UNIVERSITY OF GRANADA

Postgraduate Programme in
Software Systems Development
Departament of Lenguajes y Sistemas Informáticos

AN IMPORTANCE SAMPLING METHOD FOR ARBITRARY
BRDFs USED IN GLOBAL ILLUMINATION APPLICATIONS

PH.D. THESIS DISSERTATION
(FOR EUROPEAN MENTION) PRESENTED BY

ROSANA MONTES SOLDADO

ADVISOR
CARLOS UREÑA ALMAGRO

To my parents Antonio and Maria
Abstract

Nowadays there are many applications in the real world for Global Illumination algorithms. From modern movies’ special effects to the design of interior lighting, and even amazing video games. The scattering of light in an environment create an interesting effect on a surface: a distinctive quality of many materials. In this sense the BRDF function captures the behaviour of light when it arrives on a surface and reflects in several directions.

Our main objective of this work involves obtaining visually realistic images from computers, that accomplish a simulation of the propagation of light with numerical calculations [Mil84, WRC88, GSCH93, GTGB84, CWH93] using programs. The use of these algorithms allow us to obtain images with a more natural aspect, similar to the photographs. It is possible to simulate these characteristics with a computer using Global Illumination algorithms [Whi80, CPC84, Kaj86, WH92], that are based on following a ray path applying Monte Carlo’s method. Often these calculations present high run times, therefor we tried to implement efficient techniques suitable for them. In addition, Monte Carlo’s algorithms carry an intrinsic error in the estimator, that it is corrected using many more random samples. This, in turn, implies a high computational cost which creates a difficulty employing them in every day rendering.

Current works exploit the use of variance reduction techniques that increase the efficiency of the Monte Carlo’s estimator [VG94, Sbe96, Vea97, BSW00, ARBJ03, BGH05, JTE05, CJAMJ05, CETC06] thereby, decreasing the error using the same conditions for the number of samples and time. With the same aim, we have developed the work we present in this dissertation. More specifically, we attempt to increase the efficiency of the estimator by providing the exact information on a part of the function we want to estimate, that is, efficient sampling for the reflectance of a surface.

In order to be able to achieve this objective, we focused on examining the more representative reflectance models, including those that are the most used and also, those that describe more minutely the physical properties of a surface. During this process we became aware of the current situation: not every reflectance model has a procedure to obtain random samples according to the function itself. This is the reason why rendering systems mostly use simple and empirical models, easy to use, rather
than the more descriptive representations of the surface. In addition, it is necessary
to adapt a typical scene format to enrich the geometric information of the scene with
more specific information: vertices’ colour, the definition of the surface reflectance,
the irradiance at a point, declaration of area lights, and so on. After selecting a
representative set of surface reflectance models, we developed tools for a graphic
developer to assist in the definition and instantiation of a set of reflectance models
into scenes. Another important benefit of these tools is the ability to edit the BRDF’s
function parameters and their later visualization in two and three dimensions. At this
point in time, we had enough facilities to focus on developing a sampling algorithm
that was integrated in a light-transport simulation system.

We summarized the principal contributions of our work:

1. **State of the art for Global Illumination Reflectance Models** We undertook a study that consolidated a number of works related to the definition of
mathematical functions that represent optical properties or data from real mea-
urements: the BRDF functions. Following an unifying notation, the reader has
more capability of comparing several models, based on the knowledge of the exact
significance of its definition. In this dissertation, we also help the reader to under-
stand the reality for each decade, representing the models in a chronological form.

2. **An Scene Editor suitable for Global Illumination Systems** By means of
the use of a multi-platform tool called *Wannabe Amazing* [MUR+02] we are able
to import scenes from various well known graphic formats. After importing, we
add major information of interest in the context of realistic image synthesis. All
this information is exported to our own graphic format: *GRF*.

3. **BRDF Editor** We have developed several tools for the understanding and the
use of the BRDF’s parameters for many reflectance models. In this way, we
helped the graphic designer to select a value for each specific parameter being
the case that some of them are little intuitive. We set several different types of
graphs (in Cartesian and Polar coordinates system) and it also provides a 3D
visualization of the reflectance functions that changes by user mediation. For
real time rendering visualization of the BRDF edition, we have also implemented
the reflectance models in the GPU in Windows platform.

4. **Efficient sampling of general BRDFs** The main contribution of this work
resides in the development of an efficient sampling algorithm that is independent
of how a BRDF is modelled. Previous works provide individualized techniques of
sampling which are apt for only a subset of BRDFs [War92, LW94, NN98, AS00a].
Under no circumstances was there an exact scheme suitable for Monte Carlo
that could be used with any reflectance model. In this regard, our results
have been favourable compared with other numerical approximate techniques [LRR04, LRR05]. Initially our algorithm was conceived for analytical and general BRDF models, though it has been demonstrated to be general-purpose and applies well to acquired BRDF available data [MPBM03].

5. Efficient sampling for incident lighting and BRDF product function

It has been demonstrated that our adaptive sampling algorithm applies favourably to other techniques based on importance sampling. We refer to a sampling approach for the combined function of reflectance multiplied by irradiance expressed in a distant light illumination (an environment map) [ARBJ03, BGH05, JTE05, CJAMJ05, CETC06]. This estimator uses more information of the integrand than solely sampling the BRDF, reducing the error in the radiance computation.

This dissertation expands those ideas with more detail contents. The first chapter explains the physical phenomena of light reflection on which the computational and simplified models are based. Chapter 2 introduces crucial concepts used in Global Illumination and also the terminology that appears in this dissertation in order to understand light transport models. The next chapter gives a study of several empirical reflection models, describing an outline of how they are modelled, which parameters characterize them, their significance and how to used them in a realistic rendering system. In the following chapter, we introduce Monte Carlo methods, with the aim of computing the transport equation. We also describe major algorithms that are based on Monte Carlo, and thus need efficient sampling methods. Chapter 5 mentions variance reduction techniques and background work referred to BRDF sampling, describing some approximate solutions used with generic analytical models and acquired BRDFs. Finally in Chapter 6, a new adaptive method for general BRDFs sampling is presented with the main objective: permitting rendering systems to use complex geometry and accurate realistic reflectance models with reduced variance and no need for a user guide. Comparisons with other techniques are included. Detailed description of the implemented tools and an image gallery are included as appendices.

I hope that you find every part of the dissertation well detailed, and also forgive the limitations in expression, given that I am writing in a foreign language.

Rosana Montes
Content

1 Introduction 1
  1.1 Realistic Image Synthesis 1
  1.2 The dual nature of light 2
    1.2.1 The Propagation of Light 3
    1.2.2 Interaction of Light and Matter 3
  1.3 Representation of Surfaces 4

2 Reflectance models in Global Illumination 5
  2.1 A Light Model for Global Illumination 5
    2.1.1 Mathematical Notation 5
    2.1.2 Radiant Flux Density 8
    2.1.3 The Radiance Function 9
  2.2 Dispersion of Light in a Surface 9
    2.2.1 The Law of Reflection 10
    2.2.2 The Law of Refraction 10
    2.2.3 The Bidirectional Scattering Distribution Function 12
  2.3 The Local Reflection of Light 12
    2.3.1 The Bidirectional Reflectance Distribution Function 12
      2.3.1.1 Properties 13
      2.3.1.2 Geometry 13
      2.3.1.3 Classification 15
    2.3.2 Representations for the BRDF 16
    2.3.3 The Surface Reflectance 16
    2.3.4 The Fresnel Equations 17
      2.3.4.1 Case A: Dielectrics 18
      2.3.4.2 Case B: Electric Conductors or Metals 19
  2.4 The Radiance Equation 20

3 A Study of BRDF Models 23
  3.1 Ideal Specular Reflection (1621) 23
  3.2 Lambert (1760) 24
  3.3 Minnaert (1941) 24
  3.4 Torrance-Sparrow (1967) 26
Index of Symbols

\( S \)  Surface of interest.
\( A \)  Total surface \( S \) area.
\( dA \)  Differential of surface area.
\( dA_p \)  Differential projected surface area defined with respect to surface’s normal.
\( x \)  Surface point \( x \in S \).
\( \mathbf{w} \)  Ray direction expresses as \( \mathbb{R}^3 \) vector in a local reference system.
\( \Omega \)  The positive hemisphere of directions centred at a point domain.
\( \bar{\Omega} \)  The negative hemisphere of directions centred at a point domain.
\( S^2 \)  The unit radio sphere of directions.
\( D^2 \)  The unit disk domain. A plane that divides \( S^2 \) into two sets: \( \Omega \) and \( \bar{\Omega} \).
\( D \)  A generic domain of integration.
\( (\theta, \phi) \)  Ray direction expressed in a polar coordinate system.
\( d\sigma \)  Solid angle measure of a vector in the sphere of directions.
\( d\sigma_p \)  Projected solid angle measure of a vector in the sphere of directions.
\( P^* \)  Probability density function using \( \sigma \) measure.
\( Q^* \)  Probability density function using \( \sigma_p \) measure.
\( A_R \)  Rejected area for the rejection sampling technique.
\( i \)  Subindex that denotes incidence.
\( r \)  Subindex that denotes reflection.
\( t \)  Subindex that denotes transmission.
 CONTENT

a Subindex that denotes absorption.

d Subindex that denotes diffuse reflection.

s Subindex that denotes specular reflection.

Φ Total radiant energy (w).

dΦ Radiant Flux density area.

E(x) Irradiance or incident radiance.

M(x) Exitance or exitant radiance.

L(x, w) Radiance function (w/sr m²).

ρ Hemispherical-directional reflectivity function or albedo of the surface.

ρd Lambertian reflectivity function.

F Fresnel reflection factor.

T Fresnel transmission factor.

fr Bidirectional reflectance distribution function (BRDF).

fs Bidirectional scattering distribution function (BSDF).

fss Bidirectional subsurface reflectance distribution function (BSSRDF).

ft Bidirectional transmittance distribution function (BTDF).

λ Wavelength of light.

ν Frequency of light.

η Index of refraction.

κ Coefficient of extinction.

σa Coefficient of absorption.

σs Coefficient of specular reflection.

σt Coefficient of transmission.

ka Coefficient of ambient reflectivity.

ks Coefficient of specular reflectivity.

kd Coefficient of diffuse reflectivity.

University of Granada
n Unit vector for the normal of a surface.

r Unit vector for the tangent of a surface.

t Unit vector for perfect reflection.

h Unit vector for the halfway vector of a surface.

u Unit vector for the outgoing direction.

v Unit vector for the incoming direction.

$u_p$ Projected unit vector in $D^2$ domain.

s Random unit vector.

$\alpha$ Angle between $u$ and $r$ vectors.

$\beta$ Angle between $u$ and $h$ (same as $v$ and $h$).

$\delta$ Angle between $n$ and $h$.

$n$ Exponent parameter that controls the shape and amount of the brightness of a surface.

$\sigma_m$ Standard deviation of a density function.

$P$ Family of probability measures.

$P_u$ Probability measure function defined in $S^2$.

$p_u$ Probability density function (PDF) defined in $P_u$ with $\sigma$ measure.

$q_u$ Probability density function (PDF) defined in $P_u$ with $\sigma_p$ measure.

$w(u)$ The weight function in Monte-Carlo methods.

$I$ The integral we want to solve with Monte-Carlo methods.

$N$ Number of samples used in a Monte-Carlo estimator.

$F_N$ Monte-Carlo estimator of integral $I$.

$Q$ Estimate or exact value for integral $I$.

r.v. Abbreviation for random variable

$X$ A random variable in $(S^2, \Omega, P)$ space.

$Y$ A random variable in $(S^2, \Omega, P)$ space.

$F_X$ Cumulative distribution function (CDF) for $X$. 

R. Montes
$E(F_N)$ Average value of the Monte-Carlo estimator.

$V(F_N)$ Variance of the Monte-Carlo estimator.

$\beta(F_N)$ Bias of the Monte-Carlo estimator.

$\epsilon(F_N)$ Efficiency of the Monte-Carlo estimator.

$U$ The uniform distribution: random values with equal probability in the interval $[0, 1]$.

$\xi$ A random variable with uniform distribution.

$h$ Mapping function between $\Omega$ and $S^2$ domains.

$h^{-1}$ Inverse mapping function between $S^2$ and $\Omega$ domains.

$t(x, w)$ Ray casting function. For a given ray $w$ it returns the closest point $x$. 


University of Granada
Index of Figures

2.1 The $S^2$ domain is the set of all normalized directions. .......... 6
2.2 The local coordinate system. .................................. 6
2.3 Mapping directions from $D^2$ to $\Omega$. .......................... 7
2.4 Transmission and reflection of light. ............................ 10
2.5 The Law of Reflection of a vector. .............................. 11
2.6 The bending of rays at an interface. .............................. 11
2.7 Snell’s law. ...................................................... 12
2.8 A local coordinate system for a surface. .......................... 14
2.9 Geometry for light reflection. .................................. 14
2.10 The components of reflection. .................................. 17
2.11 Types of reflectance .......................................... 18
2.12 Surface geometry involved in the radiance computation. ......... 21

3.1 An instance of Minnaert in the GPU ............................. 25
3.2 The microfacets model. .......................................... 27
3.3 The effect of masking and shadowing light by the microfacets geometry. 28
3.4 An instance of Beard-Maxwell in the GPU ......................... 29
3.5 An instance of Phong and Blinn in the GPU ....................... 31
3.6 An instance of Cook-Torrance in the GPU ....................... 33
3.7 The cylindrical model of Poulin-Fournier ....................... 35
3.8 An instance of Poulin-Fournier in the GPU ..................... 36
3.9 An instance of Strauss in the GPU ............................... 39
3.10 The He-Torrance optical BRDF model. ........................ 40
3.11 An instance of He-Torrance in the GPU ....................... 41
3.12 An instance of Ward in the GPU .............................. 43
3.13 An instance of Lewis in the GPU .............................. 46
3.14 An instance of Schlick in the GPU ............................ 48
3.15 The subsurface model of Hanrahan. ............................ 49
3.16 An instance of Oren-Nayar in the GPU ....................... 51
3.17 Neumann & Neumann BRDF uses the disc domain. ............ 52
3.18 An instance of Lafortune in the GPU ......................... 54
3.19 An instance of Coupled in the GPU ........................... 55
3.20 Geometry for the anisotropic Ashikhmin-Shirley BRDF model. 56
INDEX OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.21</td>
<td>A layered model for polished surfaces.</td>
<td>57</td>
</tr>
<tr>
<td>3.22</td>
<td>An instance of Ashikhmin in the GPU.</td>
<td>58</td>
</tr>
<tr>
<td>4.1</td>
<td>Sampling the Ashikhmin-ShirleyBRDF.</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>A note on Lawrence’s factorization implementation.</td>
<td>73</td>
</tr>
<tr>
<td>4.3</td>
<td>A note on Lawrence’s Cascade CDF implementation. FIT (2)</td>
<td>75</td>
</tr>
<tr>
<td>4.4</td>
<td>A note on Lawrence’s Cascade CDF implementation. FIT (1)</td>
<td>77</td>
</tr>
<tr>
<td>5.1</td>
<td>The plane XY is the domain of our sampling method.</td>
<td>81</td>
</tr>
<tr>
<td>5.2</td>
<td>Distribution of 1000 samples on a disc.</td>
<td>84</td>
</tr>
<tr>
<td>5.3</td>
<td>Distribution of 2500 samples on a disc.</td>
<td>84</td>
</tr>
<tr>
<td>5.4</td>
<td>Optimal traverse of the quadtree.</td>
<td>86</td>
</tr>
<tr>
<td>5.5</td>
<td>The rejection area should be reduced.</td>
<td>88</td>
</tr>
<tr>
<td>5.6</td>
<td>The region codes.</td>
<td>89</td>
</tr>
<tr>
<td>5.7</td>
<td>Rendering images of test scenes 1 and 2.</td>
<td>93</td>
</tr>
<tr>
<td>5.8</td>
<td>GRF snapshot of scene 4.</td>
<td>93</td>
</tr>
<tr>
<td>5.9</td>
<td>Graph time versus error of test scene 1.</td>
<td>96</td>
</tr>
<tr>
<td>5.10</td>
<td>Graph time versus error of Test Scene 2 – Sphere 1.</td>
<td>98</td>
</tr>
<tr>
<td>5.11</td>
<td>The exponent in cosine lobe distributions.</td>
<td>98</td>
</tr>
<tr>
<td>5.12</td>
<td>Graph time versus error of Test Scene 2 – Sphere 2.</td>
<td>99</td>
</tr>
<tr>
<td>5.13</td>
<td>Sampling many BRDFs with Adaptive Disc.</td>
<td>101</td>
</tr>
<tr>
<td>5.14</td>
<td>Pottery scene rendered using different PDFs.</td>
<td>103</td>
</tr>
<tr>
<td>5.15</td>
<td>Graph time versus error of Test Scene 4.</td>
<td>104</td>
</tr>
<tr>
<td>5.16</td>
<td>Graph time versus error of test scene using RIS.</td>
<td>108</td>
</tr>
<tr>
<td>5.17</td>
<td>Rendered images of Merl BRDFs.</td>
<td>111</td>
</tr>
<tr>
<td>6.1</td>
<td>wannabe Amazing as a versatile tool.</td>
<td>114</td>
</tr>
<tr>
<td>6.2</td>
<td>A multiplatform GRF viewer and its counterpart in Linux.</td>
<td>115</td>
</tr>
<tr>
<td>6.3</td>
<td>MiniBpt combine GRF and BPT.</td>
<td>119</td>
</tr>
<tr>
<td>6.4</td>
<td>wannabe OP creator application.</td>
<td>120</td>
</tr>
<tr>
<td>6.5</td>
<td>Environment Map sampling utility.</td>
<td>121</td>
</tr>
<tr>
<td>6.6</td>
<td>A rejection sampling tool.</td>
<td>122</td>
</tr>
<tr>
<td>6.7</td>
<td>A BRDF viewer on the GPU (1).</td>
<td>123</td>
</tr>
<tr>
<td>6.8</td>
<td>A BRDF viewer on the GPU (2).</td>
<td>124</td>
</tr>
<tr>
<td>6.9</td>
<td>A BRDF viewer on the GPU (3).</td>
<td>124</td>
</tr>
<tr>
<td>6.10</td>
<td>Adaptive BRDF Subdivision.</td>
<td>125</td>
</tr>
<tr>
<td>6.11</td>
<td>Set of 2500 directions plot in 3D and 2D from a FIT file.</td>
<td>127</td>
</tr>
<tr>
<td>B.1</td>
<td>Synthetic image for scene CORNELL BOX.</td>
<td>139</td>
</tr>
<tr>
<td>B.2</td>
<td>Synthetic image for scene PLANTA.</td>
<td>140</td>
</tr>
<tr>
<td>B.3</td>
<td>Synthetic image for scene LIVING ROOM 1.</td>
<td>140</td>
</tr>
<tr>
<td>B.4</td>
<td>Detail image for scene LIVING ROOM 1.</td>
<td>141</td>
</tr>
<tr>
<td>B.5</td>
<td>Synthetic image for scene SIMPLESCENE.</td>
<td>141</td>
</tr>
<tr>
<td>B.6</td>
<td>Synthetic image for scene BAGUESCENE.</td>
<td>142</td>
</tr>
<tr>
<td>B.7</td>
<td>Synthetic image for scene BUZÓN.</td>
<td>142</td>
</tr>
</tbody>
</table>
INDEX OF FIGURES

B.8 Synthetic image for scene HELICÓPTERO........................................ 143
B.9 Synthetic image for scene MOTORBIKE......................................... 143
B.10 Synthetic image for scene MULTISPHERE using resampling ............. 144
B.11 Synthetic image for scene MULTISPHERE using acquired BRDFs ....... 144
B.12 Synthetic image for scene CHEVROLET..................................... 145
B.13 Synthetic image for scene PLANE_SCENE.................................. 145
B.14 Synthetic image for scene ROADWAY....................................... 146
B.15 Synthetic image for scene LIVING ROOM 2.................................. 146
B.16 Synthetic image for scene HAPPY BUDHA.................................... 147
B.17 Synthetic image for scene DRAGON.......................................... 148
B.18 Synthetic image for scene WOODMAN....................................... 149
B.19 Synthetic image for scene EMOTICO.......................................... 149
B.20 Synthetic image for scene PHLEGMATIC DRAGON......................... 150
B.21 Synthetic image for scene PHLEGMATIC DRAGON and PLANE_SCENE . 150
B.22 Synthetic image for scene BUNNY.............................................. 151
B.23 Synthetic image for scene PYRAMOB.......................................... 151
B.24 Synthetic image for scene FERTILITY......................................... 152
B.25 Synthetic image for scene FERTILITY using resampling................ 152
B.26 Synthetic image for scene CUBE............................................... 153
# Index of Tables and GRF notation

<table>
<thead>
<tr>
<th>Section</th>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Indexes of refraction for some common objects</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>Electric conductor’s properties.</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>GRF notation of <em>PerfectSpecularBRDF</em></td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>GRF notation of <em>PerfectDiffuseBRDF</em></td>
<td>24</td>
</tr>
<tr>
<td>3.3</td>
<td>GRF notation of <em>MinnaertBRDFs</em></td>
<td>25</td>
</tr>
<tr>
<td>3.4</td>
<td>GRF notation of <em>BeadMaxwellBRDF</em></td>
<td>29</td>
</tr>
<tr>
<td>3.5</td>
<td>GRF notation of <em>PhongBRDF</em></td>
<td>30</td>
</tr>
<tr>
<td>3.6</td>
<td>GRF notation of <em>BlinnBRDF</em></td>
<td>31</td>
</tr>
<tr>
<td>3.7</td>
<td>GRF notation of <em>Cook-TorranceBRDF</em></td>
<td>33</td>
</tr>
<tr>
<td>3.8</td>
<td>GRF notation of <em>Cylinders-PF-BRDF</em></td>
<td>37</td>
</tr>
<tr>
<td>3.9</td>
<td>GRF notation of <em>StraussBRDF</em></td>
<td>39</td>
</tr>
<tr>
<td>3.10</td>
<td>GRF notation of <em>HTSG-BRDF</em></td>
<td>41</td>
</tr>
<tr>
<td>3.11</td>
<td>GRF notation of <em>WardBRDF</em></td>
<td>43</td>
</tr>
<tr>
<td>3.12</td>
<td>GRF notation of <em>LewisBRDF</em></td>
<td>45</td>
</tr>
<tr>
<td>3.13</td>
<td>GRF notation of <em>SchlickBRDF</em></td>
<td>48</td>
</tr>
<tr>
<td>3.14</td>
<td>GRF notation of <em>Oren-Nayar-BRDF</em></td>
<td>51</td>
</tr>
<tr>
<td>3.15</td>
<td>GRF notation of <em>LafortuneBRDF</em></td>
<td>53</td>
</tr>
<tr>
<td>3.16</td>
<td>GRF notation of <em>ShirleyBRDF</em></td>
<td>55</td>
</tr>
<tr>
<td>3.17</td>
<td>GRF notation of <em>AshikhminBRDF</em></td>
<td>57</td>
</tr>
<tr>
<td>4.1</td>
<td>Brief summary of the BRDF properties.</td>
<td>62</td>
</tr>
<tr>
<td>4.2</td>
<td>GRF notation of <em>UniformPDF</em></td>
<td>65</td>
</tr>
<tr>
<td>4.3</td>
<td>GRF notation of <em>SpecularPDF</em></td>
<td>65</td>
</tr>
<tr>
<td>4.4</td>
<td>GRF notation of <em>HemispherePDF</em></td>
<td>67</td>
</tr>
<tr>
<td>4.5</td>
<td>GRF notation of <em>SpherePDF</em></td>
<td>67</td>
</tr>
<tr>
<td>4.6</td>
<td>GRF notation of <em>AnisotropicPDF</em></td>
<td>71</td>
</tr>
<tr>
<td>4.7</td>
<td>GRF notation of <em>FactoredPDF</em></td>
<td>73</td>
</tr>
<tr>
<td>4.8</td>
<td>GRF notation of <em>CascadeCDF</em></td>
<td>75</td>
</tr>
<tr>
<td>4.9</td>
<td>Memory requirements of uniform CDFs</td>
<td>75</td>
</tr>
<tr>
<td>4.10</td>
<td>Memory requirements of adaptive CDFs</td>
<td>76</td>
</tr>
<tr>
<td>5.1</td>
<td>The cost of precomputation the PDF.</td>
<td>90</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>5.2 Memory requirements when ( n_{\text{max}} ) varies.</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>5.3 Memory requirements of Adaptive Disc and Factored PDF for every</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \text{BRDF} ).</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>5.4 \text{GRF notation of AdaptivePDF}</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>5.5 The ( \text{BRDF parameters} ).</td>
<td>94</td>
<td></td>
</tr>
<tr>
<td>5.6 The \text{factorization of the BRDFs}.</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>5.7 Sampling times of test scene 1.</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>5.8 Average relative error of test scene 1.</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>5.9 Sampling times of test scene 2 – Sphere 1.</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>5.10 Average relative error of test scene 2 – Sphere 1.</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>5.11 Sampling times of Test Scene 2 – Sphere 2.</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5.12 Average relative error of Test Scene 2 – Sphere 2.</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5.13 Precomputation times of Test Scene 4 – pottery 3.</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>5.14 Sampling times of Test Scene 4 – pottery 3.</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>5.15 Average relative error of Test Scene 4 – pottery3.</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>5.16 Precomputation times of Test Scene 5.</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>5.17 Sampling times of Test Scene 5.</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>5.18 Average relative error of Test Scene 5</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>5.19 Test Scene 7: precomputation data for scenes with acquired reflectance</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>5.20 Test Scene 7: sampling data for scenes with acquired reflectance</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>6.1 \text{OP file example used with BPT.}</td>
<td>118</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 1

Introduction

1.1 Realistic Image Synthesis

The details of the real world are rich and varied, but sometimes we lose the habit of observing the detail. Global Illumination is a field of Computer Graphics that tries to cover up the need for realism in this area. It has the objective of generating realistic images that show most of the effects that light produces when interacting with real objects. This is a major goal, which at present, is limited because imitating nature never has been a simple task. Several aspects must be considered:

- the construction of geometric models,
- application of textures and transparencies,
- illumination and shading techniques,
- visualization in real time using hardware techniques.

Each individual item has enough potential for work and constitutes an individual field of research. Is not our intention to explain each one of them, we only require you to observe that each small improvement in each field constitutes progress for the final result.

Realistic image synthesis demands a distinct solution for different problems. Firstly, optimal geometric models need to be available for every single object in mind. Secondly, an illumination or shading model is used to compute the intensity and colour at each single point on a surface. For the later, two families of illumination algorithms exist: Local Illumination methods (LI) and Global Illumination method (GI). The difference
between them depend on whether or not there are inter-reflections between the objects.

Given a model of reality, it is necessary to elaborate an equivalent numerical one, to be used in the description of light transportation and the interaction of elements within the scene. This representation should consider the visual human system [Gla94, Bue01] for a subjective description of what the viewer interprets when she observes the scene.

We are interested in knowing about the aspect of the objects present in the scene. Their appearance, intensity and colour that she observes will depend on:

- the characteristics of the surface: colour, transparency degree, reflectance degree, etc.
- the location of the observer in the scene.
- the illumination conditions such as intensity, geometry and position of light sources,
- the environmental and atmospheric conditions (fog, smoke, etc.).

The nucleus of the shading calculations focuses on the simulation of the complex interaction of light with the objects. Complex in the sense of in reality, millions of inter-reflections take place and thus it is difficult to simulate them completely. In order to simplify this process we use theoretical cases such as ideal diffuse and ideal specular surfaces.

1.2 The dual nature of light

Since the beginning of time the existence of light is something that always has interested humanity. The propagation of light is classically described in two ways:

- Newton considered that light was formed of little particles called corpuscles;
- Huygens, based his theories in the fact that light was an undulatory phenomena.

Phenomena of transmission, reflection and refraction could be more specific under the perspective of Electromagnetic Theory. For a modern interpretation of what is happening in this regard, we use Quantum Electrodynamic field.

Until the first part of the XX century scientists were divided. Those with Newton (1665) thought that bright objects emitted particles that travelled in strait lines, and explained absorption, reflection and light refraction. Huygens (1678), Young (1839) and Fresnel (1865) found justification of diffraction, interference and polarization with
1.2. The dual nature of light

the undulatory description of light. Thus light was either a stream of particles or a
undulation of aether matter.

While all of this was happening in the field of Optics, the study of electricity and
magnetism obtained a great result. While solving the speed of the wave, Maxwell
found that light was an electromagnetic disturbance in the form of waves propa-
gated through the aether. However, the aether hypothesis was rejected by Albert
Einstein (1905). He stated that light is always propagated in empty space with a
definite velocity $c$. Furthermore, based on the work of Max Planck in Quantum
Mechanics, Einstein proposed a new form of corpuscular theory in which light
consisted of particles of energy. Each quantum of radiant energy was called photon
and its energy is proportional to its frequency $\nu$. It is not possible to know the
path of a photon, only the probability of being at a certain position. Particles
and waves in the macroscopic world appeared to be so mutually exclusive, but
merged in the submicroscopic domain. Thus photons, electrons, neutrons, and so on,
have both particle and wave manifestations, and this is a general property of materials.

1.2.1 The Propagation of Light

The light that the human eye perceives is called visible light and comprises wavelengths
from 400 to 700 nm. Superior wavelengths to 700 nm are not visible to the eye and
they correspond to the so-called infrared light. Light underneath 400 nm is called
ultraviolet. It is evident the electromagnetic nature of light. It currently correspond
to electromagnetic radiation in the narrow band of frequencies from about $3.84 \times 10^{14}$
Hz to $7.69 \times 10^{14}$ Hz.

It is a fact that light is propagated through a vacuum at a nominal value of
$2.997 \times 10^8$ m/s. Other media could introduce a shift in the velocity of the transmitted
beam. The relationship between the two velocities is given by the index of refraction $\eta$.

1.2.2 Interaction of Light and Matter

When a beam of light impinges on an object something happens on its surface that
allows us to perceive its nature and composition. We can distinguish transparent
material, such as glass, that transmits the wave through a dense medium allowing us
to see through it; translucent material, also lets light propagate through —refraction—
but it does not permit us to see other objects behind it; opaque objects always scatter
some light backward, and we call this phenomenon reflection.

The quantity of reflected or refracted light depends of the wavelength $\lambda$ but also
depends on others factors for example the surface geometry, the light and the angle of
Chapter – 1. Introduction

incidence. These considerations are taken into account in the illumination or Shading Model.

1.3 Representation of Surfaces

The phenomena of reflexion at a point, occurs in different ways, depending on the surface. The way we characterize the reflexive behaviour of the surface of the object is the bidirectional reflectance distribution function (BRDF). This mathematical function is bidirectional in the sense that its depends on two directions, one to the viewer and other to the light source, and it determines the type of material that the observer perceives.

In this manner, a BRDF or reflectance model, describes the colour and distribution of the reflected light. The choice of a particular model and parameters allows us to simulate a specific type of material and thus it is a important part of the general process of realistic image synthesis. An extended description of reflectance models, definition, classification, properties and so forth is given in next chapters.

We attempt to propitiate the use of many reflectance models (other than those used generally) in a Global Illumination system, giving a Monte-Carlo based sampling algorithm suitable for general BRDF description. In the next chapter we undertake an introduction of the terminology used throughout this dissertation, in order to understand the transportation equation for luminous radiation.
CHAPTER 2

Reflectance models in *Global Illumination*

2.1 A Light Model for Global Illumination

Two basic elements appear in any illumination process: light sources and the objects that are going to be lit up. Many authors in the literature of Global Illumination have considered that the radiant energy is quantified in bunches of particles or photons, each one with a property and a probability of interaction with matter. The particles travel in straight lines through a homogeneous medium without interfering themselves. They only interact with the objects in the scene, at that moment, the phenomena of light dispersion takes place. Then some portion of the light will be reflected, another will be transmitted and the rest will be absorbed by the object. Absorption is not taken into account (it only contributes to object’s temperature). Solely the reflected and transmitted light are considered to eventually reach the observer.

The general transport equation describes the flow of energy, and the principal task of Global Illumination algorithms is to solve it. Various mathematical concepts are needed to understand this equation. We describe them in the following sections.

2.1.1 Mathematical Notation

Let $x \in S$ be a three-dimensional point at any surface. We want to compute its colour. Similarly, directions are given in Cartesian local coordinates, they are expressed by three real values and a magnitude. In this document we use bold typeface to denote vector directions. When directions are normalized, they belong to a sphere of unit radius. That is, if $u \in S^2$ then $u(x_u, y_u, z_u)$ with $|u| = 1$. In this system the local Z
axis corresponds with the surface normal $\mathbf{n} \overset{\text{def}}{=} (0,0,1)$.

**Figure 2.1:** The $S^2$ domain is the set of all normalized directions. The XY plane divides the sphere into two equal hemispheres.

The positive hemisphere $\Omega_{+ \subset S^2}$ (fig. 2.1) is the set of directions that obey:

$$\Omega \overset{\text{def}}{=} \{ \mathbf{v} \in S^2 \mid \mathbf{v} \cdot \mathbf{n} > 0 \}.$$

The complement is the negative hemisphere $\Omega^- \subset S^2$, defined as:

$$\Omega^- \overset{\text{def}}{=} \{ \mathbf{v} \in S^2 \mid \mathbf{v} \cdot \mathbf{n} \leq 0 \}.$$

**Figure 2.2:** Angles relative to incident and reflected vectors in a local coordinate system.

Vector could also be given in polar coordinates. The polar angle $\theta_u$ indicates the angle between the vector and axis $Z$. The azimuthal angle $\phi_u$ measures the angle of the projection on plane $XY$ with respect to axis $X$. The figure shows an example 2.2. Then we note $\mathbf{u} = (\theta_u, \phi_u)$. Both representations, polar and cartesian, are equal using the following relation:

University of Granada
2.1. A Light Model for Global Illumination

\[ x_u = \cos(\phi_u) \cdot \sin(\theta_u), \]
\[ y_u = \sin(\phi_u) \cdot \sin(\theta_u), \]
\[ z_u = \cos(\theta_u). \]

A third expression for vector directions \( v \in S^2 \) is possible. If we consider a vector projection, just dropping the \( Z \) component, the result is a two-dimensional point \( v_{xy} \in D^2 \). The set for all the possible projections gives the \( D^2 \) domain:

\[ D^2 \overset{\text{def}}{=} \{ (x, y) \in E^2 \text{ t.q. } x^2 + y^2 < 1 \}. \]

There is a mapping function \( h : \Omega \rightarrow D^2 \) to express a vector as a point.

\[ h(v) = h(x_v, y_v, z_v) \overset{\text{def}}{=} (x_v, y_v) = v_{xy}. \]

**Figure 2.3**: The inverse mapping.

The inversion for this function \( h^{-1} \) takes a point in the \([-1, 1) \times [-1, 1)\) region and transforms it into a vector (see Figure 2.3).

\[ v = h^{-1}(v_{xy}) = h^{-1}(x_v, y_v) = (x_v, y_v, z_v) \mid z_v = \sqrt{1 - x_v^2 - y_v^2}. \]

The two-dimensional concept of angle is extended to three-dimensions with the solid angle. It represents a generalization of a vector to a group of directions. \( S^2 \) is equivalent to the differential surface covered by those directions. We note it \( \sigma \) and it is expressed in steradians.

\[ d\sigma(v) \overset{\text{def}}{=} \frac{dA(\theta_v, \phi_v)}{r^2} = \frac{(r \, d\theta_v) \, (r \, \sin(\theta_v) \, d(\phi_v))}{r^2} = \sin(\theta_v) \, d(\theta_v) \, d(\phi_v). \quad (2.1) \]

With the same objective, the solid projected angle \( \sigma_p \) is defined for any vector \( v \in \Omega \) as:
\[
\begin{align*}
\sigma_p(\mathbf{v}) & \overset{\text{def}}{=} (\mathbf{v} \cdot \mathbf{n}) \, d\sigma(v).
\end{align*}
\] (2.2)

Finally, a third measure is presented for points \((x, y) \in D^2\). We refer to the area measure \(dA(x, y)\). It is easy to show that the three measures are related (see Equation 2.3) and thus differential regions in \(\Omega\) project to a differential area in \(D^2\) and vice versa:

\[
\begin{align*}
dA(u_{xy}) &= dA(x_u, y_u) = dx \, dy = d\sigma_p(h^{-1}(x_u, y_u)), \\
d\sigma_p(u) &= dA(h(u)) = dA(u_{xy}).
\end{align*}
\] (2.3)

The inner product—or cosine term—could be written as:

\[
(\mathbf{v} \cdot \mathbf{n}) = \cos(\mathbf{v}) = \cos(\theta_v).
\]

Once we have a general concept about the notation we use, it is time to introduce the fundamental magnitudes that define the luminous radiation.

### 2.1.2 Radiant Flux Density

Radiant flux \(\Phi\) is the number of particles that travels through a medium. Each particle carries an energy and has an associated probability of interaction with matter. The power that leaves an object (considering a surface region) results from the power the object emits \(\Phi_e\) and some portion of energy that reflects \(\Phi_r\) from the incident flux \(\Phi_i\). That is, \(\Phi = \Phi_e + \Phi_r\). Note that we could also refer to the transmitted flux as \(\Phi_t\).

For a mathematical model, it is convenient to use the quotient of the radiant flux instead. It is defined by the International Commission of Illumination (CIE) as power emitted, transmitted or received in the form of radiation per unit area. It is expressed in \(W \cdot m^{-2}\) (watts per meter squared).

If we refer to a differential of radiant flux incident on an element of the surface containing a point \(x\), then we use the term \(E\) or \(irradiancia\). If the flux density leaves the surface, then \(M\) or \(exitance\) is used:

\[
M(x) = \frac{d\Phi(x)}{dA(x)} \quad E(x) = \frac{d\Phi_i(x)}{dA(x)}.
\]

In all cases the energy that leaves the surface \(M(x)\) takes into account some part produced by the phenomena of reflection \(M_r(x)\) and the energy that the object could emit \(M_e(x)\):

\[
M(x) = M_e(x) + M_r(x) = \frac{d\Phi_e(x)}{dA(x)} + \frac{d\Phi_r(x)}{dA(x)}.
\]
Both $M$ and $E$ functions have an equivalent definition. They consider energy at every direction and can be expressed as integral over the hemisphere. We demonstrate this in the next section.

### 2.1.3 The Radiance Function

Radiance $L$ in a given direction, at a given point of a real or imaginary surface is a quantity defined by the formula:

\[
\frac{dL_r(x, \mathbf{u})}{dA_r(x)} = \frac{d^2 \Phi(x, \mathbf{u})}{\cos(\mathbf{u}) \, dA(x)} = \frac{d^2 \Phi(x, \mathbf{u})}{dA(x) \, d\sigma_p(\mathbf{u})}.
\]

In the above formula, $d\Phi(x, \mathbf{w})$ is the radiant flux transmitted by an elementary beam passing through the given point $x$ and propagating in the solid angle $d\sigma(\mathbf{w})$ containing the given direction $\mathbf{u}$, $dA(x)$ is the area of a section of that beam containing the given point, $\theta$ is the angle between the normal to that section and the direction of the beam. Its unit is $W \cdot m^{-2} \cdot sr^{-1}$.

The differential of radiance is easily related with other magnitudes:

\[
\frac{dL_i(x, \mathbf{v})}{dA_i(x)} \cos(\mathbf{v}) = \frac{dE(x, \mathbf{v})}{d\sigma_p(\mathbf{v})},
\]

and thus,

\[
dE(x, \mathbf{v}) = dL_i(x, \mathbf{v}) \cos(\mathbf{v}) \, d\sigma_p(\mathbf{v}). \tag{2.4}
\]

If we integrate radiance over all the solid angles, we get the energy that arrives or leaves the surface per unit time and per unit area. The integral forms of exitance and irradiance are:

\[
M_r(x) = \int_{\Omega} dL_r(x, \mathbf{u}) \, d\sigma_p(\mathbf{u}), \tag{2.5}
\]

\[
E(x) = \int_{\Omega} dL_i(x, \mathbf{v}) \, d\sigma_p(\mathbf{v}). \tag{2.6}
\]

### 2.2 Dispersion of Light in a Surface

When light travels through different media, some portion of it reflects over the media and other portions are transmitted after a change of its direction (as in Figure 2.4). Phenomena of transmission, reflection and refraction are macroscopic manifestations of microscopic events.

If we know the index of refraction of both media, it is possible to calculate the quantity of energy that it is transferred. The rule used is the basic conservation of energy:

\[
\Phi_i \, A \cos(\theta_i) = \Phi_r \, A \cos(\theta_r) + \Phi_t \, A \cos(\theta_t).
\]
Chapter – 2. Reflectance models in Global Illumination

2.2.1 The Law of Reflection

The phenomena of reflection allows us to see objects that do not emit light, as long as they are illuminated. Objects reflect in different way, according to their surface. Polished surfaces reflect mainly in a single direction, the direction of ideal reflection (see Equation 2.2.1). Polished and opaque surfaces are called mirrors. The opposite of polished, is diffuse reflection, where light is reflected in all directions. Also a combination of both is possible.

The surfaces that reflect light produce it in a different ways. Reflected light goes out in only one direction. For this reason it is called ideal specular reflection. In the second case, the reflected light goes out in all directions, and so it is called ideal diffuse reflection.

Figure 2.4: Transmission and reflection of an incident beam of light.

The Law of Reflection [FvDaSFH90] state that:

1. The incident, reflected and normal vector of the mirror are on the same plane.
2. Incident and reflection angles are the same.

Given an incident direction (see Figure 2.5), the reflection direction is computed as follows:

\[ \theta_r = \theta_u, \]
\[ \phi_r = \phi_u \pm \pi, \]
\[ r = u - 2(u \cdot n)n. \]

2.2.2 The Law of Refraction

When a beam of light impinges on a interface between two transparent media with different densities (wave velocities), a new beam of light is refracted in the transmitting

University of Granada
2.2. Dispersion of Light in a Surface

Medium. A deviation occurs (see Figure 2.6) which is dependent on the incident angle and also the coefficient of refraction of the medium. For a given angle of incidence —called the limit angle— the angle of refraction is 99°. This phenomena is called total reflection.

The relationship between the difference of the incident angle and the reflected angle is a constant value used to define the index of refraction \( \eta = \frac{x_i}{x_t} \), as it is shown in Figure 2.7.

In 1621, Willebrord Snell, enunciated the laws that govern the phenomena of refraction:
2. Reflectance models in Global Illumination

1. The incident, reflected, and refracted rays all lie in the plane-of-incidence

2. The Snell’s law: \( \eta_i \sin \theta_i = \eta_t \sin \theta_t \)

2.2.3 The Bidirectional Scattering Distribution Function

Real objects are not purely reflexive or purely transmissive. The BSDF \( f_s \) or Bidirectional Scattering Distribution Function, is the mathematical function that describes the behaviour of the material that scatters some light. It is equivalent to consider four phenomena in one: two BRDFs that represent reflections one at each side of the surface, and two BTDFs for the transmissions that occur on each direction.

Both the BRDF \( f_r : \Omega_i \times \Omega_r \rightarrow \mathbb{R} \) and the BTDF \( f_t : \Omega_i \times \Omega_t \rightarrow \mathbb{R} \) could be considered as the BSDF in a lower dimension. They take positive values and use the inverse stereoradian as unit \( sr^{-1} \). Note that \( \Omega_t = \Omega \). Current work in BTDF models [Sta01] is not as broad as its counterpart the BRDF.

2.3 The Local Reflection of Light

2.3.1 The Bidirectional Reflectance Distribution Function

The Bidirectional Reflectance Distribution Function, is a parametric representation of the reflectance of a surface. This function expresses the interaction of light with matter, by relating the radiance that comes from \( \mathbf{v} \) direction, with the radiance that leaves the object in \( \mathbf{u} \) direction. This is the constant value of the following relation:

\[
dL_r(x, \mathbf{u}) \propto dE_i(x, \mathbf{v}).
\]

The BRDF is a function of six parameters, but some simplification is usually undertaken in order to reduce its complexity and dimensionality. Those assumptions are:
2.3. The Local Reflection of Light

- The distribution of reflectance does not vary along the surface. It is not necessary to express the point $x$ which is implicit in the formulation.

- Light reflects instantly and thus it is not necessary to express directly the time variable. With this simplification it is not possible to simulate the phenomenon of phosphorescence.

- The outgoing beam of light has same wavelength $\lambda$—that is, same frequency $\nu$—than the incident one; we omit the phenomenon of fluorescence.

- The parameter of radiance and the BRDF, $\lambda$ is simplified into three different value bands of colour (RGB). The wavelength is implicit in the formulation.

This let the expression of the BRDF function as $f_r(u, v) \in [0, \infty)$ or equivalently $f_r(\theta_u, \phi_u, \theta_v, \phi_v)$. In any event, it gives the ratio of radiance from a view direction $u$, that is proportional to irradiance from an infinitesimal solid angle at direction $v$:

$$f_r(u, v) = \frac{dL_r(u)}{dE_i(v)} = \frac{dL_r(u)}{L_i(v) \, d\sigma_p(v)}.$$  \hspace{1cm} (2.7)

2.3.1.1 Properties

The properties that the BRDF must have are:

1. **Symmetry.**
   
   It must obey the *Helmholtz Reciprocity Rule* [CP85], which results directly from the physics of light and states that the BRDF is symmetric relative to $u$ and $v$.

   $$f_r(u, v) = f_r(v, u) \quad \forall u, v \in \Omega.$$  

2. **Energy Conservation Law.**
   
   This also results directly from the physics of light and serves to normalize the BRDF. It states that no more than 100% of the incident radiance can be reflected at any one point [Hel25]. Thus the BRDF must satisfy:

   $$\int_{\Omega} f_r(u, v) \, d\sigma_p(v) < 1 \quad \forall u \in \Omega.$$  

2.3.1.2 Geometry

The outgoing and the incident directions are not the only ones that are used to describe or represent the BRDF function. For example, in [PF90, KM99] the binormal vector $b$ and the tangent vector $t$ define a local coordinate system (see Figure 2.8) more suitable for the illumination calculation. For a given vector $w$, we have the
following geometric constraints:

\[ \mathbf{b} \overset{\text{def}}{=} (\mathbf{w} \times \mathbf{t}), \]
\[ \cos(\theta_w) \overset{\text{def}}{=} (\mathbf{n} \cdot \mathbf{w}), \]
\[ \tan(\phi_w) \overset{\text{def}}{=} \frac{(\mathbf{b} \cdot \mathbf{w})}{(\mathbf{t} \cdot \mathbf{w})}. \]

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure28}
\caption{An alternative to the local coordinate system of Figure 2.2.}
\end{figure}

In order to use a homogeneous notation throughout this document we follow the angles, vectors and notation shown in Figure 2.9.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure29}
\caption{Geometry for light reflection used throughout this dissertation.}
\end{figure}

\[ \forall \mathbf{u}, \mathbf{v}, \mathbf{n} \in \Omega \}
\[ \left\{ \alpha \overset{\text{def}}{=} \cos^{-1}(\mathbf{r_u} \cdot \mathbf{v}), \right. \]
\[ \beta \overset{\text{def}}{=} \cos^{-1}(\mathbf{u} \cdot \mathbf{h}), \]
\[ \delta \overset{\text{def}}{=} \cos^{-1}(\mathbf{h} \cdot \mathbf{n}), \]
\[ \mathbf{h} \overset{\text{def}}{=} \frac{\mathbf{u} + \mathbf{v}}{\|\mathbf{u} + \mathbf{v}\|}. \]

The halfway vector \( \mathbf{h} \) was introduced by Blinn [Bli77] and was employed by others [CT82, ?, Rus98]. It represents the direction of maximum brightness. This approach avoids explicit computation of the reflection vector, and is simple to implement.
2.3. The Local Reflection of Light

2.3.1.3 Classification

We consider the following classification given by Ward [War92]:

- **Approximated**: The main aim is to provide a simply formulation specifically designed to mimic a kind of reflection. Consequently, a fast computational model, adjustable by parameters without a base in physics behind. Some examples are [Pho75] and Blinn [BN76].

- **Theoretical**: These models try to approximate real phenomena of light dispersion. They need more computational effort which sometimes discriminates them and they are not normally employed in rendering systems. Some examples are Cook-Torrance [CT82], Kajiya [Kaj85] and He-Torrance-Sillion-Greenberg [HTSG91]. Each theoretical model is limited to a specific type of material.

- **Experimental**: Sometimes theoretical models are not so practical. This is the reason why some types of models simulate real phenomena by using data from measurements. The BRDF can be acquired using a gonioreflectometer [War92, GA97, GA99] which manually or mechanically varies the source light and the sensor. This process could take hours and thus other techniques use digital cameras to acquire many BRDF samples with a single photograph [DGNK99, MPBM03]. This approach gives a formula which is also adjustable by parameters, but the difference is that they are designed to adjust data from measurements. Once you have the instance of your model that best represents the data, you do not need to use the data, only the model. As a result, this is simple, fast and accurate. The problem is the error introduced in data could make it difficult to find a better fit for your model. Some examples are given by [War92, Sch93, LW94].

It is possible to give a different classification considering the characteristics of the matter we want to characterize.

- **Isotropic BRDFs**: represent a kind of material whose reflection does not depend on the orientation of the surface, which is usually rough. This simplifies the BRDF that only needs three parameters \( f_r(\theta_i, \theta_r, \phi_r - \phi_i) \). Then, the BRDF obeys:

\[
\forall \alpha \in [0, 2\pi], \, \alpha > 0 \implies f_r(\theta_r, \phi_r, \theta_i, \phi_i) = f_r(\theta_r, \phi_r + \alpha, \theta_i, \phi_i + \alpha),
\]

- **Anisotropic BRDFs**: light reflects differently if the surface rotates around \( \mathbf{n} \) an angle \( \alpha \), even if the vectors \( \mathbf{u} \) and \( \mathbf{v} \) are fixed. This effect gives the streaky appearance seen on brushed metals.
2.3.2 Representations for the BRDF

Sophisticated and elaborate BRDFs often take into account the incoming and outgoing angles, wavelength, polarization, position, and time. There are many variables for computer graphics, so this can lead to lengthy computation. More simple BRDFs consider ray optics rather than wave optics when computing light transport. Ray optics treat light as non-interacting straight rays, each carrying a certain amount of energy \[ E \]. Ray optics simplify the notation and expressions of equations by neglecting wavelength. Because wavelength is neglected, so are the effects of interference, diffraction, and polarization that result from light wave interaction.

Traditionally, computer graphics systems either relied on analytical models or had to store enormous amounts of data to represent even simple BRDFs. The most simple BRDF representation stores samples of the BRDF on a regular four dimensional grid and interpolates between them. Using this method, the interpolated data is likely to be noisy and yield unsatisfactory results. Moreover, there are usually missing data points near the grazing angles. In addition to these problems, storing a complete BRDF sufficiently densely, is likely to require a lot of megabytes.

Given the size and high dimensionality of BRDFs, many techniques have been developed to store and compute them efficiently such us: fitting data to an analytical model, using spline patches, spherical harmonics base functions, spherical wavelets, and Zernike polynomials. A change of variables with one of the previously mentioned techniques can lead to a more efficient representation \[ ? \]. Using spherical harmonics to represent the BRDF is another approach. The spherical analogues of sines and cosines and are located in the frequency domain. Each part of the signal is known as a basis function and the sum of all of them reproduces the original signal precisely. Because there may be an infinite number of basis functions required to reproduce exactly the desired BRDF, usually only a subset of these basis functions are used. Employing too few basis functions can result in visual errors and using too many is computationally very expensive.

The factorization of the BRDF [LRR04] is a less expensive approach that represents the BRDF accurately and is also suitable for sampling in a Monte-Carlo rendering system. This representation is detailed in section 4.4.

2.3.3 The Surface Reflectance

The reflectance function \( \rho \) or albedo, determines the ratio of reflected energy to incident energy. It takes values between 0 and 1 and has no units. The BRDF expresses directional dependence, meanwhile the albedo is just a ratio. The reflectance at a point is computed using the integral of the BRDF for a given direction:
2.3. The Local Reflection of Light

\[ \rho(u) \overset{\text{def}}{=} \int_{\Omega} f_r(u, v) \cos(v) \, d\sigma(v). \]

The phenomenon of light scattering could be divided into three components (as shown in Figure 2.10.

![Figure 2.10: The total reflection is due to three components: uniform-diffuse, directional-diffuse and specular.](image)

The specular reflection is the phenomena applicable to mirror surfaces. It is characterized by specular reflectivity \( \rho_s(u) \in [0, 1) \) and depends on the incident angle and the Fresnel term.

The uniform-diffuse component \( \rho_d \in [0, 1) \), or Lambertian component, does not depend on the incident direction and reflects light equally in every direction. It is an ideal phenomena since there are no purely diffuse materials.

Finally, the directional-diffuse component which is not ideal and thus the more complex could consider different types of reflectance [Gla94]. Let \( \rho_{hd}(x, u) \) be the hemispherical-directional reflectance, that is, the reflectance in a given direction resulting from illumination over the hemisphere.

\[ \rho_{hd}(x, u) = \frac{L_r(x, u)}{E(x)} = \int_{\Omega} f_r(x, u, v) \, d\sigma_p(u). \quad (2.8) \]

The reciprocal quantity associated with this component of reflection is the directional-hemispherical reflectance, the integral over the entire reflection hemisphere. Both functions are the same \( \rho_{hd} = \rho_{dh} \) and thus we use a unique notation \( \rho \).

\[ \rho_{dh}(x, v) = \frac{dE_r(x, v)}{dE_i(x)} = \int_{\Omega} f_r(x, u, v) \, d\sigma_p(v). \quad (2.9) \]

2.3.4 The Fresnel Equations

When light hits a surface and reflection or transmission occurs it is not only sufficient to know the new direction of light, we also need to know the fraction of energy that
Chapter – 2. Reflectance models in Global Illumination

Figure 2.11: Both the directional-hemispherical and hemispherical-directional reflectance are the same.

is reflected or transferred. This quantity is the reflectivity $R$ and the transmittance $T$; they are solved with the Maxwell’s equations. In the case of polished surfaces, these equations are known as the Fresnel equations.

There are two types of equations: (a) for dielectrics materials, that are essentially non-magnetic, such as crystal, and (b) for electrical conductors, for example metals. In either case the polarization of light is considered and two coefficients (one perpendicular $\perp$ and other parallel $\parallel$) associated with the phenomenon of reflection $r$ and transmission $t$.

In order to comply with energy conservation $\Phi$ is equal to $F \Phi + T \Phi$. This implies that $T = 1 - F$, where $F$ —or more conveniently $F(\theta)$— is the Fresnel term also known as the Fresnel factor. In the computation of the reflected or transmitted radiance the expressions for both are:

$$L_r = F L_i,$$
$$L_t = T \left(\frac{\eta_t}{\eta_i}\right)^2 L_i.$$

2.3.4.1 Case A: Dielectrics

To compute reflectance or transmittance for a dielectric we need the index of refraction for both media. Some examples are given in Table 2.1. The coefficients for this case are:

$$r_\parallel = \frac{\eta_t \cos(\theta_i) - \eta_i \cos(\theta_t)}{\eta_t \cos(\theta_i) + \eta_i \cos(\theta_t)},$$
$$r_\perp = \frac{\eta_t \cos(\theta_i) - \eta_i \cos(\theta_t)}{\eta_t \cos(\theta_i) + \eta_i \cos(\theta_t)},$$
$$t_\parallel = \frac{2 \eta_t \cos(\theta_i)}{\eta_t \cos(\theta_i) + \eta_i \cos(\theta_t)},$$
$$t_\perp = \frac{2 \eta_t \cos(\theta_i)}{\eta_t \cos(\theta_i) + \eta_i \cos(\theta_t)}.$$
2.3. The Local Reflection of Light

<table>
<thead>
<tr>
<th>Medium</th>
<th>Index of refraction $\eta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vacuum</td>
<td>1.0</td>
</tr>
<tr>
<td>Air (at sea level)</td>
<td>1.00029</td>
</tr>
<tr>
<td>Ice</td>
<td>1.31</td>
</tr>
<tr>
<td>Water ($20^\circ$ C)</td>
<td>1.333</td>
</tr>
<tr>
<td>Quarz</td>
<td>1.46</td>
</tr>
<tr>
<td>Crystal</td>
<td>1.5 - 1.6</td>
</tr>
<tr>
<td>Diamont</td>
<td>2.42</td>
</tr>
</tbody>
</table>

Table 2.1: Indexes of refraction for some common objects.

<table>
<thead>
<tr>
<th>Material</th>
<th>Index of refraction $\eta$</th>
<th>Absorption coefficient $\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gold</td>
<td>0.37</td>
<td>2.82</td>
</tr>
<tr>
<td>Silver</td>
<td>0.177</td>
<td>3.638</td>
</tr>
<tr>
<td>Copper</td>
<td>0.617</td>
<td>2.63</td>
</tr>
<tr>
<td>Iron</td>
<td>2.485</td>
<td>3.433</td>
</tr>
</tbody>
</table>

Table 2.2: Average values for some representative metals.

The value of $F$ is a weighted sum for the perpendicular and parallel coefficients such that $s + p = 1$:

$$ F = s F_{\parallel} + p F_{\perp}, $$

where $F_{\parallel} = r_{\parallel}^2$ and $F_{\perp} = r_{\perp}^2$.

If our system does not consider polarized light then $s = p = 1/2$. Similarly the transmittance is computed as:

$$ T_{\parallel} = \left( \frac{\eta_t \cos(\theta_t)}{\eta_i \cos(\theta_i)} \right) t_{\parallel}^2, $$

$$ T_{\perp} = \left( \frac{\eta_t \cos(\theta_t)}{\eta_i \cos(\theta_i)} \right) t_{\perp}^2. $$

2.3.4.2 Case B: Electric Conductors or Metals

Electric conductors do not transmit light, instead light is absorbed and transformed in heat. To compute reflected energy we need two coefficients dependent on the wavelength: the index of reflection $\eta$ and the absorption coefficient $\kappa$ (some examples are given in Table 2.2).
The coefficients for this case are:

\[ r_2^\parallel = \frac{(\eta^2 + \kappa^2) \cos \theta_i^2 - 2\eta \cos \theta_i + 1}{(\eta^2 + \kappa^2) \cos \theta_i^2 + 2\eta \cos \theta_i + 1} \]
\[ r_2^\perp = \frac{(\eta^2 + \kappa^2) - 2\eta \cos \theta_i + \cos \theta_i^2}{(\eta^2 + \kappa^2) + 2\eta \cos \theta_i + \cos \theta_i^2}. \]

Sometimes it is not possible to know \( \eta \) and \( \kappa \) for specific material. In this case we should use the approximation to \( F \) that appears in [CT82]. The simplification consists of knowing the value of the reflectance at normal direction — \( F(0) = 1 \) always — to solve \( \eta \) and \( \kappa \). The expression of \( F \) given by Cook-Torrance is:

\[ F(\theta) = \frac{1}{2} \frac{(g - c)^2}{(g + c)^2} \left\{ 1 + \frac{c(g + c) - 1}{c(g - c) + 1} \right\}^2, \]
where \( c = \cos(\theta) \) and \( g^2 = \eta^2 + c^2 - 1 \). Using the normal incidence direction the formula reduces to:

\[ F(0) = \left\{ \frac{\eta - 1}{\eta + 1} \right\}^2, \]
and we solve \( \eta \):

\[ \eta = \frac{1 + \sqrt{F(0)}}{1 - \sqrt{F(0)}}. \]

To find an approximation for the absorption coefficient, we assume \( \eta = 1 \), which gives an alternative for the Fresnel term at normal incidence:

\[ F(0) = \frac{\kappa^2}{\kappa^2 + 4}, \]
then we solve:

\[ \kappa = 2 \sqrt{\frac{F(0)}{1 - F(0)}}. \]

### 2.4 The Radiance Equation

In this final section we present the most important equation for realistic rendering systems. It was Kajiya [Kaj86] who first introduced a formulation for this significant magnitude with a Fredholm integral of second order to represent the light transport equation (LTE). The transport equation describes the distribution of radiance at equilibrium, and thus \( L_i(x, v) = L_r(y, u') \). Where \( y \) is the closest point of intersection
2.4. The Radiance Equation

Figure 2.12: The radiance leaves a point \( y \) and arrives to a point \( x \).

returned by the ray casting function \( t \) (see Figure 2.12 for geometrical considerations).

We consider the total radiance that leaves a surface point at some direction \( L(x, u) \) to be composed of the radiance emitted by the surface \( L_e \) and the radiance that is reflected \( L_r \) (due to the effect of an incoming illumination \( L_i \) obtained by integrate over the incoming hemisphere of directions). This is a complex calculation in the sense that radiance appears in its own definition. For ease we omit the \( x \) in the following formulation:

\[
L_r(u) \overset{\text{def}}{=} \int_{\Omega} dL_r(u) \, d\sigma(v) \quad (2.10)
\]

\[
= \int_{\Omega} f_r(u, v) L_i(v) (v \cdot n) \, d\sigma(v). \quad (2.11)
\]

Sometimes it is not possible to solve analytically the reflected radiance, that is why it is approximated usually with a Monte-Carlo estimator. For clarity, we give the final expression for the radiance equation using the projected solid angle measure.

\[
L(u) = L_e(u) + L_r(u) \quad (2.12)
\]

\[
= L_e(u) + \int_{\Omega} f_r(u, v) L_i(v) \, d\sigma_p(v). \quad (2.13)
\]
A Study of BRDF Models

Shading models do not always consider reality (or physics) and it is commonly assumed that what the eye cannot see does not require the computation. We have already seen a classification for the BRDF function in the previous chapter, so the user could choose between analytical models with no theoretical qualities, or experimental ones that fit measured data. That being said, there are some more considerations to take into account in order to choose a reflection model and its parameters. This is an important task, and it deserves a chapter in this dissertation.

You should also see that the notation used here is different from that used in every original work. The truth is, that original notation differs a each author, so the reader will be satisfied to discover a way which clearly presents the formulation and make comparisons easy between different models. In order to facilitate this, we use a chronology line of presentation for each reflectance model or BRDF.

We introduce a notation based in our GRF (GRanada File) scene file format, together with the description of each reflectance model. This file is used as an input file for our Monte Carlo rendering system. For more details about this scene format see Appendix A.

3.1 Ideal Specular Reflection (1621)

In ray tracing algorithms [Whi80, Arv86] a ray is sent to the scene. On each surface it is ideally reflected until it is absorbed. Ideal specular reflection reflects in a single direction following the law of reflection (see section 2.2.1 for more details). The BRDF in this case is a delta dirac distribution, $\delta$, giving always zero, except for the reflection direction $r$. This reduces considerably the radiance computation since $L_{r,s}(x, u) = L_r(x, ru)$. 

23
Chapter – 3. A Study of BRDF Models

\[ f_{r,s}(u, v) = \rho_s(v) \delta_u(v), \]  \hspace{1cm} (3.1)

where \( \rho_s \) is the specular reflectance at the point and \( \delta \) is defined as follows:

\[ \delta_u(v) \overset{\text{def}}{=} \begin{cases} 
1 & \text{if } v = r_u, \\
0 & \text{else.}
\end{cases} \]

We give the GRF format for this BRDF. Note that with GRF we write \( \rho_s \) as \( ks \).

\textbf{PerfectSpecularBRDF} ::= \\
\textbf{brdf} \text{ specular} \text{ } \text{ks} = \text{value} .

\textbf{Table 3.1: GRF notation of PerfectSpecularBRDF}

3.2 Lambert (1760)

A diffuse surface has a BRDF that is equal for all incident and outgoing directions. This also substantially reduces the computations and thus is commonly used to model diffuse surfaces, even though there are no ideally diffuse materials in the real world. This BRDF is expressed as:

\[ f_{r,d}(u, v) = \frac{\rho_d}{\pi} \]  \hspace{1cm} (3.2)

where \( \rho_d \) is the diffuse reflectivity. It included the value \( \pi \) as a normalization, so is the result of integrate \( \cos(\theta_v) \) in the hemisphere of directions.

We give the BRDF expression using our notation (GRF) we write \( \rho_d \) as \( kd \).

\textbf{PerfectDiffuseBRDF} ::= \\
\textbf{brdf} \text{ diffuse} \text{ } \text{kd} = \text{value} .

\textbf{Table 3.2: GRF notation of PerfectDiffuseBRDF}

3.3 Minnaert (1941)

One of the first reflectance models was given by Minnaert [Min41] who wanted to model the lunar surface reflectance. This can be applied not only to the moon, but to an object where we would see a darkening of colour near edges. The model is controlled using two parameters \( \rho_d \) and an exponent responsible for darkening \( k \). The analytical expression for this BRDF is:

University of Granada
\[ f_r(u, v) = \frac{\rho_d}{\pi} (\cos(u) \cos(v))^{k-1}. \]  \hspace{1cm} (3.3)

when \( k = 1 \), this function is equivalent to the lambertian function. We give the BRDF expression using our notation (GRF), we write \( \rho_d \) as \( kd \).

\[
\text{MinnaertBRDFs} ::= \brdf \text{ minnaert } k = \text{value} \ kd = \text{value}
\]

**Table 3.3**: GRF notation of **MinnaertBRDFs**

**Figure 3.1**: The *Minnaert-BRDF* instance with our GLSL BRDF editor.
3.4 Torrance-Sparrow (1967)

This is a physical model that considers polarized light and is used for rough surfaces [TS66, TS67]. The roughness is modelled using microscopic concavities in V-form of equal length called microfacets (see Figure 3.2). Their orientation is random and is controlled by parameters, so it is possible to simulate different degrees of roughness. The complete BRDF function is as follows:

\[
f_r(u, v) = \frac{k_d}{\pi} + \frac{k_s}{4\pi (n \cdot v)} D(h) F(\theta_u) G(u, v).
\]  \tag{3.4}

Many important terms appear in the previous expression, all of them dependent of \( \lambda \) value. Let us consider them separately.

- The microfacets distribution \( D : \Omega \to \mathbb{R} \) with \( D(h) \in [0, 1] \) gives a distribution of the normals of the facets that are aligned relative to \( h \) vector and it is parameterized by \( m \). Many authors use the Gaussian distribution function but the Beckmann distribution [BS63] is more common:

\[
D(h) = \frac{1}{m^2 \cos(\delta)^4} \exp \left( \frac{\cos(\delta)^2 - 1}{m^2 \cos(\delta)^2} \right)
\]

- The Fresnel factor \( F(\theta_u) \in [0, 1] \) gives the fraction of light that is reflected on each microfacet. Its computation is a linear combination of the coefficient for each polarization state as was given in section 2.3.4 about the Fresnel equations.

  - Coefficient for perpendicular light polarization
    \[
r_\perp(\theta) = \frac{a^2 + b^2 - 2a \cos(\theta) + \cos^2(\theta)}{a^2 + b^2 + 2a \cos(\theta) + \cos^2(\theta)}
    \]
    where:
    \[
    2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2(\theta))^2 + 4\eta^2\kappa^4 + (\eta^2 - \kappa^2 - \sin^2(\theta))}
    \]
    \[
    2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2(\theta))^2 + 4\eta^2\kappa^4(\eta^2 - \kappa^2 - \sin^2(\theta))}
    \]
  - Coefficient for parallel light polarization
    \[
r_\parallel(\theta) = r_\perp(\theta) \frac{a^2 + b^2 - 2a \sin(\theta) \tan(\theta) + \sin^2(\theta) \tan^2(\theta)}{a^2 + b^2 + 2a \sin(\theta) \tan(\theta) + \sin^2(\theta) \tan^2(\theta)}
    \]
3.5. Beard-Maxwell (1973)

- The Geometric attenuation factor $G(u, v) \in [0, 1]$ expresses, for two directions, the ratio of light that is not occluded by the surface due to masking or shadowing. There are some situations in which a portion of light will not reach its destination, as it is shown in Figure 3.3. This model takes into account these effects with the following evaluation:

$$G(u, v) = \min \left\{ 1, \frac{2(n \cdot h)(n \cdot u)}{(u \cdot h)}, \frac{2(n \cdot h)(n \cdot v)}{(u \cdot h)} \right\}.$$  

![Figure 3.2: The normals of the microfacet are distributed with reference to the surface normal.](image)

The Torrance-Sparrow BRDF is one of the most complete physical reflection models for isotropic materials.

### 3.5 Beard-Maxwell (1973)

The Beard-Maxwell BRDF [MBWL73] is an empirical model that simulates, based on physical properties, a specific type of material, that is, painted surfaces. Reflection is modelled with two main components, the superficial $f_{r, sup}$ and the volumetric $f_{r, vol}$. The superficial is almost a specular reflection that takes place in the first layer of the surface. This layer is represented by microfaces whose normals are oriented following a statistic distribution $D$. The volumetric component is due to the scattering of light at subsurface level. It is produced principally by two actions: local specular reflection on the first layer (whose normal is $n$) and volumetric reflection approximation which comes from the reflection of light on the interior layers.

$$f_r(\lambda(u, v)) = f_{r, sup}(\lambda(u, v)) + f_{r, vol}(\lambda(u, v)).$$

The superficial reflection component is governed by the Fresnel term for dielectrics (using $\kappa = 0$ y $\eta = 1, 65$), and is calculated as:

$$f_{r, sup}(u, v) = \frac{F(\beta)}{F(0)} \frac{f_r(h) \cos^2(h)}{\cos(u) \cos(v)} SO(u, h, r, \Omega),$$
where $\beta$ is the bisector angle (graphically found in Figure 2.8, page 14) and

$$f_r(h) = \frac{F(0) D(h)}{4 \cos(u) \cos(v)}.$$

As with Torrance-Sparrow [TS66, TS67], the surface is modelled with microfacets so the same shadowing and masking (or obscuration) could occur. Though the light is blocked in the same form (like in Figure 3.3), the formulation that models this effect is not the same. The term $SO$ summarizes both:

$$SO(u, h, \tau, \Omega) = 1 + \frac{\theta_h}{\Omega} e^{-2\beta/\tau} \left( \frac{1}{1 + \frac{\theta_h}{\tau}} \right).$$

Some values like $\tau$, $\Omega$ take value from measured data. In recent work, Westlund and Meyer [WM02], publicised a database of 400 measured materials \footnote{NEF User Guide, 9.1 \texttt{<http://math.nist.gov/~FHunt/aapearance/nedfs.html>}} which fitted with a modified Beard-Maxwell BRDF named NEF-BM. This modification simplifies the original formulation by dropping the parameter $\phi_h$ and the $f$ and $g$ functions.

![Figure 3.3](image)

**Figure 3.3:** From left to right: *masking*, the viewer does not see all the light that is reflected by a microfacet; *shadowing*, the incident beam is blocked due to occlusion; *interreflection* light that reaches the observer comes from more than one reflection event.

The volumetric component is a simple description of the subsurface scattering of the light, where light is supposed to be depolarized. For a complete description of the formula —for $f$ and $g$ function— please refer to the original paper [MBWL73].

$$f_{r,vol}(u, v) = \frac{2 \rho_v f(\beta) g(\theta_h)}{\cos(u) \cos(v)}.$$
where $\rho_v$ is the measured reflectance of the surface when $\theta_u = \theta_v = 0$ and $f(\beta) = g(\theta_h) = 1$ is used. Finally the description of this BRDF when GRF notation is used is as follows:

\[
\text{BeardMaxwellBRDF} ::= \text{brdf bearmaxwell } \omega=value \tau=value \nreal=value \nimag=value \text{ Rs=value Rd=value Rv=value}
\]

Table 3.4: GRF notation of BeardMaxwellBRDF

Figure 3.4: The Beard-Maxwell-BRDF instance with our GLSL BRDF editor.
3.6 Phong (1975)

Phong [Pho75] is still a very popular BRDF model and it was the first description for non-lambertian surfaces. Basically, it is an empirical model which does not obey neither energy conservation nor reciprocity, but its simplicity has made it one of the most used in Computer Graphics. It depends on an \( \alpha \) angle, which is obtained after reflecting the incident direction. Essentially, this model is a simplification of the Torrance & Sparrow one, where the \( G \) and \( F \) factors are dropped and the distribution \( D \) is reduced to:

\[
D(u, v) = \cos^n(\alpha) = (\mathbf{u} \cdot \mathbf{r}_v)^n
\]

where the parameter \( n \in [0, \infty] \) characterizes the shape of the specular shine (from 0 or dull to more glossy surfaces).

Furthermore, this model could be faster with an optimization of the exponential operator, like the one given by Schlick [Sch94a]:

\[
\cos^n(\alpha) \approx \frac{\cos(\alpha)}{n - n \cos(\alpha) + \cos(\alpha)}
\]

The second and third parameters are the specular \( k_s \) and diffuse \( k_d \) constant, both taking values in \([0, 1]\), that allow a linear combination of the specular lobe with the lambertian-diffuse BRDF.

\[
f_r(u, v) = k_d (\mathbf{v} \cdot \mathbf{n}) + k_s (\mathbf{u} \cdot \mathbf{r})^n,
\]

And its GRF definition is:

\[
\text{PhongBRDF} ::= \text{brdf phong n=value ks=value kd=value}
\]

Table 3.5: GRF notation of PhongBRDF

3.7 Blinn (1977)

The Blinn BRDF, also known as the Blinn-Phong reflection model, is the standard lighting model used in both the DirectX and OpenGL rendering pipelines. Blinn [Bli77] noted that normally the direction that gave the higher reflection corresponds to the halfway \( \mathbf{h} \) between the light and view vectors. This vector is then used in conjunction
3.7. Blinn (1977)

with the normal vector to get the specular highlight.

\[ D(h) = (n \cdot h) = \cos(\delta). \]

In this way, Phong BRDF is modified replacing \((u \cdot r)\) by \((n \cdot h)\), allowing less calculations, because it is not necessary to find the reflection vector.

\[ f_r(u, v) = k_d (u \cdot n) + k_s (n \cdot h)^n \quad \text{with} \quad k_d + k_s = 1. \quad (3.6) \]

\begin{verbatim}
BLINNBRDF := brdf blinn n=value ks=value kd=value
\end{verbatim}

Table 3.6: GRF notation of BLINNBRDF

In figure 3.5 two snapshots of our BRDF editor on the GPU, show that the visual appearance of empirical reflectance models like Phong or Blinn, are similar to those of plastics objects.
3.8 Cook-Torrance (1981)

What happens in reality is neither a pure diffuse nor a pure specular surface, so a linear combination of them is not the best option. Looking for a solution using the Geometrical Optics theory, Cook & Torrance used previous works [BS63, TS67] together with some new ideas: only those microfacets oriented toward \( h \) vector contribute to the final reflection. This model [CT82] also introduces a new type of material in the field of rendering, differentiating between metallic and non-metallic surfaces. Reflection is described using three components: the ambient, diffuse and specular parts.

\[
\rho = \rho_a + k_s \rho_s + k_d \rho_d, \quad \text{donde} \quad k_s + k_d < 1.
\]

where \( k_s \) and \( k_d \) controls the fraction of energy that is specularly or diffusely reflected which is a characteristic of the material.

- The ambient term comes from an uniform distribution of the incident beam of light which originated at extended light sources. The ambient term is calculated from the specular and diffuse term in order to be energy consistent.
- The diffuse term is represented with a classical lambertian reflection.
- With the normalized halfway vector this model simulates an imaginary surface which behaves as a mirror. The specular term is the compound of the already known \( F, D \) and \( G \).

The Fresnel term represents the reflection of polished surfaces. Its formulation for dielectrics and metals were presented with detail in the previous chapter, section 2.3.4, where an optimization given by [CT82] makes this model more applicable to Computer Graphics.

Some surfaces have two or more scales of roughness, controlled by the slope \( m \), and can be modelled by using two or more distribution functions [BS63]. In such cases, \( D \) is expressed as a weighted sum of the distribution functions, each with a different value of \( m \) parameter. When \( m \) is small, the microfacet normals distribute highly directional towards the reflection direction. As \( m \) value grows, the distribution expands and becomes more uniform. Typically, \( D \) is one the following distributions:

\[
D(h) = \cos(\beta) e^{-(\frac{\tau}{m})^2} \quad \text{Gaussian},
\]

\[
D(h) = \frac{1}{m^2 \cos^4(\alpha)} e^{-[\tan(\alpha)]^2} \quad \text{Beckman for rough surfaces}.
\]

The geometrical attenuation factor \( G \) accounts for the shadowing and masking of one facet by another, as we discussed in previous sections. It definition is identical to
the Torrance-Sparrow model (see section 3.4 page 26). Finally, the specular component is:

\[ f_{rs}(u, v) = \frac{F(\beta) D(h) G(u, v)}{\pi (n \cdot u)(n \cdot v)}. \] (3.7)

Figure 3.6: The Cook-Torrance-BRDF instance with our GLSL BRDF editor.

This model combines statistics and a microfacets physical model which depends on \( \lambda \) and is based on the Fresnel equations. The main disadvantage is with other BRDF representations, the parameters are not intuitively set. The user needs to experiment with the values until a good result is found. Another drawback of the model is that the function is not plausible because for some angles this BRDF does not obey the energy conservation law. Using our GRF format, this function is described as:

\[
\text{Cook-TorranceBRDF} ::= \begin{array}{l}
\text{brdf} \text{torrance} \text{rough} = \text{value} \\
\text{nreal} = \text{value} \\
\text{nimag} = \text{value} \\
\text{ks} = \text{value} \\
\text{kd} = \text{value}
\end{array}
\]

Table 3.7: GRF notation of Cook-TorranceBRDF
3.9 Kajiya (1985)

The Kajiya model [Kaj85] implements an anisotropic method of brute force that estimates the analytical form of the reflected intensity light. It is based on Kirchoff’s approximation and a stationary base method for the approximation of the radiance equation (Eq. 2.12). Kajiya considers a simplified rough surface model and uses in its place the nearest tangent plane.

In this work, Kajiya calculates and stores the reflectance in a table each time a surface illumination is evaluated. Later, values from the table are used to perform linear interpolation. Though he uses a novel numerical technique that explores unpolarized light’s properties and the Fresnel term, this model shows a lack of many important inter-reflection phenomena. Moreover, conservation the energy is not guaranteed.

3.10 Poulin-Fournier (1990)

Poulin and Fournier [PF90] introduced a model for anisotropic materials that considered reflection and refraction of the incident light. Anisotropy is represented by a set of parallel cylinders packed tightly together such as those used in [Mil84], whose optical properties—and thus the degree of anisotropy—are described by varying the height and distance of these cylinders.

The surface is modelled by a distribution of cylinders that are added or subtracted from the base surface. The phenomenon of reflection (similarly refraction) happens at the longitudinal cut of the cylinder, more specifically, at the binormal plane BN where \( \mathbf{b} = \mathbf{t} \times \mathbf{n} \), so we assume the local system is formed by \( \mathbf{n}, \mathbf{t} \) and \( \mathbf{b} \) vectors (see Figure 2.8 at section 2.3.1.2 in page 14).

To control the degree of anisotropy, this model uses two parameters: distance between cylinders, \( d \in [0, \infty) \) and height from the base surface, \( h \in [0, 1] \). Both are unitless and appear in Figure 3.7. When \( d = 0 \), this model is the Torrance-Sparrow BRDF. When \( d = 2 \), the normal of the cylinders differentiate in \( \pi \) and we get the maximum anisotropy. However, if \( d > 2 \) then an imaginary base surface with a constant normal vector appears, and this also takes place in the reflection of light. The height of the base surface is then controlled with the \( h \) parameter. Increasing \( h \) causes the decrease of normal variation, thus softening anisotropy.

The general equation given for the calculation of the reflected intensity is now shown bellow, and its terms further described:

\[
I_r = \frac{(I_a \times l_{vi}) + (I_f \times f_{vi})}{l_v + f_v}.
\]  

(3.8)
The computation of the portion of light that is reflected has in the numerator a sum of the diffuse and specular contributions for reflection. Let us explain every term:

- $I_a$: the amount of light that is reflected by the cylinders’ visible portion.
- $I_f$: the amount of light that is reflected by the visible part of the base surface. For this, a simple computation is recommended considering $u$ and $v$ as constant vectors. This assumption is valid as long as the size of the cylinders are very small in comparison with the location of those vectors.
- $l_v$: the length of the visible part of the cylinder’ projection into the base surface. Used in the calculus of $I_a$.
- $l_{vi}$: the length of the visible and illuminated part of the cylinder’ projection into the base surface. Used in the calculus of $I_a$.
- $f_{vi}$: the length of the visible and illuminated part of the base surface. Used in the calculus of $I_a$.

This BRDF model also considers the phenomena of shadowing and masking of the cylinders (similar to what happens in the microfacets model). For this purpose, some angles should be evaluated:

- $\theta_{hs}$: Angle at which, the cylinder becomes hidden to itself from the viewer,
- $\theta_h$: and the same but considering this had happened with other cylinders.
- $\theta_{ss}$: Angle at which the cylinder starts to block light from the viewer,
- $\theta_s$: and the same but considering this had happened with other cylinders.

The length of the visible part of the projected cylinder is:
Chapter – 3. A Study of BRDF Models

\[ l_v = \frac{\sin(\theta_h - \theta_u) + \sin(\theta_{sh} - \theta_u)}{\cos(\theta_u)}. \]

For the visible and lighting part the formulation is as follows:

\[ l_{vi} = \begin{cases} \frac{\sin(\theta_u - \theta_u) + \sin(\theta_u + \theta_u)}{\cos(\theta_u)} & \text{if } \mathbf{u}, \mathbf{v} \text{ at the same side than } \mathbf{n}, \\ \frac{\sin(\theta_u - \theta_u) + \sin(\theta_v + \theta_u)}{\cos(\theta_u)} & \text{in other case}. \end{cases} \]

Let \( f \) be the total base surface length. The visible part of it, \( f_v \), is evaluated as \( f - f_h \), where \( f_h \) represents the hidden length. Similarly the illuminated part is \( f_i = f - f_s \), where the subscript \( s \) mean shadow. We can calculate \( f_{vi} \) as \( f_v + f_i \). Finally we have:

\[
\begin{align*}
f &= d - 2\sqrt{1 - h^2}, \\
f_h &= \min \left( \frac{1}{\cos(\theta_u)} - \sqrt{1 - h^2} - \left( \frac{h \sin(\theta_v)}{\cos(\theta_v)} \right), f \right), \\
f_s &= \min \left( \frac{1}{\cos(\theta_u)} - \sqrt{1 - h^2} - \left( \frac{h \sin(\theta_u)}{\cos(\theta_u)} \right), f \right).
\end{align*}
\]

We already have commented about the calculus for the reflection event. Transmission is same with a simple exception: the cylinders are negative or modelled in reverse. In each case, we have a very useful set of normals in the computation of reflection direction. By contrast, the visual result depends upon knowing how to set the parameters correctly. This model is not energy conservating.

Figure 3.8: The Poulin-Fournier-BRDF instance with our GLSL BRDF editor.
3.11. Strauss (1990)

For a description of this BRDF using our GRF notation, we have the following definition:

\[
\text{Cylinders-PF-BRDF} ::= \\
\begin{align*}
\text{brdf} & \text{ poulin d } = \text{value} \quad \text{h } = \text{value} \\
& \text{n } = \text{value} \quad \text{ks } = \text{value} \quad \text{kd } = \text{value}
\end{align*}
\]

Table 3.8: GRF notation of Cylinders-PF-BRDF

3.11 Strauss (1990)

The Strauss BRDF [Str90] is an empirical model distinguishing characterized metal and other glossy surfaces which exhibits off-specular peaks.

Strauss noted on previous BRDF models how difficult it was to select the correct values for those BRDFs when you wish to mimic the aspect of a material. For example, which is the right Phong exponent value if you want to produce half of the highlight? We already know that Phong is not physically based, and that \( ks \) and \( kd \) parameters are not constrained, but how can we relate the shiny-control shape parameter with the specularity of the surface? We can not. Another issue concern with the microfacet-based BRDF models is the need for acquired data to set the optical properties. Sometimes is not possible to represent any material, at any time and anywhere. Basically, this is the problem that Strauss found a solution for. His model is based on few and very intuitive parameters. These parameters represent different types of material properties that any designer or computer graphics user should be able to set without problems. There are five values:

- Chromatic colour \( C \) or surface colour.
- Specular colour \( C_s \) is the calculated (not a parameter) shiny colour.
- Smoothness: when \( s = 0 \) we have the perfect diffuse surface, and when \( s = 1 \), the surface behaves like a mirror. This value changes the ratio from diffuse to specular and also the shape of the shine.
- Metalness: when \( m = 1 \) the surface is full metallic, and when \( m = 0 \) is the absence of this property. It change the diffuse aspect of the resulting image and also is involved with the specular colour computation.
- Transparency: when \( t = 0 \) a totally opaque surface, when \( t = 1 \) all light not reflected will be transmitted through the media. The transmittance value (see section 2.3) used is the value of \( t \) at normal incidence direction.
Chapter – 3. A Study of BRDF Models

- Index of refraction $\eta$: used to compute the direction of the transmitted beam of light.

The Strauss reflectance model uses a lineal combination of a diffuse and specular components controlled by two coefficients $k_d$ and $k_s$. That is:

$$f_r(u, v) = k_d f_{r,d}(u, v) + k_s f_{r,s}(u, v),$$

where

$$f_{r,d}(u, v) = (u \cdot n) d r_d C,$$

$$f_{r,s}(u, v) = r_s C_s.$$

In order to obtain the values we need, the model performs some simple operations:

$$r_d = (1 - s^3)(1 - t) \quad d = (1 - ms)$$
$$r_n = (1 - t) - r_d \quad e = \frac{3}{s}$$
$$r_j = \min[1, r_n + (r_n + k_j) j] \quad k_j = 0.1$$
$$r_s = [-(h \cdot v)]^e r_j \quad j = F(\alpha) G(\alpha) G(\delta)$$

The meaning of $r_j$ is similar to the Fresnel term together with the geometric term. It uses two constant values $k_f = 1.12$ and $k_g = 1.01$, as well as a cosine term $x \in [0, 1]$.

$$F(\theta) = \frac{\frac{1}{(x-k_f)^2} - \frac{1}{k_f^2}}{\frac{1}{(1-k_f)^2} - \frac{1}{k_f^2}}$$
$$G(\theta) = \frac{\frac{1}{(1-k_g)^2} - \frac{1}{k_g^2}}{\frac{1}{(1-k_g)^2} - \frac{1}{k_g^2}}$$

The model precomputes $F$ and $G$, keeping its simplicity together with other advantages. For example, for metals, the colour of the reflection should not be the same as the colour of the surface (this is the case for non-metal materials). That is:

$$C_s = \begin{cases} 
C_1 & \text{if non-metal}, \\
C_1 + m (1 - F(\alpha)) (C - C_1) & \text{if metal}.
\end{cases}$$

where $C_1$ is the equivalent to white specular colour that it is acquired at $\pi/2$ incidence angle. In our rendering system we express the smoothness (noted as $s$), the metalness (noted as $m$) and other parameters as is shown bellow using GRF format:
3.11. Strauss (1990)

\[
\text{StraussBRDF} ::= \\
\text{brdf} \text{ strauss} \quad s = \text{value} \quad m = \text{value} \\
\text{ks} = \text{value} \quad \text{kd} = \text{value}
\]

**Table 3.9:** GRF notation of StraussBRDF

![Image of Strauss-BRDF instance with our GLSL BRDF editor.](image)

**Figure 3.9:** The Strauss-BRDF instance with our GLSL BRDF editor.
Chapter – 3. A Study of BRDF Models

3.12 He-Torrance-Sillion-Greenberg (1991)

One of the most complete and complex physical BRDF models is the one of He et al. [HTSG91]. It considers important phenomena associated with the wavelike nature of light. These are polarization, diffraction and interference. Many parameters contribute to the representation of isotropic materials. Some are geometrical, such as angles, and are illustrated on Figure 2.2, and others are physical, for example the wavelength $\lambda$ of the incident beam, the index of refraction $\eta$ (used in the Fresnel term calculation) and others parameters $\sigma$ and $\tau$ are used to characterize the roughness. These are presented in Figure 3.10.

Figure 3.10: The model uses the Geometric Optics theory.

This BRDF starts by distinguishing three components in the reflection event: pure specular $f_{r_{sp}}$, uniform diffuse reflection $f_{r_{ud}}$, and directional diffuse $f_{r_{dd}}$. Figure 2.10 has a graphical description of them.

$$f_r = f_{r_{sp}} + f_{r_{ud}} + f_{r_{dd}},$$

The authors describe implementation based on a spherical harmonic decomposition and Kirchoff’s theory. They use it to encodes $f_{r_{ud}}$ and $f_{r_{dd}}$ reflectance distribution functions for materials, and directional intensity distributions for illuminated surfaces. The perfect specular component is represented by the Fresnel equations. The expression of a reflectance component approximation:

$$f_r(u, v) \cos(v) = \sum_{l=0}^{N} \sum_{m=-l}^{l} B_{l,m}(u) Y_{l,m}(v)$$

where $Y_{l,m}$ is the normalized spherical harmonic, and $B_{l,m}$ is the coefficient.

In order to reduce the computation times —prohibitive for almost any rendering system— there was an attempt to precompute and store key value of the model in tables [HHP+92]. Again, this introduce another problem: storage cost.
This model is theoretical and is still not practical to integrate in a Monte-Carlo based system since it has no associated direct sampling method. In addition, measurements are required in order to set the parameters of this BRDF. However, it compares favourably with experimental acquired data. In our notation the parameters are:

\[
\text{HTSG-BRDF} ::= \text{brdf} \ \ \text{he} \ \ \sigma_{\text{value}} \ \ \ \text{tau} = \text{value} \\
\text{lambda} = \text{value} \ \ \text{ks}=\text{value} \ \ \text{kd}=\text{value}
\]

Table 3.10: GRF notation of HTSG-BRDF

3.13 Ward (1992)

This is an important BRDF model that introduces the experimental classification that uses no theoretical justification for the results. Firstly, Ward [War92] developed a device to acquire the reflectance of a surface sample. Secondly, he gave a mathematical description of the reflectance of anisotropic materials. Previous BRDF models that account for anisotropy were computationally expensive.

This reflection model uses a Gaussian lobe controlled by the standard deviation. For the isotropic version of the model this parameter is \(\alpha_m\). The BRDF uses the halfway vector \(\text{h}\) and two constants for the combination of the diffuse and specular reflection.
components of the reflection. This BRDF can be validated using experimental measurements. The mathematical expression is:

\[ f_{r,iso}(u, v) = \frac{k_d}{\pi} + \frac{k_s}{\cos(u) \cos(v)} \exp \left[ -\tan^2 \left( \frac{\phi}{\cos(u)} \right) \right] \frac{1}{4\pi \alpha_m^2}, \]  

(3.9)

The normalization factor \( \frac{1}{4\pi \alpha_m^2} \) guarantees the correct integration of this function on the hemisphere of directions. The model produces quite high values—which should be avoided in a rendering system—when \( \alpha_m > 0.2 \) and normalization is able to correct this.

The anisotropic version of the mathematical formulation uses an elliptic Gaussian whose axes distribute in a way that is controlled by two parameters \( \alpha_x \) and \( \alpha_y \).

\[ f_{r,ani}(u, v) = \frac{k_d}{\pi} + \frac{k_s}{\sqrt{\alpha_x \alpha_y} \sqrt{\cos(u) \cos(v)}} \exp \left[ -\tan^2 \left( \frac{\phi}{\cos(u)} \right) \right] \frac{1}{4\pi \alpha_x \alpha_y \cos(u) \cos(v)} \]  

(3.10)

An equivalent and computer-friendly version of the anisotropic model is also given by Ward:

\[ f_{r,ani}(u, v) = \frac{k_d}{\pi} + \frac{k_s}{\sqrt{\alpha_x \alpha_y} \sqrt{\cos(u) \cos(v)}} \exp \left[ -\tan^2 \left( \frac{\phi}{\cos(u)} \right) \right] \frac{1}{4\pi \alpha_x \alpha_y \cos(u) \cos(v)} \]  

(3.11)

where

\[ (h \cdot x) = \frac{\sin(\theta_u) \cos(\phi_u) + \sin(\theta_v) \cos(\phi_v)}{|h|}, \]

\[ (h \cdot y) = \frac{\sin(\theta_u) \sin(\phi_u) + \sin(\theta_v) \sin(\phi_v)}{|h|}, \]

\[ (h \cdot n) = \frac{\cos(\theta_u) + \cos(\theta_v)}{|h|}, \]

\[ |h| = \sqrt{[2 + 2 \sin(\theta_u) \sin(\theta_v) (\cos(\phi_u) \cos(\phi_v) + \sin(\phi_u) \sin(\phi_v)) + 2 \cos(\theta_u) \cos(\theta_v)]^{1/2}}. \]

This simple model accounts for anisotropy and more importantly is able to fit real data. It uses a reduced number of parameters, can be seen using our notation:
Figure 3.12: The Ward-BRDF instance with our GLSL BRDF editor. The image on the left corresponds to an isotropic instance. The image on the right has an anisotropic parametrization.

\textbf{WardBRDF} ::= \texttt{brdf ward sx=\textit{value} sy=\textit{value} ks=\textit{value} kd=\textit{value}}

Table 3.11: GRF notation of WardBRDF
3.14 Westin (1992)

As with a pendulum movement, after the inclusion of complex physically-based BRDFs [CT82, HTSG91, Kaj85], a mixture of a physical model with fast-computations was introduced by Westin et al. [WAT92]. This BRDF distinguishes three components $f_r = f_{sp} + f_{ud} + f_{dd}$. Let us start with the specular part. The reflection is dependent on the direction of incidence, the wavelength and the index of refraction is modelled with the Fresnel term.

\[ f_{sp}^{sp}(u,v) = |F(\theta_u)|^2 e^{-g(\sigma, \lambda)} S(\tau), \]

where $G$ is the geometry coefficient of Torrance-Sparrow and the distribution $D$ used is an infinite summary function:

\[ D(u, v) = \frac{\pi^2 r^2}{4\lambda^2} \sum_{m=1}^{\infty} \frac{g(\sigma, \lambda)^m e^{-g(\sigma, \lambda)}}{m!m} e^{-\frac{h(\sigma, \lambda, u, v)}{4m}}, \]

\[ h(\sigma, \lambda, u, v) = \left( \frac{2\pi\sigma}{\lambda} \right)^2 \left( (\sin(\theta_v) \cos(\phi_u) - \sin(\gamma_u))^2 + (\sin(\theta_u) \sin(\phi_u))^2 \right). \]

3.15 Lewis (1993)

Reviewing of the previous models, Lewis [Lew93] made a study of their properties. A BRDF is plausible if it is energy conservative and reciprocal. Models like Phong or
Cook-Torrance did not exhibit such properties, and so Lewis elaborated a plausible version of them.

With reference to microfacets based surfaces, the cause of plausibility is the normal distribution $D$, so it should be normalized in order that the BRDF can be used safely in any lighting algorithms. The total value of the projected microfacets areas should be same as $dA$ of the surface. If we assume energy conservation, the following should occur:

$$\int_S D(h) \cos(\theta_h) \, dA(h) = \int_\Omega D(h) \cos(\theta_h) \, d\sigma(h) = 1$$

In the case of the Phong distribution $D(u, v) = (u \cdot r_v)^n$, this integrate as this:

$$\int_\Omega c(u \cdot r_v)^n \cos(\theta_u) \, d\sigma(u) = \int_0^{2\pi} \int_0^{\pi/2} c(u \cdot r_v)^{n+1} \sin(\theta_u) \, d(\theta_u) \, d(\phi_u),$$

$$= 2 c \pi \int_0^1 u^{n+1} \, du,\quad 2 c \pi \int_0^1 u^{n+2} \, du = 2 c \pi \frac{u^{n+2}}{n+2} \bigg|_0^1 = 2 c \pi \frac{1}{n+2} = 1.$$  

The normalization factor is obtained as follows:

$$c = \frac{n + 2}{2 \pi}. \quad (3.12)$$

If we apply this normalization to the Blinn BRDF model, then energy conservation is guaranteed. The modified version is named Lewis BRDF.

$$f_r(u, v) = k_d \cos(u) + k_s \frac{n + 2}{2 \pi} \cos(\delta)^n. \quad (3.13)$$

Of course, the GRF definition is as Phong and Blinn models:

**LewisBRDF** ::=

```plaintext
brdf lewis n =value ks =value kd =value
```

Table 3.12: GRF notation of LewisBRDF
3.16 Schlick (1993)

The objective of the Schlick-BRDF [Sch93, Sch94b] is a plausible model which is based on the microfacets model and the Fresnel equations. He defines two different equations, one for homogeneous materials and the other for heterogeneous surfaces, and their parameters allow the distinction between isotropy and anisotropy.

Heterogeneous surfaces, such as plastics, exhibit diffuse and specular reflections. Usually both components are combined linearly. This is not appropriate as [SHSL97] indicates. These coefficients should not be constant—they are dependent on the incidence angle—and they should not be combined linearly—in fact they are two different phenomena—so Schlick proposes alternatives ways of representing heterogeneous surfaces: using a two layer model. The outer one for specular reflection and an the inner for diffuse subsurface scattering. In addition, homogeneous materials for example metal, should not distinguish between diffuse and specular reflections.

Homogeneous surfaces are represented by the single layer model with values \((C_\lambda, r, p)\). An heterogeneous surface should use the double layer model with values \((C_\lambda, r, p)\) and \((C'_\lambda, r', p')\). The parameters are:

- \(C_\lambda\) is the reflectance. It depends on \(\lambda\), and gives close values between 0 and 1.
- \(r \in [0, 1]\) describes the roughness. When \(r = 0\) it represent mirrors; if \(r = 1\) we have the perfect diffuse surface.

• $p \in [0, 1]$ describes the isotropy. When $p = 0$ is full anisotropy and if $p = 1$ is full isotropy.

The meaning of these parameters is the same in both the single or double model. This implies that the computation of $S(u)$ in $C_\lambda$ and $S'(u)$ is the same for $C'_\lambda$, only the parameter value differs:

• the spectral term is an approximation of the Fresnel term that is frequently used in computer graphics:

$$S(u) = C_\lambda + (1 - C_\lambda)(1 - u),$$

• the directional term accounts for emission and re-emission of light at subsurface levels

$$D(t, v, v', w) = \frac{G(v) G(v')}{4\pi v v'} Z(t) A(w) + \frac{(1 - G(v) G(v'))}{4\pi v v'} ,$$

• the geometric term: $G(v)$

$$G(v) = \frac{v}{\sqrt{r - rv + v}},$$

• for anisotropy $Z(t)$ gives the dependence on the zenithal angle:

$$Z(t) = \frac{r}{\sqrt{1 + rt^2 - t^2}},$$

• for anisotropy $A(w)$ gives the dependence on the azimuthal angle:

$$A(w) = \sqrt{\frac{p}{p^2 - p^2 w^2 + w^2}},$$

• the reflectance calculation for the single layer is:

$$R(t, u, v, v', w) = S(u) D(t, v, v', w),$$

and for the double layer we have:

$$R(t, u, v, v', w) = S(u) D(t, v, v', w) + [1 - S'(u)] S'(u) D'(t, v, v', w).$$

Finally we give the corresponding values of the reflectance arguments $(t, u, v, v', w)$. They are all angles:

$$f_r(u, v) = R(\hat{n} \cdot \hat{h}, (u \cdot \hat{h}), (v \cdot \hat{n}), (u \cdot \hat{n}), \phi_h). \quad (3.14)$$

In GRF language we have the option of expressing an heterogeneous surface by using $Ly = true$ and following this, the values of the second layer. By default $Ly = false$ and so, the surface is homogeneous.
Chapter – 3. A Study of BRDF Models

Figure 3.14: The Schlick-BRDF instance with our GLSL BRDF editor.

\[
\text{SchlickBRDF} \ ::= \text{brdf schlick SC =value \ Sr =value}
\]
\[
\text{Sp =value \ Ly =boolean}
\]
\[
[ \text{DC =value \ Dr =value \ Dp =value } ]
\]

Table 3.13: GRF notation of SchlickBRDF

3.17 Hanrahan (1993)

Hanrahan & Krueger [HK93] describe an illumination model for the simulation of light scattering at the subcutaneous layers of the human skin. Using a subsurface reflectance model, in terms of one-dimensional linear transport theory, dermis and epidermis are represented with a reflective layer and a set of subsequent layers in which absorption occurs.

The total radiance come from the outer and reflective microfacet surface, and the interior volume. The latter accounts for absorption and occasionally, re-emission of light. Usually it is assumed to be diffuse, although in this case the authors modelled an anisotropic reflectance function with a clear predominant direction for the outgoing angle.

\[
L_r(\theta, \phi, \gamma) = L_{r,s}(\theta, \phi, \gamma) + L_{r,p}(\theta, \phi, \gamma).
\]

This illumination model also includes a bidirectional transmission distribution function between the layers. The magnitude of the transmitted radiance has two com-
Heterogeneous surfaces like the human skin is modelled with sublayers.

Both these phenomena are given, the BRDF and the BTDF:

\[ f_r = f_{r,s} + f_{r,v}, \]
\[ f_t = f_{t,i} + f_{t,v}. \]

When polished surfaces are assumed, then the classical Fresnel equations are used. They provide reflectance values which have dependence on the index of refraction \( \eta_i \) and \( \eta_t \), as well as the cosines of \( \theta_i \) and \( \theta_t \) angles. They also control the quantity of light that is transferred to the volumetric model.

Subsequent derivation of the formulation in terms of the one-dimensional linear transport theory are found in [HK93].
3.18 Oren-Nayar (1994)

The Oren & Nayar model [ON94, ON95] is an improvement on the classical Lambertian interpretation for a diffuse material. In reality even diffuse materials show small highlights when the viewer direction is close to the illumination direction. This is the case of some items of clothing and even sand.

The authors use the microfacet description of the surface, but in this case the little microfaces, or grooves are purely diffuse. In addition to masking and shadowing light blocking events, the interreflections between facets are also considered, but is limited to a pair of bounces.

The assumption used, relative to the area of a single groove \(d(a)\) and the area of the surface \(dA\), is:

\[ \lambda^2 << d(a) << dA, \]

which allows calculations of the fraction of the area occupied by the grooves orientated towards a given normal direction \(a\). It is noted as \(P(a)\):

\[ P(a) = N(a) \cos(a) d(a), \]

The orientation of the grooves is distributed using a Gaussian function with a standard deviation for the angles \(\sigma_m\). Next, given an incident direction, the total reflected radiance is the integrate of grooves’ reflectance values — the projected reflected radiance — and the radiance due to the bounces between them: \(L_p(u, v, a) = L^1_p(u, v, a) + L^2_p(u, v, a)\). The BRDF results as:

\[ f_r P(a) = L_p(u, v, a) P(a) \]

The projected reflected radiance of a single event on the surface is:

\[ L^1_p(u, v, a) = \frac{\rho}{\pi} E_0 \frac{(u \cdot a)}{(a \cdot n)} \frac{(v \cdot a)}{(a \cdot n)} G(u, v, a), \]

where \(E_0\) is the irradiance evaluated at the normal direction and \(G\) is the geometrical attenuation factor, expresses as:

\[ G(u, v, a) = \min\left[1, \max\left(0, \frac{2 (u \cdot n)}{(u \cdot a)}, \frac{2 (v \cdot n)}{(v \cdot a)}\right)\right]. \]

The projected reflected radiance of two inner reflection events is:

\[ L^2_p(u, v, a) = \frac{\rho}{\pi} \int_{t=\frac{m}{2\pi}}^{1} \int_{r=\frac{m}{2\pi}}^{1} K(t, r, a) d(r) d(t), \]
Let \( w \) be the width of the V-groove, \( m_v \) the width of the visible facet and \( m_s \) the width of the adjacent facet used for masking and shadowing effects. It is supposed that the length \( l \) of the groove is much larger than its width, i.e. \( l \gg w \) and it can be viewed as a one-dimensional shape. The term \( K \), or symmetric transactional kernel, is a function of the two closest intersection points \( x \) and \( y \) between two grooves.

\[
K(x, y, a) = \frac{\pi \sin^2(2\theta_a)}{2} \cdot \frac{x y}{(x^2 + 2xy \cos(2\theta_a) + y^2)^{3/2}}.
\]

![Figure 3.16: The Oren-Nayar-BRDF instance with our GLSL BRDF editor.](image)

Two parameters are used in the Oren-Nayar BRDF, as seen with the GRF notation. The roughness is adjustable with \( \sigma \) or equivalently with \( s \in [0,1] \). The reflectivity value \( \rho \) is \( r \in [0,1] \).

\[
\text{Oren-Nayar-BRDF} ::= \ brdf \ oren \ s = \text{value} \ r = \text{value}
\]

Table 3.14: GRF notation of Oren-Nayar-BRDF

3.19 Neumann-Neumann (1996)

The Neumann & Neumann BRDF model [NN96] introduces a plausible function which is easy to integrate in a Monte-Carlo based rendering algorithm. They use a base BRDF which accounts for ideal reflections. For the description of metal, plastics,
cereal, or retroreflective material, and even the anisotropic reflectance, the authors build these BRDFs over a base model.

They introduce two key ideas. The first, is that the base BRDF uses the unit disc domain $\mathbb{D}^2$ and thus the $\sigma_p$ measure, so a cosine term is implicit. In the second one, the projection of the reflection vector is the origin of a disc $C$ whose radius is controlled by a parameter $r$. Figure 3.17 shows the geometric previously described. Since many vectors can be projected onto the disc, a distance function is defined. Note that subscript $p$ is used to denote projection.

$$m(u,v) = |v_p - r_{u,p}| = |h - (n \cdot h)n| \quad using \quad h = u + v.$$  

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure3.17}
\caption{The Neumann BRDF is based on a disc of variable radius defined over $\mathbb{D}^2$ domain.}
\end{figure}

The BRDF function is a constant value defined as:

$$f_r(u,v) = \begin{cases} 
\frac{\pi}{r} & \text{if } v_p \in C(r_{u,p}, r), \\
0 & \text{in other case.}
\end{cases}$$

Here, the surface aspect varies from perfect specular $r = 0$, to perfect diffuse $r = 2$ and allows a simple calculation of the directional-hemispherical reflectance:

$$\rho(u) = r \cdot Area(C(r_{u,p}, r)).$$

With this base model it is easy to build other BRDFs. For example a retroreflective one would simply change the $h$ of the distance function by $g = u - v$. As a consequence, light will reflect predominantly in the incident direction. Another example is a BRDF for anisotropic surfaces which change the circle $C$ by an ellipse $E$ whose axes are $a$ and $b$, and the distance function by the normal elliptic curve.

$$f_{r,aniso}(u,v) = \begin{cases} 
b & \text{if } v_p \in E(r_{u,p}, a, b), \\
0 & \text{in other case.}
\end{cases}$$

### 3.20 Lafortune (1997)

One of the most multifunctional BRDF models is the one introduced by Lafortune [LFTG97]. It was used to fit measurements from real surfaces and can be

considered as a generalization of the Phong BRDF [Pho75] that is reciprocal and obeys the energy conservation law.

Let \( R(\mathbf{u}, \mathbf{n}) \) be the reflection operator. The operator simply reflects a vector around the normal direction: \( \mathbf{r} = \mathbf{u} - 2(\mathbf{u} \cdot \mathbf{n}) \mathbf{n} \). The modified Phong expression is:

\[
f_r(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot R(\mathbf{v}, \mathbf{n}))^n = (\mathbf{v} \cdot R(\mathbf{u}, \mathbf{n}))^n. \tag{3.15}
\]

In a canonical reference system \( R(\mathbf{u}, \mathbf{n}) \) equals \((-x_u, -y_u, z_u)\), which is interpreted as a scale factor of \((-1,-1,1)\) applied to the original vector. When we apply the generalization, a matrix \( M \) is used:

\[
f_r \propto (\mathbf{v} \cdot M \mathbf{u})^n
\]

If we express the diagonal of a matrix \( M \) as a vector \( \mathbf{o} \)—note that this is a parameter of the model—a retroreflection effect is obtained with \( \mathbf{o} \) set to \((1,1,1)\).

By using more than one lobe, this BRDF is able to represent the data from acquired materials. Let \( N \) be the number of lobes, which is used in the final expression:

\[
f_r(\mathbf{u}, \mathbf{v}) = \frac{\rho_d}{\pi} + \sum_{i=1}^{N} (\mathbf{v} \cdot (x_{u_i}x_{o_i}, y_{u_i}y_{o_i}, z_{u_i}z_{o_i}))^n_i. \tag{3.16}
\]

In GRF notation we define each lobe separately. Next, we should use a combination of BRDF for any number of lobes. We write \( k_p = \rho_d \) and \((C_x, C_y, C_z) = \mathbf{o}\).

<table>
<thead>
<tr>
<th><strong>LAFORTUNEBRDF</strong></th>
<th>:=</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{brdf}</td>
<td>lafortune n =value kp =value</td>
</tr>
<tr>
<td></td>
<td>Cx =value Cy =value Cz =value</td>
</tr>
</tbody>
</table>

Table 3.15: GRF notation of LAFORTUNEBRDF

3.21 Shirley (1997)

Shirley \textit{et al.} [SHSL97] introduced a model for the representation of matte surfaces. It is called a \textit{coupled model} because it properly combines the diffuse and specular reflections. It is plausible, physically based and suited for Monte-Carlo rendering algorithms.

The model for specular reflection assumes a polished planar surface of a dielectrics in touch with the air. Some known indexes of reflection are listed on Table 2.1. The matte component is considered to be almost constant and isotropic, so both combined

R. Montes
as follows:

$$f_r(u, v) = F(\theta_u) + k R_m f(\theta_u) f(\theta_v),$$

where $k$ is a constant for normalization and $R_m$ the reflectance of the matte component given as a parameter of the model. Finally $f(\theta)$ is approximated as follows:

$$f(\theta) \propto (1 - (1 - \cos(\theta))^5).$$

Applying some restrictions the author is able to solve $k$:

$$k = \frac{21}{20\pi (1 - R_0)},$$

where $R_0$ is another parameter that varies between 0.03 and 0.06. He arrives at:

$$f_r(u, v) = [R_0 + (1 - \cos(\theta_u))^5 (1 - R_0)] f_{r,s}(u, v) + k R_m [1 - (1 - \cos(\theta_u))^5][1 - (1 - \cos(\theta_v))^5].$$

The definition for this BRDF is quite simple if we use a constant for $R_0$. 

Figure 3.19: The Coupled-BRDF instance with our GLSL BRDF editor.

\[ \text{ShirleyBRDF} ::= \]
\[ \text{brdf coupled } R_m = \text{value } [R_0 = \text{value }] \]

Table 3.16: GRF notation of ShirleyBRDF
3.22 Ashikhmin-Shirley (2000)

Ashikhmin & Shirley [AS00a] developed an anisotropic microfacets model that use simple and intuitive parameters. They also provide a method of sampling based on their BRDF function.

Two parameters define the anisotropy by controlling the halfway vector distribution of the microfacets. These parameters are related to the axes of an ellipse: $e_x$ and $e_y$ that orientate $h$ along the X and Y axes respectively as Figure 3.20 shows. An exponent value $e$ is calculated with respect to the ellipse radius and the $\phi$ angle of orientation.

![Figure 3.20](image)

The exponent values $e_x$ and $e_y$ are considered to be the length of the axis of an ellipse which gives the desired orientation.

The distribution function is as follows:

$$D(h) = \sqrt{(e_x + 1)(e_y + 1)(h \cdot n)^{e_x \cos^2(\phi_h) + e_y \sin^2(\phi_h)}}.$$  

Applying Trigonometry theory, a simplification of the exponent of $D(h)$ is achieved, and thus:

$$\cos^2(\phi) = \frac{x_h^2}{1 - z_h^2},$$

$$\sin^2(\phi) = \frac{y_h^2}{1 - z_h^2},$$

where $h = (x_h, y_h, z_h)$.  

University of Granada
In an other publication [AS00b] the authors complement the BRDF model with one for diffuse surfaces covered with a polished or painted layer. For example a wooden table with lacquer paint. At normal incidence it is diffuse but at the horizontal view it is almost specular. The Fresnel equations describe this very phenomenon, and it is modelled with a superposition of two layers: one specular and one diffuse (see Figure 3.21).

![Figure 3.21: The surface is represented by imposed layers. The outer is specular and the inner is diffuse.](image)

The BRDF is expresses as a weighted sum of the specular term and the diffuse one:

$$f_r = f_{r,s} + f_{r,d}.$$  

For the specular reflection, the distribution over $h$ is used as well as Schlick’s approximation of the Fresnel term:

$$f_{r,s}(u, v) = \frac{D(h) F(\theta_v)}{8 \pi (h \cdot u) \max(\cos(u), \cos(v))}. \quad (3.17)$$

The diffuse term guarantees the reciprocity and energy conservation properties, so the BRDF is plausible.

$$f_{r,d}(u, v) = \frac{28 \rho_d}{23 \pi} (1 - \rho_s) \left(1 - \left(1 - \frac{\cos(u)}{2}\right)^5\right) \left(1 - \left(1 - \frac{\cos(v)}{2}\right)^5\right). \quad (3.18)$$

With our GRF notation $\rho_s$ is written as $Rs$ and $\rho_d$ is $Rd$. The complete definition is:

```
ASHIKHMINBRDF ::= bref ashikhmin ex =value ey =value
Rs =value Rd =value
```

**Table 3.17:** GRF notation of ASHIKHMINBRDF
Chapter – 3. A Study of BRDF Models

Figure 3.22: The Ashikhmin-BRDF instance with our GLSL BRDF editor.


The Granier & Hiedrich BRDF [GH03] is a reflectance model specifically conceived for the representation of old metals, or dirty surfaces covered with an oily transparent exterior surface. Like their predecessors, Ward, Schlick and Lafontune, Granier & Hiedrich want a BRDF model which is simple to evaluate. Unlike them, they use the wavelength of the incident light to simulate two phenomena.

1. A change of phase between the beam of light that is reflected and the beam of the incident light that is transmitted to other layer occurs. The thinness of the cover surface influences this event. Its expression is:

\[ P(u, v, \lambda) = \pi \frac{\eta_2(\lambda) d}{\lambda} (-n \cdot h). \]

2. The index of reflection is wavelength dependent, \( \eta(\lambda) \) and produce the dispersion of light in different colours (much like a prism). This is obtained with a model of non-parallel layers, with a different alignment of the normal of the outer surface and the normal of the inner surface (see Figure 3.21).

Though the model is \( \lambda \) dependent, an initial approximation is carried out using the RGB colour model with values \( R = 645 \) nm, \( G = 525 \) nm, and \( B = 445 \) nm. Another simplification is associated with the outer layer using \( \eta_1 \) corresponding to the University of Granada
vacuum (see Table 2.1), and also $\eta_1 \leq \eta_2$. This assumption implies that transmission always happens. The final result is a BRDF that combines the reflection term $R$ and the transmission $T$ term in this way (using the Fresnel equations for dielectrics of section 2.3.4):

$$f_r(u, v, \lambda) = \frac{R(u, v, \lambda) + T(u, v, \lambda)}{\sqrt{R(u, v, \lambda) T(u, v, \lambda)}} + 2 \cos(2P(u, v, \lambda)) \sqrt{R(u, v, \lambda) T(u, v, \lambda)}.$$ (3.19)

Below we focus on the expansion of the reflectance term $R$ and for the transmission term, the reader is referred to the original paper [GH03]. $R$ is a cosine term weighted by the reflectivity —the Fresnel term using the Schlick approximation— due to direct illumination. A simplified version uses the diffuse Lambertian formulation.

$$R(u, v, \lambda) = F(-u, \lambda) (v \cdot n)^n.$$

Also in [GH03] a hardware acceleration technique that uses textures, is suggested in order to obtain low cost results.
CHAPTER 4

Sampling of the BRDF

4.1 Generic Sampling of the BRDF: Motivation

The main intention of the Monte Carlo based realistic rendering field relies on the integral calculation of the transport equation. It estimates the value of the radiance function that is carried by random paths of illumination. This stochastic process is supported by the Monte-Carlo method, as can be seeing in the next section. For this reason, random vector directions should be generated in order to simulate the outgoing flow energy correctly. This is accomplished by the BRDF sampling. Chapter 3 discusses the state of the art with reference to BRDF models. In this section talking about the BRDF sampling world, we give a summary of the aforementioned models (Table 4.1), as well as a list of their desirable properties initially given by Shirley [Shi96]. They are:

1. Physically plausible: a function that obeys non-negativity, reciprocity and energy conservation law. A BRDF with this property can be used safely in a rendering system.

2. Adjustable by parameter: the model vary with its parameters. These should be few, intuitive and easy to set.

3. Simple to compute, and thus quick to evaluate.

4. Capable of representing as many different materials as possible.

5. Capable of representing a new type of material without the need for measured parameters.

6. Capable of fitting acquired data analytically, and thus representing it with few parameters.
Chapter – 4. Sampling of the BRDF

### Table 4.1: Brief summary of the BRDF properties. Legend:

- (★) if the BRDF possesses this property; (▼) if the BRDF does not; (···) undetermined value.

<table>
<thead>
<tr>
<th>BRDF models</th>
<th>Physical</th>
<th>Plausible</th>
<th>Fresnel</th>
<th>Anisotropically Sampling</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minnaert</td>
<td>▼</td>
<td>···</td>
<td>▼</td>
<td>▼</td>
<td>Moon surface</td>
</tr>
<tr>
<td>Torrance-Sparrow</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>rough surfaces</td>
</tr>
<tr>
<td>Beard-Maxwell</td>
<td>▼</td>
<td>▼</td>
<td>▼</td>
<td>▼</td>
<td>painted surfaces</td>
</tr>
<tr>
<td>Phong-Blinn</td>
<td>▼</td>
<td>▼</td>
<td>▼</td>
<td>★</td>
<td>matte surfaces</td>
</tr>
<tr>
<td>Whitted</td>
<td>★</td>
<td>···</td>
<td>★</td>
<td>★</td>
<td>glossy</td>
</tr>
<tr>
<td>Cook-Torrance</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>metal, plastics</td>
</tr>
<tr>
<td>Kajiy</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>metal, plastics</td>
</tr>
<tr>
<td>Poulin-Fournier</td>
<td>★</td>
<td>···</td>
<td>★</td>
<td>★</td>
<td>diffuse</td>
</tr>
<tr>
<td>Strauss</td>
<td>▼</td>
<td>···</td>
<td>★</td>
<td>★</td>
<td>metal, plastics</td>
</tr>
<tr>
<td>He et al.</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>metal, plastics</td>
</tr>
<tr>
<td>Ward</td>
<td>▼</td>
<td>▼</td>
<td>▼</td>
<td>★</td>
<td>metals</td>
</tr>
<tr>
<td>Westin</td>
<td>★</td>
<td>···</td>
<td>★</td>
<td>★</td>
<td>metals</td>
</tr>
<tr>
<td>Schlick</td>
<td>▼</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>heterogeneous</td>
</tr>
<tr>
<td>Lewis</td>
<td>▼</td>
<td>★</td>
<td>▼</td>
<td>★</td>
<td>mats</td>
</tr>
<tr>
<td>Oren-Nayar</td>
<td>★</td>
<td>···</td>
<td>★</td>
<td>★</td>
<td>diffuse with gloss</td>
</tr>
<tr>
<td>Neumann</td>
<td>▼</td>
<td>★</td>
<td>▼</td>
<td>★</td>
<td>metal, plastics</td>
</tr>
<tr>
<td>Lafortune</td>
<td>▼</td>
<td>★</td>
<td>▼</td>
<td>★</td>
<td>matte surfaces</td>
</tr>
<tr>
<td>Coupled</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>matte surfaces</td>
</tr>
<tr>
<td>Ashikhmin</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>polished surfaces</td>
</tr>
<tr>
<td>Granier-Hiedrich</td>
<td>★</td>
<td>···</td>
<td>★</td>
<td>★</td>
<td>old dirty metals</td>
</tr>
</tbody>
</table>

7. The diffuse component should not be the ideal Lambertian.

8. The specular component should be described in terms of the Fresnel equations.

9. The BRDF should be appropriate for Monte-Carlo algorithms by providing a direct sampling.

As the reader can deduce, there is no single BRDF model that conforms with each of these properties. More importantly —and from the user’s point of view— if you have a BRDF that conforms with your requirements, perhaps it might be difficult to set it up with whatever Global Illumination rendering system you are using.
4.2 The Reflectance Equation and Monte-Carlo Estimation

One of the main objectives of Global Illumination relies on the evaluation of the reflectance equation (Eq. 2.10) which is in fact an alternative integral form of the BRDF definition (usually given as a differential equation). Here $L_i$ stands for incoming radiance and $L_r$ for reflected radiance. The reflectance equation is usually solved in Global Illumination by using Monte Carlo (MC) integration, because it is often impossible to obtain analytic expressions for $L_r$ or $L_i$.

When numerical integration of an arbitrary integrable (w.r.t. a measure $\mu$) function $g \in S \rightarrow \mathbb{R}$ is undertaken by using MC techniques, random samples in $S$ must be generated. Those samples are values from a random variable (or usually a random vector) generated by a probability measure $P$, which obviously obeys $P(S) = 1$ and is absolutely continuous with respect to $\mu$. The function $p \overset{\text{def}}{=} dP/d\mu$ is usually called the probability density function (PDF) of those samples. Starting with $n$ random samples (namely $\{x_0, \ldots, x_{n-1}\}$) we can build a new random variable $X_n$ whose mean is the integral we want to compute. Formally:

$$X_n \overset{\text{def}}{=} \frac{1}{n} \sum_{i=0}^{n-1} \frac{g(x_i)}{p(x_i)} \implies E(X_n) = \int_{z \in S} g(z) \, d\mu(z) \quad (4.1)$$

For equation 4.1 to be consistent, we must also ensure that $p(z) > 0$ if $g(z) > 0$. The result above allows us to obtain values from a random variable $X_n$ and use them as estimators for the integral $I = \int_S g \, d\mu$. This is done by generating sample sets whose PDF is $p$, and evaluating $X_n$ with them. The variance of $X_n$ determines the efficiency of the method, where we consider efficiency as the relationship between computational effort (usually proportional to the number of samples $n$) and variance.

It can be easily shown that, the closer is $p$ to $g/I$ the less variance we obtain (down to zero variance for $p = g/I$). Thus designing efficient MC sampling methods usually means designing good PDFs by using all available information relating to $g$.

The properties previously described allow us to compute an estimator of $L_r(u)$, as defined in equation 2.10, for a given $u \in \Omega$. In this application of MC sampling, the domain of integration is $\Omega$ and the measure is $\sigma$. We must use a set of $n$ samples $(s_1, \ldots, s_n)$, which are identically distributed random vectors defined in $\Omega$, with probability measure $P_u$ (note that we use $u$ as a subindex because the probability measure usually depends on $u$). With this set of samples, the estimator of the outgoing radiance can be obtained as:
Chapter – 4. Sampling of the BRDF

\[ L_r(u) \approx \frac{1}{n} \sum_{k=1}^{n} \frac{f_r(u, s_k) \cdot (s_k \cdot n)}{q_u(s_k)} L_i(s_k) \]  
(4.2)

where \( q_u = \frac{dP_u}{d\sigma} \) is the PDF associated with \( P_u \). Although the formulae above is the usual way of expressing MC sampling, an alternative expression can be given by using the same approach for equation 2.11 instead of 2.10. In this case, the set of samples \([(x_1, y_1), \ldots, (x_n, y_n)]\) contain random vectors in \( D \) instead of in \( \Omega \), and the estimator becomes:

\[ L_r(u) \approx \frac{1}{n} \sum_{k=1}^{n} \frac{f_r(u, s_k)}{p_u(x_k, y_k)} L_i(s_k) \]  
(4.3)

where \( s_k = h^{-1}(x_k, y_k) \). Here, the PDF \( p_u = \frac{dP_u}{d\sigma_p} = \frac{dP_u}{dA} \) is defined w.r.t. area measure \( A \), and its domain is \( D \). This alternative expression will be used in our algorithm.

Employing a random sequence of uniformly distributed samples we can obtain an estimator with high variance or error. This is due to the shape of the BRDF, as it often presents a peak around the specular direction, and this may cause the PDF \( p \) to deviate from the integrand.

Global Illumination literature describes importance sampling (IS) as a tool to reduce variance. It consists of choosing a PDF intelligently. This means that the PDF mimics the BRDF function and also the incoming radiance or a combination of both [BGH04, ?]. As has been stated, the PDF and the integrand should be similarly shaped. In standard MC image synthesis algorithms, it is often necessary to generate sample sets for integrals such as equation 2.10 without any knowledge of irradiance or other terms of the integrand, but with a known BRDF. In these circumstances, the best option is to use a PDF which is as proportional as possible to the BRDF times the cosine term.

Moreover, as sample direction generation is carried out very often, this generation process must be done in a short time. From equation 4.2 we conclude that the PDF must be evaluated, and thus we should be able to do this as fast as possible.

4.3 Direct Sampling

As has been described, in Monte Carlo algorithms for Global Illumination the generation of a random sample direction is a frequent task and thus it should be done quickly. When tackling ideal cases of reflection we can provide an analytical expression for the BRDF which is very easy to evaluate, and for which a proportional PDF exists (which
4.3. Direct Sampling

is quick to use for sampling directions). However, in order to render realistic images, we have to use realistic material models, although few of them provide a method for sampling with the desired PDF properties (that is: proportional to the BRDF and fast). Some material models are impractical in rendering systems for this reason. In this section we review several BRDF models and the corresponding PDF used to sample them.

4.3.1 Sampling Ideal Cases

A perfect diffuse BRDF reflects energy equally distributed in $\Omega$. This simple BRDF ($f_r = k_d/\pi$, where $0 \leq k_d < 1$) has an associated PDF $p_u(v) = \cos(v)/\pi$ which is exactly proportional to the BRDF times the cosine term and also independent of $u$. Generating a random vector $v$ distributed according to this PDF is simple, by using:

$$(\theta_v, \phi_v) = \left( \cos(\sqrt{\xi_1}), 2\pi \xi_2 \right)$$

where $(\theta_v, \phi_v)$ are the spherical coordinates of $v$ ($\theta_v$ is the zenith angle and $\phi_v$ is the azimuth angle), and $\xi_1$ and $\xi_2$ are two independent uniformly distributed random variables with values in $[0, 1)$. From this point we will use $\xi$ notation with this meaning.

In the case of perfect specular reflection, the BRDF can be modelled by using a Dirac delta function. This function causes the integral in equation 2.10 to be equal to a single value of $L_i$, because the reflected radiance at $u$ is caused only by the irradiance coming from $u_r$. The latter can be defined as $u_r = 2(u \cdot n)n - u$ (that is, $u$ rotated $\pi$ radians around $n$).

Therefore the PDF is a Dirac-delta function $p_u(v) = \delta_u(v)$, and thus a sample distributed according to this PDF is $u_r$ with probability 1, and any other values have probability 0. As a consequence, generating random directions for a given $u$ consists simply of producing $u_r$ for a given $u$.

### Table 4.2: GRF notation of UniformPDF

<table>
<thead>
<tr>
<th>UniformPDF</th>
<th>pdf</th>
<th>diffuse</th>
</tr>
</thead>
</table>

### Table 4.3: GRF notation of SpecularPDF

<table>
<thead>
<tr>
<th>SpecularPDF</th>
<th>pdf</th>
<th>specular</th>
</tr>
</thead>
</table>

R. Montes
4.3.2 Lobe Distribution Sampling

A well known class of BRDF is that based on cosine-lobes. These BRDF models have an associated algorithm for sampling. Within this category are Phong [Pho75], Blinn [Bli77] and their respective normalized versions delivered by Lewis [Lew93], Lafortune [LW94] and Ward [War92]. The term lobe here refers to the shape of a classic Phong-like BRDF as plotted by using spherical or polar coordinates. The single-lobe BRDF is defined as:

\[ f_r(u, v) = C(n) (v \cdot u_r)^n \]

where \( n \geq 0 \) is a parameter, and \( C(n) \) is a normalization factor which usually depends on \( n \). This ensures these BRDFs have an energy conservation property (see section 2.3.1.1).

The law of reflectivity of these BRDFs can be expressed as:

\[ \rho_n(u) = C(n)N_2(u_r, n) \]

where for any vector \( a \):

\[ N_2(a, n) \overset{\text{def}}{=} \int_{\Omega} (v \cdot a)^n (v \cdot n) d\sigma(v) \]

which is called a double axis moment. Arvo [Arv95] provided an analytical expression for \( N_2 \). It is easy to show that \( N_2(a, n) \leq N_2(n, n) \). As \( N_2(n, n) = 2\pi/(n + 2) \) (thus it can be easily evaluated). A usual option for \( C(n) \) is \( 1/N_2(n, n) \) ensuring energy conservation and yielding a simple BRDF model which can be evaluated very fast:

\[ f_r(u, v) = \frac{n + 2}{2\pi} (v \cdot u_r)^n \]

For this BRDF, a related normalized PDF can be defined as:

\[ p_a(v) = \frac{1}{N_1(u_r, n)} (v \cdot u_r)^n \]

where \( N_1 \) ensures normalization and is defined as:

\[ N_1(a, n) \overset{\text{def}}{=} \int_{\Omega} (v \cdot a)^n d\sigma(v) \]

This is called a single axis moment and, like \( N_2 \), there are analytical expressions for it. In order to obtain samples distributed according to this PDF, we first obtain a random vector \( v \) whose spherical coordinates are:

\[ (\theta, \phi) = \left( \arccos \left( \frac{\xi_1}{n+1} \right), 2\pi \xi_2 \right) \]
\[ r_v = (\sin(\alpha) \cos(\phi), \sin(\alpha) \sin(\phi), \cos(\alpha)) \]

where \( \xi_1 \) and \( \xi_2 \) are two independent uniformly distributed random variables with values in \([0,1)\). After this, we rotate \( r_v \) by using \( R \), where \( R \) is the rotation which converts \( n \) into \( u_r \), that is \( u_r = Rn \). The resulting sample is then \( s = Rr_v \). Note that this yields unit vectors outside \( \Omega \) (with a probability smaller than 1/2 of the \( Z \) component being negative), and in these cases we must reject this sample and try to produce a new one until a valid one is obtained.

The exponent \( n \) is a parameter of this sampling method, as can be seen with the GRF notation:

\[
\text{HemispherePDF} ::= \text{pdf hemisphere exp =value}
\]

**Table 4.4: GRF notation of HemispherePDF**

A frequently used variant of this PDF avoids evaluation of \( N_1 \) by using samples on the whole sphere \( S^2 \), instead of only the hemisphere \( \Omega \). This PDF is defined in \( S^2 \). However, when a sample is produced with negative a \( Z \) coordinate (that is, in \( S^2 - \Omega \), under the surface), the contribution of that sample to the integral is taken as zero. The algorithm is faster than the evaluation of \( N_1 \), and still unbiased, but has a higher variance when \( u \) approaches grazing angles. We avoid evaluation of \( N_1 \) because we take into account the part of the lobe under the surface, and as a consequence the value of \( N_1(n, n) \) is independent of \( u \) and equal to \( N_1(n, n) = \frac{2\pi}{(n+1)} \). A random vector \( s \sim p_u \), is obtained firstly from \( \Omega \) as follows:

\[
z_s = \frac{n+1}{\sqrt{\xi_1}}, \\
s = (\sqrt{1 - z_s}, 0, z_s),
\]

Next, this is rotated around \( n \) with a random angle \( \alpha = 2\pi \xi_2 \), in \([0,2\pi)\). The final direction \( r_v \) is obtained by inversely rotating \( s \). This method is implemented under the name of SpherePDF:

\[
\text{SpherePDF} ::= \text{pdf sphere exp =value}
\]

**Table 4.5: GRF notation of SpherePDF**

R. Montes
4.3.2.1 Sampling Based on the Half-Vector Angle

The reflectance models based on the half angle [Bli77, War92, KC01], generate a new direction around $h$ that should be inverted and moved to the original reference system. That is, given $u$ and the generated $h$, the final vector obtained with a sampling method based on the half-vector angle is $v = -u + 2(u \cdot h)h$. Some factors have to be taken into account to preserve the measure between both reference systems. In general, if we sample $s_i \in S$ randomly according to density $p_s$ and we want to transform it in a new sample for another space $t_i \in T$, in order to preserve measure, the PDF $p_t$ must be calculated as:

$$p_t = p_s \left| \frac{\partial s}{\partial t} \right|.$$  

This relationship comes from an area restriction, that is, a subregion $A_B \subseteq S$ which maps with subregion $B \subseteq T$ should have the same area.

$$\int_{A_B} p_s(s_1, s_2) ds_1 ds_2 = \int_{B} p_t(t_1, t_2) dt_1 dt_2.$$

The correct proportion between the measures from both domains results in:

$$p_u(v) = p_h(v) \left| \frac{d\sigma(h)}{d\sigma(u)} \right|.$$  \hspace{1cm} (4.4)

$$\frac{d\sigma(h)}{d\sigma(u)} = \frac{\sin(\theta_h) \, d\theta_h \, d\phi_h}{\sin(\theta_u) \, d\theta_u \, d\phi_u} = \frac{\sin(\theta_h) \, d\theta_h \, d\phi_h}{\sin(2\theta_h) \, 2d\theta_h \, d\phi_h}$$

$$= \frac{\sin(\theta_h)}{4 \cos(\theta_h) \sin(\theta_h)} = \frac{1}{4 \cos(\theta_h)}$$

$$= \frac{1}{4 (v \cdot h)} = \frac{1}{4 (u \cdot h)}$$

Reflectance models that use microfacet distributions to sample the specular component, calculate vector $h$ as $(\theta_h, \phi_h) = (\sqrt{\xi_1}, 2\pi \xi_2)$. Next, its reflected vector $v$ is returned with a probability of:

$$p_u(v) = \frac{n + 1}{2\pi} \frac{\cos^n(\theta_h)}{4 (v \cdot h)}$$
4.3. Ward-BRDF Direct Sampling

Not only Ward does present an isotropic and anisotropic model, but he also gives a method to sample them [War92]. He mistakenly omitted an arc angle. The corrected version appears in *Global Illumination Total Compendium* [Dut].

Ward-BRDF [War92] uses a Gaussian lobe controlled with parameters $\alpha_x$ and $\alpha_y$, which also express the degree of anisotropy. He presents a sampling method using the aforementioned parameters. The PDF has an exponential expression:

$$p_v(v) = e^{-\tan^2(\theta_h) \left( \frac{\sin^2(\phi_h)}{\alpha_x^2} + \frac{\sin^2(\phi_h)}{\alpha_y^2} \right)}$$

Ward uses $k_s$ as a constant parameter of the model to weight directions. This supposition is correct most of the time, however it does not work well for grazing angles (near 90 degrees). In a technical report [Wal05] B.Wald discusses some implementation details of Ward’s BRDF by giving them more suitable weighting in order to use them for sampling in a Monte Carlo system. The weighting function is:

$$w(v) = \frac{f_r(u,v) \cos(v)}{p_v(v)} = k_s (h \cdot u)(h \cdot n)^3 \sqrt{(v \cdot n) \left( \frac{u \cdot h}{v \cdot h} \right)}$$

Sample $v$ is obtained from the polar angles as shown:

$$\theta_h = \arctan \left( \sqrt{-\log(\xi_2) \frac{\cos^2(\phi_h)}{\cos^2(\phi_h) + \sin^2(\phi_h)}} \right)$$

$$\phi_h = \arctan \left( \frac{\alpha_x}{\alpha_y} \tan(2\pi \xi_1) \right)$$

For the isotropic Ward expression (Eq. 3.9), the random generated vector $h$ is obtained as:

$$\theta_h = \arctan \left( \sigma_m \sqrt{-\log(\xi_1)} \right),$$
$$\phi_h = 2\pi \xi_2.$$
Chapter – 4. Sampling of the BRDF

A $\mathcal{D}^2$ domain (see section 3.19). Samples $s_{xy}$ are then re-projected to the hemisphere by computing:

$$s = s_{xy} + \sqrt{1 - s_{xy}^2}$$

The resulting values follow a cosine distribution function similar to those used in diffuse objects.

4.3.5 Sampling Anisotropic BRDFs

The Ashikhmin-Shirley BRDF model [AS00a, AS00b] also has an associated method for sampling the BRDF. They start by efficiently computing a half-vector $h$ in the first quadrant of a sphere with the following density function:

$$q(h) = \frac{\sqrt{(e_x + 1)(e_y + 1)}}{2\pi} (n \cdot h) e_x \cos^2(\phi_h) + e_y \sin^2(\phi_h)$$

where $e_x$ and $e_y$ are parameters that control anisotropy, $p_u(v) \neq q(v)$, and $h$ is obtained by using these expressions:

$$\theta_h = \arccos \left( \frac{\xi_2}{\sqrt{(e_x + 1)(e_y + 1)}} \right)$$
$$\phi_h = \arctan \left( \frac{\frac{e_x + 1}{e_y + 1} \tan \left( \frac{\pi \xi_1}{2} \right)}{1} \right)$$

Following that, the outcome quadrant is randomly selected (see Figure 4.1) using a uniformly distributed $\xi_1$. The same value is used after a proper transformation—that depend on the quadrant—to preserve stratification. For example, if $\xi_1 \in [0.25, 0.5)$ then the value of $\phi$ is $1 - 4(0.5 - \xi_1)$ and it is also rotated by $\pi/2$.

Given an incoming vector $u$ and $h$ we get $v = -u + 2(u \cdot h)h$, with the corrected density in order to maintain measures (remember Equation 4.4):

$$p_u(v) = \frac{q(h)}{4 (u \cdot h)}$$

Parameters $e_x$ and $e_y$ are noted as $nu$ and $nv$ when the GRF notation is used.
4.4 General Sampling for Analytic and Acquired BRDFs

4.4.1 The Factored BRDF Method

There are published works dealing with effective importance sampling strategies for arbitrary BRDFs. In this section we focus on the BRDF factorization given by Lawrence [LRR04]. This method divides the BRDF function into the product of lower dimensional factors, stored in a tabular and compact form. Other authors have previously factorized the BRDF [KM99, MAA01], but in this work Lawrence presents a novel application of the factorization: optimal BRDF sampling for Monte Carlo algorithms. This method involves two distinct factorizations, for the purpose of subdividing the BRDF and a cosine term into factors, shown below as:

\[ f_r(u, v) \cos(v) \approx \sum_j F_j(u) \sum_k u_{jk}(\theta_w) v_{jk}(\phi_w). \]

where \( w \) is the sampled random vector. For diffuse BRDFs it is better to use \( w \) as the local representation of \( v \). Specular BRDFs will use \( w \) as \( h \) in order to align important properties such as the highlights.

The article demonstrated a sampling method for the Cook-Torrance BRDF model, Poulin-Fournier cylindrical model and the anisotropic expression of Ward. It is not only applied to those analytical functions, but also to acquired data from the

\[ \text{AnisotropicPDF} ::= \]
\[ \text{pdf anisotropic nu =value nv =value} \]

Table 4.6: GRF notation of AnisotropicPDF

4.4. General Sampling for Analytic and Acquired BRDFs
Chapter – 4. Sampling of the BRDF

public data base of Matusik [MPBM03]. The Merl BRDF DB ¹ is an approximate representation of isotropic surfaces, with resolution 90 x 90 x 180 x 180. The sampling method proposed by Matusik [MPBM03] requires even more storage than the BRDF itself (a single BRDF is 33 Mbytes).

The initial data matrix $Y$ contains $N_p \times N_u$ samples of the BRDF along the outgoing elevation angle and the outgoing azimuthal angle. The first factorization, after a reparametrization based on the half angle [Rus98], results in the product of 2D factors, stored as matrices:

$$
\begin{bmatrix}
Y
\end{bmatrix} = \begin{bmatrix}
G
\end{bmatrix} \begin{bmatrix}
F
\end{bmatrix}
\text{with } G : N_w \times J \text{ and } F : J \times N_u.
$$

The $G$ matrix multiplied by the $F$ matrix is an approximation of the $Y$ matrix. The second factorization of the view independent matrix $G$ leads to the product of two matrices, each of one dimension. The resulting $L$ factors, where $L = J \times K$, show an intuitive approximation of a specific lobe of the original BRDF. We intend to use these factors as a PDF, thus an upper limit should be calculated in order to normalize the factors.

$$
f_r(u, v) \cos(v) \approx \sum_{l=1}^{L} F'_l(u) u'_l(\theta_w) v'_l(\phi_w) = \sum_{l=1}^{L} T_l(u, \theta_w, \phi_w),
$$

Using the non-negative matrix factorization (NMF) method [?] ² we always obtain matrices with positives values. This facilitates the use of normalized factors as a cumulative distribution function (CDF). Its precomputation allows fast sampling by numerical inversion. This process begins by selecting one of these $l$ lobes that contributes the most energy for the current view $u$, according to the $F$ matrix. The CDF for this sampling is recomputed when the outgoing direction changes. Next the hemisphere is sampled according to the selected lobe $l$. We do this by sequential generation of the elevation and the azimuthal angle using the precomputed CDF for factors $u_l$ and $v_l$. The density function for the generated outgoing direction is:

$$
p_u(v) = q(w) \left\| \frac{d\sigma(w)}{d\sigma(v)} \right\| = \frac{1}{4(v \cdot h)} \sum_{l=1}^{L} T_l(u, \theta_w, \phi_w) \sum_{j} F_j(u).
$$

Each BRDF must be efficiently and independently factorized making their method difficult to use for scenes containing many materials. Some functions will require many factors in order to be properly approximated. Thus, the user has to select the value of seven parameters manually, as can be seen with our GRF notation:


University of Granada
4.4. General Sampling for Analytic and Acquired BRDFs

\[
\text{FactoredPDF} ::= \\
\text{pdf factored} \quad \text{uTh} = \text{value} \quad \text{uPh} = \text{value} \\
\quad \text{pTh} = \text{value} \quad \text{pTh} = \text{value} \\
\quad J = \text{value} \quad K = \text{value} \\
\quad \text{half} = \text{boolean}
\]

Table 4.7: GRF notation of FactoredPDF

Another drawback of this method relates to matrices $F$ and $G$: they are not reciprocal and continuity is not guaranteed. It should be noted that, for the factorized BRDF, the generation of new random directions is performed over the $S^2$ domain. This implies that if a sample is below the positive hemisphere it should be rejected, or used with zero probability. The percentage of rejected samples is something we have not studied in detail. Purely as an experiment, we consider the Ward-BRDF with low discretization and then request to 2500 samples. Local point data were exported to a FIT file. Using the information stored in the FIT file and a tool named BRDF 3D Plot (see section 6.2.4) we drew the sampled points. The results are shown in Figure 4.2 where the 66% of the samples were rejected. This sampling method is tested with the one we propose in the next chapter.

Figure 4.2: Plot in 2D and 3D of the factored Ward-BRDF. In this case 827 of 2500 samples are generated in the positive hemisphere of directions.
4.4.2 The Cascade CDF Method

An improvement of the previously presented method is the Cascade CDF method [LRR05] given also by Lawrence. This is an adaptive technique orientated to the sampling of non uniform functions. The authors apply it to environment maps (EM) and acquired BRDFs. This technique is based on sampling by inversion of the CDF. Instead of uniformly distributing the samples, it uses a second and equivalent distribution which is compact. For this to be solved, they start with a N-dimensional PDF and divide it into the product of a 1D marginal distribution $\tilde{p}$ and a set of 1D conditional distributions.

$$\tilde{p}(x) = \int_{-\infty}^{\infty} p(x, y) \, dy$$

$$p(y|x_i) = \frac{1}{x_i - x_{i-1}} \int_{x_{i-1}}^{x_i} \frac{p(x', y)}{\tilde{p}(x')} \, dx'$$

Compression is carried out using the Douglas-Peucker greedy algorithm [Ros97] which approximates a curve (in our case the CDF) employing an optimal number of segments. Applying the aforementioned to BRDF sampling the authors recommend starting from an initial dense set of data of $2048 \times 1024$ ($\theta_h, \phi_h$) uniformly-spaced samples—a resolution prohibitively expensive for the fully tabulated CDFs. In order to compute the corresponding adaptive numerical CDFs, on average, roughly 30 samples in $\theta_h$ and 10 samples in $\phi_h$ are needed. It required an average 20 minutes of processing time to compute the adaptive representation for each BRDF.

Each uniformly-sampled CDF had a resolution of $32 \times 16 \times 256 \times 32$ ($\theta_u \times \phi_u \times \theta_h \times \phi_h$) and occupied 33 MB. The Figure 4.4 shows the initial setup in 2D and 3D of the discretize structure, that is, the equivalent of sampling a diffuse BRDF with the uniform CDF. When it is applied to an acquired reflectance function, for example the specular Nickel, the distribution of the samples are completely different (see Figure 4.3).

Using different resolutions for the discretized data structure, would change the memory requirements as is shown in Table 4.9. Note that we change the $N$ parameter using $N = nTh + nPh$ value, considering a fixed discretization over the vector $u$. This means that with our GRF notation we only change the resolution over vector $h$. A boolean parameter indicates if compression should be performed.

As in [LRR05] the implementation is tested using a subset of the 100 isotropic materials of the Mitsubishi Electric Research Laboratories (MERL) BRDF data set. With these data, we achieve an average compression rate of 98% as it is presented in

\[3\] Public DB of acquired BRDFs online at: <http://www.merl.com/brdf/> [last visit November 2007]
4.4. General Sampling for Analytic and Acquired BRDFs

\[
\text{CASCADECDF} ::= \text{pdf} \text{ cascadeCDF } n\text{Th} = \text{value} \ n\text{Ph} = \text{value} \\
\hspace{2em} \text{compress} = \text{boolean}
\]

<table>
<thead>
<tr>
<th>nTh</th>
<th>nPh</th>
<th>Mem (KB)</th>
<th>Mem (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>64</td>
<td>4160</td>
<td>4.06</td>
</tr>
<tr>
<td>100</td>
<td>32</td>
<td>13200</td>
<td>12.89</td>
</tr>
<tr>
<td>64</td>
<td>100</td>
<td>25856</td>
<td>25.25</td>
</tr>
<tr>
<td>256</td>
<td>32</td>
<td>33792</td>
<td>33</td>
</tr>
</tbody>
</table>

Table 4.9: Memory requirements of Cascade CDF method when the CDF is uniformly sampled with varied resolution.

Table 4.10. In addition in the previous table, we give the amount of time that each CDF compression took, a process that is performed immediately before rendering. In the next chapter we test our adaptive sampling method against the Cascade CDF scheme for those acquired materials.

Figure 4.3: Plot in 2D of the acquired \textit{nickel.binary} BRDF. With the inversion of the stored CDFs, samples are generated according the data.
<table>
<thead>
<tr>
<th>Material</th>
<th>Compress (sec)</th>
<th>CDF (KB)</th>
<th>Compress (KB)</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>alum-bronze.binary</td>
<td>572.13</td>
<td>33792</td>
<td>998.15</td>
<td>97.03%</td>
</tr>
<tr>
<td>alumina-oxide.binary</td>
<td>577.4</td>
<td>33792</td>
<td>174.75</td>
<td>99.48%</td>
</tr>
<tr>
<td>aluminium.binary</td>
<td>568.31</td>
<td>33792</td>
<td>1150.5</td>
<td>96.58%</td>
</tr>
<tr>
<td>aventurnine.binary</td>
<td>562.68</td>
<td>33792</td>
<td>331.65</td>
<td>99.02%</td>
</tr>
<tr>
<td>beige-fabric.binary</td>
<td>539.53</td>
<td>33792</td>
<td>187.66</td>
<td>99.44%</td>
</tr>
<tr>
<td>black-fabric.binary</td>
<td>579.51</td>
<td>33792</td>
<td>263.13</td>
<td>99.22%</td>
</tr>
<tr>
<td>black-obsidian.binary</td>
<td>580.54</td>
<td>33792</td>
<td>802.2</td>
<td>97.53%</td>
</tr>
<tr>
<td>black-oxidized-steel.binary</td>
<td>575.38</td>
<td>33792</td>
<td>761.45</td>
<td>97.75%</td>
</tr>
<tr>
<td>black-phenolic.binary</td>
<td>577.38</td>
<td>33792</td>
<td>901.96</td>
<td>97.33%</td>
</tr>
<tr>
<td>black-soft-plastic.binary</td>
<td>562.24</td>
<td>33792</td>
<td>799.8</td>
<td>97.63%</td>
</tr>
<tr>
<td>blue-acrylic.binary</td>
<td>576.79</td>
<td>33792</td>
<td>382.53</td>
<td>98.87%</td>
</tr>
<tr>
<td>blue-fabric.binary</td>
<td>578.26</td>
<td>33792</td>
<td>351.58</td>
<td>98.96%</td>
</tr>
<tr>
<td>blue-metallic-paint2.binary</td>
<td>572.29</td>
<td>33792</td>
<td>1038.0</td>
<td>96.93%</td>
</tr>
<tr>
<td>blue-metallic-paint.binary</td>
<td>575.35</td>
<td>33792</td>
<td>1110.05</td>
<td>96.72%</td>
</tr>
<tr>
<td>nickel.binary</td>
<td>567.49</td>
<td>33792</td>
<td>1024.62</td>
<td>96.97%</td>
</tr>
<tr>
<td>red-plastic.binary</td>
<td>586.5</td>
<td>33792</td>
<td>275.83</td>
<td>99.18%</td>
</tr>
<tr>
<td>teflon.binary</td>
<td>561.78</td>
<td>33792</td>
<td>184.51</td>
<td>99.45%</td>
</tr>
<tr>
<td>violet-acrylic.binary</td>
<td>578.69</td>
<td>33792</td>
<td>826.69</td>
<td>97.53%</td>
</tr>
<tr>
<td>white-marble.binary</td>
<td>565.71</td>
<td>33792</td>
<td>247.79</td>
<td>99.27%</td>
</tr>
<tr>
<td>yellow-paint.binary</td>
<td>566.06</td>
<td>33792</td>
<td>145.41</td>
<td>99.58%</td>
</tr>
</tbody>
</table>

Table 4.10: Memory requirements of Cascade CDF method when the CDF is adaptively sampled reducing its storage needs.
Figure 4.4: Plot in 2D and 3D of the Cumulative Distribution Function when uniform sampling is used. In this case 5000 samples are generated in the positive hemisphere of directions.
CHAPTER 5

Efficient Sampling for Generic BRDFs

5.1 Our Algorithm

The main advantage of the method we presented here is that it can manage arbitrary BRDFs, and can be used to sample any isotropic BRDF model avoiding user guidance. It is designed for the Monte-Carlo solutions of the rendering equation and it can also be used with any 2D analytical function. The algorithm yields more samples in areas where the BRDF times the cosine term has a higher value (Figures 5.2 and 5.3), thus achieving importance sampling. It has been published in an international graphics conference, see [MULG08].

Our method is based on rejection sampling [Gen03]. This is a very simple and well known technique that yields a PDF proportional to any function \( g \in D \to \mathbb{R} \). Being able to evaluate \( g \) is the only requirement which does enable its maximum value \( m \) in \( D \) to be known. However, it runs a loop which in fact can be executed an unlimited number of times, thus it potentially yields large computing times even in the cases when \( g \) can be evaluated quickly.

The core of our approach is a hierarchical quadtree structure which can be used efficiently to obtain samples with a PDF exactly proportional to the product of the BRDF times the cosine term. The adaptive approach checks whether a region can be used safely for raw rejection sampling. This check consists of evaluating the average number \( n_t \) of trials with rejection sampling in that region. Controlled rejection time can be carried out provided we know the maximum (\( m \)) and average (\( e \)) values of \( g \) in the domain for the region. The average number of trials is \( n_t = m/e \), which can be high for \( e \ll m \). In our method, when \( n_t \) is above a threshold number \( n_{\text{max}} \) (which is an input
parameter), then the region is subdivided into four, and the criterion is applied to these four subregions. In the case that the criterion fails, the region is not subdivided. If we apply this process recursively starting from region $D^2$ (the unit radius disc centred at the origin), we obtain a quadtree which can be used efficiently to sample the BRDF. The remainder of this section describes implementation details of our method.

5.1.1 Building the Adaptive Structures

As the sampling process requires a PDF proportional to $f_r(u, \cdot)$ for arbitrary values of the view direction $u$ and for a finite collection of BRDFs in a scene, it is necessary to create a quadtree for each $(f_r, u)$ pair. In the case of $u$, a finite set of vectors $S = \{w_1, \ldots, w_n\}$ can be used. When an arbitrary $u$ is given, it is necessary to select the nearest $w_i$ to $u$ and use the corresponding structure. The error induced by using $w_i$ instead of $u$ can be reduced by using a large $n$ and uniformly distributing vectors $w_i$. Note that, since we assume the BRDF to be isotropic, it is enough for $S$ to include vectors in the plane XZ, thus a rotation must be applied to $u$ before finding the nearest $w_i$. The inverse rotation must be applied to resulting samples.

For a given quadtree, each node $i$ has an associated region $R_i \subseteq D$, which is a square area defined by:

\[
R_i = [u_i, u_i + size_i) \times [v_i, v_i + size_i),
\]

\[
R_i \cap R_j = \emptyset \quad \forall i \neq j,
\]

\[
\bigcup_{i=1}^{n} R_i = D^2.
\]

where $(u_i, v_i)$ is the lower left vertex of the region boundary and $size_i$ is the edge length. The region associated with the root node is $[0, 1)^2$, and thus $P_u(D^2) = 1$. Our method guarantees that the BRDF outside $D^2$ is always zero.

\[
f_r(u, v) > 0 \implies P_u(R) > 0 \quad \land \quad q_u(v) > 0.
\]

The algorithm creates the root node and it checks the subdivision criteria (check algorithm [1]). If a split is necessary, four new child nodes are created, each one with an associated region with an edge length half of parent’s one. Then, the process described is recursively applied to these new four nodes. The recursive algorithm ends when no split is necessary or a predefined maximal depth is reached.
5.1. Our Algorithm

Figure 5.1: The plane XY is the domain of our sampling method. Outside the disc the BRDF is always zero.

In order to check the subdivision criteria for node \( i \) these values must be computed:

\[
\begin{align*}
M_i &= \max \{ f_r(u, v_{xy}) \mid (x, y) \in R_i \} \\
I_i &= \int_{R_i} f_r(u, v_{xy}) \, dA(x, y) \\
V_i &= \int_{x,y \in R_i} f_r(u, v_{xy}) \, dx \, dy = \text{size}_i^2 \cdot M_i
\end{align*}
\]

where \( M_i \) is the maximum value of \( f_r \) in the \( i \)-th region, \( I_i \) is the integral of the BRDF in the region, \( V_i \) is the volume of the space where rejection sampling is done, and \( f_r \) is the upper-limit function:

\[
\forall R_i \subseteq D \land u \in \Omega \land v_{xy} \in R_i \left\{ \begin{array}{l}
\mathcal{F}_r(u, v_{xy}) = C, \\
f_r(u, v_{xy}) \leq \mathcal{F}_r(u, v_{xy}).
\end{array} \right.
\]

The probability for accepting a sample in the \( i \)-th region is \( \frac{I_i}{V_i} \). We also get the probability of selecting a region as it is proportional to the integral of the BRDF in that region:

\[
P_u(R_i) \overset{\text{def}}{=} \frac{1}{\int_{\Omega^2} f_r(u, v_{xy}) \, dx \, dy} \cdot I_i
\]

Both \( M_i \) and \( I_i \) can be computed by evaluating \( f_r \) on a very dense grid of points in \( R_i \) creating the quadtree, or alternatively a bottom-up approach can be used. This starts by obtaining these values at the maximum depth possible (with a high resolution grid) and then it stores them so the data can be used during tree construction. Therefore, the algorithm only requires the ability to evaluate the BRDF. In each case, it ensures that the sum of the \( I_i \) values for the four children of a parent node must be equal to that of the parent.
As has been previously stated, the subdivision criteria used must ensure that rejection sampling on the leaf nodes can be performed with an \textit{a priori} limited number of average trials $n_{\text{max}}$. This can easily be ensured by using this criterion:

$$ n_{\text{max}} \frac{I_i}{V_i} \geq 1 \quad (5.1) $$

When this inequality fails, the node must be split. In our implementation, we have used $n_{\text{max}} = 2$. The larger the $n_{\text{max}}$ the less memory that is needed (because the quadtree has less depth) and less time is used for quadtree traversal, but more time is required for rejection sampling on leaf nodes.

\begin{algorithm}
\begin{algorithmic}
  \State \textbf{DoPartition} ( ) \rightarrow
  \State \text{Let } dx, dy \text{ differential area in } R_i
  \State \text{Let } m = 0, v_1 = 0
  \For {x := u \text{ to } u + s} \Do 
    \For {y := v \text{ to } v + s} \Do 
      \State $v_1 = f_r(x, y) \times dx \times dy$
      \State $m = \max(f_r)$
    \End
  \End
  \State $v_2 = s \times s \times m$
  \State $p = v_1/v_2$
  \If {$p \leq 0.5$} \Then 
    \For {$j := 0 \text{ to } 4$} \Do 
      \State $R_i$.Child($j$) = \textbf{new} SampleQuadtreeNode
      \State $R_i$.Child($j$).DoPartition()
    \End 
  \End 
\end{algorithmic}
\end{algorithm}

\textbf{Algorithm 1: code DoPartition}

The C++ classes \textit{SampleQuadtree} and \textit{SampleQuadtreeNode} which are used for our implementation of the quadtree structure for the adaptive sampling using $D^2$ domain. The maximum depth ensures that all the regions which are used are not too small, and unable to contain a suitable number of samples.

\subsection{Adaptive Sampling of the BRDF}

Generating a random direction involves selecting a leaf node and then performing rejection sampling on that node. If the $i$-th node is a leaf node, then the probability of selecting it must be proportional to $I_i$ (more exactly it is $I_i/I_0$, if we assume the root node has index 0). A leaf node is selected following a path from the root to the leaf.
5.1. Our Algorithm

In each step, starting from the root, the precomputed integrals \( I_i \) of the descendant nodes are used randomly to choose one child to continue the path.

In order to do this, we store in each node \( i \) four values \( F_{i0}, \ldots, F_{i3} \), defined as:

\[
F_{ik} = \frac{\sum_{j=0}^{k} I_{C(i,j)}}{\sum_{j=0}^{3} I_{C(i,j)}},
\]

where \( C(i,j) \) is the index of \( i \)-th node \( j \)-th child node (note that \( F_{i3} = 1 \)). Leaf selection is then simply a loop, as it is shown in algorithm [2].

```
LeafNodeSelection() → integer
  i := 0 (index of root node)
  while i-th node is not a leaf do
    r := random value in [0, 1) (uniformly distr.)
    j := min. natural such that r < F_{ij}
    i := j
  end
  return i
```

Algorithm 2: code LeafNodeSelection

Following this, rejection sampling on the resulting \( i \)-th node is carried out. This consists of selecting a random vector \( (x, y, z) \subseteq \mathbb{R}^3 \) with uniform distribution in the prism \( R_i \times [0, M_i] \). Vector \( w_{xy} = (x, y, (x^2 + y^2)^{1/2}) \) is then obtained and the condition \( f_r(u, w_{xy}) < z \) is checked. If the comparison is true, \( w_{xy} \) is returned as the resulting sample, otherwise a new sample must be generated and checked. A sample is valid with probability \( I_i/V_i \), which is necessarily greater than \( 1/n_{\text{max}} \), because of inequality (5.1). This ensures that the average number of attempts is not greater than \( n_{\text{max}} \).

With our method, samples on disc will follow a distribution where more samples are placed in parts of the domain where the function has higher values, thus achieving importance sampling (as shown in Figures 5.2 and 5.3). In fact, the distribution obtained is exactly proportional to the BRDF.
Chapter – 5. Efficient Sampling for Generic BRDFs

Figure 5.2: Both images show a distribution of 1000 samples on the disc. The left one considers incident direction with $\theta_u = 70^\circ$ and a lobe-based BRDF with exponent $n = 50$. The image on the right uses $\theta_u = 22^\circ$ and $n = 200$.

Figure 5.3: Both images show a distribution of 2500 samples obtained with our disc method. The left one shows how the samples match the Ward-BRDF function (in red). The image on the right is the projection on the disc of those directions.
5.2 Using our PDF in a Monte Carlo Estimator

It is our intention to use our PDF with the following Monte Carlo estimator for the Equation (5.2):

\[ X_n(s_1, \ldots, s_n) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \frac{f_r(u, s_i)L_i(s_i)}{q(s_i)}. \] (5.2)

Once we have a set of random directions which are exactly proportional to a portion of the integrand of the reflectance equation, the procedure to approximate the radiance is Algorithm [3].

```
ESTIMATERADIANCE(u ∈ Ω, n ∈ Ω, N ∈ ℝ⁺, f_r, L, P_u) → ℝ

Let w_u the weighting function of P_u in D²
Let s [N] = GetNSamples(u, n, f_r, N)
Let s = 0
for i := 1 to N do begin
   s += (f_r(u, s[i]) L_i(s_i)) w_u(s_i)
end
return s/N
```

Algorithm 3: code ESTIMATERADIANCE

```
GETNSAMPLES(u ∈ Ω, n ∈ Ω, f_r, N⁺) → Sample[]

Let qt = new SampleQuadtree(u, n, f_r)
Let sample be Sample[N]
for i = 1 to N do begin
   SampleQuadtreeNode region = qt.GetRegion()
   Let s = region.DoSampling()
   Let q = region.GetPDF()
   sample[i] = new Sample(s, 1.0/q)
end
return sample
```

Algorithm 4: code GETNSAMPLES

We use the C++ class Sample to store a single direction s and its weight, which is:

\[ w_u(s) = \begin{cases} (s \cdot n)/p_u(s) & \text{if the PDF } p_u \text{ is in } Ω, \\ 1/q_u(s) & \text{if the PDF } q_u \text{ is in } D². \end{cases} \]

You should note that the new operator in the GETNSAMPLES method is totally prohibitive, since the method is called inside the for loop of ESTIMATERADIANCE. The solution is the precomputation in memory of a set of quadtrees for a number of
discrete outgoing directions. The number of quadtreese guarantees that we always find the closest structure for a given vector \( u \).

5.3 Optimal Adaptive Sampling

5.3.1 Quadtree Traversing for Many Samples

The quadtree structure is used to get a single reflected direction vector with a PDF proportional to the BRDF times the cosine term. The process (method [4]) can be repeated to obtain a finite set of samples, or we can use a modified traversal algorithm which yields a set of directions, the latter being a faster option. In the optimal traverse, each node is visited once at most, instead of visiting it \( n \) times as would be the case when using the basic approach.

Firstly, the algorithm starts by requesting \( n \) samples in the root node region and proceeds recursively. Whenever a node with index \( i \) is visited, the program must produce \( t \) random samples in \( R_i \). If the \( i \)-th node is a leaf, those \( t \) samples are obtained by rejection sampling. When the \( i \)-th node is an inner node, a partition of \( t \) is done, selecting four random integer values \( m_{i,0}, \ldots, m_{i,3} \), which assert \( m_{i,0} + m_{i,1} + m_{i,2} + m_{i,3} = t \) in such a way that the average value of \( m_{i,j} \) is \( n I_C(i,j) / I_i \). Then the algorithm is called for each \( C(i,j) \), the \( j \)-th child of the \( i \)-th node (this does not occur if \( m_{i,j} = 0 \)), and as a result we obtain in total four sets with \( t \) samples. Each leaf node \( j \) contains \( nI_i/I_0 \) samples on average, as required by importance sampling. Figure 5.4 shows a simple example. The procedure is detailed in algorithms [5] and [6].

![Figure 5.4: Quadtree traverse for 100 samples showing node probabilities and average number of samples.](image-url)

5.3.2 Efficient Use of the BRDF

Since each region should know the value of the integral \( \int_{x,y \in R_i} f_r(u, v, x, y) \, d(x) \, d(y) \) to construct adaptively a data structure, its performance should be revised. We have decided to precompute value of the BRDF when it is evaluated in \( x, y \in D^2 \), and

University of Granada
5.3. Optimal Adaptive Sampling

<table>
<thead>
<tr>
<th>SplitN(N) → N⁺ n[4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let p, q be arrays of R[4]</td>
</tr>
<tr>
<td>Set δ to N</td>
</tr>
<tr>
<td>for i=0 to 3 do begin</td>
</tr>
<tr>
<td>p[i] = Pu(Child[i]) * N</td>
</tr>
<tr>
<td>n[i] = ⌊p[i]⌋</td>
</tr>
<tr>
<td>δ = δ - n[i] ;</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>if ( δ ≥ 0 ) then</td>
</tr>
<tr>
<td>for i=0 to 3 do begin</td>
</tr>
<tr>
<td>q[i] = p[i] - n[i] ;</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>do</td>
</tr>
<tr>
<td>n[Choose(q)]++ ;</td>
</tr>
<tr>
<td>δ = δ - 1</td>
</tr>
<tr>
<td>while ( δ ≥ 0 ) ;</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>return n</td>
</tr>
</tbody>
</table>

Algorithm 5: code SplitN

consult a table for its data.

\[ \text{Image}(u, i_x, j_y) \simeq f_r(u, v_{xy}). \]

This table or \text{image-map} of the BRDF is associated with each quadtree, and it is destroyed optionally after its construction. It is used to build the quadtree with a bottom up approach, from a densely sampled domain, allowing us to reuse the already known values of the BRDF.

5.3.3 The Improved Area for Rejection Sampling

Reducing the precomputation time is not enough. We also need to reduce the sampling time given that it is the part of our method that is called so many times. Diminishing the sampling time is our main goal. Since our sampling algorithm is based on rejection sampling [Gen03], we should focus on optimizing the acceptance of a sample and thus lessening the area of rejection. The left image of Figure 5.5 gives an example of this situation. The letter A denotes acceptance area, and letter R accounts for the rejection area. Each region uses its area \((u, v) \times (u + \text{size}, v + \text{size})\) as a default domain for rejection sampling (from now on \((u, v) \times (U, V))\). Because of the relationship between the circle and the square that contains it, corner regions exhibit large areas of rejection — though they are not the only cases. Finding the optimal limit

R. Montes
Choose \((R q[4]) \rightarrow N^+\)
\[
\begin{align*}
\mathbb{R} \xi & \sim U(0,1) \\
\mathbb{R} r & = \xi \cdot (q[0]+q[1]+q[2]+q[3]) \\
\mathbb{R} s & = 0,0 \\
\text{for } i=0 \text{ to } 2 \text{ do begin} \\
& \quad s = s + q[i] \\
& \quad \text{if } (r \leq s) \text{ then} \\
& \quad \quad \text{return } i ; \\
\text{end} \\
\text{return } 3 ;
\end{align*}
\]

**Algorithm 6:** code Choose

\(( (x, y) \times (U, V) \text{ for this example) reduces } R, \text{ as is shown in the right image of Figure 5.5.} \)

We start by assigning a code to each region. Four distances are obtained \(A, B, C\) and \(D\) and with them, a binary code \(b_0 b_1 b_2 b_3\) for each region. We are only interested in the codes for those regions that should optimize their areas (see Figure 5.6).

- \(A^2 = u^2 + v^2 \quad \longrightarrow \quad b_0 = \begin{cases} 1 & \text{if } A \leq 1, \\ 0 & \text{else case.} \end{cases} \)
- \(B^2 = U^2 + v^2 \quad \longrightarrow \quad b_1 = \begin{cases} 1 & \text{if } B \leq 1, \\ 0 & \text{else case.} \end{cases} \)
- \(C^2 = u^2 + V^2 \quad \longrightarrow \quad b_2 = \begin{cases} 1 & \text{if } C \leq 1, \\ 0 & \text{else case.} \end{cases} \)
- \(D^2 = U^2 + V^2 \quad \longrightarrow \quad b_3 = \begin{cases} 1 & \text{if } D \leq 1, \\ 0 & \text{else case.} \end{cases} \)
5.3. Optimal Adaptive Sampling

The number of regions that should optimize their areas depends on the maximum depth that the quadtree reached. We explain here the case of depth three. In this case, eight regions use different domains.

- The region with code 0001 (1) changes its domain to \((x, y) \times (U, V)\).
- The region with code 0010 (2) changes its domain to \((u, y) \times (X, V)\).
- The region with code 0011 (3) changes its domain to \((u, y) \times (U, V)\).
- The region with code 0100 (4) changes its domain to \((x, v) \times (U, Y)\).
- The region with code 0101 (5) changes its domain to \((x, v) \times (U, V)\).
- The region with code 1000 (8) changes its domain to \((u, v) \times (x, y)\).
- The region with code 1010 (10) changes its domain to \((u, v) \times (x, Y)\).
- The region with code 1100 (12) changes its domain to \((u, v) \times (U, Y)\).

where:

\[
x = -\sqrt{1 - v^2} \quad y = -\sqrt{1 - u^2} \\
X = -\sqrt{1 - V^2} \quad Y = -\sqrt{1 - U^2}
\]

With this optimization we have reduced the sampling time in 90%. On average Test Scene 1 reduced from 229.58 to 8.2 seconds, and Test Scene 2 did it from 525.73 to 40.36 seconds.
Chapter – 5. Efficient Sampling for Generic BRDFs

<table>
<thead>
<tr>
<th>BRDF</th>
<th>Adaptive Disc</th>
<th>Factored PDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ashikhmin</td>
<td>57.61</td>
<td>83.46</td>
</tr>
<tr>
<td>BeardMaxwell</td>
<td>49.08</td>
<td>74.75</td>
</tr>
<tr>
<td>Blinn</td>
<td>32.12</td>
<td>45.08</td>
</tr>
<tr>
<td>Coupled</td>
<td>51.25</td>
<td>76.17</td>
</tr>
<tr>
<td>He</td>
<td>300.8</td>
<td>452.7</td>
</tr>
<tr>
<td>Lafortune</td>
<td>32.33</td>
<td>40.34</td>
</tr>
<tr>
<td>Lewis</td>
<td>32.49</td>
<td>48.86</td>
</tr>
<tr>
<td>Minnaert</td>
<td>35.33</td>
<td>54.28</td>
</tr>
<tr>
<td>Oren-Nayar</td>
<td>28.99</td>
<td>40.99</td>
</tr>
<tr>
<td>Phong</td>
<td>29.99</td>
<td>45.69</td>
</tr>
<tr>
<td>Poulin-Fournier</td>
<td>112.7</td>
<td>160.9</td>
</tr>
<tr>
<td>Schlick (D)</td>
<td>55.17</td>
<td>82.49</td>
</tr>
<tr>
<td>Schlick (S)</td>
<td>41.65</td>
<td>63.03</td>
</tr>
<tr>
<td>Strauss</td>
<td>38.82</td>
<td>56.86</td>
</tr>
<tr>
<td>Cook-Torrance</td>
<td>38.84</td>
<td>58.43</td>
</tr>
<tr>
<td>Ward</td>
<td>33.57</td>
<td>48.99</td>
</tr>
</tbody>
</table>

Table 5.1: Quadtree creation times for each BRDF model when the Adaptive Disc PDF is used. Also the factorization of the BRDF is performed in a pre-process stage. Data is relative to Test Scene 2.

5.3.4 Quadtree Set Construction Requirements

It was mentioned previously that our algorithm involves some more computations in order to represent closely any BRDF function. Table 5.1 shows information related to the cost in seconds of the precomputation for a given number of quadtree structures at varying incident angle directions. Once we have these structures on memory, they are used to estimate radiance. The values that are listed in the table correspond to the precomputation of 60 and 90 quadtrees, which are high enough to ensure a structure is available the closest to any incident direction. The average values are 62.58 and 94.69 seconds compared with 105.69, which is the cost of factored computation and precomputation of CDFs for sampling by using Lawrence’s technique. Also you may notice the extreme difference in terms of time between experimental and physically based reflection models.

Another issue concerning the requirements of our method is memory consumption. Let us consider firstly our basic algorithm with no optimization. A single quadtree represents the unit disc domain as node regions given an incident direction. Its depth, and so its memory, depends on the \( n_{\text{max}} \) parameter (see equation 5.1). Let us consider a variation of this parameter to observe how it varies with the BRDF in use. This is given in Table 5.2, page 91. Note that a value of 1.3 —the average number of tries in rejection sampling— is very restrictive, forcing to many subdivisions. On
5.3. Optimal Adaptive Sampling

<table>
<thead>
<tr>
<th>BRDF</th>
<th>nmax - Adaptive Disc PDF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.3</td>
</tr>
<tr>
<td>Ashikhmin</td>
<td></td>
</tr>
<tr>
<td>Beardmax</td>
<td>1412.8</td>
</tr>
<tr>
<td>Blinn</td>
<td>522.410</td>
</tr>
<tr>
<td>Coupled</td>
<td>322.969</td>
</tr>
<tr>
<td>He</td>
<td>1346.3</td>
</tr>
<tr>
<td>Lafontaine</td>
<td>2344.6</td>
</tr>
<tr>
<td>Lewis</td>
<td>4701.9</td>
</tr>
<tr>
<td>Minnaert</td>
<td>974.695</td>
</tr>
<tr>
<td>Oren</td>
<td>0.103</td>
</tr>
<tr>
<td>Phong</td>
<td>2455.7</td>
</tr>
<tr>
<td>Poulin</td>
<td>308.801</td>
</tr>
<tr>
<td>Schlick (D)</td>
<td>522.410</td>
</tr>
<tr>
<td>Schlick (S)</td>
<td>522.410</td>
</tr>
<tr>
<td>Strauss</td>
<td>380.730</td>
</tr>
<tr>
<td>Torrance</td>
<td>728.391</td>
</tr>
<tr>
<td>Average</td>
<td>1129.5</td>
</tr>
</tbody>
</table>

Table 5.2: Memory requirements for a single quadtree when the parameter \( n_{max} \) varies.

the other hand, a value of 3 leads to a uniform subdivision. The optimal value is \( n_{max}=2 \).

Detailed requirements for memory in both generic sampling methods—Adaptive Disc PDF and the Factored PDF—can also be found in Table 5.3. We have use 60 quadtreess and the parameter \( n_{max} \) is always set to 2. This scheme serves any BRDF model and the user does not need to adjust our method depending on the reflectance function. This is not the case with Lawrence’s factorization method which should select numerically the best parametrization for each BRDF (see Table 5.6, page 95).

We also consider the memory requirements of our method when it is used to sample acquired data from measurements. This was commented on in section 5.6 where we also compared our method with the use of sampling by inversion of the CDF—a common technique for tabular BRDFs—which usually requires more storage than the BRDF itself.

Just a note about our rendering system. The description of the the PDF method could be given by parameter in an option file (see section 6.1.3) but this will cause every BRDF of the scene to be sampled with the same method. We could assign different methods for many BRDF in the scene by using our scene description format: GRF. We have already given the expression in GRF of some of the BRDF models and PDF methods we have implemented. It is now time to show how our adaptive and general sampling method is defined with GRF:
Table 5.3: Memory requirements in KB. Both Adaptive Disc PDF and Lawrence’s factorization of the BRDF precompute and store some data.

<table>
<thead>
<tr>
<th>BRDF</th>
<th>Adaptive Disc</th>
<th>Factored PDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ashikhmin</td>
<td>6.25</td>
<td>1031</td>
</tr>
<tr>
<td>BeardMax.</td>
<td>1713.25</td>
<td>6454</td>
</tr>
<tr>
<td>Blinn</td>
<td>582.25</td>
<td>6481</td>
</tr>
<tr>
<td>Coupled</td>
<td>6.25</td>
<td>1033</td>
</tr>
<tr>
<td>He</td>
<td>2407.25</td>
<td>1034</td>
</tr>
<tr>
<td>Lafortune</td>
<td>1273.25</td>
<td>6445</td>
</tr>
<tr>
<td>Lewis</td>
<td>1279.25</td>
<td>6445</td>
</tr>
<tr>
<td>Minnaert</td>
<td>1461.25</td>
<td>1031</td>
</tr>
<tr>
<td>Oren</td>
<td>6.25</td>
<td>1033</td>
</tr>
<tr>
<td>Phong</td>
<td>1279.25</td>
<td>6445</td>
</tr>
<tr>
<td>Poulin</td>
<td>297.25</td>
<td>1038</td>
</tr>
<tr>
<td>Schlick (D)</td>
<td>342.25</td>
<td>1033</td>
</tr>
<tr>
<td>Schlick (S)</td>
<td>780.25</td>
<td>1043</td>
</tr>
<tr>
<td>Strauss</td>
<td>727.25</td>
<td>1052</td>
</tr>
<tr>
<td>Torrance</td>
<td>631.25</td>
<td>1029</td>
</tr>
<tr>
<td>Ward</td>
<td>483.25</td>
<td>1038</td>
</tr>
</tbody>
</table>

**ADAPTIVEPDF** ::=  
\[pdf \text{ adaptive nQT} = \text{value nAnisoQT} = \text{value}\]

Table 5.4: GRF notation of **ADAPTIVEPDF**
5.4 Results

In this section we provide the results of our adaptive sampling method for various reflectance models, and we compare the computing time and the average relative error we obtain for several images under different sampling strategies (PDFs). All the images were rendered using a naive path tracing algorithm under a Linux system in an AMD64 processor machine with 2GB of RAM. The test scenes are shown in Figure 5.7.

We have analysed different PDF functions (including the one we present) and have measured their performance for various BRDF models when high variation occurs: at a specular peak and also a combination with environment map sampling. Images have been obtained with a varying number of samples per pixel. The maximum quality (in terms of paths per pixel) has been used to produce a reference image.

Each image of the experiment uses each BRDF function from a list of theoretical and empirical models and for a given instance (see Table 5.5) and samples it with the following five techniques: (1) uniform sampling technique, (2) cosine lobe sampling on $S^2$ (3) the same on $\Omega$, (4) Lawrence’s factorization of the BRDF [LRR04] and (5) the proposed adaptive method on the disc. Using the maximum number of samples for every BRDF and for every PDF we get a reference image for each experiment. When we run the comparison each image yields a relative error value, computed with respect to every pixel with non-null radiance in the reference image. We take an average for the relative error of each image that were obtained with an specific method, as the final error. The smallest value, determines the best sampling scheme. In the next sections, we detail the results obtained for various test scene situations.

![Figure 5.7: Test scenes 1 and 2 with our Linux GRF viewer.](image-url)

![Figure 5.8: GRF snapshot of scene 4.](image-url)
Chapter – 5. Efficient Sampling for Generic BRDFs

<table>
<thead>
<tr>
<th>BRDF</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASHIKHMIN</td>
<td>nu = 25</td>
</tr>
<tr>
<td></td>
<td>nv = 25</td>
</tr>
<tr>
<td></td>
<td>ks = 0.39</td>
</tr>
<tr>
<td></td>
<td>kd = 0.6</td>
</tr>
<tr>
<td>BEARDMAXWELL</td>
<td>Ω = 1</td>
</tr>
<tr>
<td></td>
<td>τ = 0</td>
</tr>
<tr>
<td></td>
<td>n_{real} = 0.37</td>
</tr>
<tr>
<td></td>
<td>n_{imag} = 2.82</td>
</tr>
<tr>
<td></td>
<td>rs = 0.4</td>
</tr>
<tr>
<td></td>
<td>rd = 0.75</td>
</tr>
<tr>
<td></td>
<td>rv = 0.5</td>
</tr>
<tr>
<td>BLINN</td>
<td>n = 100</td>
</tr>
<tr>
<td></td>
<td>ks = 1</td>
</tr>
<tr>
<td></td>
<td>kd = 0.67</td>
</tr>
<tr>
<td>COUPLED</td>
<td>R_m = 1</td>
</tr>
<tr>
<td></td>
<td>R_o = 0.05</td>
</tr>
<tr>
<td>HE</td>
<td>σ = 0.5</td>
</tr>
<tr>
<td></td>
<td>τ = 3</td>
</tr>
<tr>
<td></td>
<td>λ = 800</td>
</tr>
<tr>
<td>LAFORTUNE</td>
<td>cx = -1</td>
</tr>
<tr>
<td></td>
<td>cy = -1</td>
</tr>
<tr>
<td></td>
<td>cz = 1</td>
</tr>
<tr>
<td></td>
<td>n = 100</td>
</tr>
<tr>
<td></td>
<td>ks = 1</td>
</tr>
<tr>
<td></td>
<td>kd = 0</td>
</tr>
<tr>
<td>LEWIS</td>
<td>k = 0.8</td>
</tr>
<tr>
<td></td>
<td>kd = 1</td>
</tr>
<tr>
<td>MINNAERT</td>
<td>s = 0</td>
</tr>
<tr>
<td></td>
<td>r = 1</td>
</tr>
<tr>
<td>OREN</td>
<td>n = 100</td>
</tr>
<tr>
<td></td>
<td>ks = 0.4</td>
</tr>
<tr>
<td></td>
<td>kd = 0.6</td>
</tr>
<tr>
<td>PHONG</td>
<td>d = 2</td>
</tr>
<tr>
<td></td>
<td>h = 0.01</td>
</tr>
<tr>
<td></td>
<td>n = 100</td>
</tr>
<tr>
<td></td>
<td>ks = 0.8</td>
</tr>
<tr>
<td></td>
<td>kd = 0.2</td>
</tr>
<tr>
<td>SCHLICK (D)</td>
<td>sc = 0.7</td>
</tr>
<tr>
<td></td>
<td>sr = 0.31</td>
</tr>
<tr>
<td></td>
<td>sp = 1</td>
</tr>
<tr>
<td></td>
<td>ly = true</td>
</tr>
<tr>
<td></td>
<td>dc = 1</td>
</tr>
<tr>
<td></td>
<td>dr = 0</td>
</tr>
<tr>
<td></td>
<td>dp = 1</td>
</tr>
<tr>
<td>SCHLICK (S)</td>
<td>sc = 0.7</td>
</tr>
<tr>
<td></td>
<td>sr = 0.31</td>
</tr>
<tr>
<td></td>
<td>sp = 1</td>
</tr>
<tr>
<td></td>
<td>ly = false</td>
</tr>
<tr>
<td>STRAUSS</td>
<td>s = 0.76</td>
</tr>
<tr>
<td></td>
<td>m = 0.91</td>
</tr>
<tr>
<td></td>
<td>ks = 0.64</td>
</tr>
<tr>
<td></td>
<td>kd = 0.5</td>
</tr>
<tr>
<td>TORRANCE</td>
<td>m = 0.35</td>
</tr>
<tr>
<td></td>
<td>n_{real} = 0.617</td>
</tr>
<tr>
<td></td>
<td>n_{imag} = 2.63</td>
</tr>
<tr>
<td></td>
<td>ks = 0.7</td>
</tr>
<tr>
<td></td>
<td>kd = 0.3</td>
</tr>
<tr>
<td>WARD</td>
<td>sx = 0.2</td>
</tr>
<tr>
<td></td>
<td>sy = 0.2</td>
</tr>
<tr>
<td></td>
<td>ks = 0.5</td>
</tr>
<tr>
<td></td>
<td>kd = 0.5</td>
</tr>
</tbody>
</table>

Table 5.5: List of the reflectance models, with the parametrization used through the evaluation of our model.

5.4.1 Test Scene 1: the Cornell Box

The Cornell Box\(^1\) is a well known scene. It is characterized by having a strong uniform inter-reflexion component, thus irradiance at any point is almost constant and independent of incoming direction. In this context, PDF selection does not have a great impact on variance, because which directions are sampled is not so relevant, as almost the same irradiance comes from all of them. Uniform sampling takes advantage of that fact, as seen in the time versus error comparison of Figure 5.9. Data relative to the average per BRDF sampling time and relative error appears in Tables 5.7 and 5.8 respectively.

5.4.2 Test Scene 2: the Glossy Sphere

Now we focus on sampling non-uniform scenes. We rendered a sphere lit by a single area light. We focused our measurements on the portion of the image containing the highlight on the sphere, because that is where efficiency of different sampling approaches differs the most.

5.4.2.1 Use of Constant Parameters for some PDFs

Some PDF function parameters should also be given. For example, a cosine-lobe based PDF uses an exponent parameter \(n\). Frequently, the value for \(n\) is taken from the

\(^1\) On line at: <http://www.graphics.cornell.edu/online/box/>
5.4. Results

<table>
<thead>
<tr>
<th>BRDF</th>
<th>(N_u \times N_{u_0})</th>
<th>(N_p \times N_{p_0})</th>
<th>(J \times K)</th>
<th>Reparam.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASHIKHMIN</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>1 \times 2</td>
<td>false</td>
</tr>
<tr>
<td>BEARDMAXWELL</td>
<td>16 \times 16</td>
<td>100 \times 32</td>
<td>2 \times 3</td>
<td>false</td>
</tr>
<tr>
<td>BLINN</td>
<td>16 \times 16</td>
<td>100 \times 32</td>
<td>3 \times 3</td>
<td>true</td>
</tr>
<tr>
<td>COUPLED</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>1 \times 3</td>
<td>false</td>
</tr>
<tr>
<td>HE</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>2 \times 1</td>
<td>false</td>
</tr>
<tr>
<td>LAFORTUNE</td>
<td>16 \times 16</td>
<td>100 \times 32</td>
<td>2 \times 2</td>
<td>true</td>
</tr>
<tr>
<td>LEWIS</td>
<td>16 \times 16</td>
<td>100 \times 32</td>
<td>2 \times 2</td>
<td>true</td>
</tr>
<tr>
<td>MINNAERT</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>1 \times 2</td>
<td>false</td>
</tr>
<tr>
<td>OREN</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>1 \times 3</td>
<td>false</td>
</tr>
<tr>
<td>PHONG</td>
<td>16 \times 16</td>
<td>100 \times 32</td>
<td>2 \times 2</td>
<td>true</td>
</tr>
<tr>
<td>POULIN</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>2 \times 2</td>
<td>false</td>
</tr>
<tr>
<td>SCHLICK (D)</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>1 \times 3</td>
<td>true</td>
</tr>
<tr>
<td>SCHLICK (S)</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>2 \times 3</td>
<td>false</td>
</tr>
<tr>
<td>STRAUSS</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>3 \times 3</td>
<td>false</td>
</tr>
<tr>
<td>TORRANCE</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>1 \times 1</td>
<td>false</td>
</tr>
<tr>
<td>WARD</td>
<td>16 \times 16</td>
<td>32 \times 16</td>
<td>2 \times 2</td>
<td>true</td>
</tr>
</tbody>
</table>

Table 5.6: We adjust manually the seven parameters for each BRDF factorization. They minimize the average original matrix value and its approximation.

<table>
<thead>
<tr>
<th>Samples</th>
<th>Uniform</th>
<th>CosLobe (\delta^2)</th>
<th>CosLobe (\Omega)</th>
<th>Adapt.Disc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1&lt;sup&gt;2&lt;/sup&gt;</td>
<td>0.0793</td>
<td>0.4136</td>
<td>0.3216</td>
<td>0.3396</td>
</tr>
<tr>
<td>5&lt;sup&gt;2&lt;/sup&gt;</td>
<td>0.9022</td>
<td>5.0623</td>
<td>7.6766</td>
<td>7.6304</td>
</tr>
<tr>
<td>10&lt;sup&gt;2&lt;/sup&gt;</td>
<td>3.2513</td>
<td>18.6036</td>
<td>30.5650</td>
<td>29.9153</td>
</tr>
<tr>
<td>15&lt;sup&gt;2&lt;/sup&gt;</td>
<td>7.2801</td>
<td>40.7002</td>
<td>68.6088</td>
<td>66.7410</td>
</tr>
<tr>
<td>20&lt;sup&gt;2&lt;/sup&gt;</td>
<td>12.9957</td>
<td>74.3947</td>
<td>68.6088</td>
<td>118.2702</td>
</tr>
<tr>
<td>30&lt;sup&gt;2&lt;/sup&gt;</td>
<td>28.8140</td>
<td>163.221</td>
<td>273.053</td>
<td>263.706</td>
</tr>
<tr>
<td>40&lt;sup&gt;2&lt;/sup&gt;</td>
<td>50.8346</td>
<td>286.605</td>
<td>486.620</td>
<td>483.715</td>
</tr>
<tr>
<td>50&lt;sup&gt;2&lt;/sup&gt;</td>
<td>94.1930</td>
<td>510.074</td>
<td>761.316</td>
<td>878.420</td>
</tr>
<tr>
<td>100&lt;sup&gt;2&lt;/sup&gt;</td>
<td>333.768</td>
<td>1090.65</td>
<td>2947.66</td>
<td>2882.860</td>
</tr>
</tbody>
</table>

Table 5.7: Sampling times measured in seconds for the Cornell Box. Increments of the number of samples implies longer sampling times.
The effectiveness of a sampling scheme is shown by relating the time and relative error of a given solution. Convergence to minimum error is obtained with an increase in the number of samples.

Figure 5.9: The effectiveness of a sampling scheme is shown by relating the time and relative error of a given solution. Convergence to minimum error is obtained with an increase in the number of samples.

corresponding exponent in the BRDF in use, if this BRDF is based on a cosine lobe. However, there is no information to set the PDF exponent if we sample a BRDF model which does not depend on that parameter, thus a constant must be used. This is the situation we have used for Test Scene 2 Sphere 1. In this case, we also consider equal factorization parameters for all the BRDF functions. Constant factorization is not the best case, of course, but with Sphere 1 we aim to show the situation in which the user is not need to set any parameter.

The graph plot is shown in Figure 5.10. For our comparison we have considered a simple uniform PDF—that is, almost inexpensive, but not proportional to the BRDF—cosine-lobe based PDF, very common for most analytical BRDFs and Lawrence’s factorization of the BRDFs. We have implemented two versions of the cosine-lobe PDF. One generates samples in the whole domain of the direction sphere $S^2$ and then rejects negative directions, which is very fast. The other version, a cosine-lobe PDF defined in $\Omega$, uses every sample. time 5.9 & error 5.10

5.4.2.2 Use of Adjusted Parameters for some PDFs

In order to make comparisons fairer we have found manually the exponent that yields the best match between the lobe-based PDF and each BRDF function, for Test Scene Sphere 2. Even for Phong-based BRDFs, the best $n$ for the PDF can be different to
### 5.4. Results

<table>
<thead>
<tr>
<th>SAMPLES</th>
<th>Uniform</th>
<th>CosLobe S²</th>
<th>CosLobe Ω</th>
<th>Adapt.Disc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1²</td>
<td>109.97</td>
<td>88.42</td>
<td>94.72</td>
<td>80.36</td>
</tr>
<tr>
<td>5²</td>
<td>48.30</td>
<td>74.92</td>
<td>47.50</td>
<td>18.18</td>
</tr>
<tr>
<td>10²</td>
<td>29.99</td>
<td>31.15</td>
<td>27.75</td>
<td>9.89</td>
</tr>
<tr>
<td>15²</td>
<td>23.82</td>
<td>26.85</td>
<td>27.86</td>
<td>8.58</td>
</tr>
<tr>
<td>20²</td>
<td>20.77</td>
<td>23.01</td>
<td>25.74</td>
<td>8.78</td>
</tr>
<tr>
<td>30²</td>
<td>23.84</td>
<td>19.35</td>
<td>18.52</td>
<td>8.65</td>
</tr>
<tr>
<td>40²</td>
<td>16.80</td>
<td>16.94</td>
<td>17.37</td>
<td>9.55</td>
</tr>
<tr>
<td>50²</td>
<td>8.33</td>
<td>9.03</td>
<td>17.28</td>
<td>9.98</td>
</tr>
<tr>
<td>100²</td>
<td>3.93</td>
<td>3.21</td>
<td>4.75</td>
<td>2.79</td>
</tr>
</tbody>
</table>

Table 5.8: The Cornell scene: error is calculated by the comparison of each image (for a BRDF, sampled with a PDF and using a given number of samples) with the reference image. The lowest error is achieved with our sampling scheme.

<table>
<thead>
<tr>
<th>AVERAGE</th>
<th>1²</th>
<th>5²</th>
<th>10²</th>
<th>15²</th>
<th>20²</th>
<th>30²</th>
<th>40²</th>
<th>50²</th>
<th>100²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td>0.00006</td>
<td>0.00042</td>
<td>0.0016</td>
<td>0.0034</td>
<td>0.0061</td>
<td>0.0138</td>
<td>0.0233</td>
<td>0.0379</td>
<td>15.39</td>
</tr>
<tr>
<td>CosLobe S²</td>
<td>0.00018</td>
<td>0.000298</td>
<td>0.00118</td>
<td>0.00266</td>
<td>0.00471</td>
<td>0.01062</td>
<td>0.01884</td>
<td>0.03410</td>
<td>117.74</td>
</tr>
<tr>
<td>CosLobe Ω</td>
<td>0.00025</td>
<td>0.000350</td>
<td>0.00137</td>
<td>0.00306</td>
<td>0.00544</td>
<td>0.01223</td>
<td>0.02188</td>
<td>0.03410</td>
<td>185.99</td>
</tr>
<tr>
<td>Adaptive</td>
<td>0.00026</td>
<td>0.000279</td>
<td>0.00094</td>
<td>0.00196</td>
<td>0.00356</td>
<td>0.00736</td>
<td>0.01295</td>
<td>0.02003</td>
<td>73.32</td>
</tr>
<tr>
<td>Factored BRDF</td>
<td>0.00023</td>
<td>0.000346</td>
<td>0.00135</td>
<td>0.00300</td>
<td>0.00527</td>
<td>0.01183</td>
<td>0.02096</td>
<td>0.03620</td>
<td>128.36</td>
</tr>
</tbody>
</table>

Table 5.9: Data of Test Scene 2 – Sphere 1. The unknown parameters of some sampling schemes are set to a constant value for every BRDF instance in use.

<table>
<thead>
<tr>
<th>AVERAGE</th>
<th>1²</th>
<th>5²</th>
<th>10²</th>
<th>15²</th>
<th>20²</th>
<th>30²</th>
<th>40²</th>
<th>50²</th>
<th>100²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td>225.05</td>
<td>85.38</td>
<td>59.31</td>
<td>49.27</td>
<td>44.29</td>
<td>41.06</td>
<td>29.77</td>
<td>26.72</td>
<td>22.48</td>
</tr>
<tr>
<td>CosLobe S²</td>
<td>75.87</td>
<td>59.86</td>
<td>32.65</td>
<td>29.31</td>
<td>27.65</td>
<td>26.63</td>
<td>26.24</td>
<td>25.24</td>
<td>25.07</td>
</tr>
<tr>
<td>CosLobe Ω</td>
<td>75.56</td>
<td>58.00</td>
<td>32.50</td>
<td>34.24</td>
<td>27.86</td>
<td>26.21</td>
<td>26.18</td>
<td>25.37</td>
<td>25.08</td>
</tr>
<tr>
<td>Adaptive</td>
<td>141.14</td>
<td>34.95</td>
<td>25.33</td>
<td>23.21</td>
<td>23.27</td>
<td>22.58</td>
<td>22.47</td>
<td>22.56</td>
<td>22.47</td>
</tr>
<tr>
<td>Factored</td>
<td>973.21</td>
<td>88.15</td>
<td>68.45</td>
<td>72.56</td>
<td>68.04</td>
<td>65.10</td>
<td>64.00</td>
<td>63.99</td>
<td>62.75</td>
</tr>
</tbody>
</table>

Table 5.10: Average relative error of test scene 2 – Sphere 1. We compare our method with the worst parametrization for the cosine lobe and the BRDF factorization schemes.
Chapter – 5. Efficient Sampling for Generic BRDFs

Figure 5.10: Results for Sphere 1. Optimal PDF parameters for Cosine-Lobe and the factored representation are unknown, so constant values are given.

Figure 5.11: Similar values of the Phong-exponent $n$ yield different PDF and BRDF shapes.

the BRDF’s exponent. This is because both the PDF and the BRDF include the term $(\mathbf{u} \cdot \mathbf{v})^n$. However, the BRDF also includes the cosine term $(\mathbf{v} \cdot \mathbf{n})$ whereas the PDF does not. For instance, in Figure 5.11 we have plotted the Lewis normalized Phong model [Lew93], with parameters $k_d = 0.5$, $k_s = 0.5$ and $n = 32.5$. The best PDF exponent is $n = 16$.

For the Factored PDF we have found the best factorization. It is necessary to select the value of seven PDF parameters for each BRDF. The parameters are: $N_{\theta_u} \times N_{\phi_u}$ and $N_{\theta_p} \times N_{\phi_p}$ for matrix size, $J \times K$ for the numbers of lobes that approximates the..
BRDF and whether or not to use the half-angle reparametrization. The best values are found comparing the average original matrix value with the average value from the product of the factors.

The experiment was repeated by adjusting manually the parameters in order to find the PDF shape closest to that of the BRDF. The results for this setup are plotted in Figure 5.12. In Table 5.11 we see that sampling times for these PDFs are better in some cases than our Adaptive Disc method. Nevertheless, if we consider time versus error, our algorithm performance compares favourably to those methods. Detailed error data is given in Table 5.12. You should observe that standard BRDF sampling lacks generality and cannot be used to sample many reflectance models. Our adaptive method can be used with any isotropic BRDF but also does not require manual selection of the parameters, nor knowledge of the BRDF. It simply needs the ability to evaluate the BRDF.

Figure 5.12: The effectiveness of a sampling scheme is shown by relating time and relative error to several solutions. Convergence to minimum error is obtained with an increase in the number of samples.
### Table 5.11: Sampling times of Test Scene 2 using optimal exponent values for the cosine lobe scheme. Furthermore, the factorization is manually adjusted for each BRDF.

<table>
<thead>
<tr>
<th>BRDF Type</th>
<th>1°</th>
<th>5°</th>
<th>10°</th>
<th>15°</th>
<th>20°</th>
<th>30°</th>
<th>40°</th>
<th>50°</th>
<th>100°</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td>0.00006</td>
<td>0.00012</td>
<td>0.00016</td>
<td>0.00024</td>
<td>0.00001</td>
<td>0.00138</td>
<td>0.00243</td>
<td>0.00475</td>
<td>15.399</td>
</tr>
<tr>
<td>CosLobe $S^2$</td>
<td>0.00018</td>
<td>0.000291</td>
<td>0.00114</td>
<td>0.00256</td>
<td>0.00456</td>
<td>0.01023</td>
<td>0.01853</td>
<td>0.02833</td>
<td>112.960</td>
</tr>
<tr>
<td>CosLobe $\Omega$</td>
<td>0.00025</td>
<td>0.000368</td>
<td>0.00145</td>
<td>0.00324</td>
<td>0.0077</td>
<td>0.0188</td>
<td>0.0292</td>
<td>0.03573</td>
<td>142.965</td>
</tr>
<tr>
<td>Adaptive Disc</td>
<td>0.00026</td>
<td>0.000279</td>
<td>0.00094</td>
<td>0.00196</td>
<td>0.00336</td>
<td>0.00736</td>
<td>0.01285</td>
<td>0.2063</td>
<td>73.324</td>
</tr>
<tr>
<td>Factored BRDF</td>
<td>0.00028</td>
<td>0.000321</td>
<td>0.00123</td>
<td>0.00273</td>
<td>0.00483</td>
<td>0.1083</td>
<td>0.1921</td>
<td>0.2086</td>
<td>115.394</td>
</tr>
</tbody>
</table>

Table 5.12: Test Scene 2: the error (in %) calculated by the comparison of each image (for a BRDF, sampled with a PDF and using a given number of samples) with the reference image. The lowest error is achieved with our sampling scheme.
5.4.3 Test Scene 3: Sampling many BRDFs

At this point we deal with the Dragon model from Stanford University\(^2\). The reflectance function used in this scene corresponds to Oren’s [ON94] with a rough value of 0.83, and a Strauss instance [Str90] mostly smooth for floor and wall respectively. The dragon itself has a Lafortune BRDF [LW94] with exponent \( n = 46 \).

With this mixture of BRDFs, we can compare our sampling method with uniform sampling, cosine lobe in \( \Omega \), and Lawrence’s Factored representation. For the cosine lobe and the factored BRDF we adjusted the parametrization manually to fit the shape of each BRDF instance. With only 100 samples, our algorithm gives results with less noise than the others. You can see this in Figure 5.13.

![Figure 5.13: From left to right, images corresponding to the Uniform PDF, adjusted cosine-lobe strategy in \( \Omega \), the Factored representation of the BRDF and finally our algorithm sampling. The Adaptive Disc PF shows less noise using the same number of samples as the others. The resolution is 400 \( \times \) 400 pixels. Following the same order, the sampling time is: 9.749, 29.06, 24.539 and 42.632 seconds respectively.](image)

---

\(^2\) The Stanford 3D Scanning Repository is on-line at <http://graphics.stanford.edu/data/3Dscanrep/> [last visit November 2007]

R. Montes
5.4.4 Test Scene 4: the Pottery Scene

In this section, we show results for the scene in Figure 5.4.4. In this test scene, we use three instances of the same pottery geometry. Firstly, we consider three different BRDF instances in the same scene. Secondly, we use this scene to compare our method with varying factorization approaches. Finally, we address the problem of efficient quality rendering of scenes illuminated by a distant natural illumination given in an environment map and taking into account complex reflection functions. Let us start with the configuration of this scene. The reflectance functions are:

- An instance of BeardMaxwell-BRDF [MBWL73] with $\Omega=1$ $\tau=0$ gold index of refraction and $r_s=0.7$ $r_d=0.5$ $r_v=0.2$. The CosLobe method uses an exponent of 81. The BRDF is factored using $N_{\theta_u} = 16 \times N_{\phi_u} = 16$, $N_{\theta_p} = 32 \times N_{\phi_p} = 16$ and $J \times K$ as $2 \times 3$, with a reparametrization over vector $v$.

- An instance of He-BRDF [HTSG91] with parameters: $\sigma=0.5$ $\tau = 3$ $\lambda=800$. In this case the lobe exponent is $n=81$. It is factored using $16 \times 16$ $32 \times 32$ $J=2$ and $K = 1$.

- The classical Blinn-BRDF [Bli77] instance with $n=100$ $ks=1.0$ and $kd=0.67$. The exponent for the CosLobe PDF is $n=81$ and does not correspond to the same exponent of the lobe-based BRDF. The factorization uses $16 \times 16$ $100 \times 32$ $J=2$ $K = 2$ with the half-angle reparametrization.

We render the same scene with: (1) uniform sampling, (2) standard BRDF sampling, (3) the factorization of the BRDF and (4) our method. Uniform sampling is simple and fast for computation but leads to visible noise when sampling the three functions. Cosine-lobe sampling is the most efficient sampling for the third jug, the Blinn BRDF but this scheme is not suited for non-lobe-based BRDFs. Factored BRDF could not use the same factorization parameters for all the BRDF models in the scene, so user guidance is needed.

The performance of our adaptive algorithm is even better than the aforementioned PDFs for this scene, but the real advantage is that we do not change our algorithm, or set internal parameters. We focus on in the comparison of our sampling method with the factored BRDF. For this purpose, we considered different aspects of the parametrization of the Lawrence’s method. This is: using a fixed $N_u \times N_p$ and maintaining the number of factors as in Table 5.6. The resolution increases to $32 \times 16 \times 64 \times 32$ for every function. These changes imply more memory requirements. In fact, the factorization now uses 4096 Kbytes per BRDF instance, whereas our method uses on average 564.2 Kbytes. Another consideration is the precomputation time of this scheme. On average, the factorization of the BRDF took 125.17 seconds and the adaptive disc with 60 quadtrees only 36.44 seconds. Let us call this situation pottery3. The results are summarized in Tables 5.13, 5.14 and 5.15.

University of Granada
5.4. Results

Figure 5.14: Pottery scene rendered using different PDFs. Top to bottom: uniform, cosine lobe, factored BRDF and adaptive method. With the same number of samples, 200, our approach gives less error.

<table>
<thead>
<tr>
<th>Average</th>
<th>5^2</th>
<th>10^2</th>
<th>15^2</th>
<th>20^2</th>
<th>25^2</th>
<th>30^2</th>
<th>35^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factored BRDF</td>
<td>126.193</td>
<td>128.782</td>
<td>125.661</td>
<td>127.300</td>
<td>129.731</td>
<td>127.282</td>
<td>126.941</td>
</tr>
</tbody>
</table>

Table 5.13: Precomputation times obtained for Test Scene 4.

We ran the experiments but surprisingly, the increase in memory in Lawrence’s method did not change greatly the final comparative graph 5.15. As in other cases, the factored representation is quite fast in sampling, on average just 704.42 seconds. Our method needed 2273.36 seconds. We should not draw conclusions from that. As the factored solution is an approximate one, and our adaptive sampling is an exact scheme, the average relative error tips the scales, giving a value of 35.59% for the Factored BRDF and 8.62% in our case. You should note that if our method gives less error with the same number of samples, we could use less samples to obtain the same error, as with other techniques, and all using less time.

To satisfy our curiosity we also tried another variant of the BRDF parametrization for this scene (we already did almost the same in Test Scene 2 – Sphere 1). We used

<table>
<thead>
<tr>
<th>Average</th>
<th>5^2</th>
<th>10^2</th>
<th>15^2</th>
<th>20^2</th>
<th>25^2</th>
<th>30^2</th>
<th>35^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factored BRDF</td>
<td>1.573</td>
<td>6.064</td>
<td>13.108</td>
<td>23.326</td>
<td>52.330</td>
<td>92.628</td>
<td>145.139</td>
</tr>
<tr>
<td>Adaptive Disc</td>
<td>5.001</td>
<td>19.453</td>
<td>42.775</td>
<td>81.290</td>
<td>77.883</td>
<td>311.985</td>
<td>473.752</td>
</tr>
</tbody>
</table>

Table 5.14: Sampling times (in seconds) for Test Scene 4.
The image resolution is 500 × 220

R. Montes
### Chapter – 5. Efficient Sampling for Generic BRDFs

<table>
<thead>
<tr>
<th>Average</th>
<th>(5^2)</th>
<th>(10^2)</th>
<th>(15^2)</th>
<th>(20^2)</th>
<th>(30^2)</th>
<th>(40^2)</th>
<th>(50^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FACTORED BRDF</td>
<td>38.12</td>
<td>39.06</td>
<td>38.63</td>
<td>35.04</td>
<td>35.08</td>
<td>35.08</td>
<td>38.80</td>
</tr>
<tr>
<td>ADAPTIVE DISC</td>
<td>10.65</td>
<td>8.79</td>
<td>8.43</td>
<td>8.33</td>
<td>8.23</td>
<td>8.23</td>
<td>8.19</td>
</tr>
</tbody>
</table>

**Table 5.15**: Relative error, on average, obtained for Test Scene 4.

**Figure 5.15**: The scene with the three jugs proves that our adaptive sampling method performs better than current generic sampling schemes.

16 × 16 × 32 × 16 resolution with a number of factors of 1 × 2 with \(v\) in diffuse BRDFs and 1 × 3 with \(h\) reparametrization in glossy BRDFs. We want to know whether this is a good rule but also, it enables us to easily remember the parametrization of the BRDFs. If it works, the user who sets several BRDFs in a single scene, should not worry too much. Let us name this situation **pottery1**. Despite the above the results did not change considerably.
5.5 Sampling the Product of BRDF and Lighting Functions

We present an adaptive sampling method based on a hierarchical data structure that gives directions following a PDF that is exactly proportional to the BRDF and a cosine term. Other sampling methods use adaptive hierarchical representations for different purposes, such as those dealing with importance sampling of the product of two functions: the incoming radiance $L_i$ (area lights or environment maps) and the BRDF $f_r$.

**Bidirectional Importance Sampling** (BIS) [BGH05] samples the environment map inverting its CDF and also samples the BRDF using a probability tree [MH97]. This estimator has two ways to discard points where the value of the product is too low. One approximates the product using rejection sampling, and a second and the other is a limited version which uses resampling [Tan96]. In the first case, close to 90% of the points are discarded.

**Resampling Importance Sampling** (RIS) [JTE05], tackles the problem of sampling both functions by resampling. From a proposal of $M$ samples generated according to a simpler distribution it is possible to generate $N$ samples following a second and more complex function which does not need to be normalized. When $N = M = 1$ the estimator is equivalent to importance sampling. Some heuristics exist for optimal $N$ and $M$ selection.

**Wavelet Importance Sampling** (WaIS) [CJAMJ05] generates values that are proportional to the product of two N-dimensional functions given their representation using Haar-based wavelets. The product is a new wavelet defined by the product coefficients of both wavelet representations. These coefficients are used in a hierarchical sampling method, which warps a sample set according to $f_r \cdot L_i$. The algorithm starts from a coarse level of representation towards the fine level of the wavelet tree, warping points following a probability defined by the scalar coefficients of the child nodes. This is not an exact technique for sampling both functions because their representations come from sparse levels of detail.

**A two pass method** from Cline et al. [CETC06] is a variant of WaIS where no wavelet representation is needed to guide warping. Instead, sampling uses a BSP tree that adaptively represents the environment map (EM). Exact BRDF values are stored on the region’s corners. Many subdivision criteria ensure important parts of both functions are taken into account. In the first step a hierarchical structure is created for each primary ray in the scene. Secondly it warps a uniformly distributed point to approximate $f_r \cdot L_i$ stochastically selecting a region where the EM multiplied by a bilinear interpolation of the $f_r$ corner values is used.

Our method allows the use of a complex reflectance model with Monte-Carlo importance sampling even in the absence of a specific sampling scheme for the BRDF.
As with previous commented strategies, our goal is to reduce the variance of the estimator, visible as noise in the image. Unlike those, we do not make any assumption about the incoming radiance, so it is unknown. It is fortunate that, our algorithm could be easily combined with BIS and RIS, to improve the sampling of the side of the product which is considered.

In this section, we apply adaptive sampling in combination with resampled importance sampling (RIS) [JTE05]. This robust sample generation technique is based on the importance resampling and can lead to significant variance reduction over standard Monte Carlo integration for common rendering problems.

We want to find the integral of equation (2.12) applying importance resampling of the product of two functions: incoming radiance \( L_i \) (an environment map) and the BRDF \( f_r \). In this case we need to sample directions according to both functions. Let \( q \) be the desired target PDF, that is: \( q \propto L_i \cdot f_r \). The RIS estimate uses sampling importance resampling (SIR) [Tan96] to generate samples from \( q \). To get one sample \( Y \sim q \) it uses \( M \) samples from a simpler distribution \( p \). In this case \( p \propto f_r \), the samples are obtained using our quadtree method. Following that, we select only \( N \) of the original \( M \) samples, by using a probability proportional to \( L_i \). As \( M \) becomes greater, \( Y \) follows the target distribution.

A weight \( w_i \) corrects the fact that the density of the samples only approximates \( q \). The complete RIS estimator is shown in equation (5.3).

\[
L_r(u) \approx \frac{1}{N} \sum_{i}^{N} \left( f_r(u, w_i) L_i(w_i) \frac{1}{M} \sum_{j}^{M} q(s_{ij}) \right)
\]

(5.3)

To get \( N \) samples, this estimator must generate \( N \cdot M \) values of a proposal \( s_{ij} \). There is a chance of choosing a proposal many. However, if we stratify the sampling domain we avoid the sample replication problem. We show the RIS algorithm [7] below:

<table>
<thead>
<tr>
<th>Algorithm 7: code RESAMPLEDIMPORTANCESAMPLING</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Generate M proposals (( M \geq 1 )) from the source distribution ( p ), ( (s_1, \ldots, s_m) )</td>
</tr>
<tr>
<td>2. Compute a weight for each proposal, ( w(s_j) = \frac{q(s_j)}{p(s_j)} )</td>
</tr>
<tr>
<td>3. Draw N samples ( (w_1, \ldots, w_n) ), with replacement from the proposals with probability proportional to the proposal weights.</td>
</tr>
<tr>
<td>4. Apply RIS estimator to the reflectance equation.</td>
</tr>
</tbody>
</table>

University of Granada
5.5. Sampling the Product of BRDF and Lighting Functions

<table>
<thead>
<tr>
<th>Average</th>
<th>5^2</th>
<th>10^2</th>
<th>15^2</th>
<th>20^2</th>
<th>30^2</th>
<th>40^2</th>
<th>50^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>FACTORED RIS</td>
<td>2.837</td>
<td>2.373</td>
<td>2.691</td>
<td>2.583</td>
<td>2.277</td>
<td>2.333</td>
<td>2.270</td>
</tr>
<tr>
<td>ADAPTIVE</td>
<td>44.500</td>
<td>44.499</td>
<td>44.695</td>
<td>44.429</td>
<td>44.552</td>
<td>44.268</td>
<td>44.821</td>
</tr>
<tr>
<td>ADAPTIVE RIS</td>
<td>46.298</td>
<td>46.370</td>
<td>46.369</td>
<td>46.425</td>
<td>46.294</td>
<td>46.395</td>
<td>46.346</td>
</tr>
</tbody>
</table>

Table 5.16: Precomputation times of Test Scene 5 for both methods and two variants: importance sampling and resampling importance sampling.

<table>
<thead>
<tr>
<th>Average</th>
<th>5^2</th>
<th>10^2</th>
<th>15^2</th>
<th>20^2</th>
<th>30^2</th>
<th>40^2</th>
<th>50^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIFORM RIS</td>
<td>8.187</td>
<td>29.822</td>
<td>65.593</td>
<td>121.697</td>
<td>273.865</td>
<td>486.684</td>
<td>733.209</td>
</tr>
<tr>
<td>FACTORED</td>
<td>0.873</td>
<td>3.029</td>
<td>6.568</td>
<td>11.432</td>
<td>25.195</td>
<td>44.742</td>
<td>69.992</td>
</tr>
<tr>
<td>FACTORED RIS</td>
<td>1.615</td>
<td>6.620</td>
<td>14.597</td>
<td>27.800</td>
<td>60.622</td>
<td>105.511</td>
<td>408.870</td>
</tr>
<tr>
<td>ADAPTIVE</td>
<td>1.615</td>
<td>6.620</td>
<td>14.597</td>
<td>27.800</td>
<td>60.622</td>
<td>105.511</td>
<td>408.870</td>
</tr>
<tr>
<td>ADAPTIVE RIS</td>
<td>4.363</td>
<td>17.126</td>
<td>38.099</td>
<td>67.194</td>
<td>152.204</td>
<td>271.681</td>
<td>424.314</td>
</tr>
</tbody>
</table>

Table 5.17: Sampling times (in seconds) of Test Scene 5.
The image resolution is 300 × 300.

5.5.1 Adaptive Disc applied to RIS

Complex environment maps require evaluating an expensive integral over all possible lighting directions at every surface point. The RIS estimator decouples both functions, avoiding the rotation problem [WNLH06].

Next we consider the efficiency of our method when it is combined with RIS. Then using the pottery scene we compare the following three PDFs: (1) our method combined with RIS, (2) Lawrence’s factorization method combined with RIS and (3) uniform sampling with RIS. The scene considered is quite similar to Test Scene 4, but it is smaller and has a single jug. The graph plot is shown in Figure 5.16. Associated reported data is printed in Tables 5.16, 5.17 and 5.18.

Average precomputation times are 3.531 seconds for the factorized method and 45.447 for our sampling scheme. On average, uniform sampling with RIS took 527.84 seconds in sampling giving a relative error of 11.66%. The factored sampling with RIS needed 521.77 and showed an error of 11.83%. Finally, our adaptive scheme with RIS took 1478.6 seconds and gave the lowest error of 6.04%. This experiment shows the effectiveness of our adaptive sampling algorithm, whose results are better in both cases: importance sampling and resampling importance sampling. Moreover, it could be successfully combined with BIS for the same purpose.
Chapter – 5. Efficient Sampling for Generic BRDFs

<table>
<thead>
<tr>
<th>Average</th>
<th>5°</th>
<th>10°</th>
<th>15°</th>
<th>20°</th>
<th>30°</th>
<th>40°</th>
<th>50°</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIFORM RIS</td>
<td>19.08</td>
<td>13.63</td>
<td>11.73</td>
<td>10.75</td>
<td>8.43</td>
<td>9.22</td>
<td>8.80</td>
</tr>
<tr>
<td>FACTORED</td>
<td>19.22</td>
<td>17.72</td>
<td>16.75</td>
<td>16.47</td>
<td>16.77</td>
<td>15.54</td>
<td>15.98</td>
</tr>
<tr>
<td>ADAPTIVE</td>
<td>6.70</td>
<td>5.18</td>
<td>4.86</td>
<td>4.75</td>
<td>4.67</td>
<td>4.65</td>
<td>4.72</td>
</tr>
<tr>
<td>ADAPTIVE RIS</td>
<td>11.56</td>
<td>7.36</td>
<td>5.78</td>
<td>5.04</td>
<td>4.54</td>
<td>4.09</td>
<td>3.90</td>
</tr>
</tbody>
</table>

Table 5.18: The error (in %) is calculated by comparison of each image (for a BRDF sampled with a PDF and using a given number of samples) with the reference image (one for IS and another for RIS). The lowest error is achieved with our sampling scheme and resampling.

Figure 5.16: The effectiveness of a sampling scheme is shown by relating the time and relative error of a given scheme.
5.6 Adaptive Sampling of Measured Data

As in [LRR05], we are able to sample reflectance data from measurements. We use a subset of the 100 materials of Mitsubishi Electric Research Laboratories (MERL) BRDF database\(^3\), as can be seen in Figure 5.6. These images use a resolution of 450 × 200 pixels with 5\(^2\) samples. Four sampling methods have been used:

**The Uniform PDF sampling.** Randomly sampling the hemisphere of directions with equal probability is not the best option, though it is the faster (see Table 5.20). Note the noisy images of the first column in Figure 5.6.

**The Cascade CDF with uniform sampling.** Sampling by inversion of the CDF is a common technique for sampling tabular data such as the BRDF from Merl database. The uniform Cascade CDF is an approximate method which uses a resolution of 32 × 16 × 256 × 32. This implies significantly more storage requirements than the BRDF itself (as shown in Table 5.20) and becomes prohibitive in scenes with many BRDF instances. Images rendered with this method appear in the second column of Figure 5.6.

**The compact Cascade CDF method.** The compression of the CDF tables by Lawrence [LRR05] allows use of this method without so much memory penalization. Using the Douglas-Peucker [Ros97] greedy algorithm, the CDF curve is approximated and the resulting CDF table is much smaller. We have already explained this method in section 4.4.2. As can be seen in the third column of Figure 5.6 the error is quite similar to the uniform case. The main divergence of both variants is the precomputation of the CDF and PDF tables. Table 5.19 reveals that on average 2090.57 seconds are needed to compress 33MB to a little more than 1MB by starting with a resolution of 32 × 16 × 1024 × 512. The reduction of the tables’ sizes allows fast sampling by the binary search procedure.

**The Adaptive Disc sampling.** With the precomputation of 60 quadtrees we do not need as much time as the compressed Cascade CDF nor do we require so much memory (an average of 445.33 KB) as the the uniform and even compressed CDF tables. We use more sampling time to deliver 5\(^2\) samples that best contribute to the estimator. As the fourth column of images in Figure 5.6 shows, there is almost no noise with our method.

We have shown throughout this chapter that our sampling scheme is suitable for Monte Carlo applications. We allow to sample according to a general BRDF (analytical or measured one) with a low variance. In addition, we do not overload the rendering system’s memory and user guidance is not necessary.

---

\(^3\) MERL data base: <http://www.merl.com/brdf/> [last visit November 2007]
Chapter – 5. Efficient Sampling for Generic BRDFs

<table>
<thead>
<tr>
<th>Merl BRDF</th>
<th>Uniform Time (sec)</th>
<th>Uniform Memory (KB)</th>
<th>CDF A.Disc Time (sec)</th>
<th>CDF A.Disc Memory (KB)</th>
<th>Comp. CDF Time (sec)</th>
<th>Comp. CDF Memory (KB)</th>
<th>A.Disc Time (sec)</th>
<th>A.Disc Memory (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALUM-BRONZE.BINARY</td>
<td>28.80</td>
<td>2507.26</td>
<td>70.30</td>
<td>33792</td>
<td>1984.56</td>
<td>523.28</td>
<td>223.00</td>
<td>241.01</td>
</tr>
<tr>
<td>ALUMINA-OXIDE.BINARY</td>
<td>29.55</td>
<td>2571.24</td>
<td>66.88</td>
<td>33792</td>
<td>223.00</td>
<td>241.01</td>
<td>223.00</td>
<td>241.01</td>
</tr>
<tr>
<td>BEIGE-FABRIC.BINARY</td>
<td>27.57</td>
<td>2547.19</td>
<td>65.70</td>
<td>33792</td>
<td>234.16</td>
<td>272.62</td>
<td>234.16</td>
<td>272.62</td>
</tr>
<tr>
<td>BLUE-METALLIC-PAINT2.BINARY</td>
<td>30.24</td>
<td>2482.86</td>
<td>69.97</td>
<td>33792</td>
<td>2063.55</td>
<td>721.91</td>
<td>2063.55</td>
<td>721.91</td>
</tr>
<tr>
<td>BLUE-METALLIC-PAINT.BINARY</td>
<td>30.00</td>
<td>119.78</td>
<td>71.22</td>
<td>33792</td>
<td>1685.64</td>
<td>715.91</td>
<td>1685.64</td>
<td>715.91</td>
</tr>
<tr>
<td>NICKEL.BINARY</td>
<td>30.13</td>
<td>2576.81</td>
<td>71.36</td>
<td>33792</td>
<td>2057.51</td>
<td>846.15</td>
<td>2057.51</td>
<td>846.15</td>
</tr>
<tr>
<td>RED-PLASTIC.BINARY</td>
<td>30.22</td>
<td>2550.61</td>
<td>69.63</td>
<td>33792</td>
<td>557.46</td>
<td>602.82</td>
<td>557.46</td>
<td>602.82</td>
</tr>
<tr>
<td>TEFLOF.BINARY</td>
<td>29.77</td>
<td>113.56</td>
<td>64.64</td>
<td>33792</td>
<td>299.56</td>
<td>140.48</td>
<td>299.56</td>
<td>140.48</td>
</tr>
<tr>
<td>VIOLET-ACRYLIC.BINARY</td>
<td>29.69</td>
<td>2471.41</td>
<td>67.29</td>
<td>33792</td>
<td>1647.40</td>
<td>409.40</td>
<td>1647.40</td>
<td>409.40</td>
</tr>
<tr>
<td>WHITE-MARBLE.BINARY</td>
<td>30.23</td>
<td>2531.04</td>
<td>67.09</td>
<td>33792</td>
<td>488.80</td>
<td>254.64</td>
<td>488.80</td>
<td>254.64</td>
</tr>
<tr>
<td>YELLOW-PAINT.BINARY</td>
<td>31.40</td>
<td>2534.48</td>
<td>64.54</td>
<td>33792</td>
<td>283.18</td>
<td>146.47</td>
<td>283.18</td>
<td>146.47</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>29.59</td>
<td>2090.57</td>
<td>68.06</td>
<td>33792</td>
<td>1060.62</td>
<td>445.33</td>
<td>1060.62</td>
<td>445.33</td>
</tr>
</tbody>
</table>

**Table 5.19:** Precomputation requirements (time and memory) for Cascade CDF methods and our adaptive scheme.

<table>
<thead>
<tr>
<th>Merl BRDF</th>
<th>Uniform CDF</th>
<th>Uniform A.Disc</th>
<th>Comp. CDF</th>
<th>Comp. A.Disc</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALUM-BRONZE.BINARY</td>
<td>0.35</td>
<td>1.15</td>
<td>16.03</td>
<td>16.03</td>
</tr>
<tr>
<td>ALUMINA-OXIDE.BINARY</td>
<td>0.35</td>
<td>0.91</td>
<td>16.29</td>
<td>16.29</td>
</tr>
<tr>
<td>BEIGE-FABRIC.BINARY</td>
<td>0.34</td>
<td>0.94</td>
<td>4.04</td>
<td>4.04</td>
</tr>
<tr>
<td>BLUE-METALLIC-PAINT2.BINARY</td>
<td>0.37</td>
<td>1.11</td>
<td>50.55</td>
<td>50.55</td>
</tr>
<tr>
<td>BLUE-METALLIC-PAINT.BINARY</td>
<td>0.35</td>
<td>1.11</td>
<td>4.60</td>
<td>4.60</td>
</tr>
<tr>
<td>NICKEL.BINARY</td>
<td>0.37</td>
<td>1.13</td>
<td>12.31</td>
<td>12.31</td>
</tr>
<tr>
<td>RED-PLASTIC.BINARY</td>
<td>0.37</td>
<td>0.98</td>
<td>4.31</td>
<td>4.31</td>
</tr>
<tr>
<td>TEFLOF.BINARY</td>
<td>0.34</td>
<td>0.93</td>
<td>4.25</td>
<td>4.25</td>
</tr>
<tr>
<td>VIOLET-ACRYLIC.BINARY</td>
<td>0.34</td>
<td>1.13</td>
<td>38.86</td>
<td>38.86</td>
</tr>
<tr>
<td>WHITE-MARBLE.BINARY</td>
<td>0.34</td>
<td>0.97</td>
<td>12.54</td>
<td>12.54</td>
</tr>
<tr>
<td>YELLOW-PAINT.BINARY</td>
<td>0.34</td>
<td>0.91</td>
<td>3.86</td>
<td>3.86</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>0.35</td>
<td>1.02</td>
<td>15.24</td>
<td>15.24</td>
</tr>
</tbody>
</table>

**Table 5.20:** Sampling time for Uniform PDF, both Cascade CDF methods and our adaptive sampling scheme.
5.6. Adaptive Sampling of Measured Data

Figure 5.17: Every $450 \times 220$ pixel image has used 25 samples. Four sampling schemes have been used, from left to right: Uniform PDF, Cascade CDF, Compress Cascade CDF and Adaptive Disc PDF.
CHAPTER 6

Software Utilities

6.1 Tools for Global Illumination solutions

GIIG is the acronym of Grupo de Investigación de Informática Gráfica a graphics research group belonging to the Dpto. Lenguajes y Sistemas Informáticos (University of Granada). Since its initiation many people had participated in software development. Some examples are our global illumination ray tracer (GIRT) [UPTdS92, UTR+97] and an optimal density estimation system based in the sphere cache for a photon-mapping system [Las00, LURM02a, LURM02b, GUL+04, Las04, GUL+05, GUR+06]. A global illumination system involves more than just the algorithm for radiance computation. I am referring to software utilities designed for Global Illumination, as is our case. In this part of the document, I briefly comment on the tools that support the work presented in this dissertation.

6.1.1 Wannabe Amazing

This software permit us to import scenes using standard file format (i.e. Autocad .dxf, Kinetix 3D Studio, VRML, Wavefront .obj, etc.) based on mesh representation. Our Java3D multiplatform application, named Wannabe Amazing, allows us to edit the bidirectional reflectance distribution function (BRDF) properties and also to assign different instances of the function to different surfaces in the scene. By doing, this we have an extended file format that is suitable for Global Illumination algorithms. Other relevant information that we add is the vertices normals, textures coordinates, vertices colours, the irradiance value of a vertice for an incoming set of directions, the definition of extended area lights and the reflectance of a mesh object. The output scene file is exported using Wannabe Amazing in a format named GRF (Granada File) which also includes definitions of extended light sources, location and characteristics. You should consult Appendix A for a full description of this scene format and its properties.
Further, in [MUR+02] you find a detailed consideration of the characteristics and the of this tool. We have applied these 3D virtual worlds to other fields like museum games [MM04] and bioinformatics [AMS06].

For more information, snapshots, and the latest version of the application, consult the web page <http://lsi.ugr.es/~rosana> or the URL <http://lsi.ugr.es/~rosana/software/wannabe/amazing.jar> directly.

6.1.2 GRF Viewer

The scene description using the GRF file format (described in Appendix A) allows us to enrich the geometry information with other properties for example colours, irradiance, textures and reflectance properties of a surface. With Wannabe Amazing we are able to import geometries from other formats and modify them. Using the GRF Viewer we can open files with extensions *.GRF and *.IGRF. The latter
extension denotes the inclusion of irradiance information in the scene. Next, in order to visualize the scene easily, the user can choose between many drawing options for example point clouds, wireframe or solid, displaying the normals, etc. There are two implementations of the viewer. One under the Linux platform, with full functionality, and another the multiplatform version that uses Java3D (found at <http://lsi.ugr.es/~rosana/software/wannabe/grf.jar>), devoted to running anywhere.

Figure 6.2: A multiplatform GRF viewer and its counterpart in Linux.

6.1.3 A Path Tracer for Global Illumination

BPT is the acronym for Basic Path Tracer, our C++ path tracer. This is a Global Illumination solution that produces an exact solution provided that a large number of light paths are evaluated. We implement a naïve version intended to be used as a benchmark for the different sampling approaches. It is not our intention to implement a full photosimulation system. BPT is the program used in the generation of the gallery of images of Appendix B.

The execution of BPT uses some parameters at the initialization. All of this is saved in a OP file. The resulting image from the rendering can be saved in one of the most common graphics file formats. The description of the parameters and values usable in an OP file is commented on below. An example of an OP file is given in Table 6.1.

- **BPT_SCENE**: Source geometry or grf file.
• **BPT**.IMAGE_RESOLUTION: Width and height of the resulting image.

• **BPT**.OUTPUT_IMAGE: Name and type of the exported image. Type values are: Tga, Jpg, Exr, Png, Raw.

• **BPT**.VIEWER: Viewer location in world coordinates.

• **BPT**.VLOOK: Camera look at vector.

• **BPT**.PIXEL_RADIAL: Boolean value to select uniform sampling at a pixel (antialiasing technique).

• **BPT**.PIXEL_RANDOM: Boolean value to select random sampling at a pixel (antialiasing technique).

• **BPT**.PIXEL_SAMPLE_RES: Sampling resolution at a pixel (antialiasing technique).

• **BPT**.BACKGROUND: RGB background colour (float numbers in between 0 and 1).

• **BPT**.EM_BACKGROUND: Image background or environment map name.

• **BPT**.RE_NSEC_RAYS_ROOT: Square root of the number of secondary rays generated for each primary ray. This number is used in the Monte Carlo estimator.

• **BPT**.DEPTH: Maximum depth of the recursive radiance computation.

• **BPT**.PDF_NAME: Sampling scheme or PDF used by the Monte Carlo estimator. Recognized values: Diffuse, Specular, Hemisphere, Sphere, Anisotropic, Disk, Factored, CascadeCDF.

• **BPT**.PDF_LOBE_EXPONENT: Exponent value. A parameter of the cosine lobe PDF in $\Omega$ and $S^2$.

• **BPT**.PDF_ANISOTROPY: Degree of anisotropy parallel to X axis. Parameter of PDF Anisotropic.

• **BPT**.PDF_ANISOTROPY: Degree of anisotropy parallel to Y axis. Parameter of PDF Anisotropic.

• **BPT**.PDF_MEMSTAT: Boolean value to export the memory consumption of the PDF in use. Output is in `mem.op` file.

• **BPT**.QT_PRECOMPUTED: Number of precomputed quadtrees based in the angle $\theta$. Parameter of Adaptive Disk PDF.

• **BPT**.QT_ANISOTROPIC: Number of precomputed quadtrees based in the angle $\phi$. Parameter of Adaptive Disk PDF.
6.1. Tools for Global Illumination solutions

- **BPT_QT_AVGNMAX**: Average number of attempts with the rejection sampling method. Corresponds to parameter \( n_{\text{max}} \) (see section ??) of the **Adaptive Disk PDF**.

- **BPT_QT_IMGRES**: Resolution for a dense BRDF sampling in \( \mathbb{D}^2 \) domain. Parameter of **Disk PDF**. Default value is 800 and thus the resolution is \( 800 \times 800 \) \( f_r \) values.

- **BPT_QT_RELEASE**: Boolean parameter of **Adaptive Disk PDF**. If it is set, the memory resource for a tabular BRDF is freed. Default value is false.

- **BPT_PDF_FACTORS_RESU**: A parameter of the BRDF factorization. Vector2D of integers to indicate resolution of discretization on vector \( \mathbf{u} \). For example: 16 16

- **BPT_PDF_FACTORS_RESP**: A parameter of the BRDF factorization. Vector2D of integers to indicate resolution of vector \( \mathbf{p} \) discretization. For example: 32 16

- **BPT_PDF_FACTORS_JK**: A parameter of the BRDF factorization. Vector2D of integers that gives the number of JxK factors. For example: 2 1

- **BPT_PDF_FACTORS_HALF**: Boolean parameter of the BRDF factorization. When **true** a reparametrization based on the halfway angle is used.

- **BPT_PDF_FACTORS_FILE**: The factorization of the BRDF could be off-line computed and stored in TXT files. The class constructor is created with the names of these files, and the data structures are correctly loaded for its use in BRDF sampling. For example, a parameter value of **factored_plastic** will load **factored_plastic.txt** and **factored_plastic.cdf.txt**. By default its value is **none**.

- **BPT_PDF_CDF_RES**: A parameter of the **Cascade CDF** sampling method for measured BRDFs. It is a Vector2D of integers to indicate resolution of distretization of vector \( \mathbf{h} \). Default value is 256 x 32.

- **BPT_PDF_CDF_COMPRESS**: Boolean parameter of the **Cascade CDF** sampling method. When **true** a compression of the uniformly sampled CDF is carried out in order to reduce the storage requirements. Default value is **false**.

- **BPT_RAYCASTING**: Boolean parameter. If it is set to **true** the ray casting procedure activates. This allows fast computation of the ray-scene intersection. Each ray returns as a colour the normal components of the nearest intersection point.

- **BPT_DISPLAYWINDOW**: This parameter when set to **true**, opens a window with the currently displayed rendered image.

- **BPT_CHRONO_SAMPLING**: Boolean parameter. When set to **true** the program exports the total number of seconds dedicated to BRDF sampling. The value is appended to a **samplingTime** file.
Table 6.1: An example of OP option file. Some parameters depend on the PDF selected.

- **BPT\_Chrono\_Rendering**: Boolean parameter. When set to true the program exports the total number of seconds dedicated to pixel colour computation. The value is appended to a `renderingTime` file.

- **BPT\_Chrono\_Prebuild**: Boolean parameter. When set to `true` the program exports the total number of seconds dedicated to PDF data structures computation before their use. The value is appended to a `prebuildTime` file.
6.1.4 *Mini Bpt*

This is a utility for BPT path tracer. It allows the user to read the GRF scene file (and optionally its *environment map* image or EM) in order to set some parameters correctly. The interface (see Figure 6.3) allows the user many operations:

1. Three sliders, positioned at the right side of the interface, change the values of BPTViewer. The new viewer position occurs when the Run button is pressed.

2. Three sliders, positioned at the bottom of the interface, change the values of BPT_VLOOK. The new camera’s look up vector occurs when the Run button is pressed.

3. Save button exports the image into a file with extension Tga, Jpg, Exr, Png or Raw.

4. An alternative execution of this program (without command line parameters) will open a single sphere. The menu item allows selection of a BRDF model and its parameters. The interface also lets the user selects a number of samples and call the rendering method. It results in an execution of BPT over a fixed scene and without the need for an initial options file.

![Figure 6.3: With MiniBpt we have the dual possibilities of GRF and BPT. It is able to edit the BRDF and show its rendered version over a simple sphere.](image)

6.1.5 *OP File Generator*

This is an utility implemented in Java, therefore a multiplatform application, that helps the user to execute one of our programs. Many ZEUS executables, such as BPT, have an input file with program options. As their functionality increaseds the number of options, names, meanings, values and expected type of value become known only to the programmer who made them. The user is forced to read a long paged document with the description of t parameters in order to run our system.
With Wannabe OP creator, this process is highly simplified. The interface enables and disables those parameters liable to change, explaining their significance, default and expected values. A snapshot is in Figure 6.4.

![Figure 6.4](image)

**Figure 6.4**: Utility design to help a user to create options files (OP files) which is the input of our software.

### 6.1.6 EM Viewer: Environment Map Based Sampling

As we mentioned in section 5.5.1 our adaptive sampling approach can be successfully combined with resampling importance sampling in order to have a PDF which is proportional to the product of \( f_r \cdot L_i \) — the BRDF and the environment map (EM) functions as was given in algorithm [7].

A Java utility, *EM viewer* helped us in the development and visual debugging of algorithm [7]. Samples from both individual distributions, \( f_r \) and the combination \( f_r \cdot L_i \), are drawn over the spherical EM. The number of samples for both distributions \( N \) and \( M \) can be given by sliders elements, always asserting that \( M \geq N \) applies. Other parameters are the viewer position, the environment map, and its image luminance version. This utility is only for our internal use, although can be found as an executable JAR in `<http://lsi.ugr.es/~rosana/software/wannabe/emViewer.jar>`

---

3 URL: `<http://lsi.ugr.es/~rosana/software/wannabe/opcreator.jar>`

University of Granada
6.1. Tools for Global Illumination solutions

Figure 6.5: The source distribution and the target distribution of RIS are drawn over the spherical EM.
6.2 Graphics Tools for Representation and Efficient use of Real Materials

Study and production is performed in the framework of a Global Illumination solution using Monte-Carlo methods. Our main goal consists of integrate generic reflectance definitions in image synthesis algorithms. Study of different techniques and comparison of results measuring efficiency using time, error and memory usage metrics. Implementation of various tools are multiplatform and integration in a global illumination system is done for the path-tracing program under the Linux operating system.

6.2.1 Rejection Sampling of the BRDF

Sampling according to a given distribution is the main objective of the rejection sampling algorithm. This tool is intended to select and edit different distributions in order to see how rejection sampling acts with a varying number of samples. The distributions come from two families:

- Statistics distributions: Normal, Gamma, Chi-Squared, T-Student, Beta and Logarithmic Normal.

![Figure 6.6](image-url): Editable functions are used for rejection sampling.
6.2.2 A BRDF Viewer on the GPU

A simple BRDF editor included in Wannabe Amazing, can also be used as a stand-alone multiplatform application in the form of a Java jar file. This software allows to definition and control of the appearance of a material with more than fifteen different reflectance functions. The user has different plotting options, for example Polar or Cartesian coordinate systems for 2D, or 3D viewing.

In addition, another tool has been implemented to visualize the BRDF when it is applied to geometry in real time using programmable hardware or GPU. The previous reflectance functions are now implemented as GLSL (OpenGL Shading Language)\textsuperscript{1} shaders for the Windows platform. The application controls typical OpenGL illumination settings and position, but also allows the user to modify the shader’s parameters interactively. Some captured images appeared in Figures 6.7, 6.8 and 6.9.

![Figure 6.7: Many GLSL shaders allow the user to see changes in the reflectance of a surface in real time.](image)

Figure 6.8: Many GLSL shaders allow the user to see changes in the reflectance of a surface in real time.

Figure 6.9: Environment Map applied to geometry is also shown in our windows program. The diffuseness and specularity is controlled by two slides.
6.2.3 Adaptive BRDF Subdivision

This utility permits to see the unit disc domain adaptively subdivided as we change some parameters like the BRDF model, parameters of this reflectance function, the viewer position and some other internal configuration of the quadtree construction like maximum depth. The obtained direction distribution are drawn in 2D, together with the subdivision, and in 3D overprint the BRDF plot. Samples come from the execution of our algorithm, an instance of DiskPDF, otherwise the distribution comes statically from a FIT file (see section 2.3 and 6.2.4 for more details). When the previous option is used, this program also gives the projection of those vectors on the unit disc.

A snapshot is shown in Figure 6.10. This tool is available on-line at <http://lsi.ugr.es/~rosana/software/wannabe/adaptiveDiskViewer.jar>.

![Figure 6.10](image_url) 

*Figure 6.10:* We are able to change some Quadtree parameters and see how the subdivision modifies. This also changes with the BRDF parameters and the BRDF model.
6.2.4 BRDF 3D Plot

This is a command line executable that uses an instance of a FIT file, to draw in 3D the BRDF from a fixed view direction, together with a cloud of points. These points are direction vectors with a length equal to some magnitude of interest. For example this magnitude could be the BRDF evaluated at that direction or the PDF value of that sample. The text file structure is quite simple:

1. The BRDF definition using our GRF notation (see section A.1.1).

2. A fixed view direction. After the token viewer two float values correspond to angles $\theta_u$ and $\phi_u$.

3. Between the tokens begin and end, a list of vector and value, that is for example $(x_v, y_v, z_v)$ and $f_r(u, v)$, is given.

We give a simple example with 7 elements:

\[
\begin{align*}
\text{ashikhmin } & \text{ nu } = 25 \text{ nv } = 25 \text{ ks } = 0.39 \text{ kd } = 0.6 \\
\text{viewer } & (1.33763, 344.888) \\
\text{begin} \\
& (0.344659,-0.331753,0.878151) \ 1 \\
& (0.048125,0.00957266,0.998795) \ 0.85 \\
& (0.327274,-0.329171,0.885742) \ 1 \\
& (0.0407984,0.0272606,0.998795) \ 0.85 \\
& (0.312922,-0.319833,0.894308) \ 1 \\
& (0.0272606,0.0407984,0.998795) \ 0.85 \\
& (0.30379,-0.30516,0.902546) \ 1 \\
\text{end}
\end{align*}
\]

This tool has been very useful in the debugging procedure for many of the algorithm and function implementations presented in this dissertation. We could execute it standalone as a Java3D application or from the adaptive subdivision tool of the previous section. An example is given in Figures 6.11.
6.2. Graphics Tools for Representation and Efficient use of Real Materials

Figure 6.11: Set of 2500 directions plot in 3D and 2D, from the adaptive sampling in disc of the Ward-BRDF. Definition of the BRDF and the view direction comes from a FIT file.
Conclusions and Future Work

This dissertation proposes a method to generate sample directions for global illumination according to the BRDF times the cosine function using a controlled rejection sampling approach [MULG08]. Rejection sampling allows the method to sample from any arbitrary BRDF that cannot be sampled analytically. At the same time, this work proposes a hierarchical splitting of the view-dependent incident hemisphere of an isotropic BRDF, based on a probability threshold in a pre-process that provides a lower-limit on the number of samples that need to be generated at run-time before a sample is accepted. We demonstrate that this method of sample generation is more efficient for glossy and specular BRDFs than sampling from a best fit cosine lobe, a factorized BRDF using guidance, as well as for acquired BRDFs, when sampling by inversion of the CDF.

I’m just reminded of a tale about a hermit who was running through the desert screaming: - I have an answer! I have an answer! Who has a question?. Well, I have a method... Who has an application?. The Monte-Carlo method is a powerful tool for research nowadays. The range of application is enormous:

- the simulation of galactic formation,
- the solution of systems of linear equations,
- the design of nuclear reactors,
- traffic flow,
- in radiotherapy to simulate the transport of neutrons, photons, and electrons,
- in economics, a solution for dynamically modelling complex systems in business in order to support decision making and risk analysis,
- in image synthesis Monte Carlo methods have been applied to this problem in order to reduce the amount of computation. The resulting algorithms are concurrent and suited to implementation on parallel computers.
The contribution of this work is beyond a sampling method for general BRDF models. In Chapter 3 and 4 we also give a complete study of reflectance models. Using a homogeneous notation, the reader is able to see affinities between them. Furthermore, knowing the chronology and improvements in Global Illumination is of common interest. The study is complemented with the description of the sampling techniques that can be used in each case. There are few methods that could be considered generic, and appropriate for many analytical and measured BRDFs. However those alternative schemes have been implemented in order to enable comparison with our method, detailed in Chapter 5.

Another benefit of this work is a set of graphics utilities [MUR02, MM04, AMS06]. Most of them are implemented in Java so they are multiplatform. Their description was given in Chapter 6. They assist graphics designers who work with Global Illumination. The description of a BRDF and its parameters, how to assign it to an object mesh and how to sample accordingly the BRDF function, is a very common task. Some of the utilities are for general use but others are more specific and have been used throughout all of this dissertation in order mentally visualize some ideas and present our results.

Future work

The major problem with all rejection sampling approaches is their inability to produce good sampling distributions. No practical Monte Carlo algorithm would use pure random sampling; some form of stratification or QMC is always used instead. The presented method, however, is not compatible with these approaches and some work should be undertaken in the future.

There is something wrong with the mapping between disc and hemisphere, it is not area-preserving. The cosine term is taken into account when spherical presentation is mapped onto a disc, where sampling is performed. Unfortunately, the cosine term is not taken into account when the samples are re-placed on a hemisphere. This may introduce a rather substantial bias in the regions close to equator, where the cosine terms are close to zero, and where an arbitrary BRDF may have some substantial value. In order to avoid this problem, some papers employ area-preserving spherical mappings, and build quad-tree representations directly on the hemisphere [WNLH06].

Now everything has been mentioned, but my personal feelings about this work —after so much time and effort— is that it is not the end. This is the beginning of my personal career, soon new plans for research will be on the horizon of my life.

Rosana Montes

University of Granada
GRF Scene Format

A.1 General Description

An internal scene format description, named GRF (Granada File) allows the inclusion of more detailed information concerning the illumination applied to a geometric mesh model file. That is, the vertices normals, textures coordinates, vertices colours, the irradiance value of a vertex for an incoming set of directions, the definition of extended area lights and the reflectance of a mesh object. This information is presented in the scene and is divided into sections. These are:

\[
\text{scene} ::= \text{block_list} | \text{none.}
\]
\[
\text{block_list} ::= \text{block} | \text{block block_list}
| \text{none.}
\]
\[
\text{block} ::= \text{mesh_block} | \text{lightsource_block}
| \text{include_block} | \text{transform_block}
| \text{definition_block} | \text{description_block}
| \text{none.}
\]

The geometry is given by a number of meshes. Each mesh must define a list of vertices and a list of faces. However by adding further information in each section, it gives our format the following desired properties:

1. Definitions. Each section could have an assigned identifier in order to use the section by its name.
2. Denomination. Properties by name—colours, normals, irradiance, textures, etc—will clarify the description of a mesh without the need for full description of the property, just an identifier.

3. Instantiation. Use of a previously defined section, referenced by its identifier after the clause use. For example, if the geometry of a cube is already defined and we want two more cubes, is not necessary to give the list of vertices and faces again. This reduces the amount of space needed.

4. Reusable elements. Data from others GRF files could be included in the current scene. A gallery of objects and scenes could be constructed in the base of elementary GRF files. This simplifies the description of complex scenes.

5. Properties to faces. Properties such as colours or textures are not generalized to the mesh. Individual faces could exhibit different aspects. This is particularly interesting in heterogeneous surfaces since the instance of the BRDF is another property. When the property is used at mesh level, it applies equally to every face of the mesh.

6. Geometry transformations. Basic transformations like rotation, scaling and translation are used as operators over meshes, or other imported scenes.

A full description of the GRF file format, as well as many image examples and some scenes described with GRF are available online at: <http://lsi.ugr.es/~rosana>. A brief summary of the BNF notation of this scene format description is given in the following lines.

```
lightsource_block ::= light begin region [colour_definition] end.
region ::= vector3D edge edge.
edge ::= vector3D.
include_block ::= include fileName
transform_block ::= transform aTransformation begin block_list end
aTransformation ::= aTransformation basic_transform | basic_transform
basic_transform ::= scale vector3D | rotation vector3D | translation vector3D
definition_block ::= def definitions begin block_list end
definition ::= definitions | definition basic_definitions
basic_definitions ::= identifier = body_definition
body_definition ::= texture_definition | colour_definition
```
A.1. General Description

The main contribution of this scene format is the inclusion of important information relative to the computation of the radiance function. That is: normal vector, colour or irradiance per vertex, the reflectance of the surface, and the definition and position of extended area lights. Other utilities of this format are the inclusion of the information of other GRF files, definition and use of textures, together with geometrical operations over the meshes applicable to different scopes. Note that when irradiance is included, additional information for example the observer should be given. In the previous case, the extension of the file changes to igrf.

```plaintext
A.1. General Description

```

```plaintext
| tex definition | := \( \text{texture fileName} \) |
| colour definition | := \( \text{rgb colour Value} \) |
| brdf definition | := \( \text{brdf fileName} \) |
| | \( \text{brdf brdf instance .} \) |
| pdf definition | := \( \text{pdf pdf instance .} \) |
| uso | := \( \text{texture use | colour use | brdf use | pdf use .} \) |
| texture use | := \( \text{texture identifier .} \) |
| colour use | := \( \text{colour identifier .} \) |
| brdf use | := \( \text{brdf identifier .} \) |
| pdf use | := \( \text{pdf identifier .} \) |
```

The main contribution of this scene format is the inclusion of important information relative to the computation of the radiance function. That is: normal vector, colour or irradiance per vertex, the reflectance of the surface, and the definition and position of extended area lights. Other utilities of this format are the inclusion of the information of other GRF files, definition and use of textures, together with geometrical operations over the meshes applicable to different scopes. Note that when irradiance is included, additional information for example the observer should be given. In the previous case, the extension of the file changes to igrf.

```plaintext
mesh_block ::= begin mesh verticesList facesList end mesh .
verticesList ::= begin vertexs vertices end vertexs .
vertices ::= vertex | vertex vertices | none .
vertice ::= [ normal vector3D ] [ irradiancesList ]
  / uv vector2D / vector3D
irradiancesList ::= irradiance |
  begin list (ray irradiance)* end list .
ray ::= ray vector3D .
irradiance ::= irad vector3D
vector3D ::= ( value, value, value ) .
vector2D ::= ( value, value ) .
facesList ::= begin faces faces end faces .
faces ::= [ colour_definition ] [ brdf definition ]
  [ texture_definition ] face | faces faces | none .
face ::= nvertices position* |
  ( position, position* ) .
nvertices ::= integer .
position ::= integer .
```

R. Montes
Appendix– A. GRF Scene Format

description_block ::= params begin parameters_list end
parameters_list ::= dependent viewer
viewer ::= view vector3D.

sceneName ::= fileName.
fileName ::= ( value, value, value).
colourValue ::= ” Unicode characters set “.
value ::= floatValue | integer.
floatValue ::= ( decimals “.” [decimals] [exponent] [subfix] )
| ( “.” decimals [exponent] [subfix] )
| ( decimals [exponent] [subfix] ).
decimals ::= 0..9.
exponent ::= “ e” [ “+” | “-” ] decimals.
subfix ::= “f” | “d”.
integer ::= 0..9.
identifier ::= a..z, A..Z, ,0..9, $,. 
comments ::= // string.
none ::= ;

A.1.1 List of Reflectance Functions’ BNF Description and their Parameters

brdf_definition ::= brdf fileName
| brdf brdf_instance.

brdf_instance ::= PerfectDiffuse | PerfectSpecular
| Phong | Blinn | Torrance-Sparrow
| Cilindrico-PF | HTSG |
| Ward | Lewis | Schlick
| Oren-Nayar | Strauss
| Lafortune | Shirley
| Minnaert | BeardMaxwell
| brdfs_combination | MerlDB_BRDF.

PerfectDiffuse ::= diffuse kd=value.
PerfectSpecular ::= specular ks=value.
Phong ::= phong n=value ks=value kd=value.
Blinn ::= blinn n=value ks=value kd=value.
Torrance-Sparrow ::= torrance rough=value nreal=value
nimag=value ks=value kd=value.
A.1. General Description

Cilindrico-PF ::= \( \text{poulin } d=\text{value } h=\text{value } n=\text{value } ks=\text{value } kd=\text{value.} \)

HTSG ::= \( \text{he sigma=\text{value } tau=\text{value } lambda=\text{value.}} \)

Ward ::= \( \text{ward } sx=\text{value } sy=\text{value } ks=\text{value } kd=\text{value.} \)

Lewis ::= \( \text{lewis } n=\text{value } ks=\text{value } kd=\text{value.} \)

Schlick ::= \( \text{schlick } SC=\text{value } Sr=\text{value } Sp=\text{value } Ly=value_{logico} [\text{DC}=\text{value } Dr=\text{value } Dp=\text{value.}} \)

Oren-Nayar ::= \( \text{oren } s=value \ r=value. \)

Strauss ::= \( \text{strauss } s=value \ m=value \ ks=value \ kd=value. \)

Lafortune ::= \( \text{lafortune } Cx=value \ Cy=value \ Cz=value \ n=value \ kp=value. \)

Shirley ::= \( \text{coupled } Rm=value. \)

Minnaert ::= \( \text{minnaert } k=value \ kd=value. \)

BeardMaxwell ::= \( \text{beardmaxwell } omega=value \ tau=value \ nreal=value \ nimag=value \ Rs=value \ Rd=value \ Ru=value. \)

brdfs_combination ::= \( \text{combine } base_{BRDF}, \ base_{BRDF}*. \)

MerlDB_BRDFs ::= \( \text{measured } \text{file}\text{Name.} \)

base_BRDF ::= \( \text{value } \text{identifier } | \text{none.} \)

A.1.2 List of Probability Density Functions’ BNF Description and their Parameters

pdf_definition ::= | \( pdf \ pdf\text{.instance.} \)

pdf_instance ::= Uniform_PDF \| Specular_PDF \| Sphere_PDF \| Hemisphere_PDF \| Adaptive_PDF \| Anisotropic_PDF \| Rejection_PDF \| Factored_PDF \| Cascade_CDF

Uniform_PDF ::= \( \text{diffuse.} \)

Specular_PDF ::= \( \text{specular.} \)

Sphere_PDF ::= \( \text{sphere } exp=value. \)

Hemisphere_PDF ::= \( \text{hemisphere } exp=value. \)

Adaptive_PDF ::= \( \text{adaptive } nQT=value \ nAnisoQT=value. \)

Anisotropic_PDF ::= \( \text{anisotropic } nu=value \ nv=value. \)

Rejection_PDF ::= \( \text{reject.} \)

Factored_PDF ::= \( \text{factored } uTh=value \ uPh=value \ pTh=value \ pPh=value \ J=value. \)

R. Montes
A.1.3 A First Example

This is the description using the GRF of a three dimensional cube.

```plaintext
// Wannabe Amazing sep 01, 2001, 12:58
// Canvas: Width 585 Height 415 Scale 0.2
// Box : width=2.0 height=1.0 length=0.5
// nvertexs=4 nfaces=6
// Bounding Box :
//   Lower=(-2.797451055, -3.276146411, -2.716787639)
//   Upper=(2.582348612, 2.1036532564, 2.6630120283)

def cromo = brdf ward sx=0.05 sy=0.3 ks=0.05 kd=0.4 begin
  begin mesh
    brdf cromo
    begin vertexs
      normal(0.0, 0.0, 1.0)
      (2.0, -1.0, 0.5)
      normal(0.0, 0.0, 1.0)
      (2.0, 1.0, 0.5)
      normal(0.0, 0.0, 1.0)
      (-2.0, 1.0, 0.5)
      normal(0.0, 0.0, 1.0)
      (-2.0, -1.0, 0.5)
      normal(0.0, 0.0, -1.0)
      (-2.0, -1.0, -0.5)
      normal(0.0, 0.0, -1.0)
      (-2.0, 1.0, -0.5)
      normal(0.0, 0.0, -1.0)
      (2.0, 1.0, -0.5)
      normal(0.0, 0.0, -1.0)
      (2.0, -1.0, -0.5)
    end vertexs
  end mesh
end
```

University of Granada
end vertexs
begin faces
  rgb(0.3,0.8,1.0)
  (0,1,2,3)
  (4,5,6,7)
  (7,6,1,0)
  (3,2,5,4)
  (1,6,5,2)
  (3,4,7,0)
end faces
end mesh
end

A.1.4 A Second Example

//ply2grf dragon200K.ply 20/07/2006
include "light.grf"
transform
  rotation (1.570796,0,0)
  translation(0,0.5,0)
begin
  include "light.grf"
end
transform
  rotation (1.570796,0,0)
  translation(0,0.75,-0.75)
begin
  include "wall.grf"
end
def
dragon = brdf ward sx=0.3674 sy=0.7765 ks=0.05 kd=0.4
suelo = brdf lafortune cx=-1 cy=-1 cz=1 n=20.0 kp=1
begin
  include "floor.grf"
transform
  scale (3, 3, 3)
  translation (0,-0.15, 0)

R. Montes
begin
    include "dragon200K.grf"
end
APPENDIX B

Gallery

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFS</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cornell Box</td>
<td>605</td>
<td>1000</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>N_SAMPLES</td>
<td>900</td>
<td>Sampling Time</td>
<td>397.402 secs.</td>
<td></td>
</tr>
</tbody>
</table>

Figure B.1: Importance sampling adaptive disc method applied to Cornell.
Appendix– B. Gallery

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planta</td>
<td>1184</td>
<td>1372</td>
<td>1</td>
<td>49</td>
</tr>
<tr>
<td>N.Samples</td>
<td>400</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling time</td>
<td></td>
<td>38.91</td>
<td>secs.</td>
<td></td>
</tr>
</tbody>
</table>

Figure B.2: Importance sampling adaptive disc method applied to Planta.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Living room 1</td>
<td>3311</td>
<td>4608</td>
<td>7</td>
<td>68</td>
</tr>
<tr>
<td>N.Samples</td>
<td>400</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling time</td>
<td></td>
<td>447.839</td>
<td>secs.</td>
<td></td>
</tr>
</tbody>
</table>

Figure B.3: Importance sampling adaptive disc method applied to Living room 1.

University of Granada
<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>Brdfs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Living room 1</td>
<td>3311</td>
<td>4608</td>
<td>7</td>
<td>68</td>
</tr>
<tr>
<td>N.Samples</td>
<td>400</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure B.4:** A detail of **Living room 1**.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>Brdfs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleScene</td>
<td>5086</td>
<td>7776</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>N.Samples</td>
<td>400</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure B.5:** Importance sampling *adaptive disc* method applied to **SimpleScene**.
<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BagueScene</td>
<td>6179</td>
<td>8744</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>N. Samples</td>
<td>400</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure B.6**: Importance sampling *adaptive disc* method applied to *BagueScene*.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buzón</td>
<td>9362</td>
<td>18398</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>N. Samples</td>
<td>400</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure B.7**: Importance sampling *adaptive disc* method applied to *Buzón*.  

University of Granada
Scene Vertices Faces Brdfs Meshes
Helicóptero 15199 24060 1 45
N.Samples 2500 Sampling Time 1179.44 secs.

Figure B.8: Importance sampling adaptive disc method applied to Helicóptero.

Scene Vertices Faces Brdfs Meshes
Motorbike 20001 29948 7 12
N.Samples 400 Sampling Time 196.55 secs.

Figure B.9: Importance sampling adaptive disc method applied to Motorbike.
Appendix—B. Gallery

### Figure B.10: RIS with adaptive disc method applied to Multisphere.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multisphere</td>
<td>12534</td>
<td>18696</td>
<td>4</td>
<td>84</td>
</tr>
<tr>
<td>N.Samples</td>
<td>10000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling Time</td>
<td>34178.9 secs.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Figure B.11: Multisphere with aluminia, violet-acrylic, blue-metallic-paint2 and red-plastic BRDFs.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multisphere</td>
<td>12534</td>
<td>18696</td>
<td>5</td>
<td>84</td>
</tr>
<tr>
<td>N.Samples</td>
<td>200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling Time</td>
<td>553.81 secs.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure B.12: Importance sampling *adaptive disc* method applied to *Chevrolet*.

Figure B.13: Importance sampling *adaptive disc* method applied to *PlaneScene*. 
### Appendix B. Gallery

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFS</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roadway</td>
<td>63576</td>
<td>97903</td>
<td>11</td>
<td>221</td>
</tr>
<tr>
<td>N. Samples</td>
<td>2500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling Time</td>
<td>249.494</td>
<td></td>
<td></td>
<td>secs.</td>
</tr>
</tbody>
</table>

**Figure B.14:** Importance sampling *adaptive disc* method applied to **Roadway**.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFS</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Living room 2</td>
<td>38233</td>
<td>78536</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>N. Samples</td>
<td>400</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling Time</td>
<td>332.19</td>
<td></td>
<td></td>
<td>secs.</td>
</tr>
</tbody>
</table>

**Figure B.15:** Importance sampling *adaptive disc* method applied to **Living room 2**.
### Table B.4: Meshes and BRDFs for Happy Budha

<table>
<thead>
<tr>
<th>Scene</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happy Budha</td>
<td>32328</td>
<td>67240</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>N.Samples</td>
<td>2500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling Time</td>
<td>3114.98 secs.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure B.16:** Sampling importance resampling *adaptive disc* method applied to Happy Budha using Poulin-Fournier BRDF.
### Appendix B. Gallery

![Dragon](image_url)

<table>
<thead>
<tr>
<th>Scenes</th>
<th>Vertices</th>
<th>Faces</th>
<th>BRDFs</th>
<th>Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dragon</td>
<td>100258</td>
<td>202524</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

| N.Samples | 400       | Sampling Time | 174.975 | secs.  |

Figure B.17: Importance sampling *adaptive disc* method applied to *Dragon*. 

University of Granada
Figure B.18: Importance sampling *adaptive disc* method applied to *Woodman*.

Figure B.19: Importance sampling *adaptive disc* method applied to *Emotico*. 
Appendix– B. Gallery

Scene Vertices Faces Brdfs Meshes
Phlegmatic dragon 2889278 480052 5 12

N.Samples 10000 Sampling Time 2646.73 secs.

Figure B.20: Importance sampling *adaptive disc* method applied to Phlegmatic dragon.

Scene Vertices Faces Brdfs Meshes
Phlegmatic dragon 311191 523234 8 15

N.Samples 10000 Sampling Time 8344.4 secs.

Figure B.21: Importance sampling *adaptive disc* method applied to Phlegmatic dragon & PlaneScene.

University of Granada
Scene Vertices Faces Brdfs Meshes
Bunny 35947 69451 3 3

N. Samples 400 Sampling Time 444.296 secs.

Figure B.22: Importance sampling *adaptive disc* method applied to *Bunny*.

Scene Vertices Faces Brdfs Meshes
Pyramob 18876 31460 1 1

N. Samples 5000 Sampling Time 1311.3 / 1901.55 secs.

Figure B.23: Importance sampling *adaptive disc* method applied to *Pyramob* with acquired *yellow-paint* BRDF.
Appendix– B. Gallery

Scene | Vertices | Faces | BRDFs | Meshes
--- | --- | --- | --- | ---
Fertility | 112681 | 212834 | 1 | 4

N.Samples 200 Sampling Time 22.298 secs.

Figure B.24: Importance sampling adaptive disc method applied to Fertility.

Scene | Vertices | Faces | BRDFs | Meshes
--- | --- | --- | --- | ---
Fertility | 112681 | 212834 | 1 | 4

N.Samples 200 Sampling Time 224.101 secs.

Figure B.25: RIS with adaptive disc method applied to Fertility.
Figure B.26: Importance sampling *adaptive disc* method applied to *Cube*. Left image use Phong-BRDF. Right image use acquired *nickel* BRDF.
## References


Models of light reflection for computer synthesized pictures.

Texture and reflection in computer generated images.

[BS63] Peter Beckmann and Andre Spizzichino.
The scattering of electromagnetic waves from rough surfaces.
1963.

[BSW00] Philippe Bekaert, Mateu Sbert, and Yves D. Willems.
Weighted importance sampling techniques for monte carlo radiosity.

[Bue01] Juan M. Bueno.
Reversibilidad óptica en polarización: aplicación al ojo humano.

[CETC06] David Cline, Parris K. Egbert, Justin F. Talbot, and David L. Cardon.
Two stage importance sampling for direct lighting.

[CJAMJ05] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen.
Wavelet importance sampling: efficiently evaluating products of complex functions.

[CP85] F. Clarke and D. Parry.
Helmholtz reciprocity: Its validity and application to reflectometry.

Distributed ray tracing.

A reflectance model for computer graphics.

*Radiosity and Realistic Image Synthesis*.
Reflectances and texture of real-world surfaces.

[Dut] Philip Dutre.
Global illumination compendium.
On-line at: <http://www.cs.kuleuven.ac.be/~phil/GI>
[Last visit: March 6th, 2007].

Computer graphics: principles and practice (2nd ed.).
Addison-Wesley. Massachusetts, 1990.

A goniometric optical scatter instrument for bidirectional reflectance
distribution function measurements with out-of-plane and polarimetry
capabilities.

Goniometric optical scatter instrument for out-of-plane ellipsometry
measurements.

Random number generation and Monte Carlo methods. (2nd ed.).

[GH03] Xavier Granier and Wolfgang Heidrich.
A simple layered rgb brdf model.


[GSCH93] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan.
Wavelet radiosity.

Modelling the interaction of light between diffuse surfaces.

[GUL+04] Rubén García, Carlos Ureña, Miguel Lastra, Rosana Montes, and Jorge
Revelles.
Interactive global illumination for quasi-static scenes.
In CGI’2004 Short Communications Proceedings, Creta (Greece), 2004.
REFERENCES

[GUL+05] Rubén García, Carlos Ureña, Miguel Lastra, Rosana Montes, and Jorge Revelles.
Optimizaciones en estimación de densidades para iluminación global.
CEDI’05, Thomson, September 2005.

[GUR+06] Rubén García, Carlos Ureña, Jorge Revelles, Miguel Lastra, and Rosana Montes.
Density estimation optimizations for global illumination.
In WSCG’2006 Short Communications Proceedings, 2006.

[Hel25] H.V. Helmholtz.
Treatise on physiological optics.
1(17), 1925.

A fast and accurate light reflection model.

Reflection from layered surfaces due to subsurface scattering.

A comprehensive physical model for light reflection.

[JTE05] David Cline Justin Talbot and Parris K. Egbert.
Importance resampling for global illumination.
In Rendering Techniques 2005 (Eurographics Symposium on Rendering), pages 139–146, June 2005.

Anisotropic reflectance models.

The rendering equation.

[KC01] Szirmay-Kalos László Kelemen Csaba.
A microfacet based coupled specular-matte brdf model with importance sampling.

University of Granada
REFERENCES


Hardware rendering with bidirectional reflectances. 

[Las00] Miguel Lastra. 
Síntesis eficiente de imágenes fotorealistas por simulación del modelo de partículas de la luz. 

[Las04] Miguel Lastra. 
Stochastic Rendering Techniques for Complex Environments. 

Making shaders more physically plausible. 

[LFTG97] Eric P. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. 
Non-linear approximation to reflectance functions. 

[LRR04] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. 
Efficient importance sampling using a factored representation. 

[LRR05] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. 
Adapative numerical cumulative distribution functions for efficient importance sampling. 
In Eurographics Symposium on Rendering, 2005.

[LURM02a] Miguel Lastra, Carlos Ureña, Jorge Revelles, and Rosana Montes. 
Density stimation on the tangent plane for radiosity. 
In 1er Simposium Iberoamericano de Informatica Grafica (SIACG’2002), Guimaraes, Portugal, July 2002.

[LURM02b] Miguel Lastra, Carlos Ureña, Jorge Revelles, and Rosana Montes. 
A particle-path based method for monte carlo density estimation. 
In Eurographics Workshop on Rendering. Poster presentations, Pisa, Italy, June 2002.

[LW94] Eric P. Lafortune and Yves D. Willems. 
Using the modified phong reflectance model for physically based rendering.
REFERENCES


[MAA01] Michael D. McCool, Jason Ang, and Anis Ahmad.
Homomorphic factorization of BRDFs for high-performance rendering.

Bidirectional reflectance model validation and utilization.

Probability trees.

[Mil84] Hoffman Miller.
Illumination and reflection maps: Simulated objects in simulated and real environments.

The reciprocity principle in lunar photometry.

[MM04] Rosana Montes and Francisco J. Melero.
A multimedia 3d game for museums.

A data-driven reflectance model.

[MULG08] Rosana Montes, Carlos Ureña, Miguel Lastra, and Rubén García.
Generic brdf sampling: A sampling method for global illumination.

[MUR+02] Rosana Montes, Carlos Ureña, Jorge Revelles, Miguel Lastra, and Rubén García.
Wannabe amazing, una herramienta de trabajo para iluminación global.
In Jornadas Regionales de Informática Gráfica, pages 147–158, Jaen, Spain, February 2002.
REFERENCES

[NN96] Laszlo Neumann and Attila Neumann. 
A new class of brdf models with fast importance sampling. 

[NN98] Laszlo Neumann and Attila Neumann. 
New simple reflectance models for metals and other specular materials. 

Generalization of lambert’s reflectance model. 

Visual appearance of matte surfaces. 

[PF90] Pierre Poulin and Alain Fournier. 
A model for anisotropic reflection. 

Illumination for computer generated pictures. 

[Ros97] Paul L. Rosin. 
Techniques for assessing polygonal approximations of curves. 

A new change of variables for efficient brdf representation. 

[Sbe96] Mateu Sbert. 
The use of global random directions to compute radiosity — Global Monte Carlo techniques. 

[Sch93] Christophe Schlick. 
A customizable reflectance model for everyday rendering. 
In Fourth Eurographics Workshop on Rendering, Eurographics’93, pages 73–84, June 1993.

R. Montes
REFERENCES

[Sch94a] Christophe Schlick.
A fast alternative to phong’s specular model.

[Sch94b] Christophe Schlick.
An inexpensive brdf model for physically-based rendering.

[Shi96] Peter Shirley.
Monte carlo methods for rendering.

[SHSL97] Peter Shirley, Helen Hu, Brian Smits, and Eric Lafortune.
A partitioners’ assessment of light reflection models.

[Sta01] Jos Stam.
An illumination model for a skin layer bounded by rough surfaces.

[Str90] Paul S. Strauss.
A realistic lighting model for computer animators.


Off-specular peaks in the directional distribution of reflected thermal radiation.

Theory for off-specular reflection.
In *Journal of the Optical Society of America*, number 9, September 1967.

[UPTdS92] Carlos Ureña, Jose Parets, Juan C. Torres, and Vicente del Sol.
An object-oriented aproach to ray tracing image synthesis implementations.

[UTR+97] Carlos Ureña, Juan C. Torres, Jorge Revelles, Pedro Cano, Vicente del Sol, and Marcelino Cabrera.
Girt: Un sistema orientado a objetos para simuracion precisa de la iluminacion.
REFERENCES

[Ve97] Eric Veach. 
Robust Monte Carlo methods for light transport simulation. 

[VG94] Eric Veach and L. Guibas. 
Bidirectional estimators for light transport. 

[Wa05] Bruce Walter. 
Notes on the ward brdf. 

Measuring and modelling anisotropic reflection. 

Predicting reflectance functions from complex surfaces. 

Irradiance Gradients. 

An improved illumination model for shaded display. 

A brdf database employing the beard-maxwell reflection model. 

[WNLH06] Rui Wang, Ren Ng, David Luebke, and Greg Humphreys. 
Efficient wavelet rotation for environment map rendering. 
In Eurographics Symposium on Rendering, 2006.

A ray tracing solution for diffuse interreflection. 