IMAGE-BASED EXTRACTION OF MATERIAL REFLECTANCE PROPERTIES OF A 3D OBJECT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET ERKUT ERDEM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

JULY 2003

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Volkan Atalay
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Volkan Atalay _____

Assoc. Prof. Dr. Sibel Tarı _____

Assoc. Prof. Dr. Yasemin Yardımcı _____

Assist. Prof. Dr. Uğur Güdükbay _____

Dr. Adem Yaşar Mülayim _____

ABSTRACT

IMAGE-BASED EXTRACTION OF MATERIAL REFLECTANCE PROPERTIES OF A 3D OBJECT

Erdem, Mehmet Erkut

M.Sc., Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Volkan Atalay

July 2003, 45 pages

In this study, an appearance reconstruction method based on extraction of material reflectance properties of a three-dimensional (3D) object from its two-dimensional (2D) images is explained. One of the main advantages of this system is that the reconstructed object can be rendered in real-time with photorealistic quality in varying illumination conditions. Bidirectional Reflectance Distribution Functions (BRDFs) are used in representing the reflectance of the object. The reflectance of the object is decomposed into diffuse and specular components and each component is estimated seperately. While estimating the diffuse components, illumination-invariant images of the object are computed from the input images, and a global texture of the object is extracted from these images by using surface particles. The specular reflectance data are collected from the residual images obtained by taking difference between the input images and corresponding illumination-invariant images, and a Lafortune BRDF model is fitted to these data. At the rendering phase, the diffuse and specular components are blended into each other to achieve a photorealistic appearance of the reconstructed object.

Keywords: photorealism, bi-directional reflectance distribution functions, appearance reconstruction, particle systems, three-dimensional modeling, texture mapping, computer vision.

ÖZ

ÜÇ BOYUTLU NESNELERİN MADDE YANSITIRLIK ÖZELLİKLERİNİN İMGEYE DAYALI OLARAK ÇIKARILMASI

Erdem, Mehmet Erkut
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Assoc. Prof. Dr. Volkan Atalay

Temmuz 2003, 45 sayfa

Bu çalışmada, üç boyutlu bir nesnenin iki boyutlu imgelerinden nesnenin maddesel yansıtırlık özelliklerinin çıkarılmasına dayanan bir görüntü geriçatımı metodu anlatılmaktadır. Bu sistemin önemli avantajlarından biri, geriçatılan nesnenin değişken ışıklandırma koşullarında fotogerçekçi olarak eş zamanlı oynatılabilmesidir. Nesnenin yansıtırlık özellikleri Çift-yönlü Yansıtırlık Dağılım Fonksiyonları (ÇYDF) kullanılarak ifade edilmektedir. Nesnenin yansıtırlığı düzgün (specular) ve dağınık (diffuse) bileşenlere ayrıştırılmakta ve her bir bileşen ayrı ayrı hesaplanmaktadır. Dağınık (diffuse) bileşenlerin hesaplanması için girdi imgelerinden nesnenin ışıklandırmadan bağımsız imgeleri oluşturulur ve bu imgelerden yüzey parçacıkları kullanılarak objenin geniş çaplı doku kaplaması çıkartılır. Girdi imgelerinden karşılık gelen ışıklandırmadan bağımsız imgelerin çıkartılmasıyla elde edilen artık (residual) imgelerden düzgün (specular) yansıtırlık verisi toplanır ve bu veriye bir Lafortune Çift-yönlü Yansıtırlık Dağılım Fonksiyon modeli bindirilir. Oynatılma aşamasında, geriçatılan nesnenin fotogerçekçi görüntüsünü elde etmek için dağnık (diffuse) ve düzgün (specular) bileşenler bütünleştirilir.

Anahtar Kelimeler: fotogerçeklik, çift-yönlü yansıtırlık dağılım fonksiyonları, görüntü geriçatımı, parçacık sistemleri, üç boyutlu modelleme, doku kaplama, bilgisayarla görme.

To my family

ACKNOWLEDGMENTS

I would like to thank to Volkan Atalay for his supervision, guidance and support throughout the development of this thesis. I would also like to thank my family for their encouragement and assistance to finish my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Three dimensional (3D) computer graphics has a wide range of application areas varying from 3D virtual environments, computer games to visualisation of cultural heritage. With the use of Internet, 3D visualisation of the products becomes very important for e-commerce applications. The most common way of creating 3D models is manual design. However, it is cost expensive and time consuming. Therefore, several techniques for automatic reconstruction of 3D models of real objects are proposed. This requires a process of reconstructing both geometry and the appearance of the object. Excessive research has been done for obtaining the geometry of an object, each using different approaches such as range imaging, shape from silhouette, shape from shading, etc. However, there are relatively fewer number of studies on obtaining appearance of the objects. While the results obtained by using simple appearance models are far from photo-realism, real-time rendering of the objects is not possible when complex models are used.

Physically, the interaction of light with a surface can be described as the photon hitting the surface and the resulting photon leaving that surface. This can be parametrized with 12 degrees of freedom as follows:

$$(x, y, t, \theta, \phi, \lambda)_{in} \rightarrow (x, y, t, \theta, \phi, \lambda)_{out}$$

where $(x, y)_{in}$ is the position the photon hits with an incident angle $(\theta, \phi)_{in}$ and wavelength λ_{in} at time t_{in}, and $(x, y)_{out}$ is the position the photon leaves with an exitant angle $(\theta, \phi)_{out}$ and wavelength λ_{out} at time t_{out}.

After ignoring the dependences on time and wavelength $(t_{in}, \lambda_{in}, t_{out}, \lambda_{out})$, the appearance of a volume under arbitrary illumination conditions can be described as a function of eight degrees of freedom. We can also simplify this 8D function to 6D by ignoring subsurface scattering, i.e we assume the incoming light does not leave the surface from a different position ($(x, y)_{in} = (x, y)_{out}$). This 6D function can be described as either a Bidirectional Reflectance Distribution Function (BRDF) that varies for each surface point (i.e a Spatially-Varying BRDF) or a texture map that varies with the illumination and viewing directions (i.e a Bidirectional Texture Function, BTF). If we ignore the dependence on illumination direction, we can come up with light fields and its variants (surface light fields, lumigraphs, etc.). In addition, if we assume that the surface is perfectly flat and diffuse (i.e texture map), we can reach to two degrees of freedom. An extension to texture map is bump map where the surface normals of each point can vary. The taxonomy described here is inspired from [1]. The overall taxonomy of appearance modeling presented here is given in Figure 1.1.

As it is clear from the given taxonomy, the most common method for appearance reconstruction is texture map due to its simplicity [3, 4, 5, 6, 2, 7]. In texture mapping related studies, the model is represented as a triangular mesh, and each triangle is associated with one of the images for texture extraction. However, this approach may cause discontinuities on the triangle boundaries if adjacent triangles are associated with different images. To overcome this problem, mostly blending [5, 6] is used. Also alternatively, surface particles concept can be used in extracting the texture [2]. However, the main weakness of texturing is that it does not capture the true physical characteristics of the object surface, i.e. it ignores changes in illumination and viewing conditions. Therefore, some more complex but physically more accurate models are introduced. In this study, we use a physically more plausible and popular model, Bidirectional Reflectance

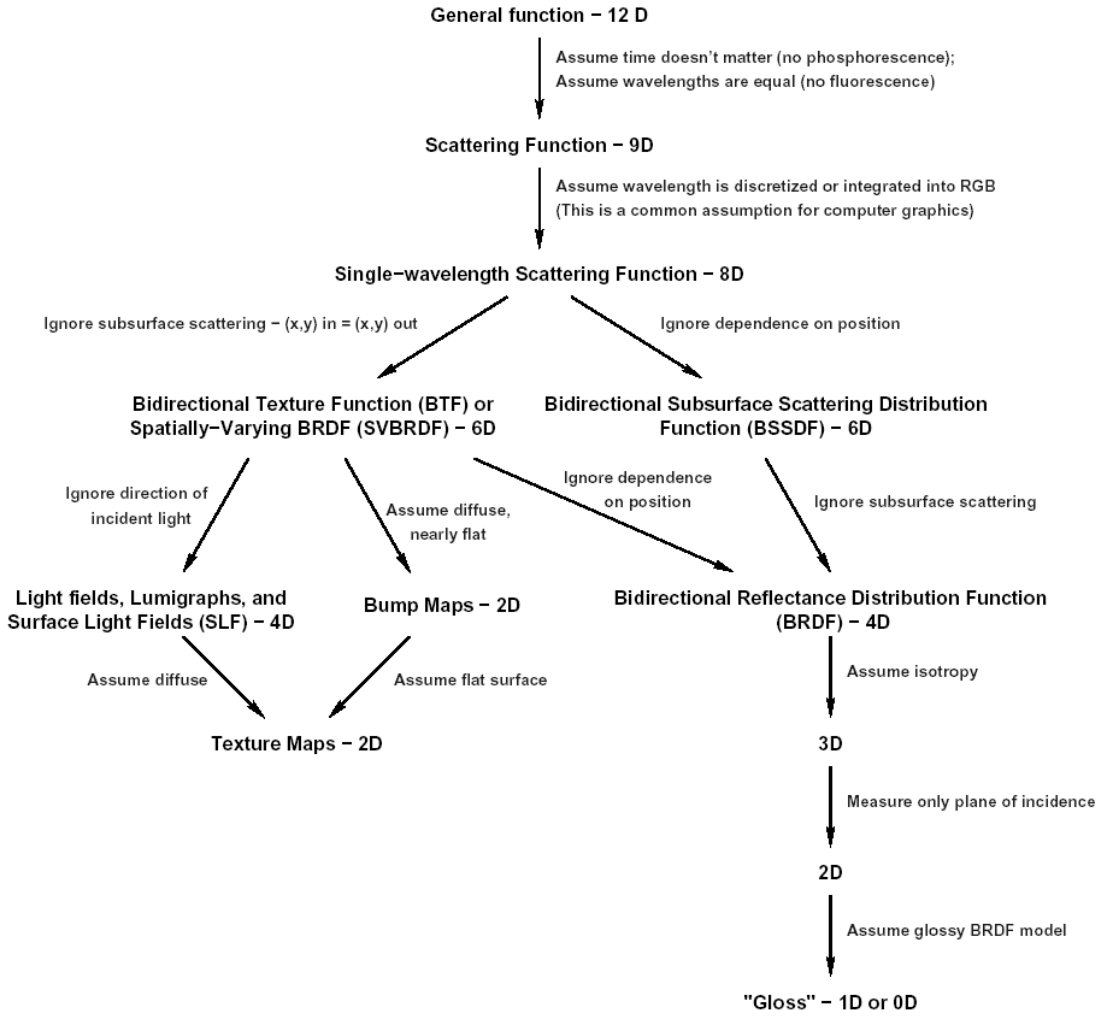


Figure 1.1: Taxonomy of Appearance Modeling (from Appearance Models for Computer Graphics and Vision Lecture Notes [1]).

Distribution Functions (BRDFs) to represent the appearance of the object.

Traditionally, BRDFs are measured by using special devices known as gonioreflectometers. However, recently image-based techniques are introduced where there is no need to use any special device. In general, these techniques are used for modeling any homogenous or spatially varying surface. For example, Debevec et al. [8] use BRDFs for rendering architectures in varying illumination conditions. Even BRDFs can be used for modeling the human skin [9, 10]. These image based measurement techniques can be mainly grouped in two categories. While some try to acquire the reflectance property from just one image [11, 12], the others try to capture this information by using multiple views [8, 9, 13, 14].

1.2 Purpose and Improvements Achieved

In this study, our goals are extraction of material reflectance property of a 3D object from a set of images when the geometry of the object is known and rendering the reconstructed object in real-time with photorealistic quality in varying illumination conditions. This technique can be easily adopted to image-based model reconstruction frameworks such as given in [2] to improve the appearance quality of the obtained 3D models. In this study, we work on artificial models since we do not have necessary equipments in our laboratory.

Since in most of the local reflection models used in computer graphics, reflection is considered as a combination of a diffuse and aspecular component [15], the key idea of the described method which makes real-time rendering possible is decomposing reflectance into diffuse and specular components and estimating these components separately. While we are storing the diffuse component in a global texture, the specular component is represented as a single BRDF. This process can be thought as using Spatially-Varying BRDFs with homogeneous specular components. The system diagram of appearance reconstruction phase is as shown in Figure 1.2. The diffuse components of the object are estimated

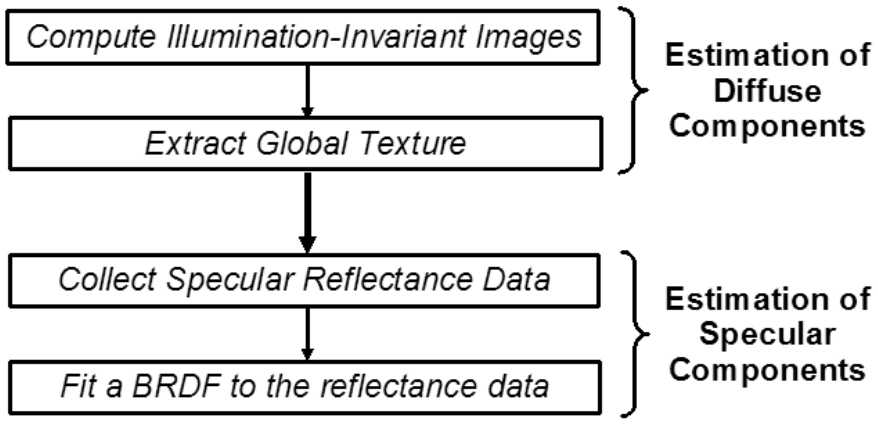


Figure 1.2: Overall system diagram of appearance reconstruction phase.

by using the illumination-invariant images of the object. These images are computed from the input images as stated in the work of Rocchini et al. [6]. Then, the

global texture of the object is extracted from these images by using the surface particles concept proposed by U. Yilmaz [7].

The specular reflectance data are collected from the residual images obtained by taking difference between the input images and corresponding illumination-invariant images. Collecting reflectance data is performed by a similar process to the one proposed by Lensch et al. [13], but since the residual images are used, the collected reflectance data contains only specular components. Then, a BRDF model is fitted to these data.

Evaluation of BRDFs is very expensive and time consuming, therefore previous frameworks based on appearance reconstruction of 3D objects using BRDFs or its extensions do not support real-time rendering [13, 14]. Recently, there is some research on real-time rendering of BRDFs where texture maps are used to approximate the behaviour of BRDF models [16, 17]. The main contribution of this study is that the reconstructed 3D object can be rendered in real-time with photo-realistic quality. Rendering of the reconstructed object is achieved in two passes by blending diffuse and specular components. In the first pass, the object is rendered using global texture containing the diffuse component, and in the second pass the BRDF representing the specularity of the object is rendered by the method proposed by Kautz et al. [16]. The system diagram of real-time rendering process is as shown in Figure 1.3.

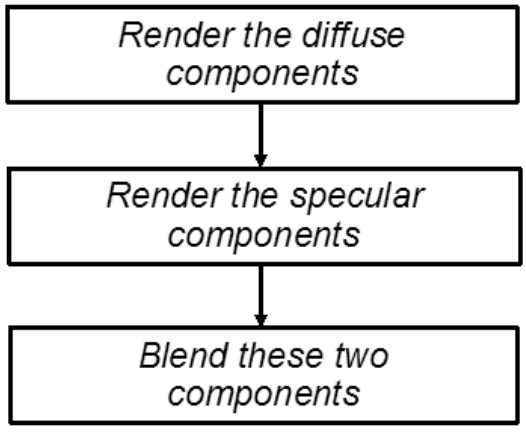


Figure 1.3: Overall system architecture of real-time rendering process.

1.3 Organization of the Thesis

The organization of the thesis is as follows: In Chapter 2, reflectance and BRDFs are decribed. In Chapter 3, we discuss the details of our appearance reconstruction method, estimation of diffuse and specular components. In Chapter 4 interactive rendering process is presented in detail. Chapter 5 presents the experimental results and finally in Chapter 6 the proposed system is discussed.

CHAPTER 2

REFLECTANCE AND BIDIRECTIONAL REFLECTANCE DISTRIBUTION FUNCTIONS

2.1 Reflection

In general, when light interacts with an object, three kinds of interactions may occur. The object may absorb some amount of light, or may transmit some of it to another medium. In addition, some of the light may be reflected from the object. These interactions depend on physical characteristics of both the object and the incoming light. For example, the amount of reflection of the incoming light when it interacts with an opaque object like sand is different than the amount when it interacts with a smooth reflective surface like a mirror. However, since the light is a kind of energy, from the conservation of the energy rule, the amount of incoming light must equal to the sum of absorbed light, transmitted light and the reflected light.

As it may be noticed, reflection has an important role in appearance modeling. In this study, we use BRDFs to represent the reflectance property of the object. The use of BRDFs is becoming an important topic in computer graphics and in short they are the functions that describes how light is reflected when it interacts

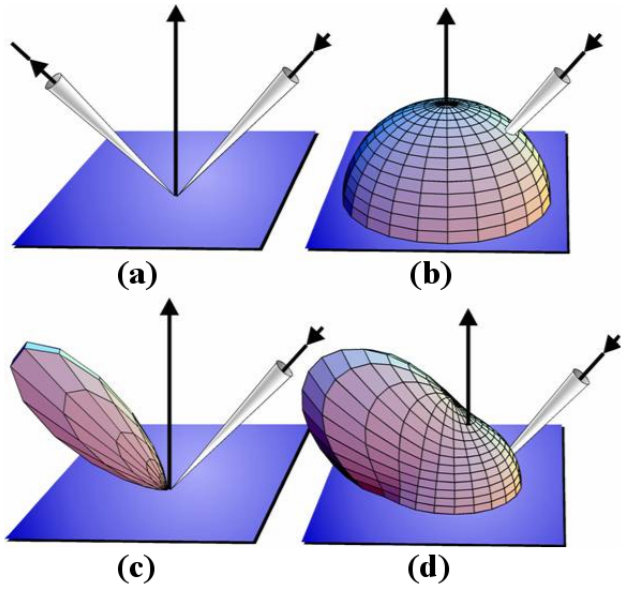


Figure 2.1: BRDFs of (a) ideal specular, (b) ideal diffuse, (c) rough specular, and (d) nonideal diffuse surfaces.

with a surface. While BRDFs ignore some other concepts such as subsurface scattering, fluorescence, phosphorescence and polarization, they still give more realistic results than the other conventional methods.

2.2 Bidirectional Reflectance Distribution Functions (BRDFs)

The amount of reflection depends on the material property of the object. In nature, materials show mostly either diffuse or specular reflection characteristics. While a perfect mirror surface reflects incoming light in one direction (Figure 2.1a), a rough specular surface reflects light in a distribution centered on the optical reflection direction (Figure 2.1c), or alternatively for an ideal diffuse surface, the incident light is scattered equally in all directions (Figure 2.1b), for the nonideal cases the reflection is based on a distribution (Figure 2.1d). Furthermore, the amount of reflection is a function of incoming light direction, viewing direction and the wavelength of the incoming light. As it can be observed in Figure 2.2, the hightlights in the object shifts whenever the light source position

(a) (b) (c)

Figure 2.2: Effect of viewing and incoming light directions on reflectance: (a) original scene, (b) scene when light source position changes, (c) scene from a different view.

is changed while the observer and the object remain fixed, or the observer moves to another position while the light source and the object remain fixed.

From the previous explanations, we can use the following function notation for BRDF, $f_\lambda(\mathbf{u}, \mathbf{v})$ where λ is the wavelength of the incoming light; $\mathbf{u}$ is the incoming light direction; and $\mathbf{v}$ is the viewing direction. Since in our study, we work on RGB color space, we can omit the wavelength (λ) in the function notation and use the BRDF as a 4D function for each color channel ($f_R(\mathbf{u}, \mathbf{v})$, $f_G(\mathbf{u}, \mathbf{v})$, and $f_B(\mathbf{u}, \mathbf{v})$).

Finding an efficient way to represent BRDFs is another difficult problem. In literature, there are mainly four classes for representation of BRDFs:

- **Tabular Representation:** In early studies, tabular representation of BRDFs are used to store the sampled values. Sampling reflectance values requires using special devices called gonioreflectometers that consist of a movable light source and a sensor. Inspite of its simplicity, since general BRDFs are 4D functions, the tables occupy large amount of storage.

Furthermore, sampling reflectance values is a relatively slow process.

- **Splines:** Splines can also be used to model BRDFs. The idea is by using control points, smooth parts can be represented with a spline. However, determining splines are relatively complex and fitted splines may not be so accurate.

- **Basis Functions:** BRDFs can be represented by means of some basis functions. These basis functions can be spherical harmonics, Zernike polynomials and as well as Fourier basis and wavelets.

- **Parametric Models:** In literature, several parametric BRDF models are proposed. Every model is formed to represent the nature of reflectance by using some physical principles. The main benefit of using parametric BRDF models over other representations is that it requires much less storage (i.e. only the fitted parameters). Another advantage is that these parameters can be determined by using small number of reflectance samples. On the other hand, these parametric models may not approximate all possible reflectance functions to a desired accuracy. Therefore, each model has its own weaknesses and strengths.

In this study, we choose to represent reflectance by a parametric model. Over many alternatives, Lafortune BRDF model [18] is used in our computations. It defines a multimodal distribution of reflectance with the following representation:

$$f(\mathbf{u}, \mathbf{v}) = \rho_d + \sum_i \left[C_{x,i}(u_x v_x + u_y v_y) + C_{z,i} u_z v_z\right]^{N_i} \tag{2.1}$$

where $\mathbf{u}$ is the incoming light direction, $\mathbf{v}$ is the viewing direction, i is the index of a lobe, ρ_d is the diffuse component, N_i is the specular exponent, $C_{x,i}$ and $C_{z,i}$ are the weighting factors where the ratio between $C_{x,i}$ and $C_{z,i}$ indicates the off-specularity of lobe i of the BRDF f.

As stated in the original work by Lafortune [18], the model has the following features:

- It has a compact representation, each primitive function is described by two coefficients and an exponent.
- The model can represent complex reflectance phenomena such as off-specular reflection, retro-reflection, anisotrophy and non-Lambertian diffuse reflection.
- The noise in the raw reflectance data can be handled by the functions.
- The functions are physically plausible.
- The model can be used in both local and global illumination algorithms efficiently.
- The model can approximate many number of reflectance function accurately.

In this study, we only use one lobe representation, so the coefficients ρ_d, N, C_x and C_z represent reflectance property of a single material.

CHAPTER 3

APPEARANCE RECONSTRUCTION

In this study, we assume the object to be reconstructed has homogeneous material property. By using the fact that reflection is a combination of diffuse and specular components, we estimate these components for each surface point individually. Basically, while the diffuse components are stored in a global texture, a single BRDF is fitted to represent the specularity of the whole object.

3.1 Estimation of Diffuse Components

Diffuse reflection is the view-independent component of the reflection. When light interacts with the surface of an object, the incident light is scattered in various directions. For the ideal case which assumes Lambertian surface, light is scattered equally in all directions. In our computations, we also assume a Lambertian surface.

In general, the diffuse component of a surface point is estimated as the minimum color value among the pixels in the acquired input images where the corresponding surface point is visible [19]. But, this initial estimation is inaccurate when the surface point belongs to a shadow area in one of the input images. In our approach, we store the diffuse component of each surface point in a global texture. This texture is extracted by using surface particles concept as proposed

in [2]. However, initially, an unshading phase proposed by Rocchini et al. [6] is applied to input images to remove the illumination effects like shadows and specular highlights. This unshading phase requires a system setup where six point light sources are placed around the camera at the known positions and during each shot only one of the light sources is activated and six images are acquired for each view.

3.1.1 Computing Illumination-Invariant Images

For each view, the illumination effects such as shadows, specular highlights, etc. can be eliminated by inspecting pixel values in the images. While the pixels having lower intensities correspond to shadow areas, the saturated pixels correspond to specular highlights. To obtain the corresponding illumination-invariant images for each view, we need to compute the diffuse component ρ_d of each surface point p that is visible in that view. The diffuse component can be reconstructed by assuming a Lambertian surface, i.e the intensity of the reflected ray is proportional to the cosine angle between the surface normal $\mathbf{n}$ and the incoming light direction $\mathbf{u}$. This can be formulated as a linear system of equations $\rho_d \mathbf{u}_i \cdot \mathbf{n} = c_i$ where $\mathbf{u}_i$ is incoming light direction, $\mathbf{n}$ is the surface normal and c_i is the observed color value in input image i as follows:

$$\rho_d \begin{bmatrix} u_{1x} & u_{1y} & u_{1z} \\ u_{2x} & u_{2y} & u_{2z} \\ & \vdots & \\ u_{ix} & u_{iy} & u_{iz} \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_i \end{bmatrix} \tag{3.1}$$

After removing the shadows and specular highlights in the input images, for a surface point this linear equation can be solved if at least three different color values are observed in the input images for each view. If this is not the case, we could not determine values of some of the pixels in the output image. While Rocchini [6] estimates the values of these pixels by interpolating from neighboring pixels, we do not need to fully reconstruct the illumination-invariant image since we use surface particle concept in texture extraction.

3.1.2 Extracting Texture for Diffuse Components

Once the illumination-invariant images are computed, the diffuse component of the object's surface appearance is stored in a global texture. While constructing the texture, the surface particles concept is used similar to one described in [2]. The model is considered to be a composition of surface particles with three attributes:

1. position, $\mathbf{P}(x, y, z)$,
2. normal, $\mathbf{N}(x, y, z)$,
3. color, $\mathbf{C}(r, g, b)$.

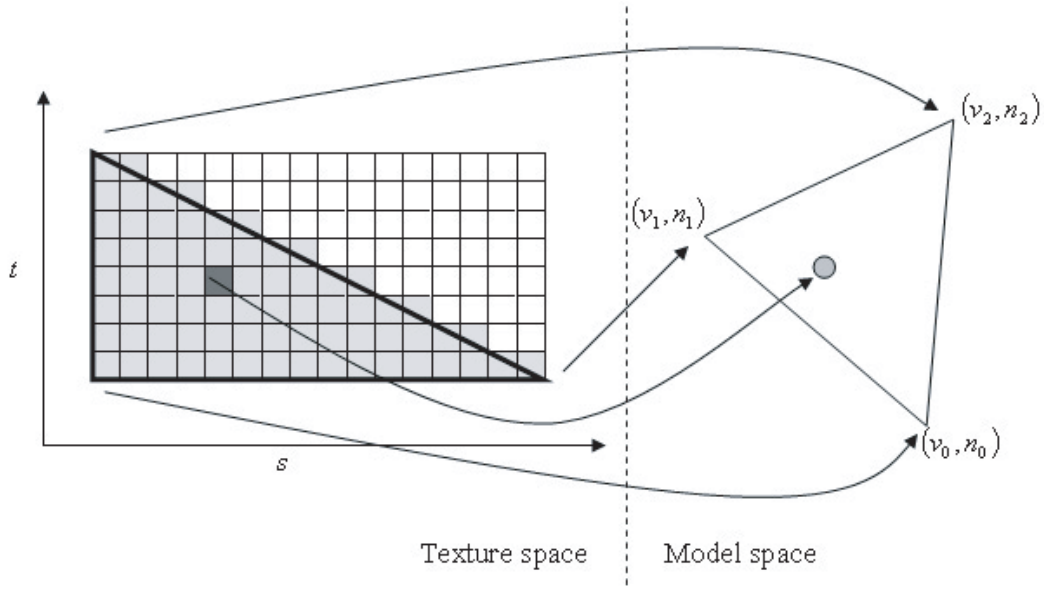


Figure 3.1: Association between the texture map and a surface particle (from [2]).

The main idea of using surface particle in texture extraction is that instead of assigning triangles to images, particles are assigned to images. By this way, the discontinuities on the triangle boundaries are eliminated. Each surface particle is associated with a pixel on the texture as shown in Figure 3.1. The position and the normal of each particle is determined from the texture space by using bilinear mapping Equation 3.2 and Equation 3.3 where s and t are the coordinates

of the texture element, $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ are the vertices of the triangle associated with that texture patch, and $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2$ are the normal vectors of the vertices of the triangle.

$$\mathbf{P}(x, y, z) = (1 - s - t)\mathbf{v}_0 + (s)\mathbf{v}_1 + (t)\mathbf{v}_2, \tag{3.2}$$

$$\mathbf{N}(x, y, z) = (1 - s - t)\mathbf{n}_0 + (s)\mathbf{n}_1 + (t)\mathbf{n}_2, \tag{3.3}$$

The color value for that particle can be determined from the image where the particle is visible and whose normal vector produces the minimum angle with the particle normal. The particle is projected to this image and the corresponding color in the image is assigned to the corresponding pixel in the global texture.

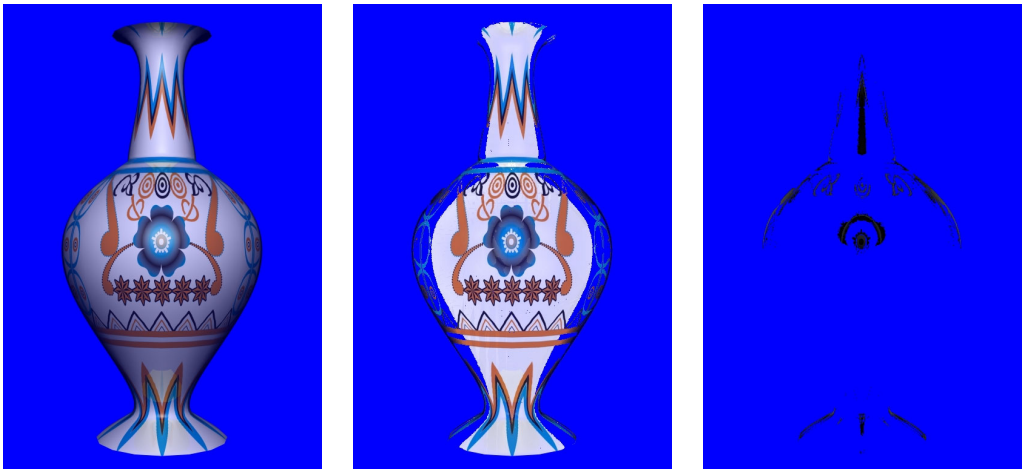
3.2 Estimation of Specular Components

Specular reflection is the view-dependent component of the reflection. Estimation of specular component requires mainly two steps: collecting the reflectance data from the images and fitting a single BRDF to the reflectance data.

3.2.1 Collecting Reflectance Data

During the collection of reflectance data, for each surface particle generated in the estimation of diffuse component phase, the corresponding reflectance data are collected from the residual images obtained by taking intensity difference between the input images and corresponding illumination-invariant images. For example, the residual image in Figure 3.2(c) is obtained from the input image and the corresponding illumination-invariant image in Figure 3.2(a) and (b).

The process of collecting reflectance data is similar to the one proposed by Lensch et al. [13], but since we decompose the reflectance as the combination of diffuse and specular components, the reflectance data obtained from the residual images contain only the specular reflectance. The idea is that for a set of surface



(a) (b) (c)

Figure 3.2: (a) One of the six input images, (b) computed illumination-invariant image, (c) corresponding residual image.

points in the object, the corresponding radiance samples are collected from each residual image where the point is visible.

The overall collecting reflectance data process is illustrated in Figure 3.3: For each surface particle P, a list of radiance samples R_j, each composed of the outgoing radiance r, the local viewing direction $\mathbf{v}$ and the local incoming light direction $\mathbf{u}$, is generated from residual images $S = \{I_0, \ldots, I_k\}$ where P is visible. The visibility test can be easily done by inspecting the angle between surface particle normal and the camera direction vector.

The local incoming light direction $\mathbf{u}$ and the viewing direction $\mathbf{v}$ can be easily determined for each image by using the position of light source $\mathbf{p}_l$, camera position $\mathbf{p}_c$, position $\mathbf{x}$ of the surface particle P and matrix $\mathbf{M}$ representing the local tangent space (see Appendix B) that the surface particle defines as follows:

$$\mathbf{u} = \mathbf{M}(\mathbf{p}_l - \mathbf{x}) \quad \mathbf{u} = \mathbf{M}(\mathbf{p}_c - \mathbf{x}) \tag{3.4}$$

The corresponding radiance value r can be determined by projecting $\mathbf{x}$ to the residual image. The radiance value r is calculated as follows:

$$r = \frac{c}{d^2} \tag{3.5}$$

where c is the color value at the projected pixel and d is the distance from the light source to surface particle P.

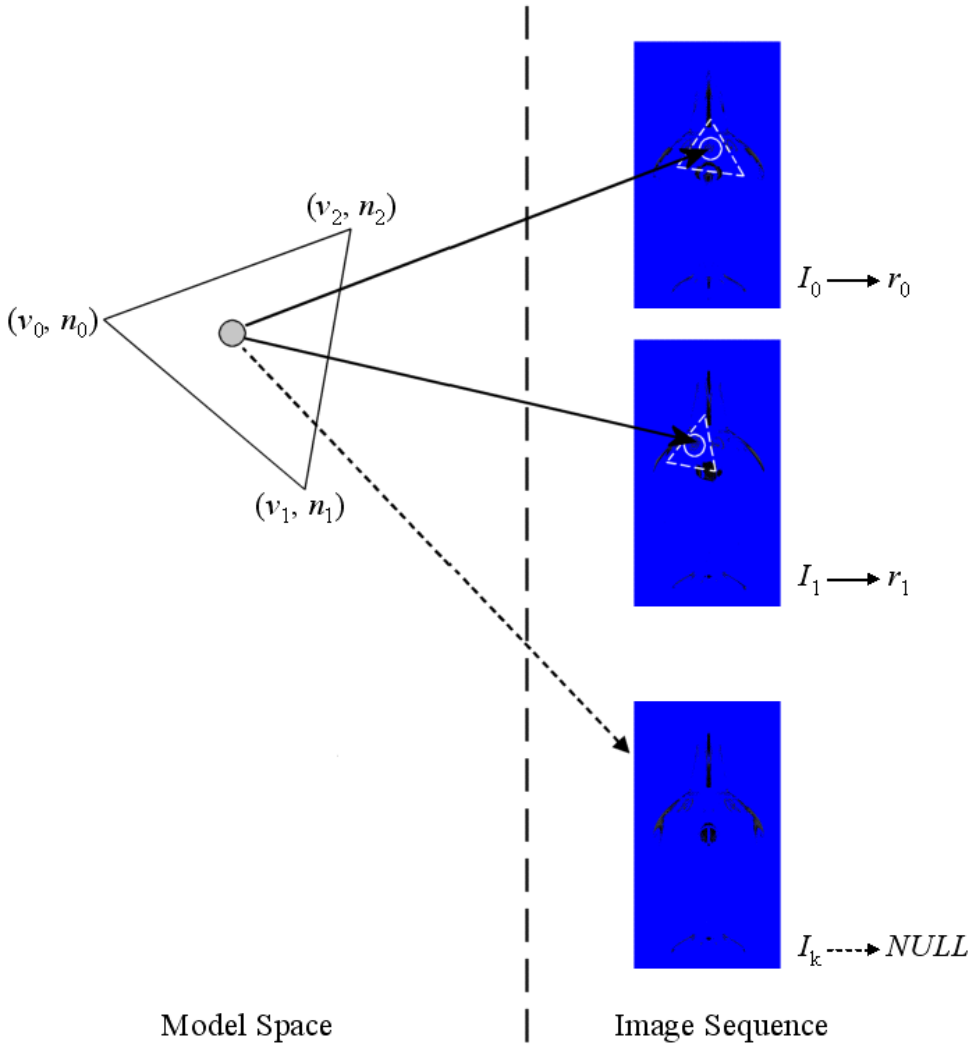


Figure 3.3: Collecting the reflectance data.

The overall algorithm for collecting the reflectance data from residual images is given in Algorithm 1.

Algorithm 1 Collecting Reflectance Data.

For all surface particle P at the position $\mathbf{x}$ and having normal $\mathbf{n}$

Generate the list of radiance samples for the surface particle P

For all residual image I_j where the surface particle P is visible

Project surface particle P to residual image I_j

Extract radiance r from the image I_j

Calculate local incoming light direction $\mathbf{u}$ and viewing direction $\mathbf{v}$

Generate the radiance sample R_j, $(r, \mathbf{u}, \mathbf{v})$

Add radiance sample R_j to the radiance samples list

End for

End for

3.2.2 BRDF Fitting

As mentioned in Section 2, we use Lafortune BRDF model to represent the specularity of the object. Lafortune model has four parameters as indicated in Equation 2.1. Since we work on RGB color space, three different models are used to represent the reflectance for each color channel. However, ρ_d is assumed to be zero since we are fitting only the specular data obtained from residual images. The remaining parameters can be determined by using a non-linear optimization method called Levenberg-Marquardt optimization technique. As stated in the original work by Lafortune [18], this method is well-suited for fitting non-linear BRDFs. In our computations, we use Levenberg-Marquardt implementation available in MATLAB Optimization Toolbox.

Levenberg-Marquardt optimization method requires an initial guess for each parameter. For all the reconstructed objects, the parameters are initiliazed with the values $C_x = -1, C_z = 1$ and $N = 10$, and according to reconstruction results, fitted parameters from this initial guess are observed to be fairly good in representing the specularity of each object.

CHAPTER 4

RENDERING

4.1 Interactive Rendering

Rendering of the reconstructed object is implemented by C programming language. Additionally, the following libraries are used:

- Graphics Library Utility Toolkit (GLUT) for user interface design,
- Open Graphics Library (OpenGL) for graphics rendering,
- JpegLib for image file operations.

In the rendering phase, we need to combine diffuse and specular components of the object to imitate the reflectance behaviour of its surfaces [15]. Therefore, the rendering is performed in two passes. In the first pass, the object is rendered using global texture containing the diffuse component, and in the second pass, the specular component is added to the diffuse component by blending. Since specular component is represented by a single BRDF, we need to render the model using fitted parameters.

4.1.1 Rendering the object with diffuse components

While rendering the object with only diffuse components, texture mapping is performed using the routines provided by OpenGL. The diffuse components are

stored as a patchwork of texture patterns as shown in Figure 4.1. Each texture pattern is associated with a triangle on the model and in the rendering phase each triangle is rendered with the associated texture coordinates in the patchwork.

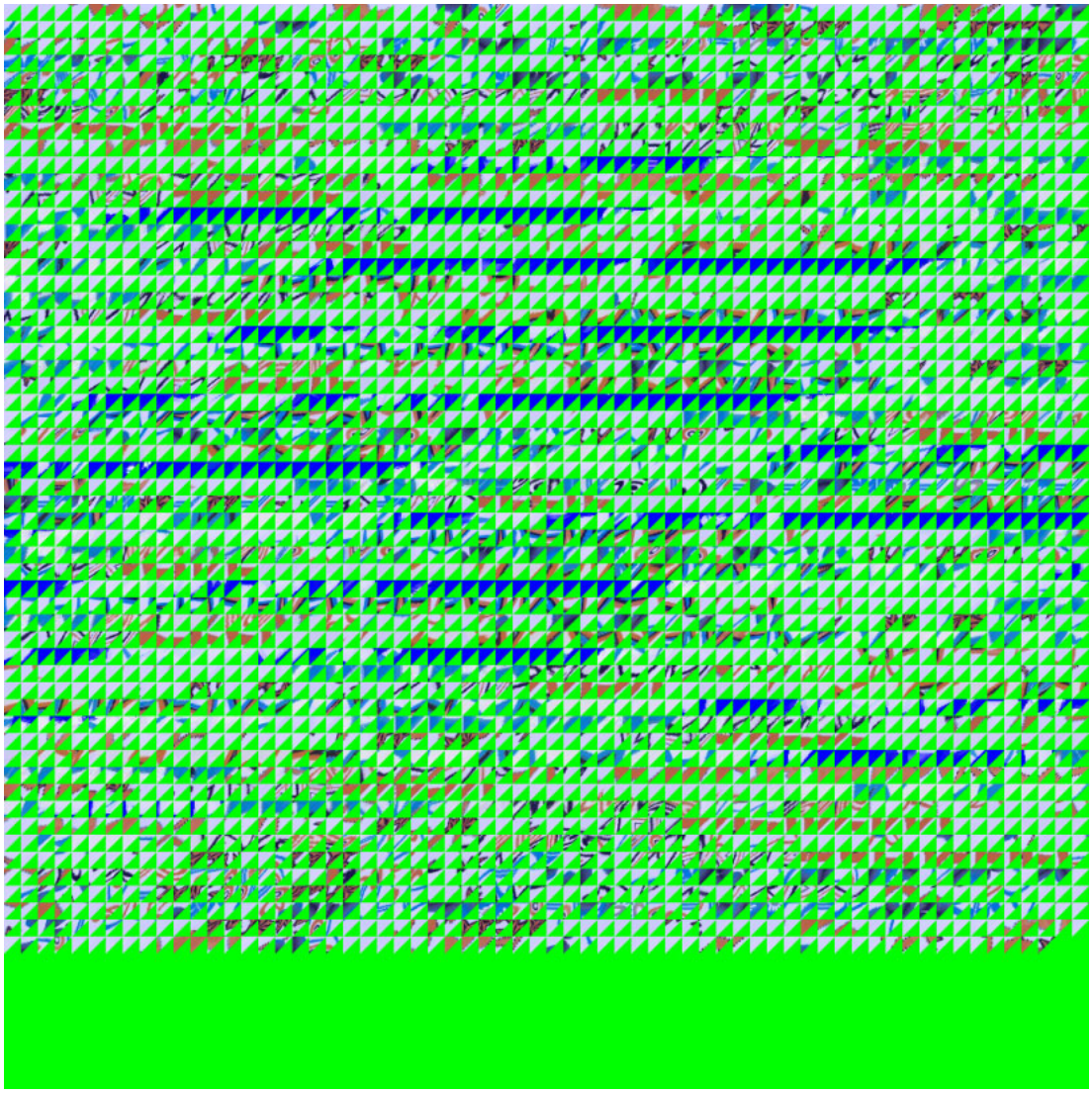


Figure 4.1: Patchwork of texture patterns.

In Figure 4.2, an example reconstruction result with rendering only the diffuse components is shown.

Figure 4.2: Example reconstruction by rendering only the diffuse components.

4.1.2 Rendering the object with specular components

Rendering the object with only specular components requires rendering the object according to the parametric BRDF obtained in the fitting the reflectance process. Normally, rendering of a model using parametric representation of BRDFs requires evaluation of BRDF for each surface point for the current viewing and incoming light directions. In real-time rendering, these evaluations must be recomputed when the view or the position of the light source changes. But this results very low rendering speeds in current graphics hardware. There is some research on real-time rendering of BRDFs where texture maps are used to approximate the behaviour of BRDF models [16, 17]. In our work, we chose to use Kautz and McCools [16] method which is based on separable decompositions of BRDFs. It is a practical method and does not require new types of programmable graphics cards as the work by McAllister do [17]. The details of this method is described in Appendix A.

In Figure 4.3, same object in Figure 4.2 is rendered with only the specular components.

Figure 4.3: Example reconstruction by rendering only the specular components.

Figure 4.4: Example final reconstruction of the object by rendering the object by blending the diffuse and specular components.

The final reconstruction result of the object is shown in Figure 4.4 where the diffuse and specular components of the object are blended. By this way, a more photo-realistic appearance of the object is obtained.

Real-time rendering of a single object is performed with around 20 frames per second. This rendering speed depends on the number of objects in the current scene, i.e. the rendering speed decreases with the increase in the number of objects in the environment.

CHAPTER 5

EXPERIMENTAL RESULTS

In this chapter, the reconstruction results of four objects ("coke", "cologne", "vase", "woodenpot") are shown and discussed.

5.1 Measuring the Quality of the Reconstructed Appearance

There are several parameters that affect the quality of the reconstructed appearance of an object. The main ones are:

- quality of the image acquisition,
- quality of the calibration,
- quality of the reconstructed geometry.

For our study, since we work on artificial objects formed by a 3D modeling tool, the quality of the reconstructed appearances of the objects mainly depends on the reconstructed geometry.

Measuring the quality of the reconstructed appearance can be performed according to two ways: visual and quantitative comparisons. Quantitative comparison methods try to evaluate the similarity between the acquired images and the final rendered images of the reconstructed object according to some image

comparison metrics. In literature, several different metrics are used for image comparison: Minkowski metrics (L_p), Hausdorff distance (HD), normalized cross correlation ratio (CO), root mean square error (RMS), and direct difference error (DDE) [20, 21, 22, 7].

- **Minkowski metrics (L_p).**

  Minkowski metrics are a family of distance measurements which are generalized from the Euclidean distance formula. For two pixels A_{ij} and B_{lm} which have k color channels, the general Minkowski distance is defined as in Equation 5.1.

$$L_p(A_{ij}, B_{lm}) = \left(\sum_{t=1}^{k} |(A_{ij})_t - (B_{lm})_t|^r \right)^{1/r} \tag{5.1}$$

  When p is equal to 2, this distance is called Euclidean or L_2 distance, and when p is equal to 1, it is called a city-block or L_1 (or sometimes a Manhattan or taxi-cab) distance [21].

- **Hausdorff distance (HD).**

  Hausdorff distance between two images is computed as the maximum distance of the first image to the nearest point in the second image [20]. More formally, directed Hausdorff distance between the images A and B is defined as follows:

$$h(A, B) = \max_{i,j} \{ \min_{B_{lm} \in B} \{ d(A_{ij}, B_{lm}) \} \} \tag{5.2}$$

  where d is a metric that computes the distance between the pixel A_{ij} and the image B. Over several alternatives, Euclidean or city-block distances can be used as d.

  From Equation 5.2, it can be easily seen that directed Hausdorff distance is not symmetric (i.e $h(A, B) \neq h(B, A)$). Therefore, normalized Hausdorff distance between two images is computed as follows:

$$HD(A, B) = \max\{h(A, B), h(B, A)\} \tag{5.3}$$

- **Normalized cross correlation ratio (CO).**

  Cross correlation is a standard method of estimating the degree to which two images are correlated. Its definition is given in Equation 5.4.

$$CO(A,B) = 1 - \frac{\sum_{i,j=0}^{N} A_{ij}B_{ij}}{\sqrt{\sum_{i,j=0}^{N} A_{ij}^2 \sum_{i,j=0}^{N} B_{ij}^2}} \tag{5.4}$$

- **Root mean square error (RMS).**

  Root mean square is the mean of the sum of the squares of the differences between the values of pixels in two images. It is defined as in Equation 5.5.

$$RMS(A,B) = \frac{1}{\sqrt{N}}\sqrt{\sum_{i,j=0}^{N}(A_{ij} - B_{ij})^2} \tag{5.5}$$

- **Direct difference error (DDE).**

  Direct difference error (DDE) is an approximation of RMS. While calculating the distance, instead of using root of square differences, the absolute value of differences are used. Therefore, it is computationally less complex than root mean square operation. It is defined as in Equation 5.10.

$$DDE(A,B) = \frac{1}{N}\sum_{ij=0}^{N}|A_{ij} - B_{ij}| \tag{5.6}$$

If the images are similar, RMS and Hausdorff distance work well for image comparison [20]. In this study, two different quality metrics are defined according to root mean square and direct difference errors. For comparisons, the images that are used for reconstruction and the artificial correspondings of them are used.

Definition 5.1 *Given an image sequence $S = A^0, A^1, ..., A^{M-1}$, and a reconstruction R with artificially rendered corresponding images $S^R = B^0, B^1, ..., B^{M-1}$, the quality of the reconstruction Q_1^R is defined as in Equation 5.9. G is the maximum intensity value and N is the number of pixels.*

$$Q_1^R = \left(100 - \frac{\sum_{i=0}^{M-1} RMS_i}{M}\right) \tag{5.7}$$

$$RMS_i = \frac{RMS(A^i, B^i)}{G} \times 100\% \tag{5.8}$$

Definition 5.2 *Given an image sequence* $S = A^0, A^1, ..., A^{M-1}$, *and a reconstruction* R *with artificially rendered corresponding images* $S^R = B^0, B^1, ..., B^{M-1}$, *the quality of the reconstruction* Q_2^R *is defined as in Equation 5.9.* G *is the maximum intensity value and* N *is the number of pixels.*

$$Q_2^R = \left(100 - \frac{\sum_{i=0}^{M-1} DDE_i}{M}\right) \tag{5.9}$$

$$DDE_i = \frac{DDE(A^i, B^i)}{G} \times 100\% \tag{5.10}$$

5.2 Experimental results

In this section, the reconstruction results of four objects ("coke", "cologne", "vase", "woodenpot") are shown and discussed. The experiments are performed on a personal computer with 1 GB of RAM, Intel PIV 2GHz processor and 64MB frame buffer. In appearance reconstruction, for each object, totally 120 images are acquired (from 20 different views, 6 different images where the position of the light source is changed). For the objects "coke", "cologne", "vase" and "woodenpot", the reconstruction results are shown in Figures 5.1, 5.2, 5.3, and 5.4 respectively. The figures are composed of some of the acquired images, artificial corresponding of them, and the difference images.

Figure 5.1: Reconstruction results of the "coke" object (a) original images, (b) reconstruction results, (c) difference images.

(a)

(b)

(c)

Figure 5.2: Reconstruction results of the "cologne" object (a) original images, (b) reconstruction results, (c) difference images.

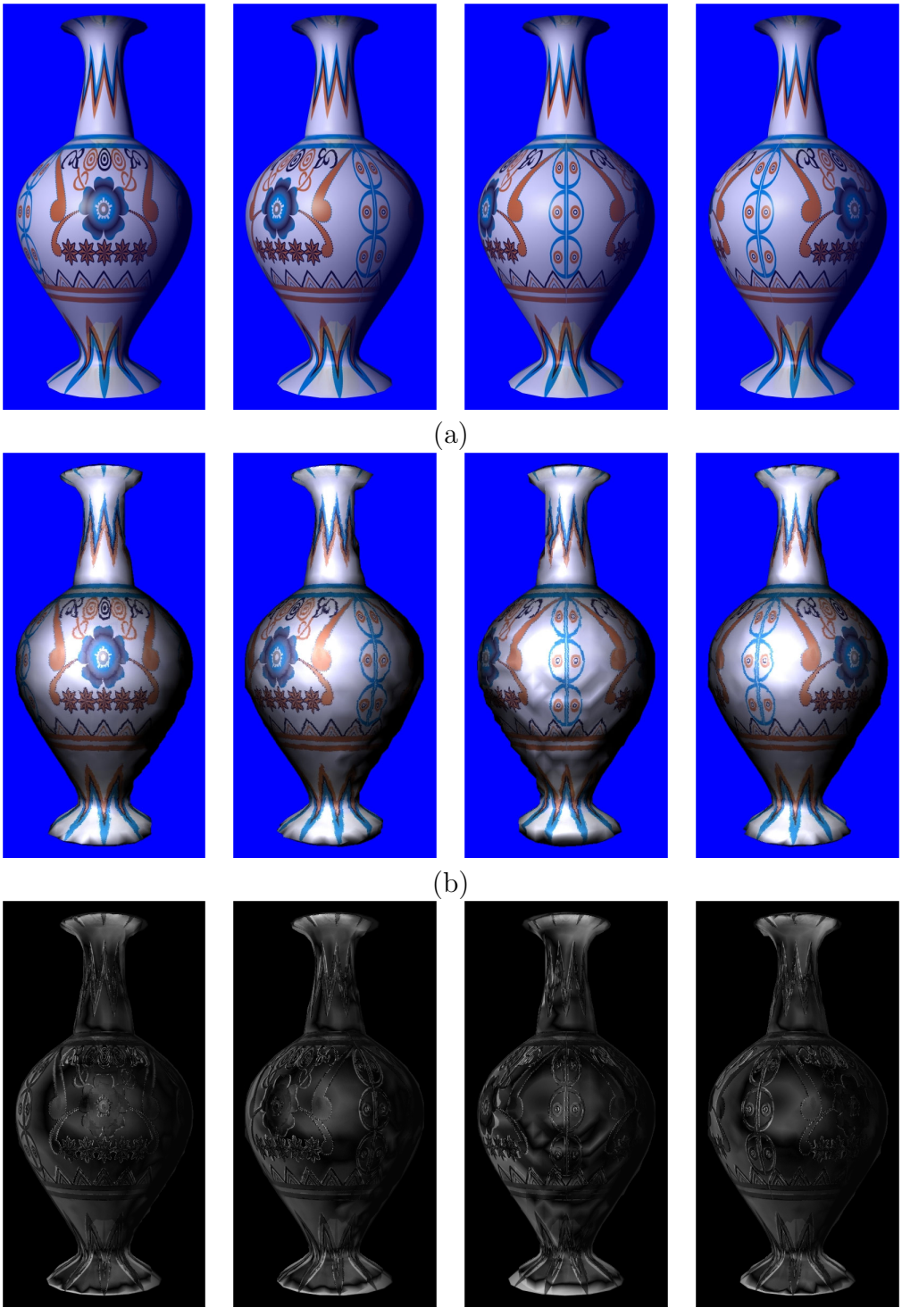


Figure 5.3: Reconstruction results of the "vase" object (a) original images, (b) reconstruction results, (c) difference images.

Figure 5.4: Reconstruction results of the “woodenpot” object (a) original images, (b) reconstruction results, (c) difference images.

Table 5.1: Measured quality of the reconstruction for objects.

Model	Q_1^R	Q_2^R
pepsi	90.483766	90.352217
cologne	92.955578	95.031953
vase	89.262464	88.827280
woodenpot	92.154319	93.308473

In Table 5.1, the results of the error analysis for the objects "coke", "cologne", "vase", "woodenpot" are listed. From the experiments, we observe that the quality measures are very high for the reconstructed objects. As stated in Section 5.1, the quality of the reconstructed appearance mainy depends on the reconstructed geometry. Apart from this, some of the error comes from the assumption that objects have homogeneous specularity, i.e. the amount of the specular reflection of the object depends on a single BRDF determined in the appearance reconstruction process.

If we examine the reconstruction results qualitatively, we can see that for highly specular objects "coke" and "woodenpot", our appearance reconstruction method captures the specular reflectance property of these objects well. For the "vase" object, again we come up fairly good results. On the other hand, the reconstructed appearance of the "cologne" object shows different reflectance behaviour than its original. The reason is that we assume the reconstructed object has a homogeneous specularity. Since the object is composed of a plastic top and a translucent glass body, the specularity of the dominant part of the object i.e. the plastic top is assigned to whole object.

In Table 5.2, the parameters of the BRDFs of each object are listed. While finding these parameters, the initial guess $C_x = -1, C_z = 1$ and $N = 10$ is used.

Table 5.2: Parameters of the BRDFs of each object.

Model	C_x	C_z	N
pepsi	-0.8962	0.7587	10.1119
cologne	-0.8874	1.1162	18.0888
vase	-1.2127	0.5653	20.3645
woodenpot	-1.4299	-0.7763	9.7362

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 Summary

In this study, we describe a method for image-based photorealistic appearance reconstruction of three-dimensional objects. This appearance reconstruction method is based on extraction of material reflectance properties of the object from its images. BRDFs are used in representing the reflectance of the object. The reflectance of the object is decomposed into diffuse and specular components and these components are estimated seperately. While the diffuse components are stored in a global texture, the specular component is represented as a single BRDF. This process can be thought as using Spatially-Varying BRDFs with homogeneous specular components.

Estimation of the diffuse components of the object mainly depends on the idea of computing illumination-invariant images of the object from input images [6]. By this way, the illumination effects such as shadows and highlights are eliminated from the input images. After computing these illumination-invariant images, a global texture is generated from these images by using the surface particles concept proposed by U. Yilmaz [7]. The main idea of using surface particles in texture extraction is that instead of assigning triangles to images, particles are assigned to images and therefore, the discontinuities on the triangle boundaries

can be eliminated.

The specular reflectance data are collected from the residual images obtained by taking difference between the input images and corresponding illumination-invariant images. While collecting reflectance data, a similar process to the one proposed by Lensch et al. [13] is used, but since we use residual images, the collected reflectance data contains only specular components. Then, a Lafortune BRDF model is fitted to these data.

By the decomposition of diffuse and specular components, real-time photo-realistic rendering of the objects become possible. The rendering is achieved in two passes by blending diffuse and specular components. In the first pass, the object is rendered using global texture containing the diffuse component, and in the second pass the BRDF representing the specularity of the object is rendered by the method proposed by Kautz et al. [16].

6.2 Conclusions

The described appearance reconstruction method is tested with artificial data since some necessary equipments are not available in our laboratory. The geometry of the objects are obtained using the 3D reconstruction method described in [2]. Our method is evaluated both quantitatively and qualitively. For the quantitative analysis of the method, two different quality measures are defined. According to the results of the experiments, we come up fairly good reconstruction results.

In computer graphics, local reflection models decomposes reflectance into diffuse and specular components. This decomposition idea is also used in some appearance reconstruction studies [14, 19] previously. However, the main reason of using such a decomposition in our study is that since real-time rendering of BRDFs become possible by the work of Kautz, and we represent the specular components by a single BRDF, we come up with a real-time rendering feature of which, up to our knowledge, other appearance reconstruction frameworks based on BRDFs have a lack of. Therefore, our method can be used in interactive media

like computer games to provide more photorealistic solutions.

6.3 Future Work

As a futurework, we plan to test our appearance reconstruction method for real-world objects. Since our method depends on studies which works well for real-world examples, we believe that we can obtain good appearance reconstruction results for real-world objects.

A more complete extraction of reflectance properties can also be done by fitting multiple BRDFs for each material exist in the object and associating each surface point with these BRDFs. However, the main problem is efficiently rendering of these multiple BRDFs in real-time. There is a continuous research on this subject [17]. With the improvements on hardware programmable shading technology in the new graphics cards (Nvidia GeForce4 family, etc.), real-time rendering of an object using spatially-varying BRDFs will be supported by hardware. Therefore, we think that these rendering techniques will be embedded into appearance reconstruction frameworks.

REFERENCES

[1] Appearance models for computer graphics and vision, cs448c lecture notes, http://graphics.stanford.edu/courses/cs448c-00-fall, as accesed on, July 2003.

[2] U. Yilmaz, A. Y. Mulayim, and V. Atalay. Reconstruction of three dimensional models from real images. In *Proceedings of the International Symposium on 3D Data Processing Visualization and Transmission*, pages 554–557, 2002.

[3] F. Schmitt and Y. Yemez. 3d color object reconstruction from 2d image sequences. In *Proceedings of International Conference on Image Processing*, pages 65–69, October 1999.

[4] H. P. A. Lensch, W. Heidrich, and H-P. Seidel. Automated texture registration and stitching for real world models. In *Proceedings of Pacific Graphics*, pages 317–337, October 2000.

[5] H. Lensch, W. Heidrich, and H.-P. Seidel. A silhouette-based algorithm for texture registration and stitching. *Journal of Graphical Models*, 63(4):245–262, July 2001.

[6] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. Acquiring, stitching and blending diffuse appearance attributes on 3d models. *Visual Computer*, 18(3):186–204, 2002.

[7] U. Yilmaz. Appearance reconstruction of three dimensional models from real images. Master's thesis, Department of Computer Engineering, Middle East Technical University, January 2002.

[8] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs from. In Alyn Rockwood, editor, *Siggraph99, Annual Conference Series*, pages 215–224, Los Angeles, 1999. Addison Wesley Longman.

[9] P. Debevec, T. Hawkins, C. Tchou, H. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proceedings of ACM SIGGRAPH 2000*, pages 145–156, 2000.

[10] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, K. E. Torrance, and D. P. Greenberg. Image-based brdf measurement including human skin. In *Proceedings of the 10th Eurographics Workshop on Rendering.*, pages 131–144, 1999.

[11] Ron O. Dror, Edward H. Adelson, and Alan S. Willsky. Recognition of surface reflectance properties from a single image under unknown real-world illumination. In *Proceedings of the IEEE Workshop on Identifying Objects Across Variations in Lighting: Psychophysics and Computation*, 2001.

[12] Samuel Boivin and Andr Gagalowicz. Image-based rendering of diffuse, specular and glossy surfaces from a single image. In *Proceedings of ACM SIGGRAPH 2001*, 2001.

[13] Hendrik Lensch, Michael Goesele, Jan Kautz, Wolfgang Heidrich, and Hans-Peter Seidel. Image-Based reconstruction of spatially varying materials. In *Proceedings of the 12th Eurographics Workshop on Rendering*, pages 104–115, 2001.

[14] Yoichi Sato, Mark D. Wheeler, and Katsushi Ikeuchi. Object shape and reflectance modeling from observation. *Computer Graphics*, 31(Annual Conference Series):379–388, 1997.

[15] Alan Watt. *3D Computer Graphics*. Third edition, 2000.

[16] J. Kautz and M. D. McCool. Interactive rendering with arbitrary brdfs using separable approximations. In *Proceedings of the 10th Eurographics Workshop on Rendering*, pages 281–292, 1999.

[17] David K. McAllister, Anselmo Lastra, and Wolfgang Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. In *Proceedings of Graphics Hardware*, 2002.

[18] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. *Computer Graphics*, 31(Annual Conference Series):117–126, 1997.

[19] K. Nishino, Z. Zhang, and K. Ikeuchi. Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis. In *Proc. of Eighth IEEE International Conference on Computer Vision (ICCV '01)*, pages 599–606, July 2001.

[20] V. Di Gesu and V. Starovoitov. Distance-based functions for image comparison. *Pattern Recognition Letters*, 20(2):207–214, 1999.

[21] R. C. Veltkamp and M. Hagedoorn. Shape similarity measures, properties and constructions. In *Visual Information and Information Systems*, pages 467–476, 2000.

[22] V. Atalay, A. Y. Mulayim, O. Ozun, and U. Yilmaz. Otomatik uc boyutlu nesne modelleyici - internet uzerinde tanitim ve pazarlamada ucuncu boyuta dogru. In *Bilisim 2000*, Eylul 2000.

[23] Mark A. DeLoura, editor. *Game Programming Gems*. Charles River Media, Inc., 2000.

APPENDIX A

INTERACTIVE RENDERING WITH ARBITRARY BRDFs USING SEPARABLE APPROXIMATIONS

Kautz and McCools [16] method is based on separable decompositions of BRDFs. Separable decompositions approximate a high-dimensional function f using a sum of products of lower-dimensional functions g_k and h_k:

$$f(x, y, z, w) \approx \sum_{k=1}^{N} g_k(x, y) h_k(z, w) \tag{A.1}$$

In their work, Kautz and McCool stated that $N = 1$ has proven to be visually adequate for many BRDFs. So the process turns out to be seperating the 4D function as products of two 2D functions. Finding functions g_1 and h_1 is performed by using Normalized Decomposition. For the approximation,

$$f_P(\mathbf{x}, \mathbf{y}) \approx \tilde{f}_{P1}(\mathbf{x}, \mathbf{y}) = \mathbf{g_1}(\mathbf{x})\mathbf{h_1}(\mathbf{y}) \tag{A.2}$$

Then the functions g_1 and h_1 can be determined by using the following formulas:

$$g_1(\mathbf{x}) = (\int_{\mathbf{Y}} |\mathbf{f_P}|^{\mathbf{p}}(\mathbf{x}, \mathbf{y})\mathbf{dy})^{\frac{\mathbf{1}}{\mathbf{p}}}, \quad h_1(\mathbf{y}) = \frac{\mathbf{1}}{|\mathbf{X}|} \int_{\mathbf{X}} \frac{\mathbf{f_P(x,y)}}{\mathbf{g_1(x)}}\mathbf{dx}. \tag{A.3}$$

A.1 Rendering Algorithm

The overall interactive rendering algorithm for a single BRDF can be performed by the following steps:

1. **Sampling BRDF in each dimension and constructing the corresponding 2D matrix.**

   BRDFs are 4D functions that depend on local viewing direction $\mathbf{v}$ and incoming light direction $\mathbf{u}$. We can represent these directions in spherical coordinates $\mathbf{u} = (\theta_i, \phi_i)$ and $\mathbf{v} = (\theta_o, \phi_o)$. By selecting an specific angle as the sampling resolution, we can easily construct the corresponding sampled BRDF matrix by using spherical coordinates. For example, in Figure A.1, the BRDF is mapped to a 2D matrix representation with a sampling rate of 2x in both θ and ϕ.

$$
\begin{array}{c}
 \\
\theta_{o0},\phi_{o0} \\
\theta_{o0},\phi_{o1} \\
\theta_{o1},\phi_{o0} \\
\theta_{o1},\phi_{o1}
\end{array}
\begin{array}{c}
\begin{array}{cccc}
\theta_{i0},\phi_{i0} & \theta_{i0},\phi_{i1} & \theta_{i1},\phi_{i0} & \theta_{i1},\phi_{i1}
\end{array} \\
\begin{bmatrix}
\mathrm{F}(\theta_{i0},\phi_{i0},\theta_{o0},\phi_{o0}) & \mathrm{F}(\theta_{i0},\phi_{i1},\theta_{o0},\phi_{o0}) & \mathrm{F}(\theta_{i1},\phi_{i0},\theta_{o0},\phi_{o0}) & \mathrm{F}(\theta_{i1},\phi_{i1},\theta_{o0},\phi_{o0}) \\
\mathrm{F}(\theta_{i0},\phi_{i0},\theta_{o0},\phi_{o1}) & \mathrm{F}(\theta_{i0},\phi_{i1},\theta_{o0},\phi_{o1}) & \mathrm{F}(\theta_{i1},\phi_{i0},\theta_{o0},\phi_{o1}) & \mathrm{F}(\theta_{i1},\phi_{i1},\theta_{o0},\phi_{o1}) \\
\mathrm{F}(\theta_{i0},\phi_{i0},\theta_{o1},\phi_{o0}) & \mathrm{F}(\theta_{i0},\phi_{i1},\theta_{o1},\phi_{o0}) & \mathrm{F}(\theta_{i1},\phi_{i0},\theta_{o1},\phi_{o0}) & \mathrm{F}(\theta_{i1},\phi_{i1},\theta_{o1},\phi_{o0}) \\
\mathrm{F}(\theta_{i0},\phi_{i0},\theta_{o1},\phi_{o1}) & \mathrm{F}(\theta_{i0},\phi_{i1},\theta_{o1},\phi_{o1}) & \mathrm{F}(\theta_{i1},\phi_{i0},\theta_{o1},\phi_{o1}) & \mathrm{F}(\theta_{i1},\phi_{i1},\theta_{o1},\phi_{o1})
\end{bmatrix}
\end{array}
$$

Figure A.1: Mapping of BRDF to a 2D matrix representation.

2. **Finding 2D functions G and H by using Normalized Decomposition.**

   The standard parametrization of a BRDF is with respect to local viewing direction $\mathbf{v}$ and incoming light direction $\mathbf{u}$ which are represented in spherical coordinates. A unit length vector $\mathbf{a}$ can be expressed in spherical coordinates $\theta(\mathbf{a}), \phi(\mathbf{a})$ relative to the local surface coordinate system $\{\mathbf{n}, \mathbf{t}, \mathbf{s}\}$ as follows (see Figure A.2):

$$
\begin{aligned}
&cos(\theta(\mathbf{a}) = \mathbf{n} \cdot \mathbf{a}, \\
&tan(\phi(\mathbf{a}) = (\mathbf{s} \cdot \mathbf{a})/(\mathbf{t} \cdot \mathbf{a}).
\end{aligned}
\tag{A.4}
$$

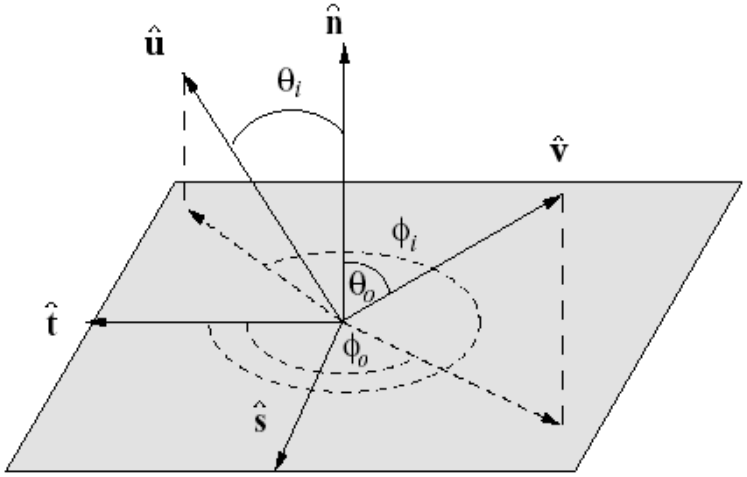


Figure A.2: Local surface coordinate system.

But this parametrization does not always provides good results. To further improve the decomposition results, Gram-Schmidt Halfangle-Difference (GSHD) vector reparametrization can be used. GSHD parametrized the BRDF with respect to the halfway vector **h** (the vector halfway between the incoming light vector **u** and the outgoing view vector **v**) and a difference vector **d**. The difference vector **d** can be thought as the incoming light direction **u** relative to a new coordinate system $\{\mathbf{n}, \mathbf{t}', \mathbf{s}'\}$ where **h** corresponds to the pole of the new coordinate system. Then the vectors **h** and **d** can be determined as follows:

$$\begin{aligned} \mathbf{h} &= norm(\mathbf{u}+\mathbf{v}), \quad & \mathbf{t}' &= norm(\mathbf{t} - (\mathbf{t}\cdot\mathbf{h})\mathbf{h}), \\ \mathbf{s}' &= \mathbf{h}\times\mathbf{t}', & \mathbf{d} &= [\mathbf{u}\cdot\mathbf{h}, \mathbf{u}\cdot\mathbf{s}', \mathbf{u}\cdot\mathbf{t}']^T. \end{aligned} \tag{A.5}$$

The effect of GSHD reparametrization can seen in Figure A.3.

3. **Creating cube-maps for the functions G and H.**

   The algorithm for creating cube-maps for functions G and H can be seen in Algorithm 2.

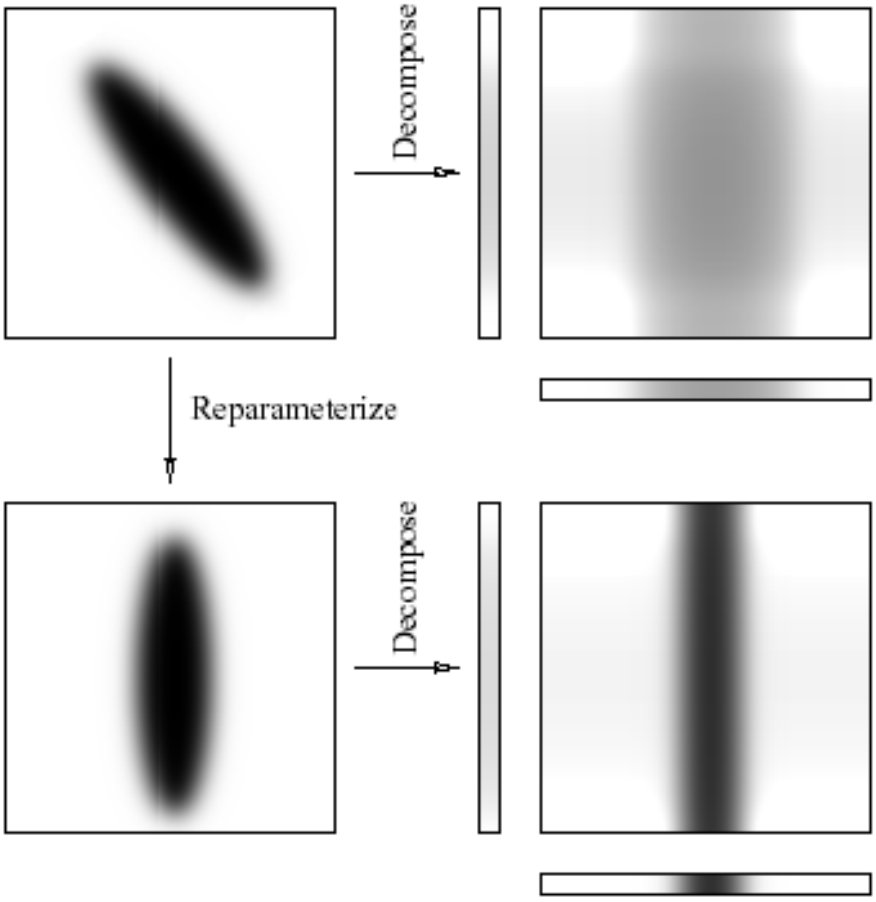


Figure A.3: Improving results of decomposition through reparametrization.

Algorithm 2 Creating Cube-maps

for all face of the cube-map **do**
  for all texel of the face **do**
    if texel is on bottom half of hemisphere **then**
      assign (0,0,0) RGB value into texel and **continue**
    end if
    compute the vector from the origin through the texel center
    compute the spherical coordinate (θ, ϕ) representation of the vector
    compute the corresponding RGB value using bilinear interpolation
    write the RGB value into the texel
  end for
end for

4. **Interactive Rendering**

   In this phase, texture coordinates for the functions G and H are recalculated for the current viewing and light directions. The approximate BRDF is determined by using the cube-maps and multi-texturing feature of OpenGL and the object is rendered accordingly.

APPENDIX B

TANGENT (TBN) SPACE

This section is inspired from Game Programming Gems [23].

Tangent space is a local coordinate space defined for a point on a surface. It uses the surface normal **n** as the +Z axis and represents a space that lies tangent to that surface as shown in Figure B.1. In order to generate the tangent space, two unique vectors are enough and the third vector can be generated from these two. Since for the polygonal models only +Z (**n**) is known, infite number of tangent spaces are exist. Therefore, +Y axis in the model space can be chosen as the second vector. The third vector serving as +X can be determined with the cross product of +Y with +Z. Then, a new +Y axis generated by taking cross product of +Z with +X. Finally, normalizing these vectors gives us the three columns of matrix **M** representing the local tangent space at the surface point.

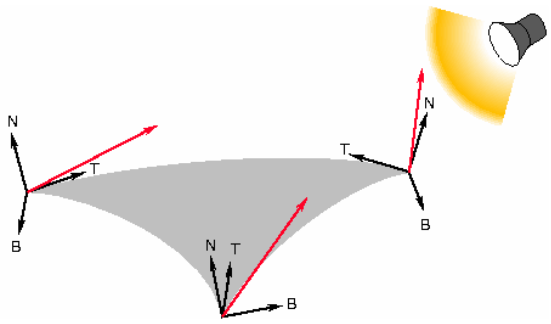


Figure B.1: Tangent space defined on vertices of a polygon.