Homomorphic Factorization of BRDF-based Lighting Computation

Lutz Latta and Andreas Kolb
Faculty of Media-Information Science
University of Applied Sciences Wedel, Wedel, Germany

Abstract

Several techniques have been developed to approximate Bidirectional Reflectance Distribution Functions (BRDF) with acceptable quality and performance for realtime applications. The recently published *Homomorphic Factorization* by McCool et al. is a general approximation approach that can be used with various setups and for different quality requirements.

In this paper we propose a new technique based on the Homomorphic Factorization. Instead of approximating the BRDF, our technique factorizes the full lighting computation of an isotropic BRDF in a global illumination scenario. With this method materials in complex lighting situations can be simulated with only two textures by using commonly available computation capabilities of current graphics hardware.

The new technique can also be considered as a generalized approach to several environment map prefiltering techniques. Existing prefiltering techniques are usually limited to specific BRDFs or require advanced hardware capabilities like 3D texturing. With the factorization only common 2D textures are required.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, Shading, Shadowing, and Texture I.3.3 [Computer Graphics]: Picture/Image Generation

Keywords: Illumination, Reflectance & Shading Model, Rendering, Rendering Hardware, Texture Mapping

1 Introduction

Several techniques for integrating more realistic material and lighting models in realtime rendering based on *Bidirectional Reflectance Distribution Functions (BRDF)* have been developed in recent years.

These approaches build upon the BRDF lighting function

$$\mathbf{L}_o(\hat{\boldsymbol{\omega}}_o) = \int_\Omega BRDF\left(\hat{\boldsymbol{\omega}}_i, \hat{\boldsymbol{\omega}}_o\right) \mathbf{L}_i(\hat{\boldsymbol{\omega}}_i) \cos(\theta_i)\, d\hat{\boldsymbol{\omega}}_i \qquad (1)$$

which describes the relation between incoming light $\mathbf{L}_i$ from direction $\hat{\boldsymbol{\omega}}_i = (\theta_i, \phi_i)$ over the hemisphere Ω and the outgoing light $\mathbf{L}_o$ in direction $\hat{\boldsymbol{\omega}}_o$. The symbol θ represents the elevation angle, whereas ϕ stands for the angle in the (local) tangent plane. According to Lambert's law $\mathbf{L}_i(\hat{\boldsymbol{\omega}}_i)\cos(\theta_i)d\hat{\boldsymbol{\omega}}_i$ describes the amount of light per unit area arriving at the point on the surface of the illuminated object. Note that $\hat{\boldsymbol{\omega}}_o$ is taken from the hemisphere Ω since for viewing directions below the horizon no reflection occurs.

The computation required for Equation (1) is quite complex and can not be implemented efficiently during realtime rendering.

A major research effort focuses on the approximation of the high-dimensional BRDF using several lower dimensional textures (see e.g. [Kautz and McCool 1999a; McCool et al. 2001]). These techniques assume a relatively simple lighting environment consisting of a few point light sources, reducing Equation (1) to a discrete sum. Usually several textures are generated that approximate certain materials for a single point light source. Utilizing standard texture mapping techniques, these textures realize a lookup table functionality for BRDF values. However, once for each light source the reconstruction of the BRDF and mapping of the resulting textures is required. Thus these approaches are not suitable for complex lighting environments.

Other publications, e.g. [Heidrich and Seidel 1999; Kautz and McCool 2000], discuss the integration of more complex lighting environments. They use a simplified version of the BRDF itself in order to be able to pre-compute environment maps that are used to approximate an object's reflectance in realtime. The computation of these environment maps usually incorporates filtering applied to the initial environment lighting data. The filter actually replaces the complex and general BRDF description of an object's reflectance behavior, thereby restricting the BRDF to special function classes.

This paper introduces a technique that combines both major research directions stated above. Our approach uses the BRDF separation introduced by [McCool et al. 2001], but in contrast to their original simplification of Equation (1) we separate the complete lighting function. The major difference to [McCool et al. 2001] is the use of global parameterizations instead of local ones. We introduce several possible global parameterizations to be used in the approximation of the lighting function. Our technique allows the integration of arbitrary BRDFs in arbitrary lighting environments in realtime applications using standard textures mapping.

The new technique is currently restricted to isotropic BRDFs and static lighting environments. Compared to prefiltering techniques our method does involve more computational effort in the pre-processing phase.

In section 2 we give an overview of existing works on BRDF approximation and prefiltering techniques. Section 3 describes our approach in detail. First we summarize McCool et al.'s Homomorphic Factorization approach which we use as approximation technique. Then we give a detailed description of the new technique to approximate the lighting function (1), including aspects of parameterizations, sampling the lighting function and rendering. Our results are presented in section 4, and section 5 states several future research directions in this context.

2 Prior Work

In this section we give a brief overview of the various approaches to make the BRDF-based lighting Equation (1) accessible to realtime rendering. These techniques either restrict the lighting environment

Copyright © 2002 by the Association for Computing Machinery, Inc.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212-869-0481 or e-mail permissions@acm.org.
© 2002 ACM 1-58113-521-1/02/0007 $5.00

and focus on the BRDF approximation (section 2.1) or use simple classes of BRDF functions and focus on the integration of more complex lighting environments using prefiltering techniques (section 2.2). Additionally in section 2.3 some recent approaches in the area of *surface light fields* that relate to our results are summarized.

2.1 Texture based BRDF Approximation

A texture based BRDF approximation technique applicable in realtime rendering was developed by [Heidrich and Seidel 1999] to pre-compute several analytical BRDFs into textures.

Based on this work, [Kautz and McCool 1999a] propose a generalized approach for the approximation of arbitrary BRDFs. They expand a technique introduced by [Fournier 1995] to approximate the BRDF with a sum of products of two 2D functions using *Singular Value Decomposition (SVD)*. The function values are stored in textures $t_{j,1}, t_{j,2}$ yielding the following approximation:

$$BRDF(\hat{\omega}_i, \hat{\omega}_o) \approx \sum_{j=1}^{J} t_{j,1}\left(\pi_1(\hat{\omega}_i, \hat{\omega}_o)\right) t_{j,2}\left(\pi_2(\hat{\omega}_i, \hat{\omega}_o)\right) \quad (2)$$

where π_1, π_2 appropriately map the four-dimensional parameter space onto a two-dimensional one. For $J = 1$ [Kautz and McCool 1999a] propose the accelerated *Normalized Decomposition* technique.

The restriction to pairs of factors with the same parameters is removed by the *Homomorphic Factorization* algorithm introduced by [McCool et al. 2001]. They approximate the BRDF by the product of J two-dimensional functions t_j, each pre-computed into a texture. Their general parameterization can be stated as

$$BRDF(\hat{\omega}_i, \hat{\omega}_o) \approx \prod_{j=1}^{J} t_j\left(\pi_j(\hat{\omega}_i, \hat{\omega}_o)\right) \quad (3)$$

The approximation is computed by taking the logarithm on both sides of the equation, and then solving the resulting linear problem with an iterative solver.

2.2 Prefiltering

Techniques for the approximation of complex lighting effects are often based on environment maps. Generally, these methods generate a 2-dimensional filter kernel from a BRDF. In many cases the BRDF is isotropic and radially symmetric w.r.t. the reflected viewing direction. The application of the filter to an environment map approximates the lighting computation with the illumination based on the original environment map. For an overview of the most common prefiltering techniques see [Kautz et al. 2000].

[Heidrich and Seidel 1999] build upon the well known Phong illumination model [Phong 1975]. They use Phong lobes to approximate isotropic BRDFs. Phong lobes are radially symmetric w.r.t. the reflected viewing direction and do not depend upon the view elevation. This technique stores the diffuse and specular terms in separate textures. Additionally a Fresnel term is included at runtime.

[Kautz and McCool 2000] developed a prefiltering technique that uses arbitrary shaped, radially symmetric lobes. Using a single lobe approximation, 3D textures are needed to store the approximated lighting functions yielding *glossy environment maps*. For simple BRDFs it is also possible to use a 2D texture. Furthermore an approximation technique based on several lobes is introduced, where each lobe has to be rendered separately.

[Cabral et al. 1999] propose a technique using a set of prefiltered environment maps. Each of the maps is computed for a fixed point of view. For an arbitrary viewpoint the precomputed environment maps of the three closest points of view are warped and blended to compute an approximation of the environment map used for rendering.

[Kautz et al. 2000] introduced a unified approach to the prefiltering techniques proposed by [Miller and Hoffman 1984], [Heidrich and Seidel 1999], [Kautz and McCool 2000] and [Cabral et al. 1999]. Based on this generalized point of view an approximation of the anisotropic BRDF by [Banks 1994] is given. Additionally [Kautz et al. 2000] introduce a hardware accelerated approach to realize the different prefiltering processes.

2.3 Surface Light Fields

In contrast to a BRDF, a surface light field $f(r,s,\theta,\phi)$ describes the radiance of a surface point with parameters (r,s) in viewing direction $\hat{\omega} = (\theta,\phi)$. One major task in the context of light fields is rendering an object, for which f is known, in realtime (e.g. see [Levoy and Hanrahan 1996]).

[Chen et al. 2001] propose a texture based approximation which is very similar to the BRDF approximation technique introduced by [Kautz and McCool 1999a]. They also use a SVD to approximate the light field data yielding

$$f(r,s,\theta,\phi) \approx \sum_{j=1}^{J} t_{j,1}(r,s) t_{j,2}(\theta,\phi) \quad (4)$$

which is, in spite of the parameterization, the same as Equation (2).

Our approach has something in common with Chen et al.'s procedure since they also apply a BRDF approximation technique to a different lighting problem.

3 Lighting Function Factorization

In this section we describe in detail our technique to approximate the lighting function (LF). We first discuss the Homomorphic Factorization (HF) technique introduced by [McCool et al. 2001] and describe the modifications needed to approximate the complete lighting function. Further sections focus on parameterization aspects (section 3.2), the sampling of the lighting function (section 3.3) and rendering details (section 3.4).

The main goal of our approach is the factorization of the complete lighting function

$$\mathbf{L}_o(\hat{\mathbf{v}}, \hat{\mathbf{n}}, \hat{\mathbf{t}}) = \int_{\Omega} BRDF\left(\hat{\boldsymbol{\omega}}(\hat{\mathbf{l}}, \hat{\mathbf{n}}, \hat{\mathbf{t}}), \hat{\boldsymbol{\omega}}(\hat{\mathbf{v}}, \hat{\mathbf{n}}, \hat{\mathbf{t}})\right) \mathbf{L}_i(\hat{\mathbf{l}}) \left(\hat{\mathbf{n}} \cdot \hat{\mathbf{l}}\right) \, d\hat{\mathbf{l}} \quad (5)$$

where $\mathbf{L}_i(\hat{\mathbf{l}})$ describes the amount of incoming light from direction $\hat{\mathbf{l}}$. The two vectors $\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}$ describe the normal and a tangent of the surface. The function $\hat{\boldsymbol{\omega}}(\hat{\mathbf{a}}, \hat{\mathbf{n}}, \hat{\mathbf{t}})$ computes the spherical coordinates for a vector $\hat{\mathbf{a}}$ relative to the coordinate frame defined by $\{\hat{\mathbf{n}}, \hat{\mathbf{t}}, \hat{\mathbf{n}} \times \hat{\mathbf{t}}\}$. All other vectors are given in world space.

Note that in contrast to the representation of the BRDF in Equation (1), we use world coordinates in Equation (5). Thus we have to assume that all vectors lie on the whole sphere not only the hemisphere Ω. The integration is again on the hemisphere since $\hat{\mathbf{n}} \cdot \hat{\mathbf{l}} \geq 0$ is a meaningful assumption.

Applying the Homomorphic Factorization algorithm with the lighting function for isotropic BRDFs, simplifies Equation (5) since isotropic BRDFs do not depend on the tangent $\hat{\mathbf{t}}$. Replacing $\hat{\mathbf{t}}$ by a function $\hat{\mathbf{t}}(\hat{\mathbf{n}})$, which computes a valid tangent for a given normal yields

$$\mathbf{L}_o(\hat{\mathbf{v}},\hat{\mathbf{n}}) = \int_\Omega BRDF\left(\hat{\boldsymbol{\omega}}(\hat{\mathbf{l}},\hat{\mathbf{n}},\hat{\mathbf{t}}(\hat{\mathbf{n}})),\hat{\boldsymbol{\omega}}(\hat{\mathbf{v}},\hat{\mathbf{n}},\hat{\mathbf{t}}(\hat{\mathbf{n}}))\right)\mathbf{L}_i(\hat{\mathbf{l}})\left(\hat{\mathbf{n}}\cdot\hat{\mathbf{l}}\right)\,d\hat{\mathbf{l}} \tag{6}$$

which depends only on two directional vectors: the viewing direction and the surface normal. Both can be expressed in spherical coordinates with two angles. Therefore the function has four degrees of freedom, just like a position-invariant BRDF.

We compute the tangent vector as

$$\hat{\mathbf{t}}(\hat{\mathbf{n}}) = \frac{\hat{\mathbf{x}} - (\hat{\mathbf{x}}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}}}{\|\hat{\mathbf{x}} - (\hat{\mathbf{x}}\cdot\hat{\mathbf{n}})\,\hat{\mathbf{n}}\|}$$

where $\hat{\mathbf{x}}$ denotes the horizontal direction in viewing coordinates. This leads to singularities as $\hat{\mathbf{n}} \approx \hat{\mathbf{x}}$, placing them in visually unimportant regions.

3.1 Approximation

The approximation of the lighting function is based on the Homomorphic Factorization approach introduced by [McCool et al. 2001]. Analogous to the HF approximation in Equation (3) we approximate the lighting function in Equation (6) in world coordinates using normal and reflection vectors. We derive the general parameterization for isotropic materials as

$$\mathbf{L}_o(\hat{\mathbf{v}},\hat{\mathbf{n}}) \approx \prod_{j=1}^{J} t_j\left(\pi_j(\hat{\mathbf{v}},\hat{\mathbf{n}})\right)$$

For now, we use the following effective and simple parameterization

$$\mathbf{L}_o(\hat{\mathbf{v}},\hat{\mathbf{n}}) \approx t_1(\hat{\mathbf{n}})t_2(\hat{\mathbf{r}}_{\hat{\mathbf{v}}}) \tag{7}$$

which is based on the normal and the viewing direction reflected at the normal $\hat{\mathbf{r}}_{\hat{\mathbf{v}}} = 2\,(\hat{\mathbf{n}}\cdot\hat{\mathbf{v}})\,\hat{\mathbf{n}} - \hat{\mathbf{v}}$ (see also section 3.2 for other parameterizations).

Both functions, t_1 and t_2, can be pre-computed into textures using a mapping of a direction vector onto texture coordinates. Several such texture mappings exist in environment mapping techniques, e.g. spherical, parabolic or cubic maps.

The HF algorithm works by taking the logarithm of both sides of Equation (7) yielding

$$\bar{\mathbf{L}_o}(\hat{\mathbf{v}},\hat{\mathbf{n}}) \approx \bar{t_1}(\hat{\mathbf{n}}) + \bar{t_2}(\hat{\mathbf{r}}_{\hat{\mathbf{v}}})$$

Here $\log(a)$ is written as $\bar{a}$ (for more detailed description see [McCool et al. 2001]).

Now the approximation can be expressed as a linear equation system. In general a number of LF samples are used to fill the constant vector of the equation system, and all the texels of the two approximation textures are unpacked into the solution vector. The coefficient matrix of the equation system now maps each LF sample (one per row) to the appropriate texels. To achieve sub-pixel precision, the exact texture coordinates derived from the directions of the current sample are used to compute bilinear weights for the four surrounding texel positions.

In short the whole equation system can be written as

$$\left[\bar{\mathbf{L}_o}\right] = \left[A_1 A_2\right]\begin{bmatrix}\bar{t_1}\\ \bar{t_2}\end{bmatrix}$$

This system is probably under-constrained. On the one hand some texels might not be mapped according to the texture mapping technique, e.g. texels outside the inner circle in case of parabolic maps [Heidrich and Seidel 1998]. On the other hand depending on the sampling technique some texels representing valid directions might not be constrained by any LF sample. To insure that all texels are constrained a Laplace operator is introduced which adds additional dependencies to the equation system.

$$\begin{bmatrix}\bar{\mathbf{L}_o}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}A_1 & A_2\\ \lambda L_1 & 0\\ 0 & \lambda L_2\end{bmatrix}\begin{bmatrix}\bar{t_1}\\ \bar{t_2}\end{bmatrix} \tag{8}$$

where the scalar λ controls the influence of the Laplacian. Note that the Laplace operator will smooth the resulting texture, thus possibly increases the overall approximation error.

We use the Quasi-Minimal Residual (QMR) algorithm [Freund and Nachtigal 1992] to iteratively solve the equation system. The coefficient matrix might be quite large. Since there are only a few texels mapped in each row of the matrix, it is quite sparse and can either be stored in a compressed format or computed on the fly during the solving process.

The computation time of iterative solvers heavily depend on the starting vector. The initial vector is taken as the average of all data implied by the LF samples to each texel. In cases where a texel has no constraints, a weighted sum of LF samples closest to the considered texel is computed, using the inverse distances as weights. We further discuss the choice of sampling directions in section 3.3.

Beside the use of a different (global) parameterization, the need to span a whole sphere instead of a hemisphere also distinguishes our approach from the original HF method. [McCool et al. 2001] use parabolic maps (see [Heidrich and Seidel 1998]) during the approximation. The whole sphere could be described using two parabolic maps. But since nearly 22% of the texture area remain unused, we prefer cube maps to describe the LF function.

Another advantage of cube maps lies in the application of the Laplacian. [McCool et al. 2001] state, that the Laplacian should ideally be applied to the BRDF, i.e. to the lighting function in our situation. Since this is relatively expensive, the Laplacian is applied to the texture. Adopted to our situation, the Laplacian would have to be applied to both parabolic maps independently. This, however, may lead to discontinuities for directions ($\hat{\boldsymbol{\omega}}_i$ or $\hat{\boldsymbol{\omega}}_o$) close to the horizon, i.e. directions that are represented by texels close to the inner circles of the parabolic maps. For cube maps the neighborhood of texels for adjacent spacial directions can easily be established (see Figure 1).

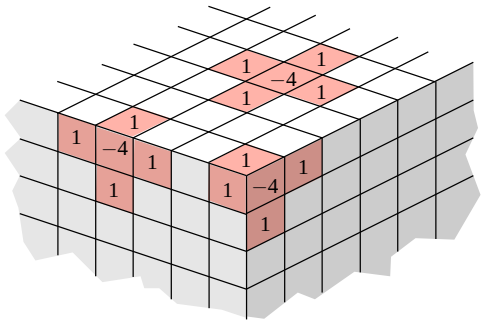

Figure 1: The Laplace kernel applied to several cube map texels, each representing a different spacial direction.

3.2 Other Parameterizations

The above normal-reflection parameterization (NR) with two textures, one dependent on the normal and the other dependent on the reflection vector, is only one of several possible parameterizations.

The use of both these vectors is motivated by the properties of the lighting function with simple BRDFs. In general, the texture dependence on the normal vector tends to approximate the amount of light resulting from the diffuse part of the BRDF, and the reflection vector texture will contain the specular highlights.

Using the Phong BRDF or BRDFs with a rotational symmetry around the reflection vector (as in the prefiltering technique by [Kautz and McCool 2000]) the parameterization can achieve optimal results. Other more complex isotropic BRDFs will be approximated as well as possible during the solver run.

In addition to the two textures above a third one can be used that is dependent on the viewer direction. This normal-reflection-view parameterization (NRV) results in slightly less statistical error, but does not improve significantly the visual appearance with most BRDFs.

Alternatively two additional textures can be used that work similarly to the Gram-Schmidt halfangle-difference (GSHD) parameterization of the BRDF separation introduced by [Kautz and McCool 1999a]. The halfangle vector $\hat{\mathbf{h}}$ is usually computed between the viewing direction and the light direction. In the global illumination scenario a dominant light direction $\hat{\mathbf{l}}_{dom}$ is used instead of the direction to a single light source. The resulting halfangle vector $\hat{\mathbf{h}}$ is transformed to the local surface coordinate frame $\{\hat{\mathbf{n}}, \hat{\mathbf{t}}(\hat{\mathbf{n}}), \hat{\mathbf{b}}\}$, where $\hat{\mathbf{b}} = \hat{\mathbf{n}} \times \hat{\mathbf{t}}(\hat{\mathbf{n}})$ is the orthogonal complement.

$$\hat{\mathbf{h}}' = \frac{\hat{\mathbf{l}}_{dom} + \hat{\mathbf{v}}}{\|\hat{\mathbf{l}}_{dom} + \hat{\mathbf{v}}\|} \qquad \hat{\mathbf{h}} = \begin{pmatrix} \hat{\mathbf{h}}' \cdot \hat{\mathbf{t}}(\hat{\mathbf{n}}) \\ \hat{\mathbf{h}}' \cdot \hat{\mathbf{b}} \\ \hat{\mathbf{h}}' \cdot \hat{\mathbf{n}} \end{pmatrix}$$

In our global lighting approach the difference vector $\hat{\mathbf{d}}$ resembles the dominant light direction in the local surface frame:

$$\hat{\mathbf{d}} = \begin{pmatrix} \hat{\mathbf{l}}_{dom} \cdot \hat{\mathbf{t}}(\hat{\mathbf{n}}) \\ \hat{\mathbf{l}}_{dom} \cdot \hat{\mathbf{b}} \\ \hat{\mathbf{l}}_{dom} \cdot \hat{\mathbf{n}} \end{pmatrix}$$

This normal-reflection-halfangle-difference parameterization (NRHD) is most useful if light comes predominantly from one general direction (e.g. from the sun). In the extreme case of having light coming only from one direction, the quality of the approximation is similar to a BRDF factorization with the GSHD parameterization. In other situations the light from other directions than the dominant one will be approximated as well as possible.

3.3 Sampling the Lighting Function

Sampling of the lighting function involves two major issues: which samples to consider in the factorization and how to compute these samples based on incoming light information $\mathbf{L}_i(\hat{\mathbf{l}})$ (see Equation (6)).

When using measured BRDFs, only samples that represent directions for which the BRDF is known can be used. To solve this problem, unknown BRDF values have to be interpolated from nearby samples. Because interpolation on a four-dimensional function is costly, we simply use McCool's original Homomorphic Factorization technique to approximate the BRDF and use the reconstructed BRDF in the LF integral. This reconstruction can be done with floating point precision and does not need to be quantized for storage in textures as in the case of hardware accelerated BRDF rendering.

The computation of a LF sample requires the evaluation of the integral in Equation (6). In our case the incoming light function $\mathbf{L}_i(\hat{\mathbf{l}})$ is discretely and regularly defined as a cubic environment map.

A simple variant of the integration might simply sum up the discrete lighting information weighted with the related BRDF value. Actually, this is incorrect, since a texel covers a non-constant area on the environmental sphere (see Figure 2).

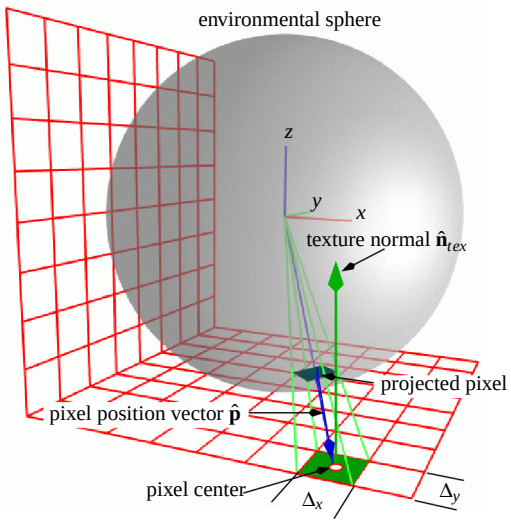

Figure 2: Setup for computing the texel weights for the lighting integration for the environment map at $z = -1$.

A better approach takes the relevance of an environmental texel into account, since boundary texels are less important than texels in the environment map's center. The projected texel area w.r.t the area of the unit sphere should be used as additional weight. This yields the following sampling equation (the BRDF arguments have been simplified):

$$\mathbf{L}_o(\hat{\mathbf{v}}, \hat{\mathbf{n}}) = \frac{\sum_{\hat{\mathbf{l}}} BRDF(\hat{\mathbf{v}}, \hat{\mathbf{l}}) \mathbf{L}_i(\pi^e(\hat{\mathbf{l}})) g(\pi^e(\hat{\mathbf{l}})) \left(\hat{\mathbf{n}} \cdot \hat{\mathbf{l}}\right)}{\sum_{\hat{\mathbf{l}}} g(\pi^e(\hat{\mathbf{l}}))} \tag{9}$$

where π^e maps the directional vector $\hat{\mathbf{l}}$ to cube map texture coordinates and g is the projected texel area.

To derive the weight g we assume for simplification, that the textures of the cubic environment map are placed at $x = \pm 1, y = \pm 1$ and $z = \pm 1$ and that they are parameterized over $[-1,1]^2$ (see Figure 2). Considering the environment map located at $z = -1$ and the *differential* texel with center (x_0, y_0), the projected texel area on the environmental sphere with radius 1 is given by

$$\left(\frac{-\hat{\mathbf{p}}}{\|\hat{\mathbf{p}}\|} \cdot \hat{\mathbf{n}}_{tex} \right) \frac{dxdy}{r^2} = \frac{dxdy}{(x_0^2 + y_0^2 + 1)^{3/2}}$$

where $\hat{\mathbf{p}} = (x_0, y_0, -1)$ is the position vector of the texel center, $\hat{\mathbf{n}}_{tex} = (0,0,1)$ is the environment map's normal and r is the distance of the texel center to the origin, i.e. $r = \|\hat{\mathbf{p}}\|$.

Applying this result to a texel with extension Δ_x and Δ_y in $x-$ and $y-$ direction respectively, we get

$$g(x_0, y_0) = \int_{x_0 - \frac{\Delta_x}{2}}^{x_0 + \frac{\Delta_x}{2}} \int_{y_0 - \frac{\Delta_y}{2}}^{y_0 + \frac{\Delta_y}{2}} \frac{1}{(x^2 + y^2 + 1)^{3/2}} \, dxdy \tag{10}$$

Since the analytical solution to Equation (10) involves the evaluation of several trigonometric functions we work with the following approximation without significant impact on the results:

$$g(x_0, y_0) \approx \left(\frac{-\hat{\mathbf{p}}}{\|\hat{\mathbf{p}}\|} \cdot \hat{\mathbf{n}}_{tex} \right) \frac{\Delta_x \Delta_y}{r^2} = \frac{\Delta_x \Delta_y}{(x_0^2 + y_0^2 + 1)^{3/2}} \tag{11}$$

Figure 3 shows the function in Equation (11) for weighting the texels.

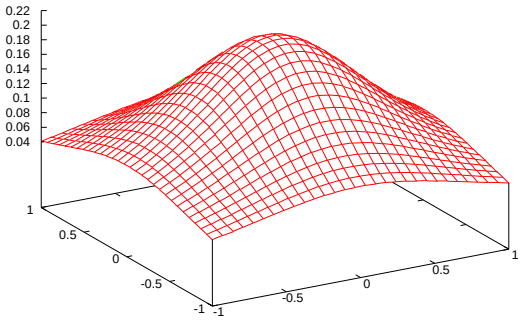

Figure 3: The weight function used for integrating the lighting samples for cubic maps.

Due to extreme BRDF values or light environment values some LF samples might get very large. To avoid precision problems in the factorization step we clamp the LF sample values. Theoretically we could clamp to $[0,1]$ since, in contrary to BRDF approximation, damping through the $\hat{\mathbf{n}} \cdot \hat{\mathbf{l}}$ factor does not occur. However, we found that allowing slightly greater values up to 4 leads to better approximation results.

The selection of LF samples used in the factorization should map as regularly as possible to the texels in the resulting textures used to approximate the lighting equation (6). To achieve this, the inverse mappings $\pi_j^{-1},\ j = 1,2$ are applied to all texels. The related directions are then used to compute the LF samples, if the directional combinations is valid. Actually, we use $\hat{\mathbf{v}}$ instead of $\hat{\mathbf{r}}_{\hat{\mathbf{v}}}$ which makes no difference for the regularity of the resulting LF samples w.r.t. the textures.

3.4 Rendering

The reconstruction of the factorization result in current graphics hardware raises several problematic issues. While the factorization can be done with high precision floating point values, the resulting textures have to be converted to low precision (e.g. 8 bit) integer representation. In what follows, we will regard the hardware texture processing pipeline as a fixed-point arithmetic unit working with values in the range $[0,1]$.

The factorization textures need to retain as much precision as possible during the LF reconstruction. A first step to achieve this goal is the normalization of the textures to the maximum value in each texture. To correct for the normalization a constant correction color is multiplied with the textures during the rendering. This correction color can be included in the computation as constant per-vertex color.

If the correction value appears to be greater than one, it can not be represented by a color. To reach an optimal result in this case, the correction value should be split into two factors. One below one, that is used as per-vertex color, and another constant value that is multiplied as late as possible in the texture pipeline, in order to avoid premature clamping. Most current graphics hardware allows this post-scaling with constant values of 1, 2 or 4 (e.g. OpenGL extension `GL_ARB_texture_env_combine`, also part of OpenGL 1.3, see [SGI 2002]). Recent programmable hardware allows even higher powers of two, though 4 is usually an acceptable compromise as higher values lead to a loss of precision in the lower bit ranges.

Sometimes these scaling factors are not sufficient to restore the original intensity range of the map. In such cases the textures need to be pre-scaled before the rendering, possibly immediately after the factorization. The pre-scaling needs to find an acceptable compromise between avoiding contouring artifacts in lower intensity areas and risking clamping of higher intensities during the texture pipeline computations. Due to our experience this compromise has to be found for each factorization result individually. Currently we use an interactive adjustment of the scaling factors which allows immediate visual quality judgement.

Now some notes on the rendering performance of the method. On most current graphics hardware the two textures required for the NR parameterization can be applied in a single pass. The rendering then differs from a standard Phong-lit material only at the vertex processing and the texture application. The vertex processing needs to compute the two texture coordinate sets instead of the diffuse and specular Phong computation. For the texture coordinate computation both normal and viewing direction are required in world coordinates. The normal is usually needed for Phong lighting in eye coordinates. This transformation can now be replaced by the world coordinates computation. The viewing direction has to be transformed and then used for the computation of the reflection vector. Using a fully programmable vertex processor (e.g. with vertex programs/shaders as described by [Lindholm et al. 2001]) the required number of instructions per vertex is nearly equal to Phong lighting, especially when using cube maps, because the computed vectors can be used immediately as texture coordinates. If no programmable vertex processor is available, one can apply appropriate texture coordinate generation functionality (e.g. in OpenGL with `GL_NORMAL_MAP` and `GL_REFLECTION_MAP`) in combination with texture matrix transformations to map the directional vectors in world coordinates.

The application of two textures might be the performance bottle necks in most situations with limited pixel processing and memory transfer resources for each object in a scene. Since the textures are often quite small or can be used at a low mip-map level due to their smoothness, memory transfers should be cacheable in most texture caches. Therefore rendering the objects sorted by texture should have an even higher priority than usual. Note that standard techniques to generate mip-map levels might lead to incorrect reconstruction results, as the filtering can cause the loss of highlights and non-linear properties. This can be avoided by using mip-map textures generated during a multi-resolution factorization as described in the next section. However even a theoretical incorrect mip-map filtering should give a reasonable approximation and reduces aliasing artifacts significantly.

4 Results

Our algorithm has been tested with several isotropic BRDFs and light environments. The analytical Cook-Torrance [Cook and Torrance 1981] and Modified Phong [Lafortune and Willems 1994] BRDF have been used as well as a reconstruction of the measured BRDF "Cayman", a highly specular car lacquer [Cornell University 2002]. Among the used light environments is the synthetical environment map "Loch" by Jeff Heath (also used in [Kautz et al. 2000]) and high-dynamic range light probe measurements in St. Peter's Basilica, Rome and the Uffizi Gallery, Florence by [Debevec 2002].

The computational effort for the approximation falls into two categories: Sampling the lighting function, i.e. computing the integral over the incoming light for each sample, and performing the factorization.

The LF sample locations have been determined by reverse-projection of the texels (see section 3.3). Having a cube map with 64 texels per edge, leads to about $\frac{1}{2}(64 \times 64 \times 6)^2$ texels (the squar-

ing is due to two textures and the $1/2$ is due to invalid normal-view combinations), which is just over 300 million samples. This is generally above current workstation computation and memory capabilities. To reduce the number of samples we use a virtual texture size of 12 or 16 steps per edge, resulting in several hundred thousand to a millon samples. This texture size is only used during sampling, not for the real textures during the factorization itself, leading to regularly but not densely mapped texels. Problems that are due to this lower sampling density can be resolved, in most cases, by using a multi-resolution algorithm, starting a factorization with a cube map edge size of 16. Then the result of this factorization is interpolated to a map of twice the size and used as starting solution for another factorization run, again with the same samples as in the first run. This can be repeated until the desired map size (e.g. 64) is reached.

The environment map representing the incoming light function is usually scaled to a resolution between 20^2 and 64^2 per cube face. This leads to several thousand light samples. For each LF sample only about half of the light samples need to be considered due to invalid combinations of the normal and light direction. For each light value an appropriate BRDF value needs to be evaluated, resulting in a total of several hundred of millions to a billion BRDF samples taken during the LF sampling. This usually consumes the most computational effort of the approximation process. On a current AMD Athlon 1.4 GHz workstation this takes between a quarter and a full hour in our research implementation.

The effort of the factorization itself depends on the size of the coefficient matrix and the convergence of the whole equation system. While the coefficient matrix depends only on the number of LF samples and the texture size, the convergence behavior cannot be predicted easily and varies greatly for each BRDF and light environment setup. On the aforementioned system the factorization usually takes between 10 and 30 minutes. Note that all values given here apply to high-quality approximations, for previewing purposes considerably smaller map sizes and sample numbers need to be taken, thereby immediately accelerating the process.

4.1 Examples

Here are some examples of our test cases. Figure 4 shows the factorization of the LF with the Cook-Torrance BRDF (set up to simulate copper) in a lighting environment defined by the high dynamic range environment map of the Uffizi Gallery, shown in the top-left corner as a unfolded cube map. On the right hand side, the two resulting textures are shown. The top one is parameterized by the normal vector, while the middle one is parameterized by the reflection vector. Below them the normalization correction color is displayed (see section 3.4). A rendered teapot with applied textures is shown in the bottom-left corner.

To be able to judge the quality of the approximation the exact lighting computation with the copper BRDF has been carried out for each vertex on a very fine tessellated model (almost per-pixel computation) in Figure 5. It can be seen that the approximation cannot capture all properties of the BRDF completely and the typical Cook-Torrance BRDF color shift at the edge of the highlight is slightly less intense in Figure 4.

To test the factorization results with a highly specular, almost mirror-like BRDF, the Modified Phong BRDF has been used with a specular exponent of 400. The incoming light is sampled from the environment map of St. Peter's Basilica. Figure 6 shows the results with the same layout as above.

The previous examples use real, measured light environments as incoming light function. In order to use standard images with only a limited dynamic range, it is advisable to enhance highlights while sampling the environment map. In Figure 7 the factorization of a Cook-Torrance BRDF (with gold setup) has been conducted with the artificial, rendered "Loch" environment map. At the direction of

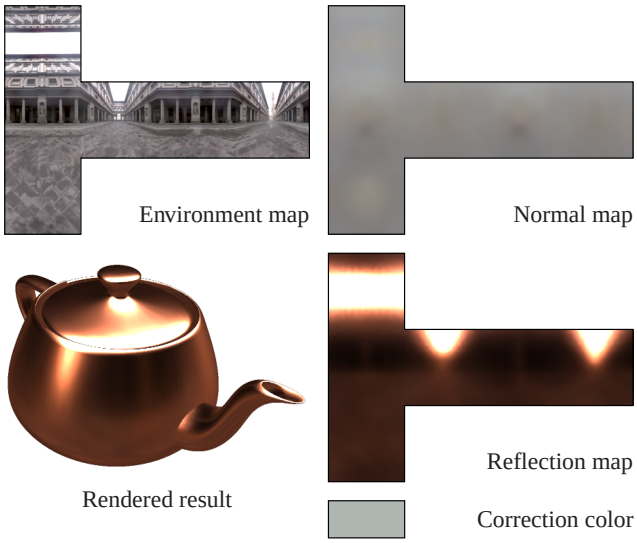

Figure 4: Factorization of copper BRDF with Uffizi environment map

Figure 5: Teapot with exact lighting computation

the sun and its reflection in the sea, the map has been made brighter by several powers of ten to simulate the much higher intensity of light arriving from these directions.

Finally Figure 8 shows an example using a measured BRDF, "Cayman" car lacquer. The light environment is defined by a high dynamic range light measurement of a forest scene.

4.2 Error Measurements

Several error measurements have been taken for the above shown factorizations. The error measurements in Table 1 are abbreviated as follows: Average (Avg) and maximum (Max) absolute (Abs) error and relative (Rel) error, all between the exact measurements and the reconstruction from double precision floating point values. RMS I is the root mean squares error of the same difference, while RMS II is the error of the reconstruction from 8-bit integer textures. All values are the arithmetic averages over the RGB components. The errors have been measured with the same samples that have been used for the factorization.

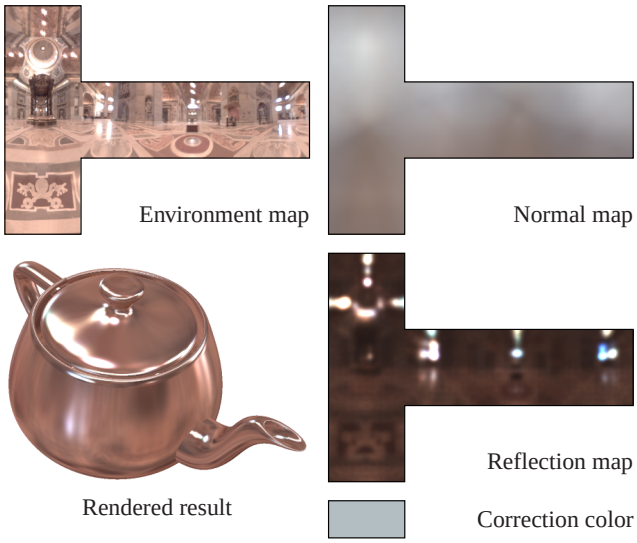

Figure 6: Highly specular Modified Phong material with St. Peters Basilica environment map

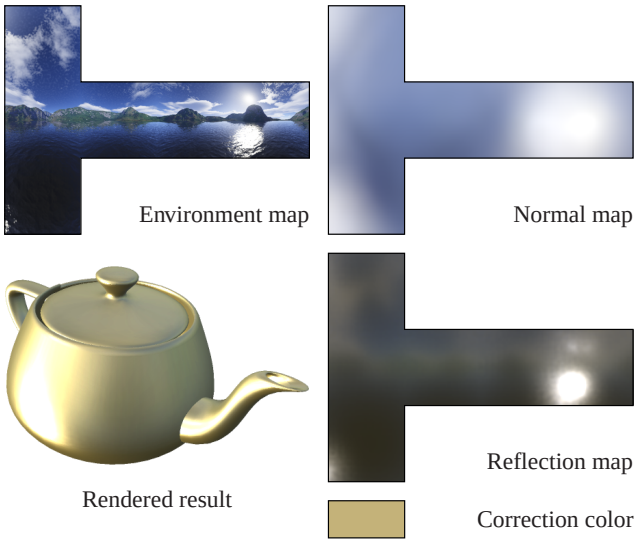

Figure 7: Gold material with Loch environment map

Material	Copper	Gold	Gold	Phong	Cayman
Environ.	Uffizi	Loch	Loch	St.Pet.	Forest
Param.	NR	NR	NRV	NR	NR
Avg Abs	0.1298	0.0726	0.0724	0.1626	0.0460
Max Abs	0.2246	0.1015	0.0982	0.3518	1.0804
Avg Rel	3.3554	3.0664	3.0965	3.9328	0.3120
Max Rel	36.9487	3.9212	3.8807	19.2405	35.8461
RMS I	0.3073	0.1887	0.1892	0.2895	0.0635
RMS II	0.9881	0.3290	0.3319	0.4774	0.1022

Table 1: Error measurements

Besides the four examples shown in section 4.1 also values for the gold BRDF with the NRV parameterization are given. The visual appearance does not differ from the NR parameterization and the error measurements show only small differences as well. Surprisingly the RMS error is even slightly higher, which might be caused by the higher influence of the Laplacian in the equation system due to three textures rather than two.

Some of the error measurements, especially the maximum relative error are quite high. This has also been observed by [McCool et al. 2001] with the BRDF factorization and can be attributed to large peaks in the data. The measured light environments with high-dynamic range lead to very high intensity differences in the lighting function after the computation with a highly specular BRDF. The synthetical environment map with less dynamic-range has considerably lower maximum error values.

For quality judgements the original BRDF factorization technique has been compared to a LF factorization with a single-light source. Using a Cook-Torrance BRDF the BRDF factorization has a RMS error of 0.0435, the LF with normal-reflection parameters 0.1110, the LF with the additional halfangle-difference parameters 0.0781. Due to the different application of the techniques this comparison has only limited significance.

Of course, we would like to compare our technique with other environment map prefiltering techniques, especially [Kautz and McCool 2000]. Unfortunately, no comparable data sets are currently available.

5 Future Work

Our technique offers a unified approach to the simulation of complex lighting and material. Since we separate the complete lighting, the incoming light situation needs to be static to a certain extent in order to be visually plausible. A rather simple improvement is the use of additional dynamic light sources based on the standard Phong model. Alternatively dynamic light sources with the original BRDF factorization might be added during rendering.

As stated above, the pre-computational costs to approximate the lighting function is relatively high. Both, computing the LF samples and the final factorization, e.g. solving the linear system, contribute to this costs. One could attempt to solve the linear system once by finding a pseudo inverse matrix to the one in Equation (8). Changing the environment or the BRDF would still need the computation of the lighting function, but the factorization would be very cheap. However, such a “global” solution will not give as good approximations as the ones computed for each individual BRDF and incoming light function.

Trying to reduce the effort needed to sample the lighting function using Monte Carlo-like approaches might help to decrease the computing time for the LF sampling.

Another principle direction of future research is the introduction of anisotropic materials. This would involve the factorization of the lighting equation (5), forcing us to add another texture. The major problem lies in the general use of world coordinates in the lighting computation. Thus a technique has to be found to map the local tangent vector to world coordinates.

Acknowledgments

We found it very helpful to have the chance to discuss our research results with Jan Kautz and his colleagues of the Computer Graphics Group at the Max-Planck Institut für Informatik in Saarbrücken, Germany and the Computer Graphics Group at the Saarland University. Also we would like to thank Michael McCool for his suggestions on further optimization and improvements of our approach.

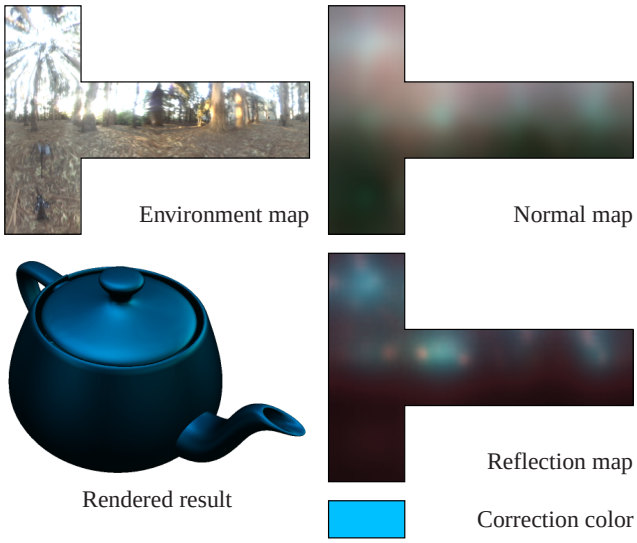

Figure 8: Measured Cayman BRDF with forest environment map

References

BANKS, D. 1994. Illumination in diverse codimensions. In *ACM Proceedings SIGGRAPH*, Addison Wesley, vol. 28, ACM, 327–334.

CABRAL, B., OLANO, M., AND NEMEC, P. 1999. Reflection space image based rendering. In *ACM Proceedings SIGGRAPH*, Addison Wesley, vol. 33, ACM, 165–170.

CHEN, W.-C., GRZESZCZUK, R., AND BOUGUET, J.-Y. 2001. Light field mapping: Hardware-accelerated visualization of surface light fields. In *SIGGRAPH 2001 Course Notes "Acquisition and Visualization of Surface Light Fields"*.

COOK, R., AND TORRANCE, K. 1981. A reflectance model for computer graphics. In *ACM Proceedings SIGGRAPH*, Addison Wesley, vol. 15, ACM, 307–316.

CORNELL UNIVERSITY, 2002. Program of computer graphics online resource: Reflectance data. `http://www.graphics.cornell.edu/online/measurements/reflectance/index.html`.

DEBEVEC, P., 2002. Online resources. `http://www.debevec.org`.

FOURNIER, A. 1992. Filtering normal maps and creating multiple surfaces. Tech. Rep. TR-92-41, Department of Computer Science, University of British Columbia, Vancouver, BC, Canada.

FOURNIER, A. 1992. Normal distribution functions and multiple surfaces. In *Proc. Graphics Interface, Workshop on Local Illumination*, Morgan Kaufmann, Canadian Human-Computer Communications Society, 45–52.

FOURNIER, A. 1995. Separating reflection functions for linear radiosity. In *EUROGRAPHICS Rendering Workshop*, Eurographics, 296–305.

FREUND, R. W., AND NACHTIGAL, N. M. 1992. QMR: a quasi-minimal residual method for non-Hermitian linear systems. In *Iterative Methods in Linear Algebra*, R. Beauwens and P. de Groen, Eds. Elsevier Science Publishers, 151–154.

HEIDRICH, W., AND SEIDEL, H.-P. 1998. View-independent environment maps. In *Eurographics/SIGGRAPH Workshop on Graphics Hardware*, Addison-Wesley, 39–45.

HEIDRICH, W., AND SEIDEL, H.-P. 1999. Realistic, hardware-accelerated shading and lighting. In *ACM Proceedings SIGGRAPH*, Addison Wesley, vol. 33, ACM, 171–178.

KAUTZ, J., AND MCCOOL, M.-D. 1999. Hardware rendering with bidirectional reflectances. Tech. rep., Department of Computer Science, University of Waterloo.

KAUTZ, J., AND MCCOOL, M.-D. 1999. Interactive rendering with arbitrary BRDFs using separable approximations. In *EUROGRAPHICS Rendering Workshop*, Eurographics, 253–253.

KAUTZ, J., AND MCCOOL, M. D. 2000. Approximation of glossy reflection with prefiltered environment maps. In *Proc. Graphics Interface*, Morgan Kaufmann, Canadian Human-Computer Communications Society, 119–126.

KAUTZ, J., VAZQUEZ, P., HEIDRICH, W., AND SEIDEL, H.-P. 2000. A unified approach to prefiltered environment maps. In *EUROGRAPHICS Rendering Workshop*, Eurographics, 185–196.

LAFORTUNE, E. P., AND WILLEMS, Y. D. 1994. Using the modified phong BRDF for physically based rendering. Tech. Rep. CW197, Dept. of Computer Science, Leuven, Belgium.

LAFORTUNE, E., FOO, S.-C., TORRANCE, K., AND GREENBERG, D. 1997. Non-linear approximation of reflectance functions. In *ACM Proceedings SIGGRAPH*, Addison Wesley, vol. 31, ACM, 117–126.

LEVOY, M., AND HANRAHAN, P. 1996. Lightfield rendering. In *ACM Proceedings SIGGRAPH*, Addison Wesley, vol. 30, ACM, 31–42.

LINDHOLM, E., KILGARD, M., AND MORETON, H. 2001. A user-programmable vertex engine. In *ACM Proceedings SIGGRAPH*, Addison Wesley, vol. 35, ACM.

MCCOOL, M.-D., ANG, J., AND AHMAD, A. 2001. Homomorphic factorization of BRDFs for high-performance rendering. In *ACM Proceedings SIGGRAPH*, Addison Wesley, vol. 35, ACM, 171–178.

MILLER, G., AND HOFFMAN, R. 1984. Illumination and reflection maps: Simulated objects in simulated and real environments. In *SIGGRAPH 1984 Course Notes "Advanced Computer Graphics Animation"*.

PHONG, B. T. 1975. Illumination for computer generated pictures. *Communications of the ACM 18*, 6, 311–317.

SGI, 2002. OpenGL extension registry. `http://oss.sgi.com/projects/ogl-sample/registry`.