, the two objects in figure 2.5 would appear equally smooth upon touching them, but the object on the right is rougher on a microscopic scale.

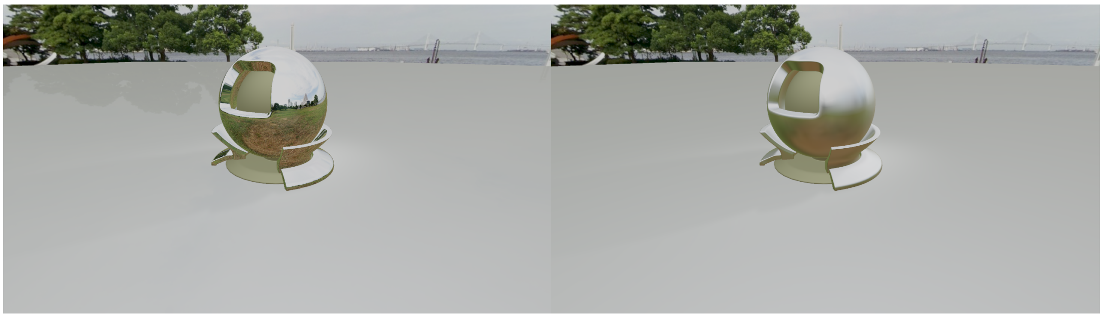

Figure 2.5: Microscopic surface roughness

[3] In this case reflection is scattered uniformly over all directions.

2.2.3. Refraction

What happens to the refracted light is determined by the material's absorptivity and composition. Metals have very high absorption rates, so almost no refracted light comes out of the material. On the other hand, materials like glass have very low absorption rates, so the refracted light is able to pass through the entire volume of the object without being absorbed.

Between those two extremes are *dielectric* materials. Dielectrics absorb some amount of light, but their composition does not allow light to pass through the material uninterrupted, like in glass. Instead, light is scattered against particles beneath the object's surface, and some light gets re-emitted out of the same surface. This phenomena is called *subsurface scattering* or *diffusion*.

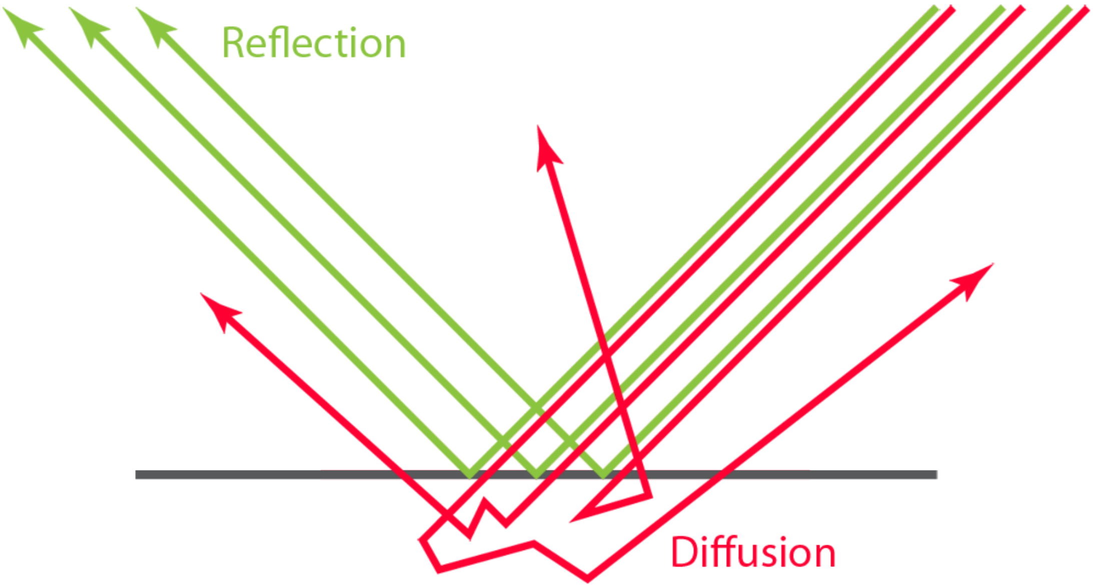

Figure 2.6: Diffusion and reflection

The modeling of diffusion in shading depends on the scale of scattering beneath the surface. If the distance between surface entry and exit points for all rays is smaller than the size of a pixel, the contribution of diffusion can be computed locally. Otherwise, special non-local subsurface scattering rendering techniques should be used [1]. Examples of such materials are human skin, wax and any sufficiently thin dielectric material (Figure 2.7).

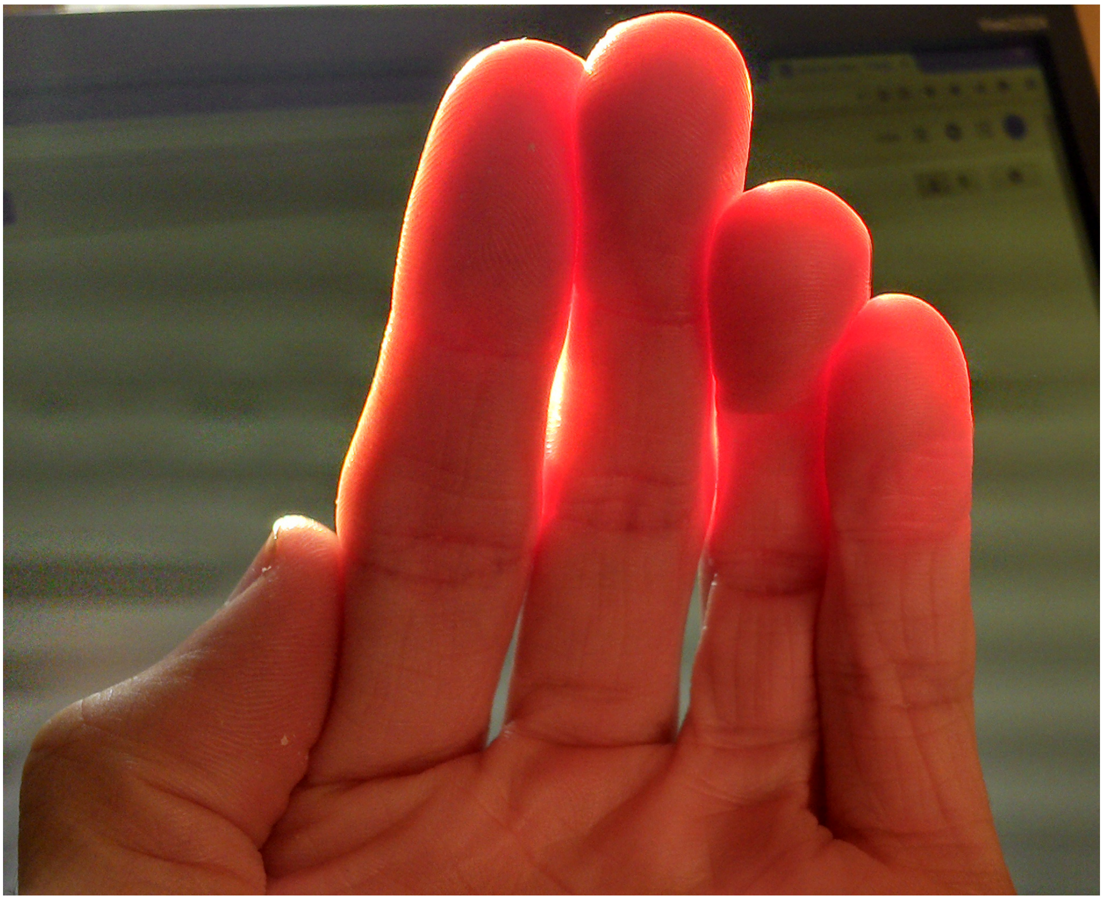

Figure 2.7: Subsurface scattering is cleary visible on human skin

3. The mathematical model

This chapter will give a quantitative mathematical model for physically based shading.

3.1. Reflectance equation

The main radiometric quantity used for shading purposes is *radiance* (symbol L). Although radiance is a spectral quantity, for rendering purposes it is most often stored as an RGB triplet. The outgoing radiance at a certain point (L_o) is a function of the ingoing radiance at that point (L_i). The model most commonly used to describe this is the *reflectance equation*:

$$L_o(\boldsymbol{\omega}_o) = \int_\Omega f(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \otimes L_i(\boldsymbol{\omega}_i)(\underline{\boldsymbol{n} \cdot \boldsymbol{\omega}_i})\, d\boldsymbol{\omega}_i \tag{3.1}$$

Outgoing radiance is given for an outgoing direction $\boldsymbol{\omega}_o$. For local illumination shading $\boldsymbol{\omega}_o$ is usually the view vector $\boldsymbol{v}$, a unit vector pointing from the shading position to the camera. Ingoing radiance $\boldsymbol{\omega}_i$ is the light vector $\boldsymbol{l}$, pointing from the shading position to the light source. The domain of the integral Ω is a hemisphere defined by the surface normal $\boldsymbol{n}$.

As seen in the equation, outgoing radiance is proportional to the sum of the incoming radiance from every direction on the hemisphere weighted by the angle of incidence between the incoming ray and the surface normal (the *cosine factor*)[1]. The ingoing radiance is also weighted by the *bidirectional reflectance distribution function* $f(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i)$ (the BRDF). This function takes as input incoming and outgoing ray, and outputs a weight of how much the incoming ray is contributing to the final outgoing radiance.[2] The differences between PBR models come from the choice and the construction of the BRDF.

[1] The underlining on $\underline{\boldsymbol{n} \cdot \boldsymbol{\omega}_i}$ represents clamping to 0. This is done because angles whose cosine is less than 0 are outside of the hemisphere around the surface normal.

[2] The function outputs an RGB vector of weights. Hence, the $\otimes$ symbol represents per component vector multiplication.

3.2. The BRDF

The BRDF can be interpreted as specifying the distribution of scattered and reflected light for a given incoming light direction, and is parameterized by four angles[3]. To be physically plausible, the BRDF must be *reciprocal*:

$$f(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) = f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$$

and must *conserve energy*:

$$\forall \boldsymbol{\omega}_i, \int_{\Omega} f(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \underline{(\boldsymbol{n} \cdot \boldsymbol{\omega}_i)} \, d\boldsymbol{\omega}_o \leq 1$$

The integral of the BRDF multiplied by the cosine factor over all possible outgoing directions $\boldsymbol{\omega}_o$ must not exceed one, for all possible ingoing directions $\boldsymbol{\omega}_i$. This simply means that the amount of outgoing light must not exceed the amount of incoming light.

As seen in figure 2.6, reflection and subsurface scattering are very different phenomena, and often each one is modeled by a separate BRDF. The term describing subsurface scattering is called the *diffuse term*, while the term describing reflection is called the *specular term*. The two most common BRDFs in real time production shading are Lambert's BRDF for the diffuse and Cook-Torrance for the specular term.

3.3. Cook-Torrance specular BRDF

The Cook-Torrance [2] specular BRDF was conceived in 1981 and has since been widely used in off-line and film rendering. In recent years graphics processors have become powerful enough to run the Cook-Torrance model in real time, which caused the wide adoption of PBR in real time and video game rendering.

3.3.1. Microfacet theory

The Cook-Torrance BRDF is based on the microfacet theory [3]. In this theory surfaces are treated as a collection of tiny optically flat surfaces; microfacets. Each of these microfacets has its surface normal, $\boldsymbol{m}$. Furthermore, when shading a surface point, the incoming light vector $\boldsymbol{l}$ and the outgoing view direction $\boldsymbol{v}$ are given. What needs to be computed is the fraction of light coming from the $\boldsymbol{l}$ direction that is reflected in the $\boldsymbol{v}$

[3] $\boldsymbol{\omega_i}$ and $\boldsymbol{\omega_o}$ are unit vectors, and as such are defined by their two polar angles, so four angles in total. It is also worth noting that BRDFs are often parametrized by fewer angles, for example Lambert's BRDF does not depend on $\boldsymbol{\omega_i}$ and Cook-Torrance is isotropic.

direction. Only the microfacets whose surface normal $\boldsymbol{m}$ is equal to the *half vector* $\boldsymbol{h}$ between $\boldsymbol{l}$ and $\boldsymbol{v}$ can contribute to the reflection.

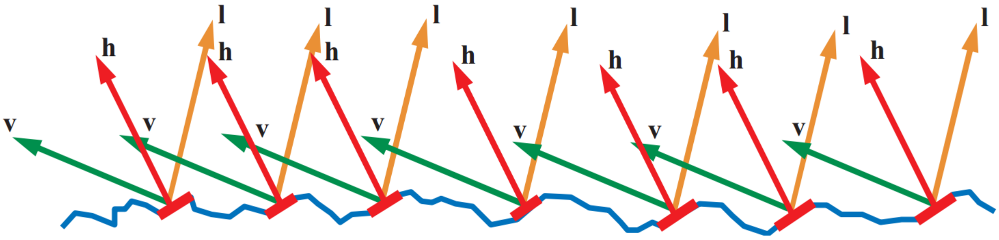

Figure 3.1: Only the microfacets for which $\boldsymbol{m} = \boldsymbol{h}$ can contribute to the reflection

The incoming light to the microfacet and the outgoing light from the microfacet can be obstructed by the nearby deformities on the surface. These phenomena are called shadowing and masking, respectively. Shadowed and masked microfacets do not contribute to the reflection.[4] All microfacets that contribute to the reflection are called *active microfacets*. Cook-Torrance BRDF is derived from these assumptions and has the following form:

$$f_{Cook-Torrance}(\boldsymbol{v}, \boldsymbol{l}) = \frac{F(\boldsymbol{l}, \boldsymbol{h})G(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{h})D(\boldsymbol{h})}{4(\boldsymbol{n} \cdot \boldsymbol{l})(\boldsymbol{n} \cdot \boldsymbol{v})}$$

In the equation, $F(\boldsymbol{l}, \boldsymbol{h})$ is the Fresnel reflectance term, $G(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{h})$ is the shadowing and masking term, $D(\boldsymbol{h})$ is the normal distribution term and $4(\boldsymbol{n} \cdot \boldsymbol{l})(\boldsymbol{n} \cdot \boldsymbol{v})$ is the normalization factor.

3.3.2. Fresnel reflectance

Fresnel equations describe behaviour of light when passing through materials with varying indices of refraction. For shading purposes, the Fresnel term computes the percentage of reflected light as a function of the light's angle of incidence and the material's index of refraction. At the angle of $0°$ each material reflects a specific amount of light, $\boldsymbol{F}_0$, which is its base reflectivity[5]. For angles up to approximately $45°$ the reflectivity remains relatively constant, and then it rapidly approaches 100% as the angle

[4] The light from masked microfacets bounces off the surface multiple times and eventually does bounce off it, possibly contributing to the reflection. In most cases not accounting for this does not cause visible errors.

[5] Usually called metallic or specular color, this value is a representation of the material's index of refraction.

approaches $90°$. This effect can be seen in figure 3.2; the plastic ball does not reflect the environment around its center area, but does near the edges.

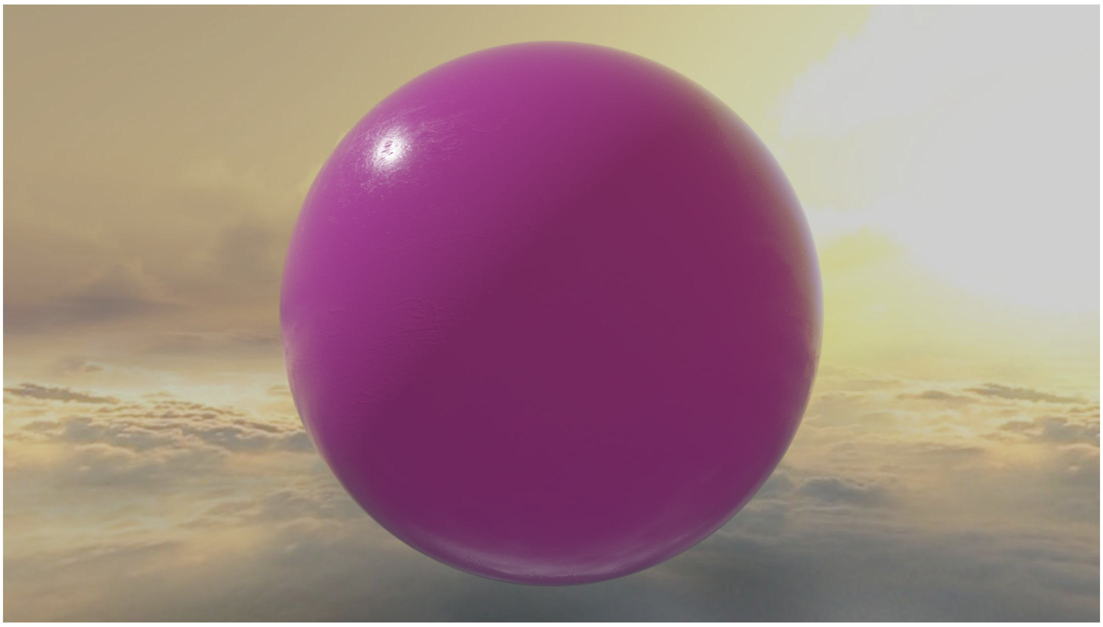

Figure 3.2: Fresnel effect can be seen towards the edges of the ball

The approximation used to model this behavior is the Schlick's function:

$$F_{Schlick}(\boldsymbol{F}_0, \theta) = \boldsymbol{F}_0 + (1 - \boldsymbol{F}_0)(1 - \cos(\theta)^5)$$

where in the case of specular BRDFs and microfacets the angle of incidence is:

$$\cos(\theta) = \boldsymbol{l} \cdot \boldsymbol{h}$$

$$F_{Schlick}(\boldsymbol{F}_0, \boldsymbol{l}, \boldsymbol{h}) = \boldsymbol{F}_0 + (1 - \boldsymbol{F}_0)(1 - (\boldsymbol{l} \cdot \boldsymbol{h})^5)$$

For some materials base reflectance can be a function of the incoming light's wavelength. This phenomenon gives metals like gold and copper its color (Figure 3.3). In PBR this is modeled by making $\boldsymbol{F}_0$ an RGB triplet.

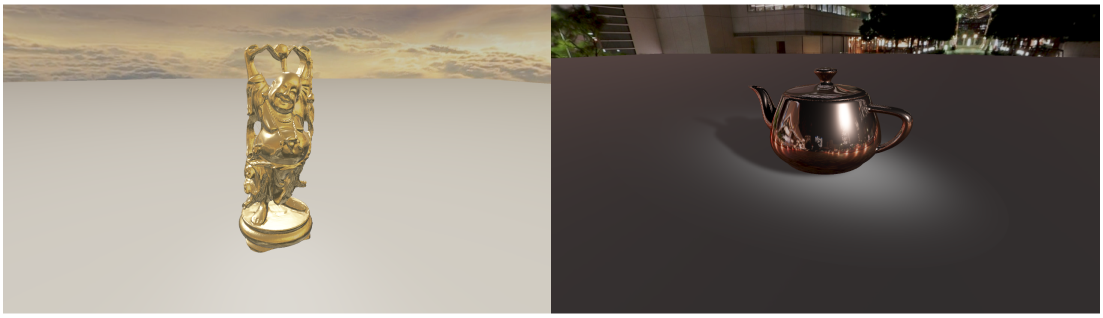

Figure 3.3: Gold (left) and copper (right) have colored reflections

3.3.3. Normal distribution term

The normal distribution term, or NDF, is a scalar function that gives the percentage of microfacet normals oriented towards the direction $\boldsymbol{h}$. This percentage tends to be higher if $\boldsymbol{h}$ is closer to the macroscopic surface normal $\boldsymbol{n}$. Models for the normal distribution function are usually parameterized by surface roughness[6] α, a value between 0 and 1[7]. The NDF must contain a normalization factor in order to conserve energy, which is derived from the following constraint [4]:

$$\int_{\Omega} D(\boldsymbol{h})\underline{(\boldsymbol{n} \cdot \boldsymbol{h})}\, d\boldsymbol{\omega}_i = 1$$

In the isotropic case all NDF models have the normalization factor of $\frac{1}{\alpha^2\pi}$. The most used NDF model is the Trowbridge-Reitz GGX [5]:

$$D_{GGX}(\boldsymbol{h}, \alpha) = \frac{\alpha^2}{\pi((\boldsymbol{n} \cdot \boldsymbol{h})^2(\alpha^2 - 1) + 1)^2}$$

Another prominent model is Beckmann's NDF:

$$D_{Beckmann}(\boldsymbol{h}, \alpha) = \frac{1}{\pi\alpha^2(\boldsymbol{n} \cdot \boldsymbol{h})^4} \exp\left(\frac{(\boldsymbol{n} \cdot \boldsymbol{h})^2 - 1}{\alpha^2(\boldsymbol{n} \cdot \boldsymbol{h})^2}\right)$$

The effects of the roughness parameter can be seen in figure 3.4:

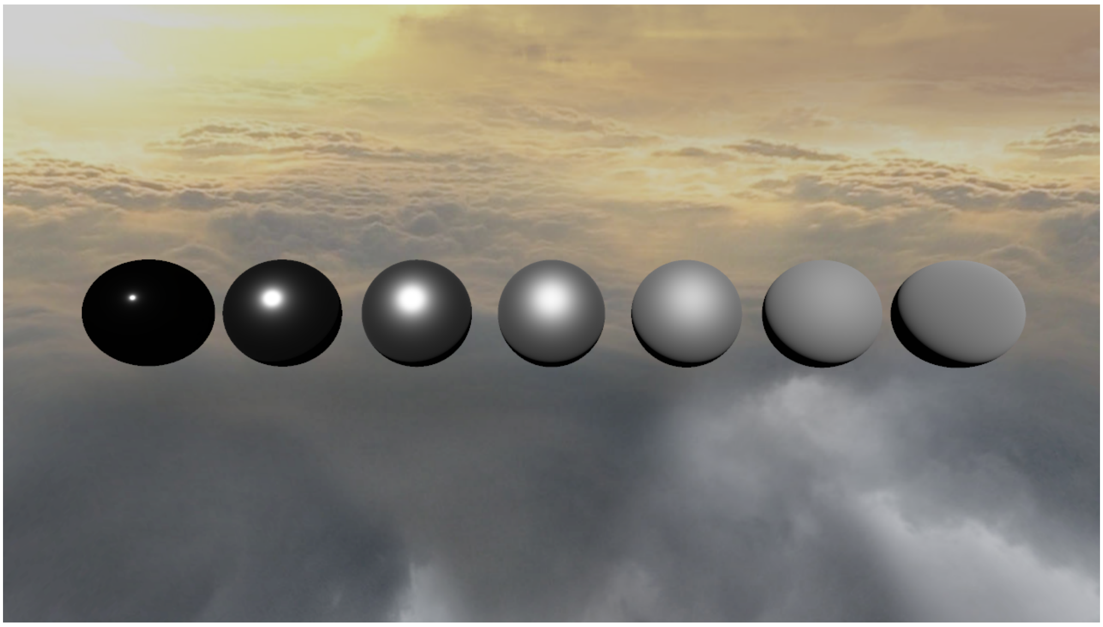

Figure 3.4: Rougness from 0 (left) to 1 (right)

[6] Anisotropic BRDFs have two surface roughnesses, α_x and α_y.

[7] α of 0 represents an optically flat surface, and α of 1 represents maximally rough surface.

3.3.4. Geometric term

The geometric, or the shadow masking term, is a function that outputs the probability that microfacets with the given normal $\boldsymbol{h}$ will be able to reflect light from incoming to outgoing direction. In other words, the probability that microfacets are not masked or shadowed by other microfacets. As with the NDF, there are many theoretical models for this function [6]. The most prominent model is the Smith's function, which splits G into light and view components:

$$G_{Smith}(\boldsymbol{l}, \boldsymbol{v}, \boldsymbol{h}) = G_1(\boldsymbol{l})G_1(\boldsymbol{v})$$

Where G_1 can be some other model, like GGX:

$$G_{GGX}(\boldsymbol{e}) = \frac{2(\boldsymbol{n} \cdot \boldsymbol{e})}{(\boldsymbol{n} \cdot \boldsymbol{e}) + \sqrt{\alpha^2 + (1 + \alpha^2)(\boldsymbol{n} \cdot \boldsymbol{e})^2}}$$

or Beckmann:

$$c = \frac{\boldsymbol{n} \cdot \boldsymbol{e}}{\alpha\sqrt{1 - (\boldsymbol{n} \cdot \boldsymbol{e})^2}}$$

$$G_{Beckmann}(\boldsymbol{e}) = \begin{cases} \frac{3.535c + 2.181c^2}{1 + 2.276c + 2.577c^2} & \text{if } c < 1.6 \\ 1 & \text{if } c \geq 1.6 \end{cases}$$

With this the analysis of the Cook-Torrance specular BRDF is complete. The difference between Beckmann and GGX terms can be seen in figures 3.5, 3.6 and 3.7.

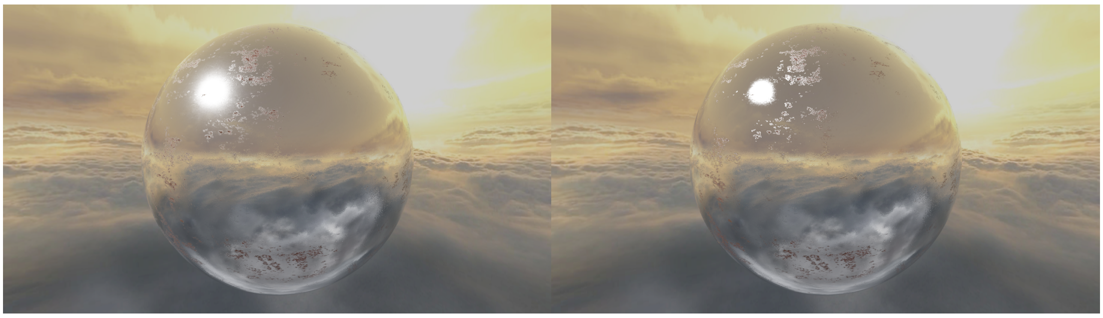

Figure 3.5: GGX (left) and Beckmann (right) on a slightly rusted steel material

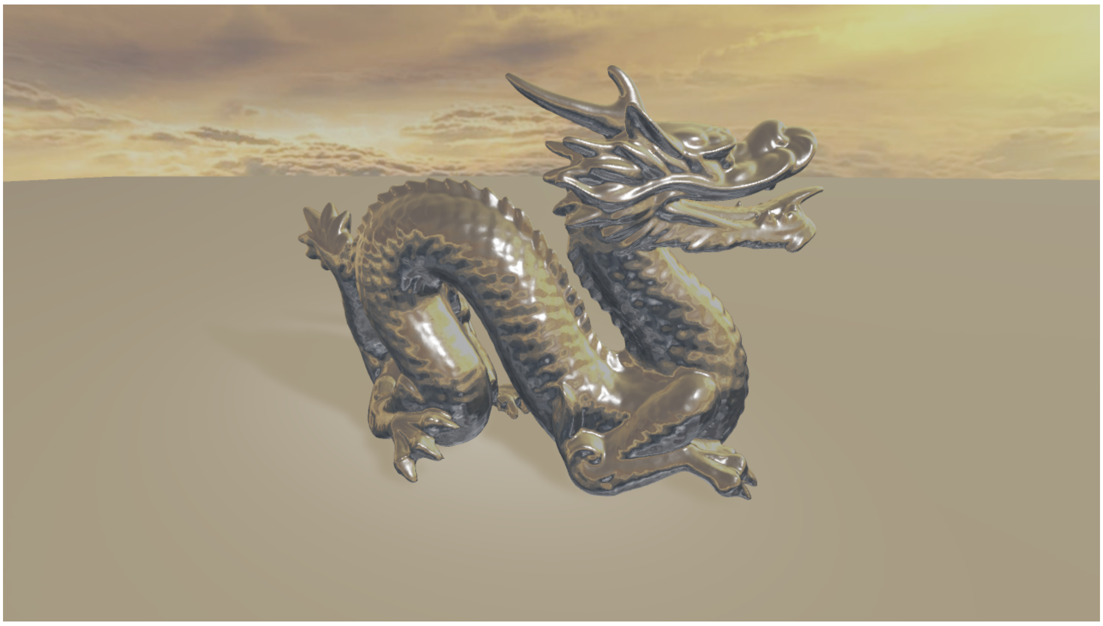

Figure 3.6: GGX terms on iron material

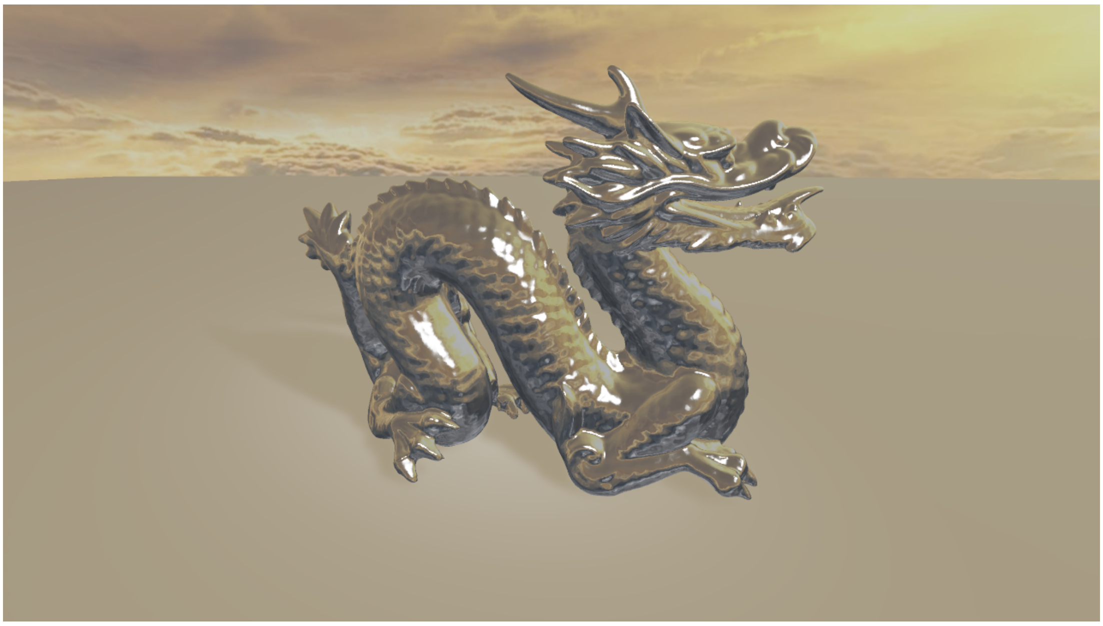

Figure 3.7: Beckmann terms on iron material

3.4. Lambert's diffuse BRDF

In real time rendering, the most often used diffuse BRDF has been Lambert's BRDF:

$$f_{Lambert}(\boldsymbol{v}, \boldsymbol{l}) = \frac{\boldsymbol{c}_{diff}}{\pi} \tag{3.2}$$

This BRDF is in fact a constant, meaning that it models a uniform diffusion; incoming light scatters over all possible directions equally likely. $\boldsymbol{c}_{diff}$ is the fraction of light that will scatter out of the surface. This parameter is often called albedo or diffuse color, and it closely corresponds to what is usually thought of as material's color.

The BRDF used in the reflectance equation consists of the specular and the diffuse BRDFs, but they must be mixed in such a way that energy conservation is not broken. This can be achieved in a multitude of ways, most commonly linearly interpolating between the BRDFs with a factor that is proportional to the reflectivity of the object at the given shading point. A factor that satisfies this property is the diffuse Fresnel term[8]:

$$\cos(\theta) = \boldsymbol{v} \cdot \boldsymbol{n}$$

$$F_{diff}(\boldsymbol{F}_0, \boldsymbol{v}, \boldsymbol{n}) = \boldsymbol{F}_0 + (1 - \boldsymbol{F}_0)(1 - (\boldsymbol{v} \cdot \boldsymbol{n})^5)$$

Finally, the full BRDF is:

$$f(\boldsymbol{v}, \boldsymbol{l}) = f_{Lambert}(\boldsymbol{v}, \boldsymbol{l})(1 - F_{diff}) + f_{Cook-Torrance}(\boldsymbol{v}, \boldsymbol{l})F_{diff}$$

In figure 3.8 the value of $\boldsymbol{F}_0$ is changing from 0 on the left to 1 on the right with constant values for diffuse color and roughness. It can be seen how the sphere's appearance changes from dielectric to metallic.

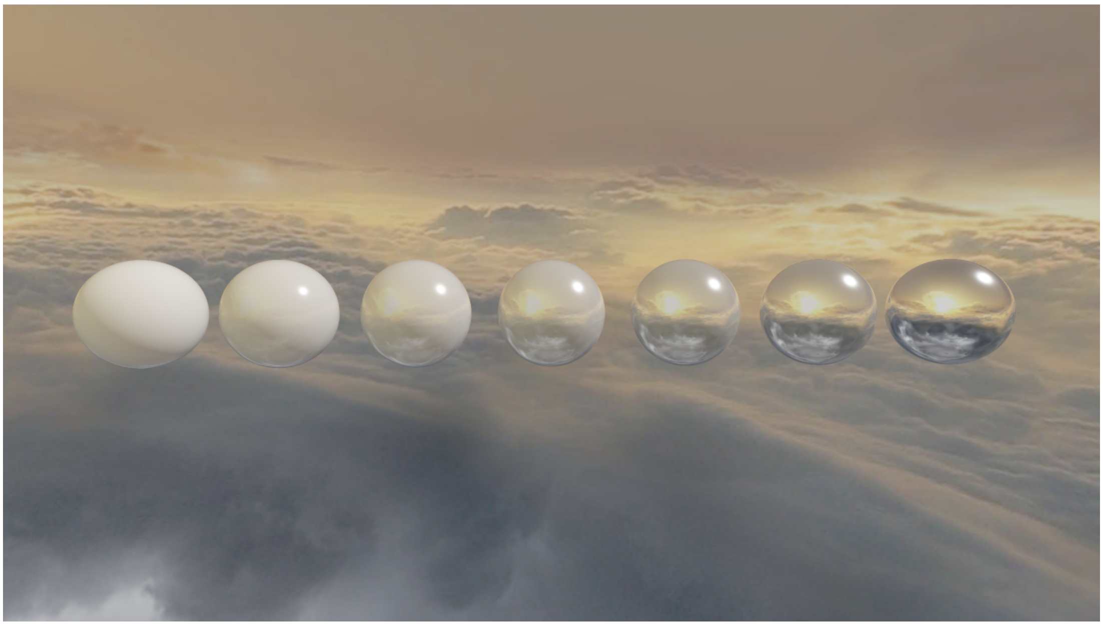

Figure 3.8: Changing $\boldsymbol{F}_0$ with constant $\boldsymbol{c}_{diff}$

[8] The difference between diffuse and specular Fresnel terms is that the specular term uses $\cos(\theta) = \boldsymbol{l} \cdot \boldsymbol{h}$, and diffuse uses $\cos(\theta) = \boldsymbol{v} \cdot \boldsymbol{n}$.

4. Implementing PBR

This chapter will give an implementation example for the mathematical models discussed in the previous chapter.

4.1. Gamma and high dynamic range

Before going through implementation details, the importance of gamma correction and high dynamic range (hereafter: HDR) must be stressed.

4.1.1. Gamma correction

Textures containing color data (diffuse and specular textures) are encoded in the non-linear sRGB space, as this is more suitable for viewing by human eyes [7]. These textures need to be linearized before inputting them into the shading equations or the results will be incorrect. Similarly, the final color data must be gamma corrected before outputting it to the monitor for the user to see. The effects of gamma correction can be seen in figure 4.1, where on the left image the boundary between lit and unlit area is too soft.

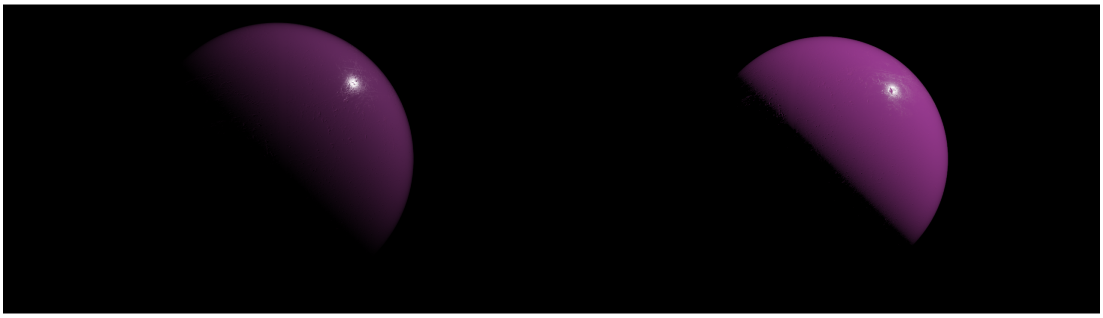

Figure 4.1: Incorrect gamma (left) and correct (right)

Linearization is done by raising the color values to the power of gamma, or raising them to the reciprocal of gamma in the case of gamma correction.

4.1.2. High dynamic range

On a sunny day, the intensity of light coming from the Sun is thousands od times grater than the intensity of any other light source. This wide intensity range is impossible to model using the standard 8 bit per channel light accumulation buffers. This problem can be solved by using high dynamic range. Instead of 8 bits per channel, a 16 bit (or grater) floating point accumulation buffer is used. Unlike the 8 bit buffers, floating point buffers allow the accumulated value to be grater than 1. After the light accumulation step, the unbounded values in the buffer must be remapped to $[0, 1]$ range for monitor output. Functions used to achieve this are called tone maps, an example of which is:

$$T(\boldsymbol{c}_{hdr}) = 1 - \exp(-\boldsymbol{c}_{hdr} * exposure)$$

The $exposure$ parameter is analogous to exposure in photography i.e. the amount of light per unit area. This parameter can be derived from average screen brightness or can be left to artists for tweaking. The effects of HDR and tone mapping can be seen in figure 4.2.

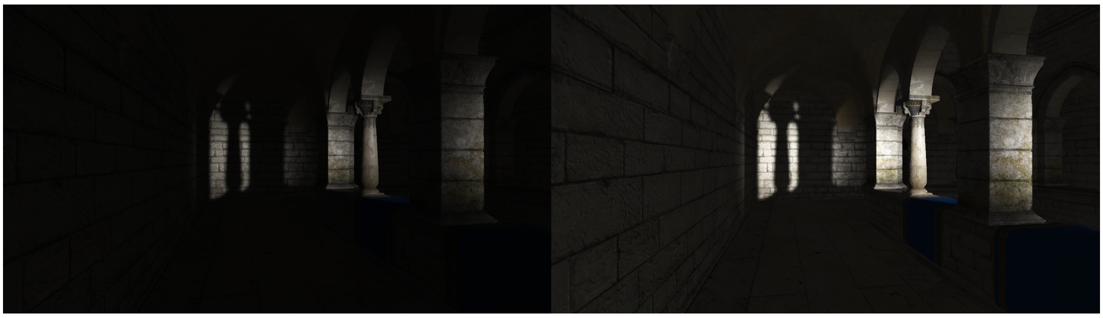

Figure 4.2: HDR with low exposure (left) and high exposure (right)

4.2. Solving the reflectance equation

This section will give some possible solutions and approximations to the rendering equation.

4.2.1. Punctual light sources

To fully solve the reflectance equation, light incoming from all possible directions around the normal-defined hemisphere should be integrated, which is obviously not possible in real time. The equation can be solved for punctual light sources, like point, spot and directional lights. When lit by a punctual light, every shading point receives light from a single direction $\boldsymbol{l}_c$:

$$L_i(\boldsymbol{l}) = 0, \forall \boldsymbol{l} \neq \boldsymbol{l}_c \tag{4.1}$$

Applying (4.1) to (3.1) removes the integral:

$$L_o(\boldsymbol{v}) = f(\boldsymbol{v}, \boldsymbol{l}_c) \otimes L_i(\boldsymbol{l}_c)(\underline{\boldsymbol{n} \cdot \boldsymbol{l}_c}) \tag{4.2}$$

Punctual lights are parametrized by $\boldsymbol{c}_{light}$, the color a white lambertian surface would have when lit by the light from the direction perpendicular to the surface normal.

$$\boldsymbol{l}_c = \boldsymbol{n} \tag{4.3}$$

$$\boldsymbol{c}_{light} = \int_{\Omega} f_{Lambert}(\boldsymbol{v}, \boldsymbol{\omega}_i) \otimes L_i(\boldsymbol{\omega}_i)(\underline{\boldsymbol{n} \cdot \boldsymbol{\omega}_i})\, d\boldsymbol{\omega}_i \tag{4.4}$$

Inserting (3.2) into (4.4) with $\boldsymbol{c}_{diff} = 1$:

$$\boldsymbol{c}_{light} = \frac{1}{\pi} \int_{\Omega} L_i(\boldsymbol{\omega}_i)(\underline{\boldsymbol{n} \cdot \boldsymbol{\omega}_i})\, d\boldsymbol{\omega}_i \tag{4.5}$$

Applying (4.1) and (4.3) to (4.5):

$$\boldsymbol{c}_{light} = \frac{1}{\pi} L_i(\boldsymbol{l}_c) \tag{4.6}$$

$$L_i(\boldsymbol{l}_c) = \pi \boldsymbol{c}_{light} \tag{4.7}$$

Inserting (4.7) into (4.2) gives the final punctual light shading equation:

$$L_o(\boldsymbol{v}) = \pi f(\boldsymbol{v}, \boldsymbol{l}_c) \otimes \boldsymbol{c}_{light}(\underline{\boldsymbol{n} \cdot \boldsymbol{l}_c}) \tag{4.8}$$

The equation above must be computed for each punctual light on the scene, and the results must be summed. To model light attenuation, $\boldsymbol{c}_{light}$ can be made into a function of distance to the light source:

$$\boldsymbol{c}_{light}(d) = \frac{\boldsymbol{c}_{light_0}}{a_0 d^2 + a_1 d + a_2}$$

4.2.2. Indirect lighting

Lighting a scene with only punctual lights can be problematic, as unlit areas would remain completely dark. Likewise, highly specular objects would scatter almost no light when their surface normal is at a high angle to the light vector. These problems can be seen in figure 4.3.

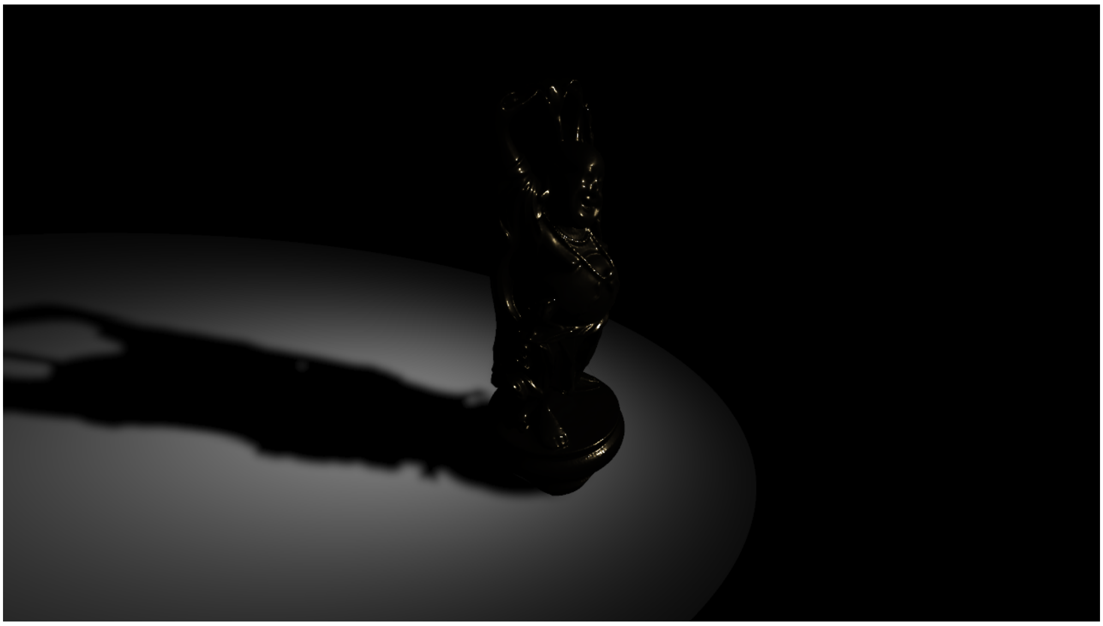

Figure 4.3: Highly specular material lit by a single spot light

Possible solutions to this problem include ambient and environment based lighting. Ambient lighting can be as simple as adding a constant value to the results of shading in order to brighten the scene, but this approach does not produce attractive results.[1]

On the other hand, environment lighting techniques can produce good results [8]. The environment cube map (skybox) can be treated as a radiance map. Each pixel of a radiance map represents a light that is infinitely far away and whose color ($\boldsymbol{c}_{light}$) is the pixel's color. By applying these arguments and Lambert's BRDF, the reflectance equation becomes a function of only the surface normal:

$$L_o(\boldsymbol{n}) = I(\boldsymbol{n})\boldsymbol{c}_{diff} \tag{4.9}$$

$$I(\boldsymbol{n}) = \frac{1}{\pi} \int_{\Omega} L_i(\boldsymbol{\omega}_i)(\underline{\boldsymbol{n} \cdot \boldsymbol{\omega}_i})\, d\boldsymbol{\omega}_i \tag{4.10}$$

[1] Other options for ambient lighting are light probes and spherical harmonics, but these techniques are outside of the scope of this paper.

$I(\boldsymbol{n})$ is irradiance, the sum of all incoming light hitting the hemisphere defined by $\boldsymbol{n}$. This value can be precomputed for various values of $\boldsymbol{n}$. The results of the precomputation are stored in the irradiance map, a cube map that has the value of $I(\boldsymbol{t})$ at texture coordinates $\boldsymbol{t}$.

In order to compute $I(\boldsymbol{n})$, equation (4.10) is expressed in polar coordinates:

$$I(\boldsymbol{n}) = \frac{1}{\pi} \int_{\phi} \int_{\theta} L_i(\phi_i, \theta_i) \cos(\theta_i) \sin(\theta_i) \, d\theta_i \, d\phi_i \tag{4.11}$$

The integrals are solved by applying Monte Carlo integration on each one:

$$I(\boldsymbol{n}) = \frac{1}{\pi} \frac{2\pi}{N_1} \frac{\pi}{2N_2} \sum^{N_1} \sum^{N_2} L_i(\phi_i, \theta_i) \cos(\theta_i) \sin(\theta_i) \tag{4.12}$$

$$I(\boldsymbol{n}) = \frac{\pi}{N_1 N_2} \sum^{N_1} \sum^{N_2} L_i(\phi_i, \theta_i) \cos(\theta_i) \sin(\theta_i) \tag{4.13}$$

Where N_1 and N_2 are the number of samples for each angle. The incoming radiance $L_i(\phi_i, \theta_i)$ is a sample of the radiance map at texture coordinates $\boldsymbol{t}$, where:

$$\boldsymbol{t} = (\sin(\theta)\cos(\phi), \sin(\theta)\cos(\phi), \cos(\theta))$$

The effects of irradiance mapping can be seen in figure 4.4, where the unlit areas can be seen receiving illumination from the environment map.

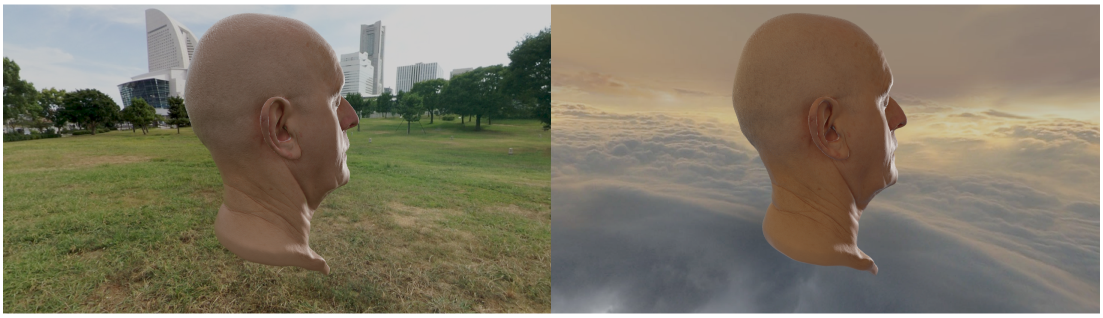

Figure 4.4: Same object in different environments

It is not possible to precompute specular image based lighting as it depends on view direction [9] [10]. In this case, the radiance map should be sampled at runtime and the specular BRDF should be computed for each sample. However, this is not possible in real time applications. The often used alternative is computing classic environment mapping for specular objects, and blurring the reflection depending on the material's roughness.[2] The blurring can be achieved by computing the mipmaps of

[2] Rougher materials should have blurrier reflections.

the environment map, and then simply selecting the mipmap level based on the pixel roughness during environment map sampling. The results of this method can be seen in figure 4.5.

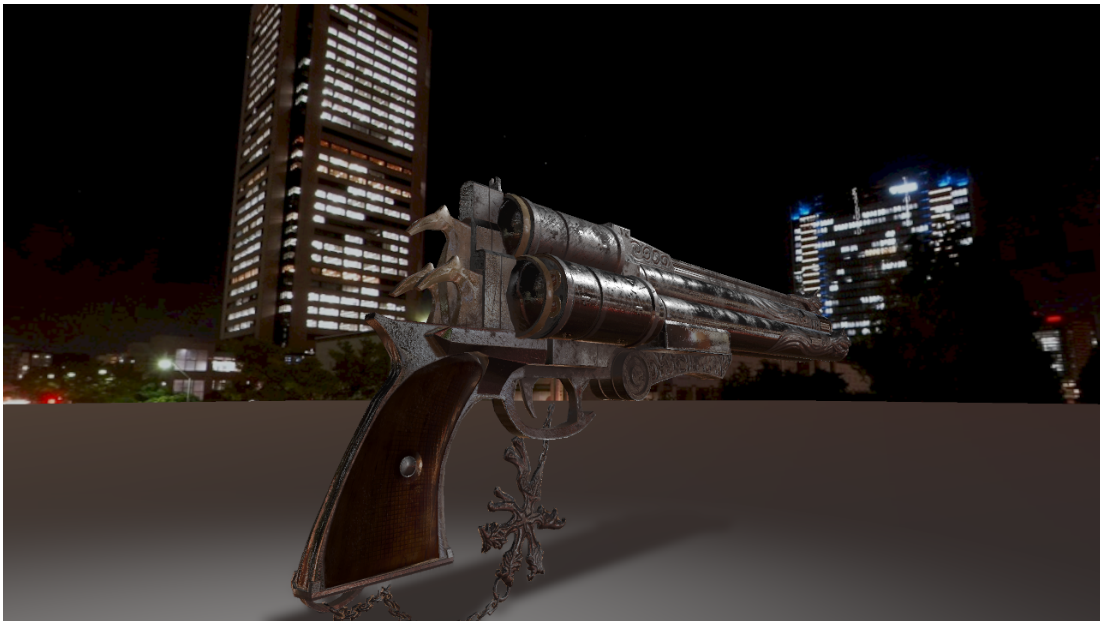

Figure 4.5: Specular reflections

4.3. Comparison to Blinn-Phong model

The outlined PBR shading model is parameterized by albedo, specular and surface roughness. Although Blinn-Phong is parametrized by similar sounding diffuse, specular and glossiness[3], these groups of parameters have vastly different meanings. For this reason, using the same parameters as input to both models and comparing results is not valid. A separate set of parameters should be made for each model in order to fully utilize the graphical fidelity each model can produce.

Furthermore, there is no empirical way to compare these models. The comparison is usually drawn by comparing the perceived visual correctness of rendered images. With these notes in mind, figures 4.6 through 4.9 present identical scenes rendered with each model.

[3] Glossines could be described as inverse roughness.

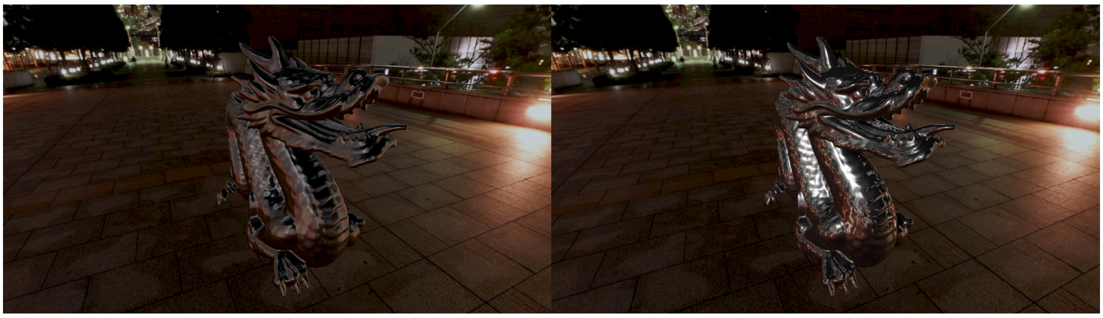

Figure 4.6: Blinn-Phong on the left and PBR on the right

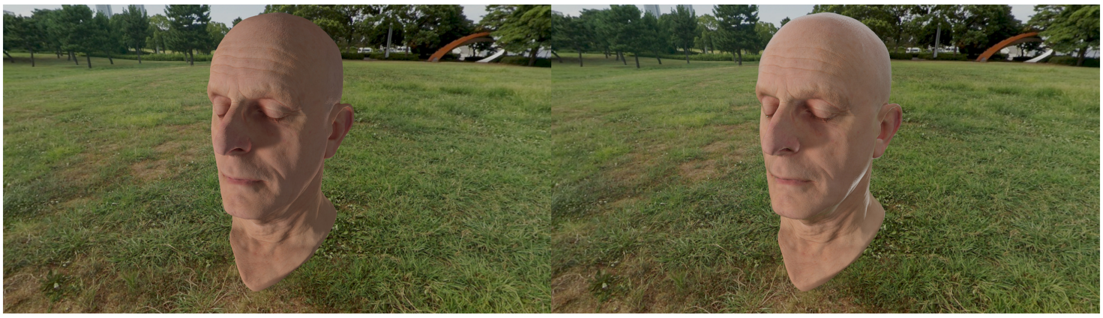

Figure 4.7: Blinn-Phong on the left and PBR on the right

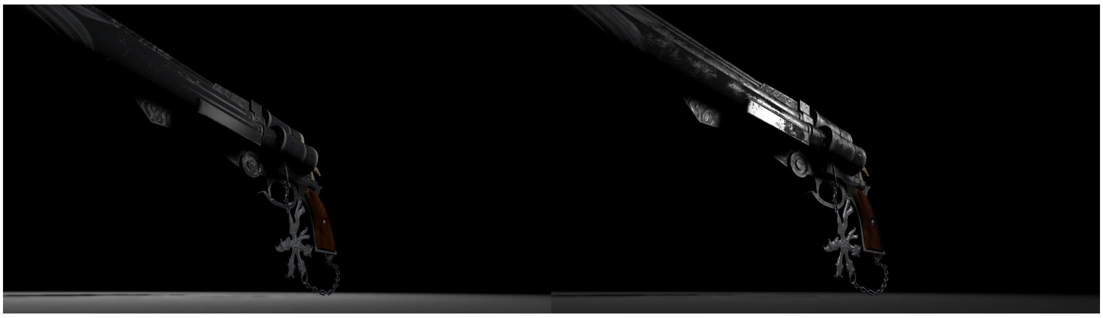

Figure 4.8: Blinn-Phong on the left and PBR on the right

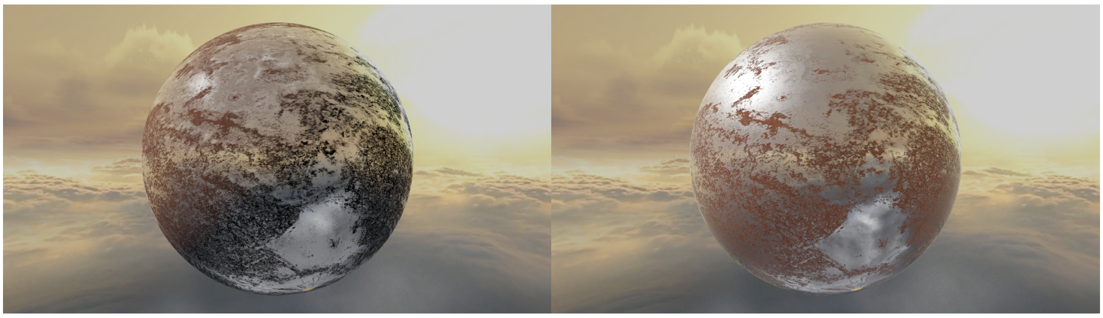

Figure 4.9: Blinn-Phong on the left and PBR on the right

5. Conclusion

In the recent several years, due to the increase in graphics processing capabilities, physically based rendering has seen wide adoption in real time productions such as video games. Because of this industry interest, the developments in the field of PBR have become rapid. The performance difference between the basic PBR and Blinn-Phong shading models is negligible on any modern PC grade GPU, as it amounts to several more shader instructions for the PBR model. These facts testify to the superiority of PBR to other, non physically based shading models. The drawbacks of PBR include its relatively low applicability to highly stylized graphics (although in recent years there were many successful games using stylized graphics with PBR). Furthermore, Blinn-Phong's small performance edge over PBR might be a concern in highly constrained hardware circumstances, like mobile platforms. This paper only goes through the basic ideas of this vast field, and the reader is encouraged to pursue further reading on this topic.

BIBLIOGRAPHY

[1] S. Green, “GPUGems Chapter 16. Real-Time Approximations to Subsurface Scattering,” 2007. [Online]. Available: http://http.developer.nvidia.com/GPUGems/gpugems_ch16.html

[2] R. Cook and K. E. Torrance, “A reflectance model for computer graphics,” SIGGRAPH, 1981. [Online]. Available: http://inst.cs.berkeley.edu/~cs294-13/fa09/lectures/cookpaper.pdf

[3] N. Hoffman, “Physically-Based Shading Models in Film and Game Production,” SIGGRAPH, 2010. [Online]. Available: http://renderwonk.com/publications/s2010-shading-course/hoffman/s2010_physically_based_shading_hoffman_a_notes.pdf

[4] N. Reed, “How Is The NDF Really Defined?” 2013. [Online]. Available: http://www.reedbeta.com/blog/hows-the-ndf-really-defined/

[5] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, “Microfacet Models for Refraction through Rough Surfaces,” Program of Computer Graphics, Cornell University, 2007. [Online]. Available: https://www.cs.cornell.edu/~srm/publications/EGSR07-btdf.pdf

[6] B. Karis, “Specular BRDF Reference,” 2013. [Online]. Available: https://graphicrants.blogspot.hr/2013/08/specular-brdf-reference.html

[7] L. Gritz and E. d’Eon, “GPUGems 3 Chapter 24. The Importance of Being Linear,” 2007. [Online]. Available: http://http.developer.nvidia.com/GPUGems3/gpugems3_ch24.html

[8] M. Alamia, “Physically Based Rendering,” 2011. [Online]. Available: http://www.codinglabs.net/article_physically_based_rendering.aspx

[9] ——, “Physically Based Rendering: Cook–Torrance,” 2011. [Online]. Available: http://www.codinglabs.net/article_physically_based_rendering_cook_torrance.aspx

[10] P. Hennessy, “Implementation Notes: Runtime Environment Map Filtering for Image Based Lighting,” 2015. [Online]. Available: https://placeholderart.wordpress.com/2015/07/28/implementation-notes-runtime-environment-map-filtering-for-image-based-lighting/

Physically based rendering

Abstract

Physically based rendering is defined. Basic and qualitative descriptions of light and material interactions are outlined. Physically based shading models and with their mathematical formulations are discussed. Real time implementation techniques for physically based rendering and image based lighting are presented.

Keywords: physically based rendering, physics of light, bidirectional reflectance distribution function, image based lighting

Fizikalno utemeljeno iscrtavanje

Sažetak

Fizikalno utemeljeno iscrtavanje je definirano. Dani su osnovni kvalitativni opisi interakcije svjetla i materijala. Fizikalno utemeljeni modeli sjenčanja, te njihove matematičke formulacije su komentirane. Tehnike ugradnje fizikalno utemeljenog iscrtavanja i osvjetljenja temeljenog na slikama su objašnjene.

Ključne riječi: fizikalno utemeljeno iscrtavanje, fizika svjetla, funkcija distribucije dvosmjerne refleksije, osvjetljenje temeljeno na slikama