

Image-Based Reconstruction of
Spatially Varying Materials

Hendrik P. A. Lensch, Jan Kautz,
Michael Goesele, Wolfgang Heidrich,
Hans-Peter Seidel

MPI-I-2001-4-001

May 2001

FORSCHUNGSBERICHT RESEARCH REPORT

MAX-PLANCK-INSTITUT
FÜR
INFORMATIK

Stuhlsatzenhausweg 85 66123 Saarbrücken Germany

Author's Address

Hendrik P. A. Lensch, Jan Kautz, Michael Geosele,
Wolfgang Heidrich, Hans-Peter Seidel
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany
{lensch, kautz, goesele, hpseidel}@mpi-sb.mpg.de
heidrich@cs.ubc.ca

Keywords

Reflectance & Shading Models, Physically Based Modeling, Computer Vision, Illumination, Image-Based Rendering, Rendering, Texture Mapping

1 Introduction

The use of realistic models for all components of image synthesis is a fundamental prerequisite for photorealistic rendering. This includes models for the geometry, light sources, and cameras, as well as materials. As more and more visual complexity is demanded, it is more and more often infeasible to generate these models manually. Automatic and semi-automatic methods for model acquisition are therefore becoming increasingly important.

In this paper we concentrate on the acquisition of realistic materials. In particular, we describe an acquisition process for spatially varying BRDFs that is efficient, reliable, and requires little manual intervention. Other methods described in the literature (see Section 2 for an overview) are either focusing on homogeneous materials, or make assumptions on the type of material to be measured (e.g. human faces). In our work, we measure spatially varying BRDFs without making any additional assumptions. In particular, our contributions are

- a robust and efficient BRDF fitting process that clusters the acquired samples into groups of similar materials and fits a Lafortune model [11] to each group,
- a method that projects every sample texel into a basis of BRDFs obtained from the clustering procedure. This projection accurately represents the material at that point and results in a compact representation of a truly spatially varying BRDF.

We require only a relatively small number of high-dynamic range photographs (about 20-25 images for one object), thereby speeding up the acquisition phase.

As a result of the fitting, clustering, and projection process, we obtain a compact representation of spatially varying materials that is well suited for rendering purposes (see Figure 9.1 for an example). The method works both for objects consisting of a mixture of distinct materials (e.g. paint and silver, see Figure 9.3), or for smooth transitions between material properties.

In the following we first review some of the previous work in this area, before we discuss the details of our own method. We start by describing the acquisition of the measurement data (Section 3), explain the resampling of this data into our data structures (Section 4), the BRDF fitting and material clustering steps (Sections 5 and 6), and finally a method for projecting the materials into a basis of BRDFs (Section 7). Section 8 briefly describes our rendering method. In Section 9 we present our results and then we conclude in Section 10.

2 Related Work

The representation of real-world materials has recently received a lot of attention in the computer graphics community. The approaches can be grouped into three different categories: light field and image database methods with static illumination, dense sampling of the light and viewing directions to generate a tabular representation of the BRDF, and finally the fitting of reflection models, often based on a sparser set of samples. This last approach is the one we take and extend to spatially varying BRDFs.

In the first category, there has been a number of approaches ranging from a relatively sparse set of images with a geometric model [4] over the Lumigraph [7] with more images and a coarser model to the light field [13] with no geometry and a dense image database. Recently surface light fields [27, 18] have become popular, which feature both a dense sampling of the directional information and a detailed geometry. In contrast to these approaches, bidirectional texture functions [1] also work for changes in the lighting conditions, although at very high storage costs. In our work we use an algorithm similar to the function quantization approach proposed by Wood et al. [27] to resample the image data into a compact representation.

The traditional approach for dense sampling of reflectance properties is to use specialized devices (gonioreflectometers), that position both a light source and a sensor relative to the material. These devices can only obtain one sample for each pair of light and sensor position and are therefore relatively slow.

More recently, image-based approaches have been proposed. These methods are able to acquire a large number of samples at once. For example, Ward Larson [25] uses a hemispherical mirror to sample the exitant hemisphere of light with a single image. Instead of using curved mirrors, it is also possible to use curved geometry to obtain a large number of samples with a single image. This approach is taken by Lu et al [15], who assume a cylindrical surface, and Marschner et al. [17] who obtain the geometry using a range scanner. Our method is similar in spirit to the method of Marschner et al., but we are also dealing with spatially varying BRDFs and we are fitting a reflection model rather than using a tabular

form in order to achieve a compact representation.

A number of researchers have also described the fitting of reflection models to the acquired sample data [2, 11, 22, 25, 28]. Of these methods, the ones by Ward Larson [25] and Lafortune et al. [11] do not consider spatial variations. Sato et al. [22] fit a Torrance-Sparrow model [24] to the data, and consider high-frequency variations for the diffuse part but only per-triangle variations for the specular part. This is also the case for the work by Yu et al. [28], which also takes indirect illumination into account. In our work, we perform the measurements in a darkened, black room, so that there is no indirect light coming from the outside of the object. Indirect light within the object is assumed to be negligible, which excludes the use of objects with extreme concavities.

Debevec et al. [2] describe a method for acquiring the reflectance field of human faces. In one part of their work they fit a specialized reflection model for human skin to the measured data (consisting of about 200 images). Both specular and diffuse parameters of the reflection model can vary rapidly across the surface, but other parameters like the de-saturation of the diffuse component at grazing angles are constant and only apply to human skin. In our work we try to avoid making assumptions on the kind of material we are measuring.

Several different representations have been used for fitting BRDF data. In addition to the models used for measured data (e.g. Koenderink et al. [10], Lafortune [11], Torrance-Sparrow [22, 28], Ward [25]), Westin et al. [26] have used spherical harmonics for projecting simulated BRDF data. In our work we use the Lafortune model because it is compact, well suited for optimization algorithms, and capable of representing interesting BRDF properties such as off-specular peaks and retro-reflection.

3 Acquisition

We obtain the 3D models with a structured light 3D scanner and a computer tomography scanner both generating dense triangle meshes. The triangle meshes are smoothed [5, 9], manually cleaned, and decimated.

All images are acquired in a measurement lab using a professional digital camera. An HMI metal halide bulb serves as point light source for the BRDF measurements. The interior of the photo studio is covered with dark and diffusely reflecting felt to reduce the influence of the environment on the measurements.

Several views of each object are captured with different camera and light source positions. For each view we acquire three sets of images: two images to recover the light source position, one image of the object’s silhouette to register the 3D model with the images. We then acquire a high dynamic range image [3] of the object lit by the point light source by taking a series of photographs with varying exposure time.

In addition, a series of calibration images of a checkerboard pattern is taken whenever the lens settings are changed. The calibration method proposed by Zhang [29] is used to recover the intrinsic camera parameters. Another high dynamic range image of a gray card with known camera and light position is taken in order to compute the brightness of the light source.

To register the images with the 3D model we use a silhouette-based method [12] that yields the camera position relative to the object. The light source position is triangulated based on the reflections in a number of mirroring steel balls. The details of that approach will be described elsewhere.

4 Resampling of Radiance Values

After acquisition of the geometric model, high-dynamic range image recovery, and registration, it is necessary to merge the acquired data for further processing. For each point on the model’s surface we collect all available information using two data structures.

The first one is a so called *lunitexel* denoted by \mathcal{L} , which is generated for every visible surface point. Each lunitexel stores the geometric and photometric data of one point, i.e. its position \vec{x} and the normal \hat{n} in world coordinates¹. Linked to the lunitexel is a list of radiance samples \mathcal{R}_i , each representing the outgoing radiance r of the surface point captured by one image plus the direction of the light \hat{u} and the viewing direction \hat{v} . \hat{u} and \hat{v} are both given in the local coordinate frame of the surface point spanned by \hat{n} and a deterministically constructed tangent and bi-normal.

A lunitexel can be seen as a very sparsely sampled BRDF. We define the error between a given BRDF f_r and a lunitexel \mathcal{L} as:

$$E_{f_r}(\mathcal{L}) = \frac{1}{|\mathcal{L}|} \sum_{\mathcal{R}_i \in \mathcal{L}} s \cdot I(f_r(\hat{u}_i, \hat{v}_i)u_{i,z}, r_i) + D(f_r(\hat{u}_i, \hat{v}_i)u_{i,z}, r_i), \quad (4.1)$$

where $|\mathcal{L}|$ stands for the number of radiance samples linked to the lunitexel, $I(r_1, r_2)$ is a function measuring the intensity difference, and $D(r_1, r_2)$ measures the color-difference. We introduce the weight s , to be able to compensate for noisy data (e.g. a slightly wrong normal resulting in a wrong highlight). We always set $s \leq 1$. Please note that since r represents radiance and not reflectance, the BRDF has to be multiplied by the cosine between the normal and the local light direction, which is u_z .

¹hats denote unit vectors and arrows denote vectors of arbitrary length.

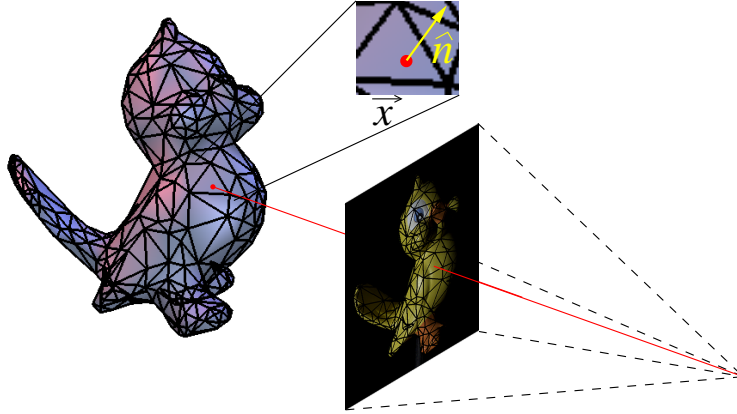


Figure 4.1: *The correspondence between pixel position and point position \vec{x} on the object is computed by tracing a ray through the image onto the object. At every \vec{x} a local normal \hat{n} can be computed from the triangle's vertex normals.*

4.1 Assembling Lumitexels

Collecting all radiance samples for a lumitexel requires a resampling of the input images for the particular point on the surface. At first, one has to determine the set of surface points for which a lumitexel should be generated. In order to obtain the highest quality with respect to the input images, the sampling density of the surface points must match that of the images.

Every triangle of the 3D model is projected into each image using the previously determined camera parameters. The area of the projected triangle is measured in pixels and the triangle is assigned to the image I_{best} in which its projected area is largest. For every pixel within the triangle in I_{best} a lumitexel is generated.

The position \vec{x} of the surface point for the lumitexel is given by the intersection of the ray from the camera through the pixel with the mesh (see Figure 4.1). The normal \hat{n} is interpolated using the triangle's vertex normals.

A radiance sample \mathcal{R}_j is now constructed for each image I_j in which \vec{x} is visible from the camera position and the surface point is lit by the point light source. The vectors \hat{u}_j and \hat{v}_j can be directly calculated. The associated radiance is found by projecting \vec{x} onto the image plane and retrieving the color c_j at that point using bilinear interpolation. Note, that for I_{best} no bilinear interpolation is necessary and c_{best} can be obtained without resampling since \vec{x} exactly maps to the original pixel by construction. The radiance r_j of the radiance sample \mathcal{R}_j is obtained by scaling c_j according to the brightness of the light source and the squared distance from the light source to \vec{x} .

5 BRDF Fitting

In this section we will first detail the Lafortune BRDF model [11] that we use to fit our given lumitexels. Then we will explain how this fit is performed using Levenberg-Marquardt optimization.

5.1 Lafortune Model

BRDFs are four-dimensional functions that depend on the local viewing and light direction. The dependence on wavelength is often neglected or simply three different BRDFs are used for the red, green, and blue channel. We use the later approach.

Instead of representing a measured BRDF as a 4D table the measured samples are in our case approximated with a parameterized BRDF. This has two advantages. Firstly, the BRDF requires much less storage since only the parameters are stored and secondly, we only require a sparse set of samples that would not be sufficient to faithfully represent a complete tabular BRDF.

Many different BRDF models have been proposed (e.g. [24, 25]) with different strengths and weaknesses. Our method may be used together with any parameterized BRDF model. We have chosen the computationally simple but general and physically plausible Lafortune model [11] in its isotropic form:

$$f_r(\hat{u}, \hat{v}) = \rho_d + \sum_i [C_{x,i}(u_x v_x + u_y v_y) + C_{z,i} u_z v_z]^{N_i}, \quad (5.1)$$

This model uses only a handful of parameters. \hat{u} and \hat{v} are the local light and viewing directions, ρ_d is the diffuse component, N_i is the specular exponent, the ratio between $C_{x,i}$ and $C_{z,i}$ indicates the off-specularity of lobe i of the BRDF. The sign of $C_{x,i}$ makes the lobe i either retro-reflective (positive $C_{x,i}$) or forward-reflective (negative $C_{x,i}$). The albedo of the lobe i is given by the magnitude of the parameters $C_{x,i}$ and $C_{z,i}$. From now on we will denote the BRDF with $f_r(\vec{a}; \hat{u}, \hat{v})$, where \vec{a} subsumes all the parameters of the model, i.e. $\rho_d, C_{x,i}, C_{z,i}$, and N_i . In the case of only one lobe \vec{a} is 12-dimensional (4 parameters for each color channel).

5.2 Non-Linear Fitting

The Lafortune BRDF is non-linear in its parameters, which means that we have to use a non-linear optimization method to fit the parameters to the given data. As in the original work by Lafortune et al. [11], we use the Levenberg-Marquardt optimization [20] to determine the parameters of the Lafortune model from our measured data. This method has proven to be well-suited for fitting non-linear BRDFs.

Instead of BRDF samples we use radiance samples as our input data, which means we are not directly fitting the BRDF $f_r(\vec{a}; \hat{u}, \hat{v})$ but the radiance values $f_r(\vec{a}; \hat{u}, \hat{v})u_z$ to the radiance samples \mathcal{R}_i in order to avoid the numerically problematic division by u_z .

We also ensure that the fitting process works well and does not get stuck in undesired local minima by initializing the fitting routine with parameters that correspond to an average BRDF.

The Levenberg-Marquardt optimization outputs not only the best-fit parameter vector \vec{a} , but also a covariance matrix of the parameters, which provides a rough idea of the parameters that could not be fitted well. This information is used in our splitting and clustering algorithm, as explained in the next section.

6 Clustering

In this section we will explain how we cluster the given lumitexels so that each cluster C_i corresponds to one material of the object. Given a set of BRDFs $\{f_i\}$, each cluster C_i consists of a list of all the lumitexels \mathcal{L}_i for which f_i provides the best approximation. Determining these clusters is a problem closely related to vector quantization [6] and k -means clustering [14, 16], both of which work in affine spaces. Unfortunately, we do not have an affine space when clustering BRDF samples, and we are therefore employing a different method.

The general idea is to first fit a BRDF f_r to an initial cluster containing all the data. Then we generate two new BRDF models f_1 and f_2 using the covariance matrix from the fit (explained in more detail below) representing two new clusters. The lumitexels \mathcal{L}_i from the original cluster are then distributed according to the distance $E_{f_1}(\mathcal{L}_i)$ and $E_{f_2}(\mathcal{L}_i)$ into the new clusters. We then recursively choose another cluster, split it, and redistribute the lumitexels and so on. This is repeated until the desired number of materials is reached, as detailed in Section 6.4.

6.1 Lumitexel Selection

The fitting procedure described in Section 5 performs a relatively large number of operations per radiance sample. Thus, it is expensive to fit a BRDF using all lumitexels (and all radiance samples contained in the lumitexels) generated by the assembling procedure. Instead, it is sufficient to consider only a few thousand lumitexels at the beginning. Later on, we increase the number for an accurate fit.

A first, naive approach to choosing this subset for fitting selects every n -th lumitexel regardless of its reliability or possible contribution. However, as stated in [28] and [23], for a robust estimation of the specular part of a BRDF it is very important to include radiance samples within the specular lobe of the material. Unfortunately, these brightest pixels statistically also carry the largest error.

Following these ideas we select more lumitexels in areas where a highlight is likely to occur. These areas are determined by the surface normal, the light source

position and a synthetic BRDF with a broad highlight.

6.2 Splitting

Fitting just a single BRDF to the initial cluster of course is not sufficient if the concerned object consists of more than one material. In order to decide which cluster to split, we compute the following error for all clusters C_j :

$$E(C_j) = \sum_{\mathcal{L}_i \in C_j} E_{f_r}(\mathcal{L}_i) \quad \forall C_j. \quad (6.1)$$

The cluster C_j with the largest error will be split into two new clusters each with a different BRDF. Further materials can be extracted by further splitting the clusters.

But how do we split a cluster? The BRDF fitted to a cluster represents the average material of the lumitexels in that cluster. Fitting the BRDF using the Levenberg-Marquardt algorithm (see Section 5) will also provide us with the covariance matrix of the samples. The eigenvector belonging to the largest eigenvalue of this matrix represents the direction in which the variance of the samples is highest, and is therefore a good choice for the direction in which the parameter space is to be split.

Let \vec{a} be the fitted parameter vector of the BRDF $f(\vec{a}; \hat{u}, \hat{v})$ for cluster C . \vec{e} denotes the eigenvector belonging to the largest eigenvalue λ of the corresponding covariance matrix. We then construct two new BRDFs:

$$f_1(\vec{a} + \tau\lambda\vec{e}; \hat{u}, \hat{v}) \quad \text{and} \quad f_2(\vec{a} - \tau\lambda\vec{e}; \hat{u}, \hat{v}), \quad (6.2)$$

where τ is a scaling factor to adapt λ to a moderate value. Two new clusters C_1 and C_2 are generated by distributing every lumitexel \mathcal{L}_i of cluster C either to C_1 if $E_{f_1}(\mathcal{L}_i) < E_{f_2}(\mathcal{L}_i)$, or to C_2 otherwise. In the next step, f_1 and f_2 are fitted to best approximate the lumitexels in the new clusters.

6.3 Reclustering

Because the parameters of the BRDF fitted to a multi-material cluster are not necessarily the center of the parameters of the contained materials and due to improper scaling of λ and other reasons like noise, the performed split will not be optimal and the two new clusters may not be clearly separated, e.g. in the case of two distinct materials some lumitexels belonging to one material may still be assigned to the cluster of the other material.

A better separation can be achieved by iterating the procedure of distributing the lumitexels \mathcal{L}_i based on $E_{f_1}(\mathcal{L}_i)$ and $E_{f_2}(\mathcal{L}_i)$, and then fitting the BRDFs again.

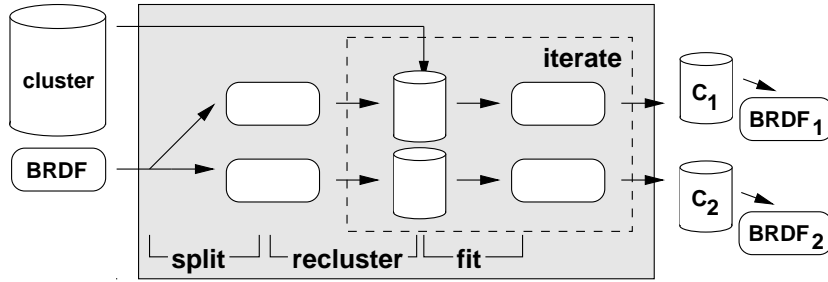


Figure 6.1: *Split-recluster-fit process (SRF). The initial BRDF is split into two new BRDFs using the covariance matrix. The lumitexels from the initial cluster are distributed according to their distance to the BRDFs. Then we fit the BRDF again to each new cluster. We now iterate the reclustering and fitting until the resulting BRDFs and cluster have converged.*

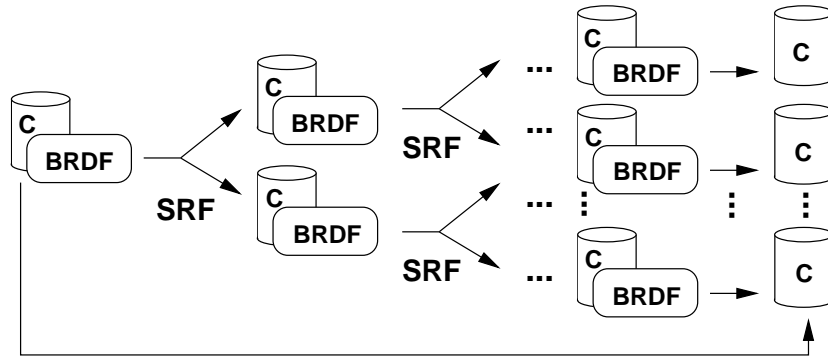


Figure 6.2: *The complete splitting and reclustering algorithm including the global reclustering, which is similar to the recluster-fit iteration, only that all lumitexels are distributed among all clusters.*

The iteration stops when the number of lumitexels in the generated cluster does not change any more. In our experiments this reclustering operation leads to a clear separation of materials and is done after each split. The split-recluster-fit (SRF) process is visualized in Figure 6.1.

When more than two clusters have been generated by successive binary splits and a new material is clearly distinguished, it is helpful to clean the other clusters, which were not involved in the last split, from all lumitexels belonging to the newly discovered material. This can be done in a global reclustering step by redistributing all initial lumitexels \mathcal{L}_i to the cluster C_j with

$$j = \underset{k}{\operatorname{argmin}} E_{f_k}(\mathcal{L}_i). \quad (6.3)$$

And again, the BRDFs of all involved clusters have to be refitted. This global reclustering is repeated several times to clearly separate the materials. We stop this iteration when the percentage of change is smaller than some ϵ , or a maximum number of iterations is reached. The complete splitting and reclustering algorithm is depicted in Figure 6.2.

6.4 Termination of the Splitting Process

We still have to decide when to stop the splitting process. To do this we require the user to input the estimated number of materials $|M|$. We stop the splitting and clustering process after $2|M| - 1$ clusters have been created. We use this additional number of clusters to compensate for the often noisy and not absolutely accurate radiance samples (e.g. slightly wrong normals, noise in the images, mis-registration, etc.).

This means that we do not have a one to one mapping between actual materials and clusters. This is not crucial since the projection, which we will present in the next section, uses a weighted sum of several BRDFs to accurately represent every lumitexel.

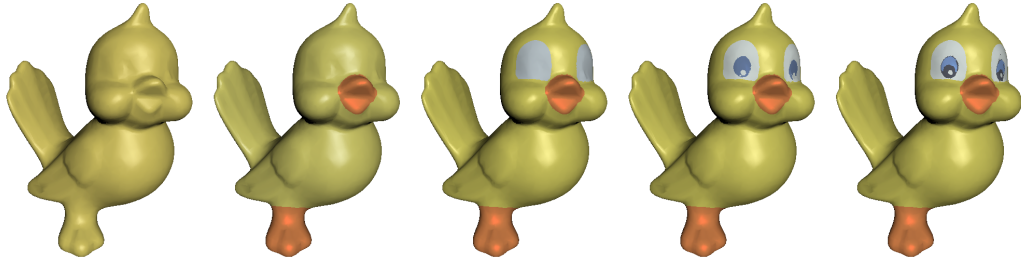


Figure 6.3: *The clustering process at work. In every image a new cluster was created. The object was reshaded using only the single BRDFs fitted to each cluster before the projection into a basis of multiple BRDFs.*

7 Projection

As can be seen in Figure 6.3 the representation of an object by a collection of only a few clusters and BRDFs make the virtual object look flat because real surface exhibit changes in the reflective properties even within a single material. These changes cannot be represented by a single BRDF per cluster since all lumitexels within the cluster would be assigned the same BRDF parameters.

To obtain truly spatially varying BRDFs we must find a specific BRDF for each lumitexel. But the sparse input data does not allow to fit a reliable or even meaningful BRDF to a single lumitexel because each lumitexel consists of only a few radiance samples. In addition, you would need to acquire a highlight in every lumitexel to reliably determine the specular part, as already explained in Section 6.1.

The solution is to project each lumitexel into a basis of BRDFs (see Section 7.1). The BRDF f_{π_i} of a lumitexel \mathcal{L}_i is represented by the linear combination of m BRDFs f_1, f_2, \dots, f_m :

$$f_{\pi_i} = t_1 f_1 + t_2 f_2 + \dots + t_m f_m, \quad (7.1)$$

with t_1, t_2, \dots, t_m being positive scalar weights. This forces the space of solutions (i.e. the possible BRDFs for a pixel) to be plausible since the basis BRDFs are already reliably fitted to a large number of radiance samples.

Given the BRDFs, the weights have to be determined for each lumitexel. Let $r_{j=1\dots|\mathcal{L}_i|}$ be the radiance values of the lumitexel \mathcal{L}_i . The weights are found by a least square optimization of the following system of equations using singular-value decomposition:

$$\begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{|\mathcal{L}_i|} \end{pmatrix} = \begin{pmatrix} \tilde{f}_1(\hat{u}_1, \hat{v}_1) & \tilde{f}_2(\hat{u}_1, \hat{v}_1) & \cdots & \tilde{f}_m(\hat{u}_1, \hat{v}_1) \\ \tilde{f}_1(\hat{u}_2, \hat{v}_2) & \tilde{f}_2(\hat{u}_2, \hat{v}_2) & \cdots & \tilde{f}_m(\hat{u}_2, \hat{v}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{f}_1(\hat{u}_{|\mathcal{L}_i|}, \hat{v}_{|\mathcal{L}_i|}) & \tilde{f}_2(\hat{u}_{|\mathcal{L}_i|}, \hat{v}_{|\mathcal{L}_i|}) & \cdots & \tilde{f}_m(\hat{u}_{|\mathcal{L}_i|}, \hat{v}_{|\mathcal{L}_i|}) \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{pmatrix}, \quad (7.2)$$

with $\tilde{f}(\hat{u}, \hat{v}) := f(\hat{u}, \hat{v})u_z$. Compared to the non-linear fitting of BRDF model parameters (see Section 5.2), we now have a linear problem to solve with a smaller degree of freedom and even more constraints. Above equation shows only the system for one color channel, whereas the weights t_i have to be the same for all channels. In contrast to this, BRDF parameters would require a distinct set of parameters per channel.

The least square solution may contain negative values for some t_k . But negative weights may result in an oscillating BRDF that represents only the given radiance sample accurately but will produce unpredictable values for other viewing and light directions, we therefore set t_k to zero and compute another least square solution for the remaining t 's, until all t 's are positive.

7.1 Basis BRDFs

The next question is how to determine the set of basis BRDFs. Since the changes of the surface properties within one material tend to be small, a distinct set of basis BRDFs is assigned to each cluster.

Finding the optimal set of BRDFs f_1, f_2, \dots, f_m , that minimizes the error

$$E_\pi(C) = \frac{1}{|C|} \sum_{\mathcal{L}_i \in C} E_{f_{\pi_i}}(\mathcal{L}_i) \quad (7.3)$$

for a cluster C , where f_{π_i} denotes the least square projection of the lumitexel \mathcal{L}_i as defined in Equation 7.1, is a problem of principal function analysis (PFA) (see [27]). Principal function analysis is closely related to principal component analysis (PCA) with the important difference that functions f_m are optimized instead of vectors. Unfortunately, the PFA does not reduce to a simple eigenvalue problem as PCA does. To minimize $E_\pi(C)$, we again perform a least square optimization using the Levenberg-Marquardt method, this time fitting m BRDFs simultaneously. Within each iteration we recompute the projection f_{π_i} of lumitexel \mathcal{L}_i into the currently estimated basis.

As for every optimization problem the initial parameters (BRDFs) are quite important. For a given cluster C , we use the following BRDFs as a basis:

- f_C , the fitted BRDF of cluster C ,
- the BRDFs of spatially neighboring clusters to match lumitexels at cluster boundaries,
- the BRDFs of similar clusters with respect to the material,
- and two BRDFs based on f_C , one with slightly increased and one with decreased diffuse component ρ_d and exponent N .

In our experiments it turned out that this initial basis together with the projection already produces very good results with small errors. In most cases the PFA computed almost negligible changes to the initial BRDFs. This is to be expected because the initially chosen basis constructed through splitting and clustering already approximates the material properties quite well.

8 Rendering

As explained in Section 4.1 we know the position of every lumitexel, as well as the triangle it belongs to and the 2D coordinates within that triangle.

This information can then be used to generate an index texture for the full object. For every texel, that texture contains an index to the cluster it belongs to. Then we generate a weight texture map for every cluster that stores the weights resulting from the projection into the basis BRDFs. The parameters for the basis BRDFs of every cluster are stored in a small table.

Raytracing such an object is very simple, since for every point on the object that is raytraced we can simply look up the cluster the texel belongs to. Then we evaluate the basis BRDFs for the local light and viewing direction and compute the weighted sum using the weight texture map. So rendering basically reduces to evaluating a few BRDFs per pixel. Another big advantage of this representation is that mip-mapping can easily be used. Since the weighted sum is just a linear operation, the weights of neighboring texels can simply be averaged to generate the next coarser mip-map level.

If the original images are of high resolution and hence the object is sampled very densely, point sample rendering using forward projection is a viable alternative. It completely avoids the generation of texture maps and the resulting data can be used with almost no further processing. This method is used to display our results.

9 Results

We applied our algorithm to three different objects consisting of different materials with varying reflection properties in both the diffuse and the specular part. The model of the angels was generated by extracting an isosurface of a computer tomography scan. The geometry of all other models was captured using a structured light 3D scanner. Some statistics about the meshes and the number of acquired views are listed in Table 9.1. Acquisition of 20 views (each needing about 15 photographs) takes approx. 2.5h. The high dynamic range conversion and the registration with the 3D model takes about 5h but is a completely automated task. The clustering and the final projection takes about 1.5h.

In Figure 6.3 you can see how five successive split operations partition the lumitexels of the bird into its five materials. The splits were performed as described in Section 6. Only the per-cluster BRDFs determined by the clustering process are used for shading, making the object look rather flat. After performing the projection step every lumitexel is represented in a basis of four BRDFs, now resulting in a much more detailed and realistic appearance, see Figure 9.2.

The bust in Figure 9.1 shows another reconstructed object with very different reflection properties. The bronze look is very well captured.

In Figure 9.3 you can see a comparison between an object rendered with an acquired BRDF (using the projection method) and a photograph of the object. You can see that they are very similar, but differences can be seen in highlights and in places where not enough radiance samples were captured. Capturing more samples will increase the quality. Furthermore our rendering algorithm does not include shadow computation, e.g. making the grooves in the hair look flat.

Another difference is due to the fact that the diffuse color of one lumitexel may not be represented in any of the constructed clusters because the number of lumitexels belonging to the same material can be so small that they nearly vanish in the mass of lumitexels of the cluster they are currently assigned to. This effect can for example be observed at the mouth of the larger angel which in reality exhibits a much more saturated red, see Figure 9.3.

In Table 9.1 we list RMS errors computed between all the radiance samples

model	T	V	L	R	C	B	1-RMS	C-RMS	P-RMS	F-RMS
angels	47000	27	1606223	7.6	9	6	.2953	.1163	.1113	.1111
bird	14000	25	1917043	6.3	5	4	.1513	.0627	.0387	.0387
bust	50000	16	3627404	4.2	3	4	.1025	.0839	.0583	.0581

Table 9.1: *This table lists the number of triangles (T) of each model, the number of views (V) we used to reconstruct the spatially varying BRDFs, the number of acquired lumitexels (L) and the average number of radiance samples (R) per lumitexel, the number of partitioned material clusters (C), the number of basis BRDFs (B) per cluster, the RMS error for a single average BRDF (1-RMS), the RMS error when using per-cluster BRDFs, the RMS error after projecting every lumitexel into the basis of BRDFs, and finally the RMS error after doing a PFA on the basis BRDFs and projecting every lumitexel into the new basis.*

of a model and the reconstructed BRDFs. You can see that the error considerably decreases when going from one average BRDF to per-cluster BRDFs and then to per-pixel BRDFs (using projection). As already mentioned the PFA only slightly changes the RMS error.

Generally it can be said that for all the models only a few clusters were needed to accurately represent all the materials since the projection takes care of material changes. In our experiments even Lafortune BRDFs consisting of a single lobe were sufficient to form good bases for the clustering and projection.

The projection method also compensates for imprecise normals, and hence no refitting of the normals is needed. Using exactly reconstructed normals for example by applying a shape-from-shading approach such as the one by Rushmeier et al. [21] may yield even better results.

Due to the lack of a test object that had a single base color but varying specularly, we experimented with artificially generated data. The tests proved that our clustering algorithm is also able to clearly distinguish materials that have the same color but different specularly, even when noise was introduced in the data.



Figure 9.1: A bronze bust rendered with a spatially varying BRDF, which was acquired with our reconstruction method.



Figure 9.2: This image shows the bird with the spatially varying BRDF determined by projecting each lumitexel into a basis of BRDFs. Note the subtle changes of the materials making the object look realistic.

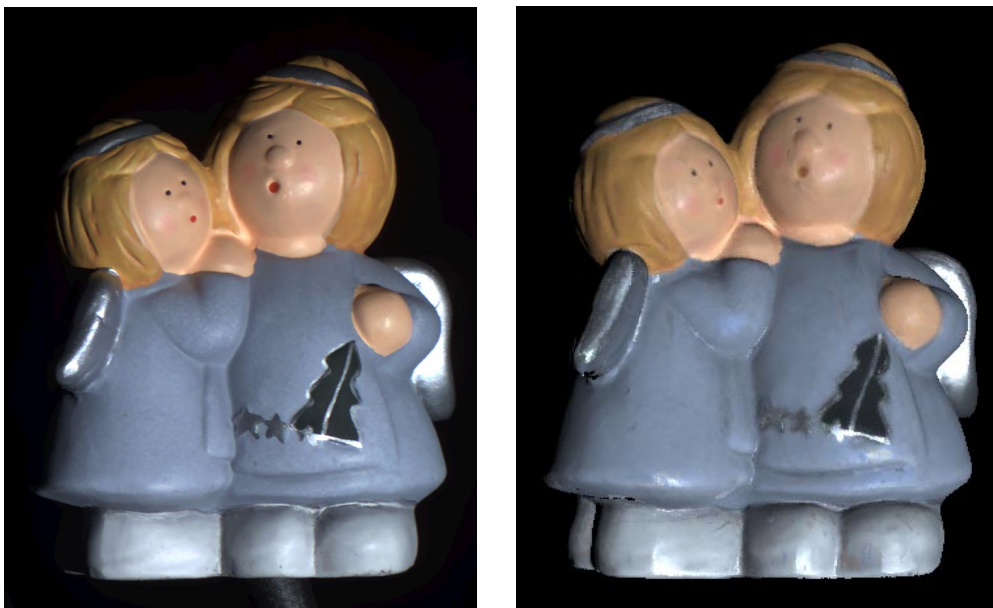


Figure 9.3: *Left side: Photograph of model. Right side: Model with acquired BRDF rendered from nearly the same view with similar lighting direction. The difference in the hair region is due to missing shadow computations during rendering.*

10 Conclusions and Future Work

We have presented an algorithm and demonstrated a system for reconstructing a high-quality spatially varying BRDF from complex solid objects using only a small number of images. This allows for accurately shaded, photorealistic rendering of these objects from new viewpoints and under arbitrary lighting conditions.

The output of our algorithm also allows to modify the object's geometry while preserving material properties, since the fitted BRDFs are represented on a per-textel basis and do not change with the geometry.

Both the number of input views required by our algorithm and the size of the output data are very small compared to previous approaches for representing real-world objects, like surface light fields or reflection fields which needed up to 600 images [27].

We have demonstrated the quality and accuracy of our approach, by applying it to different objects. The resulting spatially varying BRDFs accurately represent the original materials.

Until now interreflections within the object are not considered, but it should be easy to remove the effects of interreflections by simulating secondary reflection using the results obtained by the presented algorithm, or e.g. using techniques from [19].

We also want to investigate the possibility to do hardware accelerated rendering with the spatially varying BRDFs. Since our data can be represented as texture maps and the Lafortune model is computationally fairly simple, this should be easily possible, e.g. using techniques from [8] or from [27].

11 Acknowledgements

We would like to thank Kolja Kähler, Christian Rössel, Mario Botsch and the IMP Erlangen for acquiring the 3D meshes used in this paper. Thanks also to Hartmut Schirmacher for proofreading and valuable discussions.

Bibliography

- [1] K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, January 1999.
- [2] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the Reflectance Field of a Human Face. In *Proc. SIGGRAPH*, pages 145–156, July 2000. ISBN 1-58113-208-5.
- [3] P. Debevec and J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proc. SIGGRAPH*, pages 369–378, August 1997.
- [4] P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH*, pages 11–20, August 1996.
- [5] M. Garland and P. Heckbert. Surface Simplification Using Quadric Error Metrics. In *Proc. SIGGRAPH*, pages 209–216, August 1997.
- [6] A. Gersho and R. Gray. *Vector Quantization and Signal Compression*. Kluwer Acad. Publishers, 1992.
- [7] S. Gortler, R. Grzeszczuk, R. Szelinski, and M. Cohen. The Lumigraph. In *Proc. SIGGRAPH*, pages 43–54, August 1996.
- [8] J. Kautz and H.-P. Seidel. Towards Interactive Bump Mapping with Anisotropic Shift-Variant BRDFs. In *Eurographics/SIGGRAPH Hardware Workshop*, pages 51–58, August 2000.
- [9] L. Kobbelt. Discrete fairing. In *Proc. of the 7th IMA Conf. on the Mathematics of Surfaces*, pages 101–131, 1996.
- [10] J. Koenderink, A. van Doorn, and M. Stavridi. Bidirectional Reflection Distribution Function expressed in terms of surface scattering modes. In *Proc. 4th Europ. Conf. on Computer Vision*, pages 28–39, 1996.

- [11] E. Lafortune, S. Foo, K. Torrance, and D. Greenberg. Non-Linear Approximation of Reflectance Functions. In *Proc. SIGGRAPH*, pages 117–126, August 1997.
- [12] H. Lensch, W. Heidrich, and H.-P. Seidel. Automated Texture Registration and Stitching for Real World Models. In *Pacific Graphics '00*, pages 317–326, October 2000.
- [13] M. Levoy and P. Hanrahan. Light Field Rendering. In *Proc. SIGGRAPH*, pages 31–42, August 1996.
- [14] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. on Information Theory*, IT-28:129–137, 1982.
- [15] R. Lu, J. Koenderink, and A. Kappers. Optical Properties (bidirectional reflectance distribution functions) of velvet. *Applied Optics*, 37(25):5974–5984, September 1998.
- [16] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, volume 1, 1967.
- [17] S. Marschner, S. Westin, E. Lafortune, K. Torrance, and D. Greenberg. Image-based BRDF Measurement Including Human Skin. In *10th Eurographics Workshop on Rendering*, pages 131–144, June 1999.
- [18] G. Miller, S. Rubin, and D. Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. In *9th Eurographics Workshop on Rendering*, pages 281–292, June 1998.
- [19] S. Nayar, K. Ikeuchi, and T. Kanade. Recovering Shape in the Presence of Interreflections. In *IEEE Int. Conf. on Robotics and Automation*, pages 1814–1819, 1991.
- [20] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press, 1992. ISBN 0-521-43108-5.
- [21] H. Rushmeier, G. Taubin, and A. Guézic. Applying Shape from Lighting Variation to Bump Map Capture. In *8th Eurographics Workshop on Rendering Workshop*, pages 35–44, June 1997.
- [22] Y. Sato, M. Wheeler, and K. Ikeuchi. Object Shape and Reflectance Modeling from Observation. In *Proc. SIGGRAPH*, pages 379–388, August 1997.

- [23] H. Schirmacher, W. Heidrich, M. Rubick, D. Schiron, and H.-P. Seidel. Image-Based BRDF Reconstruction. In *Proc. of the 4th VMV Conference*, pages 285–292, November 1999.
- [24] K. Torrance and E. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57(9), 1967.
- [25] G. Ward Larson. Measuring and Modeling Anisotropic Reflection. In *Proc. SIGGRAPH*, pages 265–272, July 1992.
- [26] S. Westin, J. Arvo, and K. Torrance. Predicting Reflectance Functions From Complex Surfaces. In *Proc. SIGGRAPH*, pages 255–264, July 1992.
- [27] D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. Surface Light Fields for 3D Photography. In *Proc. SIGGRAPH*, pages 287–296, July 2000.
- [28] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse Global Illumination: Recovering Reflectance Models of Real Scenes From Photographs. In *Proc. SIGGRAPH*, pages 215–224, August 1999.
- [29] Z. Zhang. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Int. Conf. on Computer Vision*, pages 666–673, September 1999.