國立交通大學

多媒體工程研究所

碩士論文

在光源及視點空間最佳重新參數化的雙向貼圖函數模型

Bidirectional Texture Function Modeling Using Optimized Reparameterization in the Lighting and Viewing Space

研究生：方奎力

指導教授：林文杰　教授

中華民國 九十七 年十月

Bidirectional Texture Function Modeling Using Optimized Reparameterization in the Lighting and Viewing Space

Student: Kuei-Li Fang Advisor: Dr. Wen-Chieh Lin

Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University

ABSTRACT

We present a new approach to model BTF data using multivariate SRBF kernels. More importantly, we propose a data-dependent reparameterization method in which optimal reparameterization in the lighting and viewing space that minimizes the modeling error is obtained. To speed up our BTF modeling computation and improve spatial coherence of acquired BTF model, we utilize a mipmap hierarchy in our BTF model, which can be nicely integrated with graphics hardware when rendering BTF data. Our experiments show that the optimal reparameterization does provide better accuracy than a fixed reparameterization such as the half-way reparameterization. In addition, real-time rendering of BTF from novel viewing or lighting direction that is not recorded in the original BTF database can be achieved easily.

Acknowledgments

I would like to thank Ph.D. Candidate Yu-Ting Tsai who provides the idea of optimal reparameterization and helpful suggestions on implementation. I also want to thank my advisors Dr. Wen-Chieh Lin who gives me a lot guidance, inspirations, useful ideas, and teach me the attitude of hard-working. Thanks to my colleagues: Shang-Wen Wang, Wei-Chien Cheng, Chin-Hsian Chang, Yueh-Tse Chen, Hsin-Hsiao Lin, Ying-Tsung Li, Zhi-Han Yan, Ming-Shang Hsu, Ying-Shou Lan, and Jia-Ru Lin for their assistances and discussions. It is honor to work with all of you. At last, I would like to thank my parents for their unselfish love and support.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Rendering materials with complex meso structures and reflectance behaviors used to be a great challenge in computer graphics. It is hard to describe reflectance behavior with simple analytic models, so recent rendering algorithms synthesize high-quality images from precomputed results or measured reflectance data. These image-based rendering algorithms have tremendous advances over the last decades. Dana et al.[4] introduced the bidirectional texture function (BTF) to model the appearance of materials in different viewing and lighting directions. A BTF is a 6D function that stores spatially-varying reflectance appearance of materials. Therefore, images rendered from measured BTFs can faithfully exhibit complex lighting effects and detailed meso-structures of real-world objects, including the micro-geometry of rough surfaces, self-shadows, and multiple light scattering.

As BTF is a collection of discrete texture data with enormous data size, how to efficiently compress and model BTF data is a great challenge. This challenge has stimulated recent development of approximation algorithms for large-scale surface appearance models[1, 3, 4, 6, 11]. The appearance of a BTF depends on various physical parameters. This high-dimensional data is too complicated to be presented by a simple analytic model, so we introduce a sum-of-product model to describe complex anisotropic reflectance behaviors of BTF data. More specifically, we decompose a reflectance field as a linear combination of multivariate spherical radial basis functions (SRBFs), while each multivariate SRBF is constructed from the product of univariate SRBF[1].

Transforming the parameters of a reflectance function into another parametric space, which we called reparameterization[7, 11, 17, 21] is a well-known method to improve approximation efficiency; however, only fixed function reparameterization was introduced in previous articles. In this thesis, we propose a data-dependent reparameterization method. We propose to find the optimized reparameterization function, so we use a parametric representation in viewing and lighting space to model the BTF data and integrate parameters optimization into our framework. Fixed functions reparameterization, such as half-way vector becomes special cases in our general framework.

In order to exploit spatial coherence in BTF data more efficiently, we also utilize the mipmap hierarchy in our framework. We find the optimized representation of the highest level in mipmaps at first, then propagate results from higher level to lower level. As a result, we can make our optimization process much faster and still remain great quality. With this hierarchy, hardware support such as mipmaps can also been used at run time stage.

The contributions of this thesis can be summarized as:

- A novel functional representation based on the linear combination of multivariate spherical radial basis functions is introduced to accurately model the anisotropic behavior of measured reflectance fields.
- An automatic reparameterization framework is proposed to learn the best parameter transformation function from a given reflectance data set.
- A multi-scale optimization algorithm for spatially-varying materials is presented to exploit the spatial coherence among data and accelerate the approximation process.

The rest of this thesis is organized as follows: Chapter 2 reviews previous methods for

[1] The univariate SRBFs refer to the original SRBFs in [19]

modeling surface reflectance. Chapter 3 illustrates the proposed algorithm for modeling BTF. Chapter 4 shows our experimental results. In the end, conclusion and future work are discussed in Chapter 5.

CHAPTER 2

Previous Work

In this section, we give an brief introduction of BTF in section 2.1, then review some reparameterization methods in section 2.2 and recent approximation algorithms for reflectance fields in section 2.3

2.1 Bidirectional Texture Function

BTF is a 6D function $T(x, y, \eta_i, \eta_o) = T(x, y, \theta_i, \phi_i, \theta_o, \phi_o)$ proposed by Dana et al.[4], where x, y, θ_i, ϕ_i, θ_o, ϕ_o denote the spatial coordinates, illumination and view directions in spherical coordinate, respectively. The BTF data is a set of images taken from different illumination and view directions. Without regard to spatial dimensions, we can take BTF as ordinary reflectance function such as BRDF $\rho(\theta_i, \phi_i, \theta_o, \phi_o)$.

Due to enormous data size of BTF, how to compress and model BTF becomes an important issue in BTF manipulation. Recent researches focus on recovering ways to model these BTF data. Vasilescu and Terzopoulos[1] organized BTF textures as a tensor, and then extracted basis matrices in each mode by traditional PCA. Wang et al.[20] also proposed an out-of-core and block-wise tensor by using an optimal N-mode SVD algorithm. These dimensionality reduction techniques still need to use interpolation between each basis to fill the hole between sparsely observed view and lighting directions and only allowing one of lighting or viewpoint variation.

Zickler et al.[21] modeled spatially-varying BRDFs as continuous RBF functions via half-way reparameterization[17]; However, Szymon[17] assumed the BRDF model are isotropic, so half-way reparameterization is not suitable for modeling anisotropic reflectance behaviors. In our work, we propose a BTF model reparameterized by optimal parameters in illumination and view space.

2.2 Reparameterization of Reflectance Functions

Reflective[16] and half-way[17] reparameterization for BRDFs are well-known methods to model highly specular materials. Recently, Lawrence et al.[11] and Zickler et al.[21] applied these reparameterization to approximate spatially-varying BRDFs. In general, reparameterization is beneficial reducing the dimensionality and greatly increasing the data coherence of reflectance functions.

Previous reparameterization techniques are limited to fixed or heuristic transformation functions; however, it is unknown which parameter transformation is suitable for the reflectance data. In this thesis, we propose to find the optimized reparameterization function from data, with restricted functional forms.

2.3 Modeling of Reflectance Data

Modern approximation algorithms for reflectance fields can be summarized into two main categories: non-parametric models and functional linear models.In non-parametric models, an appropriate basis is learnt from data for an accurate representation rather than pre-defined basis functions. Clustering and dimensional reduction techniques are the most popular methods in computer graphics, e.q. matrix factorization[11, 14], principal component analysis[3, 8, 18], tensor approximation[1, 20] and vector quantization[13]. Although non-parametric methods are entirely data-driven models that yield accurate and flexible representations, the compressed data is still huge; It is difficult to render BTF in real time with non-parametric models. Also, interpolation methods are still needed when rendering BTF from novel illumination and view direction. In contrast to non-parametric models, our method provides not only higher compression ratios but also real-time rendering performance under similar image quality; Our reparameterization function can represent reflectance data into continuous functions, so interpolation is not needed in our method.

The main concept of functional linear models is to expand a radiance function as a sum of pre-defined basis functions. The choice of an appropriate basis significantly influences rendering performance and image quality. Previous articles have described reflectance fields using parametric kernels[6, 9], radial basis functions[21], spherical harmonics[16], and wavelets[2]. In general, kernel and radial basis functions provide better run-time performance and quality for all-frequency materials. However, they usually require computationally expensive non-linear optimization for parameter estimation, which becomes even awful and impractical when modeling BTF; In this thesis, we adopt the mipmap hierarchy and the expectation maximization(EM) algorithm in our framework to accelerate the non-linear optimization.

CHAPTER 3

Approach

This chapter describes our algorithm. First, we give an overview of our approach in section 3.1. Next, we describe the background of our algorithm(section 3.2), including multivariate SRBF(section 3.2.1), optimal reparameterization(section 3.2.2). At last, we introduce our implementation issues(section 3.3), including mipmap construction(section 3.3.1), fitting with mipmap hierarchy(section 3.3.2), smooth terms(section 3.3.3), and our rendering system (section 3.3.4).

3.1 Overview

Figure 3.1 shows an overview of our approach, which consists of BTF modeling and BTF rendering.

$$T(x, y, \eta_i, \eta_o) \tag{3.1}$$

A BTF is a 6D function $T(x, y, \eta_i, \eta_o)$, where x and y denote the spatial coordinates and η_i and η_o denote the illumination and view directions. Finding multivariate SRBF expansion of BTF data with optimal parameters is the main task in BTF modeling.

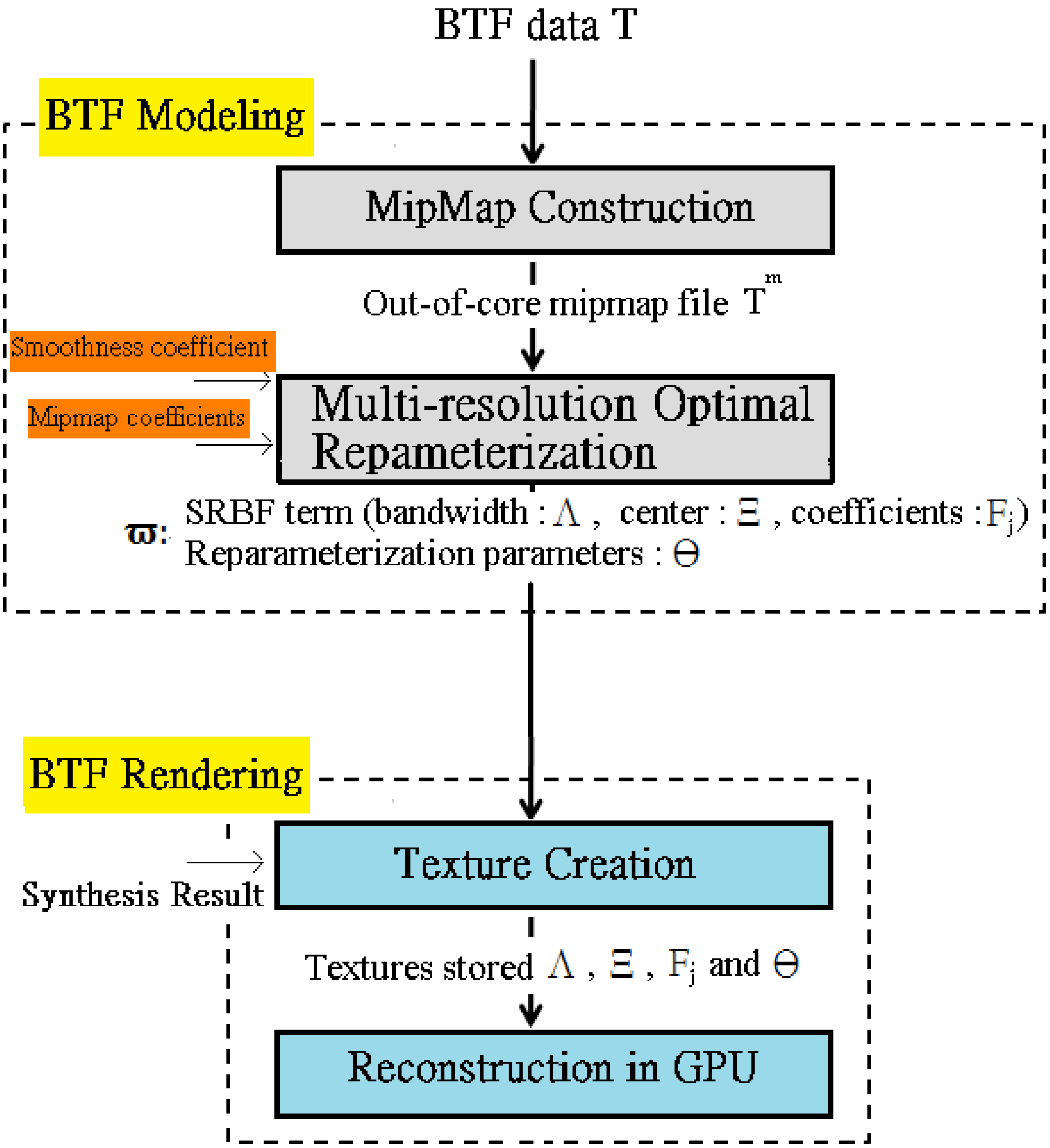

BTF data T
BTF Modeling
MipMap Construction
Out-of-core mipmap file T^m
Smoothness coefficient
Mipmap coefficients
Multi-resolution Optimal Repameterization
ϖ: SRBF term (bandwidth : Λ , center : Ξ , coefficients : F_j)
Reparameterization parameters : Θ
BTF Rendering
Synthesis Result
Texture Creation
Textures stored Λ , Ξ , F_j and Θ
Reconstruction in GPU

Figure 3.1: Approach Overview

We first represent the input BTF $T(x, y, \eta_i, \eta_o)$ as mipmap format $T^l(x, y, \eta_i, \eta_o)$ where l denotes the mipmap level, and $T^0(x, y, \eta_i, \eta_o)$ is the original input data; Like regular mipmap structure, width and height of level $k+1$ is half of level k.

In mipmap hierarchy fitting, we optimize coefficient, center, bandwidth of SRBF kernel and Θ of reparameterization function using the gradient-decent method; We find the optimal result of higher level in mipmap and propagate them to lower level until all levels are obtained.

We construct mipmap textures stored representation results and throw these textures into GPU memory. At BTF rendering stage, we compute illumination direction η_i and view direction η_o at each pixels, and reconstruct BTF T by looking up the computed BTF model.

3.2 Background

We introduce background of our approach in this section. In section 3.2.1, we introduce Multivariate SRBF which is the extend form of the original SRBF introduced by Tsai and Shih[19]. In section 3.2.2, we introduce our optimal reparameterization method.

3.2.1 Multivariate SRBF

SRBFs[19] are circularly axis-symmetric functions defined on the unit sphere S^m. Let η and ξ be two points on S^m, and θ be the geodesic distance between η and ξ, i.e. $\theta = \arccos(\eta \cdot \xi)$, then a univariate SRBF is defined as a function $G(cos\theta) = G(\eta \cdot \xi)$. Here is an example of univariate SRBFs called Gaussian SRBF kernel:

$$G^{Gau}(\eta \cdot \xi; \lambda) = e^{\lambda(\eta \cdot \xi)} e^{-\lambda}, \tag{3.2}$$

where λ is bandwidth and ξ is center of the SRBF kernel. Similar to RBFs in R^{m+1}, a spherical function $F(\eta)$ can be represented as

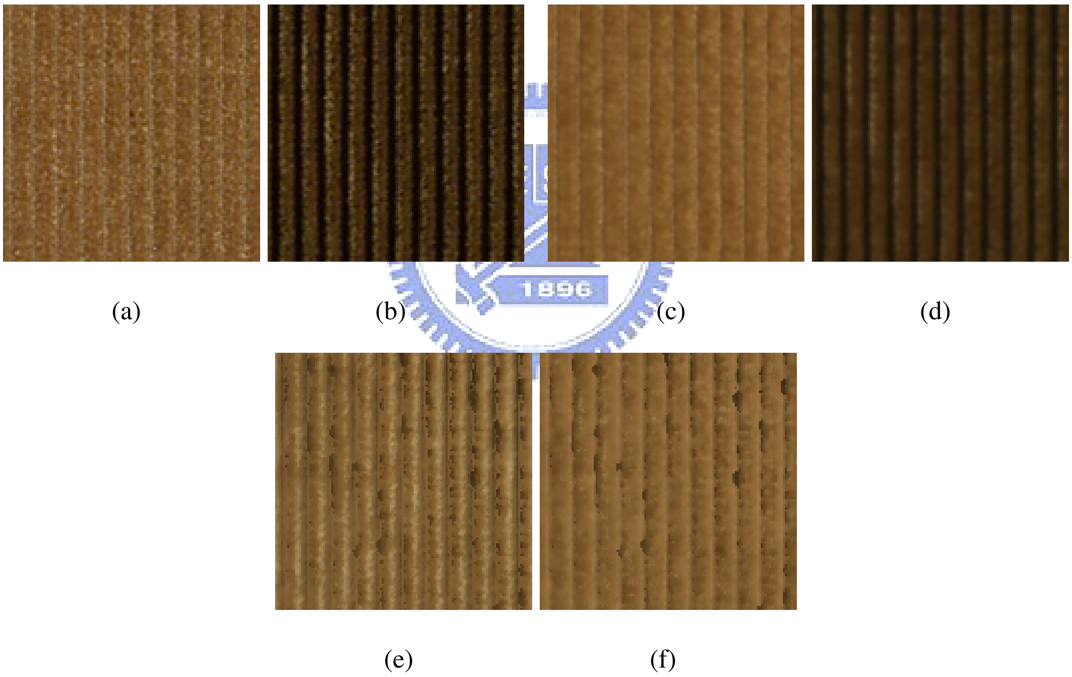
$$F(\eta) = \sum_{j=1}^{n} F_j G(\eta \cdot \xi_j; \lambda_j) \tag{3.3}$$

As BTF data is too complicate to be represented by univariate SRBFs, we combine sum-of-product of univariate SRBFs to form multivariate SRBFs. An example of multivariate SRBFs using the Gaussian SRBF kernel is listed below:

$$G^{M-Gau}(\{\eta_k \cdot \xi_k; \lambda_k\}_{k=1}^{K}) = e^{\sum_k \lambda_k(\eta_k \cdot \xi_k) - \lambda_k} \tag{3.4}$$

We use multivariate Gaussian SRBF kernel to model a spherical function $F(\eta)$ as shown in Equation 3.5, where J is the number of multivariate Gaussian SRBF kernels and K is number of variables in multivariate Gaussian SRBF kernel.

$$F(\eta) = \sum_{j=1}^{J} F_j G^{M-Gau}(\{\eta_k \cdot \xi_{j,k}; \lambda_{j,k}\}_{k=1}^{K}); \tag{3.5}$$



(a) (b) (c) (d)

(e) (f)

Figure 3.2: Comparison of BTF data and representation by Gaussian SRBFs and multivariate Gaussian SRBFs. (a)(b) are textures of corduroy data from Bonn BTF database. (c)(d) are reconstructed results by multivariate Gaussian SRBFs via half-way reparameterization. (e)(f) are reconstructed results by Gaussian SRBFs. Twenty SRBF kernels($J = 20$) are used in (c)(d)(e)(f), Three multivariate modes($K = 3$) are used in (c)(d).

3.2.2 Optimal Reparameterization

Reparameterization is a popular method in improving approximation efficiency. Let $T(\eta_i, \eta_o)$ be a BTF data without spatial varying, where η_i and η_o denotes the illumination directions and view directions. We would like to find an optimized reparameterization function $\psi(\eta_i, \eta_o|\Theta_k)$, with parameter set Θ_k, such that another function $F(\psi(\eta_i, \eta_o|\Theta_k))$ can approximates $T(\eta_i, \eta_o)$ efficiently.

As mention above, we can model $T(\eta_i, \eta_o)$ using multivariate Gaussian SRBF kernel as follows:

$$T(\eta_i, \eta_o) = F(\psi(\eta_i, \eta_o|\Theta_k)) = \sum_{j=1}^{J} F_j G^{M-Gau}(\{\psi(\eta_i, \eta_o|\Theta_k) \cdot \xi_{j,k}; \lambda_{j,k}\}_{k=1}^{K}) \qquad (3.6)$$

In this work, we use a linear combination form showed below as our reparameterization transform function.

$$\psi(\eta_i, \eta_o|\Theta_k) = a_k\eta_i + b_k\eta_o \qquad (3.7)$$

We define our objective function E_b below:

$$E_b = \sum_{\iota,\upsilon}(T(\eta_i^\iota, \eta_o^\upsilon) - \sum_{j=1}^{J} F_j G^{M-Gau}(\{\psi(\eta_i^\iota, \eta_o^\upsilon|\Theta_k) \cdot \xi_{j,k}; \lambda_{j,k}\}_{k=1}^{K}))^2 \qquad (3.8)$$

Then, we optimize Θ_k using this objective function E_b until we get the optimal transfer function.[1]

3.3 Implementation Issues

In this section, we describe issues when we implement our method. First, We introduce how we get a proper initial guess(section 3.3.1) and how we construct mipmap file of BTF data(section 3.3.2), then how to model BTF data with mipmap hierarchy(section 3.3.3) and how to smooth our BTF model in spatial dimension(section 3.3.4) is followed, at last, rending part(section 3.3.5) including reconstructed results and texture synthesis is listed.

[1] We use 'L-BFGS'(http://www.alglib.net/optimization/lbfgs.php) which is a software package for solving non-linear optimization problems to solve our optimization problems

3.3.1 Initial Guess

As our optimization problem involves many high-dimensional variables, a good initial is important. We adopt standard expectation maximization(EM) algorithm to obtain an initial guess for our optimization problem. This is based on the assumption that each SRBF kernel is independent. As this assumption does not hold when using SRBF to model BTF, we only use EM to obtain the initial guess.

In the E step, we evaluate a probability density function P defined as follows:

$$P((j,k)|\psi(\eta_i^\iota,\eta_o^\upsilon|\Theta_k),\Xi,\Lambda) = \frac{F_j G^{M-Gau}(\psi(\eta_i^\iota,\eta_o^\upsilon|\Theta_k)\cdot\xi_{j,k};\lambda_{j,k})}{\sum_{j=1}^{J} F_j G^{M-Gau}(\{\psi(\eta_i^\iota,\eta_o^\upsilon|\Theta_k)\cdot\xi_{j,k};\lambda_{j,k}\}_{k=1}^{K})} \tag{3.9}$$

where η_i^ι and η_o^υ are respectively the ι-th illumination direction and the υ-th view direction and Ξ and Λ are sets of SRBF centers $\xi_{j,k}$ and bandwidth $\lambda_{j,k}$.

In the M step, we maximize the exception value Q as follows:

$$Q = \sum_{\iota,\upsilon}\sum_{j=1}^{J}\sum_{k=1}^{K}\lambda_{j,k}T(\eta_i^\iota,\eta_o^\upsilon)(\psi(\eta_i^\iota,\eta_o^\upsilon|\Theta_k)\cdot\xi_{j,k})P((j,k)|\psi_k(\eta_i^\iota,\eta_o^\upsilon|\Theta_k),\Xi,\Lambda) \tag{3.10}$$

If we assume that the distribution P is fixed when optimizing Θ_k, Equation 3.10 can be further simplified into

$$Q = \sum_{\iota,\upsilon}\sum_{j=1}^{J}\sum_{k=1}^{K}\lambda_{j,k}T(\eta_i^\iota,\eta_o^\upsilon)(\psi(\eta_i^\iota,\eta_o^\upsilon|\Theta_k)\cdot\xi_{j,k})P((j,k)|\eta_{\iota\upsilon},\Xi,\Lambda) \tag{3.11}$$

where $\eta_{\iota\upsilon}$ is the reparameterized direction of η_i^ι and η_o^υ in previous iteration. Traditional gradient-based non-linear optimization process then can be applied to update Θ_k.

After we get Θ_k from EM step, we can take the result into objective function E_b in Equation 3.8 and optimize Θ_k to get optimal reparameterization parameters.

3.3.2 Mipmap Construction

When a BTF data first into our system, we construct mipmap data of each BTF textures using Lanczos resampling[5], the Lanczos filter is defined below:

$$L(x) = \begin{cases} \frac{asin(\pi x) sin(\frac{\pi}{a}x)}{\pi^2 x^2}, & -a < x < a, x \neq 0 \\ 1, & x = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.12}$$

Where a is set to be three in our method. We store the BTF mipmap data into an out-of-core three dimensional array. The first dimension is the mipmap level from highest level(contains only one texel) to lowest level(original texture size); The second dimension is spatial dimension according to the position of textel in the texture; The third dimension is illumination and view directions.

3.3.3 Multi-resolution Optimal Repameterization

Let $\varpi(x, y, l)$ be the BTF model parameters which contains scale coefficients F_j and centers ξ and bandwidths λ of SRBF kernels and reparameterization parameters θ in spatial position (x, y) of mipmap level l. As Figure 3.3 shows, we use $\varpi(x/2, y/2, l)$ as the initial guess to find optimized $\varpi(x, y, l + 1)$. To keep ϖ varying slowly when mipmap level changes, we set a mipmap constrain E_m showed in Equation 3.13 into our objective function E_b in Equation 3.8, where c_{mip} is the weight of mipmap constrain. Our method ensures ϖ change slightly when mipmap level changes, so we can use hardware acceleration to smooth mipmap level switch and still remain great quality.

$$E_m = c_{mip} * \sum_{x,y,l} (\varpi(x, y, l + 1) - \varpi(x/2, y/2, l))^2 \tag{3.13}$$

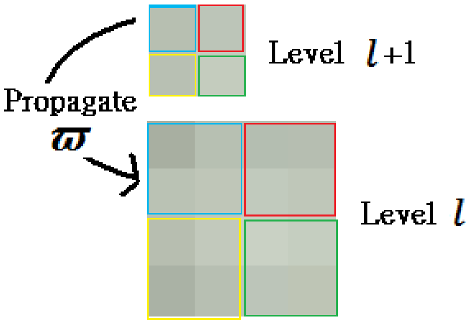


Figure 3.3: We propagate ϖ of mipmap level l+1 to mipmap level l

3.3.4 Smooth Terms

To assure rendering quality, SRBF and reparameterization coefficients need to be continuous in spatial domain. We enforce the spatial smoothness constrains E_s on the SRBF coefficients and reparameterization parameters:

$$E_s = c_{Smooth} * \sum_{x,y,l} \sum_{dx=-2}^{2} \sum_{dy=-2}^{2} (\varpi(x,y,l) - \varpi(x+dx, y+dy, l))^2, \tag{3.14}$$

where c_{Smooth} is used to control the weights of neighbor differences in objective functions.

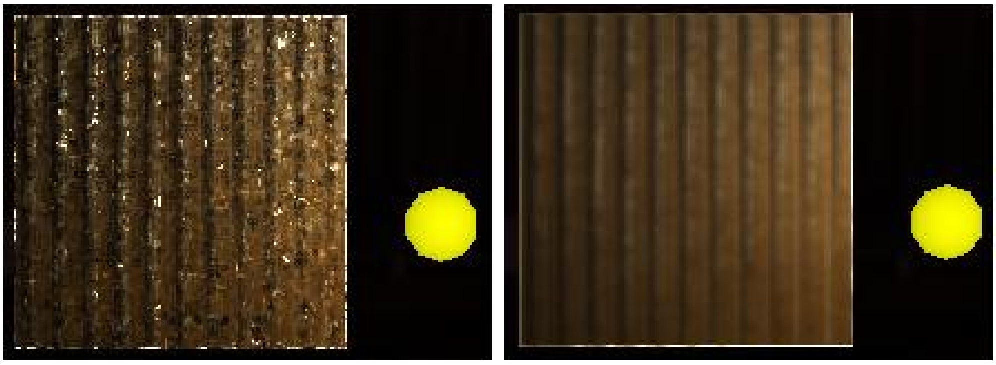

(a) (b)

Figure 3.4: (a)(b) are rendering results without and with smooth terms. Yellow sphere is the point light source located above the right side of mesh.

3.3.5 Optimization

Combining Equations 3.8 3.13 3.14, we have an objective function E:

$$E = E_b + E_m + E_s \tag{3.15}$$

We use the limited memory Broyden-Fletcher-Goldfarb-Shanno(BFGS) method supported by a software package called L-BFGS(from http://www.alglib.net/optimization/lbfgs.php) to solve this optimization problem.

3.3.6 Rendering System

In rendering part, we send result of BTF representation to GPU by mipmap textures. At run time, we derive illumination direction η_i and view direction η_o of each pixel and reconstruct the illumination result from Equation 3.5. As reconstruction is the bottleneck of rendering performance, we use smooth terms in representation stage to make SRBF and reparameterization coefficients continuous in spatial, so we can using texture filter from hardware support to do linear interpolation for SRBF and reparameterization coefficients.

To improve rendering quality, BTF synthesis is necessary. We use the method proposed by Lefebvre el at.[12] to synthesize BTF. We treat the original BTF data $T(x, y, \eta_i, \eta_o)$ as the appearance vector, and use the tensor approximation method to reduce the dimensions of $T(x, y, \eta_i, \eta_o)$ and apply anisometric synthesis method mentioned in [12].

CHAPTER 4

Results

4.1 Data Acquisition

All the BTF data are from BTF Database Bonn(http://btf.cs.uni-bonn.de/). The CORDUROY data and the IMPALLA data and the WOOL data consists of 6561 images, 256x256 resolution, 81 view and 81 light directions. The CEILING data consists of 6561 images, 800x800 resolution, 81 view and 81 light directions, and we resample the CEILING data to 256*256 resolution using Lanczos filter.

4.2 Univariate SRBF vs. Multivariate SRBF

At first, we compare the results of our optimal parameterized BTF model using univariate SRBF and multivariate SRBF. Figure 4.1 shows the RMS error comparison of univariate SRBF and multivariate SRBF. We calculate RMS errors of reconstructed results in different resolution levels of the Mipmap data of original BTF data from all lighting and viewing directions. The reconstructed results are shown in Figure 4.2. Although BRDF data can be modeled efficiently using univariate SRBF in [19], it is difficult to model BTF data using univariate SRBF. One can observe that there is obvious error in univariate results Figure 4.2(e)(f)(g)(h) while the image quality is much better in multivariate SRBF results Figure 4.2(i)(j)(k)(l).

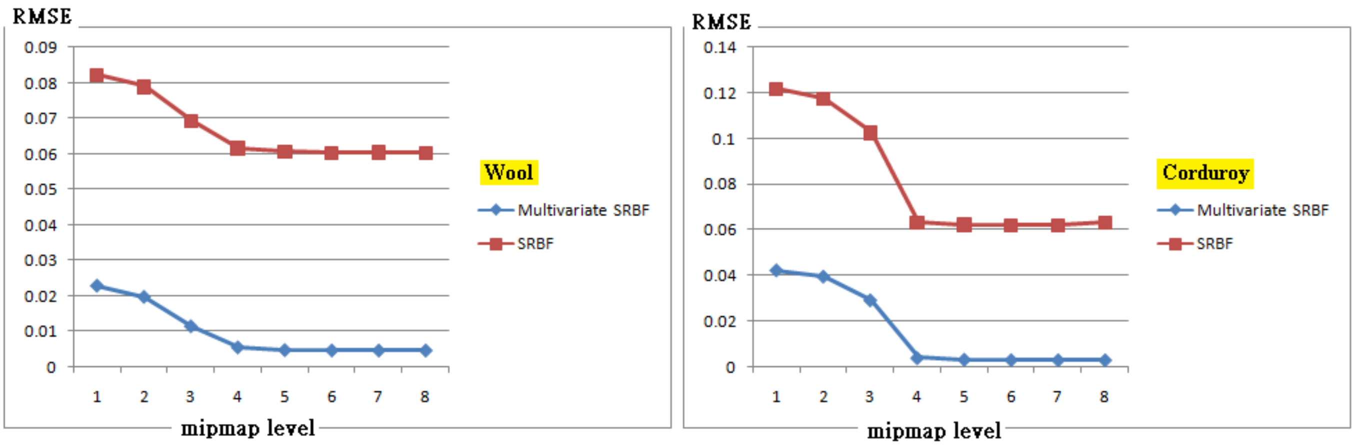
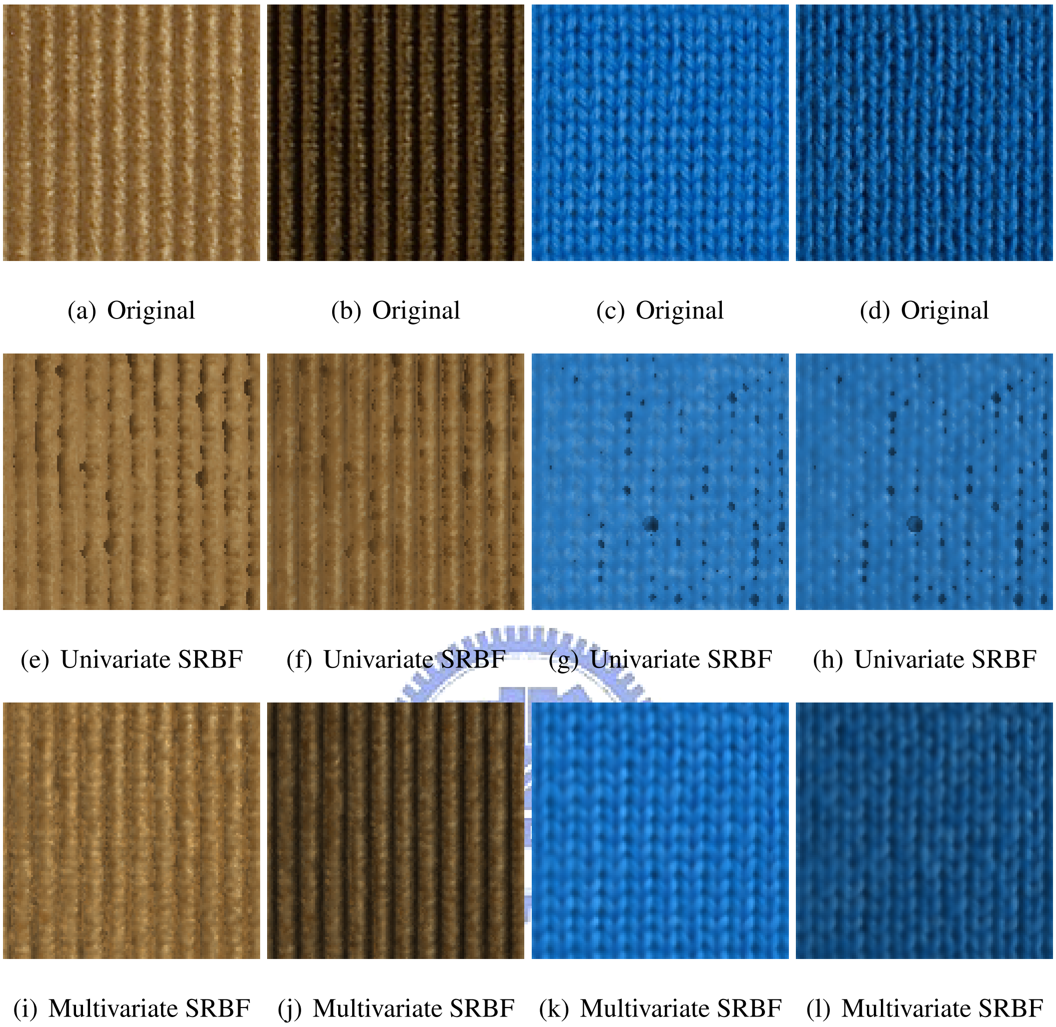


Figure 4.1: RMS error comparison of univariate SRBF and multivariate SRBF; X-axis is mipmap level and Y-axis is RMS error



(a) Original (b) Original (c) Original (d) Original

(e) Univariate SRBF (f) Univariate SRBF (g) Univariate SRBF (h) Univariate SRBF

(i) Multivariate SRBF (j) Multivariate SRBF (k) Multivariate SRBF (l) Multivariate SRBF

Figure 4.2: Reconstructed results comparison of univariate SRBF and multivariate SRBF. (a)(b)(c)(d) are textures of corduroy and wool data from Bonn BTF database, respectively. (e)(f)(g)(h) are reconstructed results using Gaussian SRBFs via optimal reparameterization. (i)(j)(k)(l) are reconstructed results using multivariate Gaussian SRBFs via optimal reparameterization. Twenty SRBF kernels($J = 20$) are used in (e)(f)(g)(h)(i)(j)(k)(l), three multivariate modes($K = 3$) are used in (i)(j)(k)(l).

4.3 Half-way Reparameterization vs. Optimal Reparameterization

We show the improvement of our optimal reparameterization compared to half-way reparameterization. The RMS error comparison is listed in Figure 4.3 and the reconstructed results are showed in Figure 4.4 and Figure 4.5. These experiments are set with mipmap hierarchy and smooth terms, so the result would be smooth in spatial domain. As the ceiling BTF data have less height variation in different spatial position, the reflectance behavior is much simpler, and the RMS error is smaller than the others. Although the RMS error of our optimal reparameterization improves only about ten percents than half-way reparameterization, our optimal reparameterization keeps more details. According to error figure of reconstruction results, errors of half-way reparameterization most are in self-shadows of data because it assumes the data are isotropic in half-way reparameterization, so our optimal reparameterization suits for isotropic and anisotropic data.

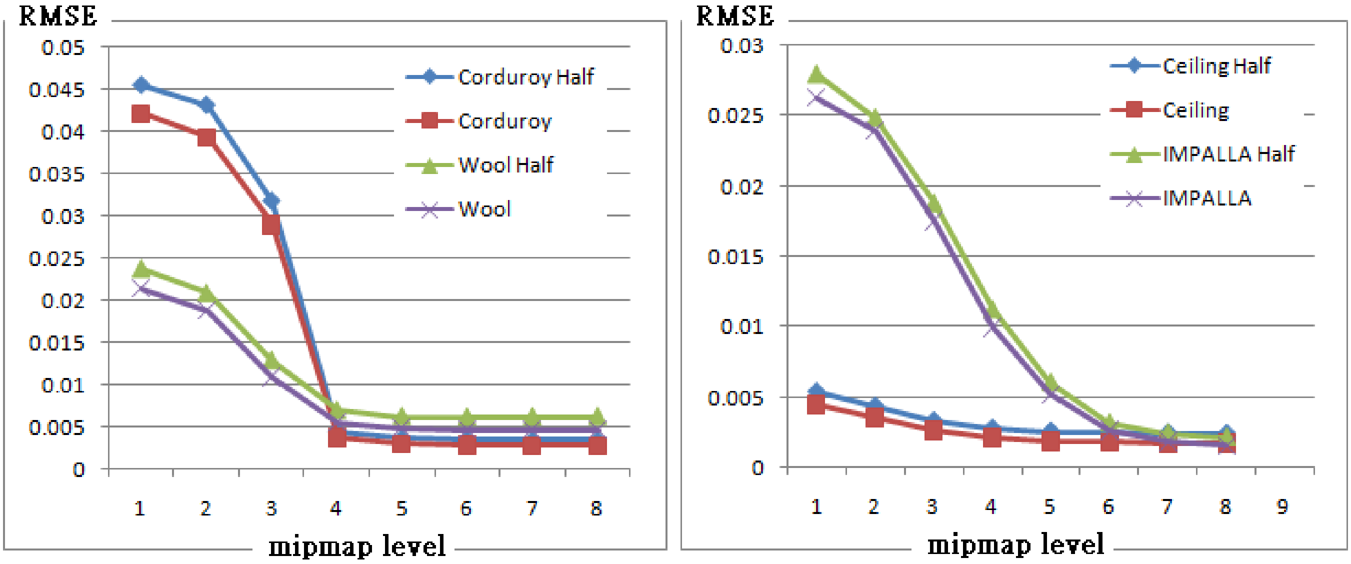
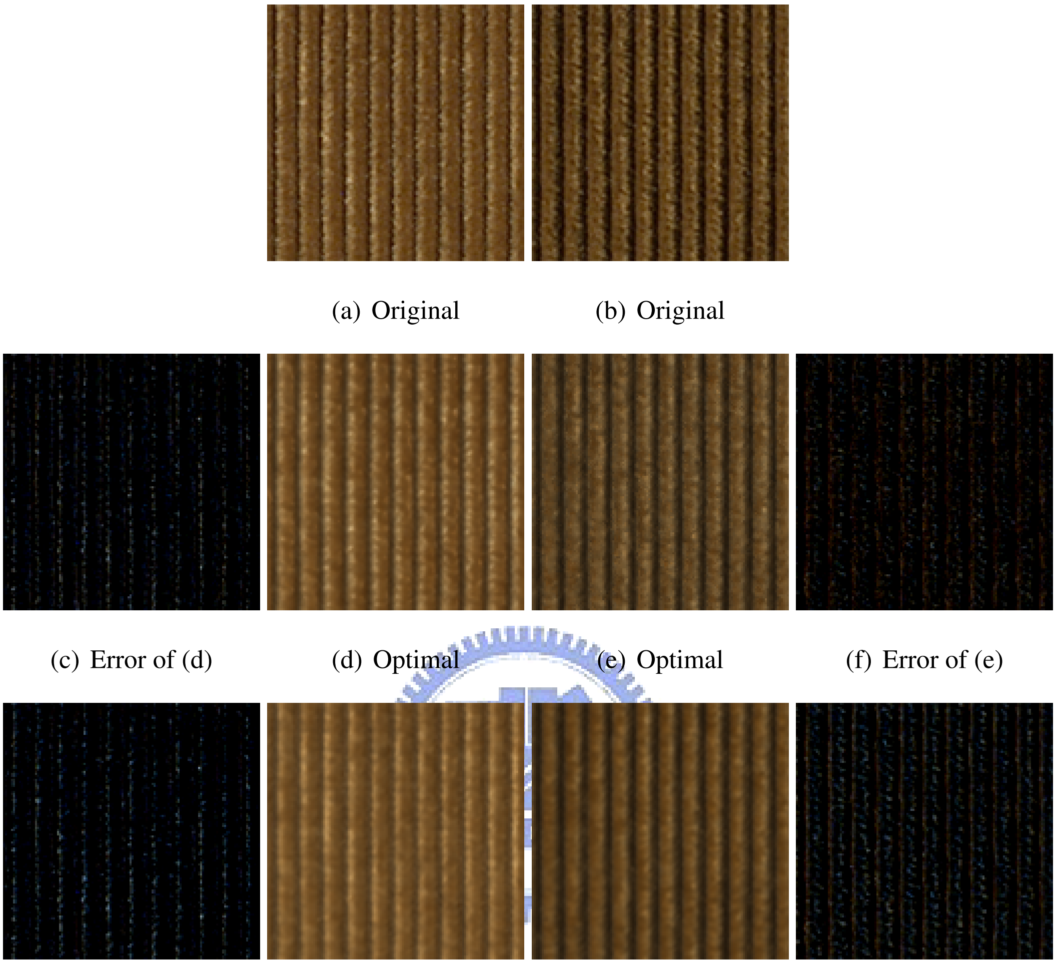


Figure 4.3: RMS error comparison of representation results using our optimal reparameterization and half-way reparameterization; X-axis is mipmap level and Y-axis is RMS error

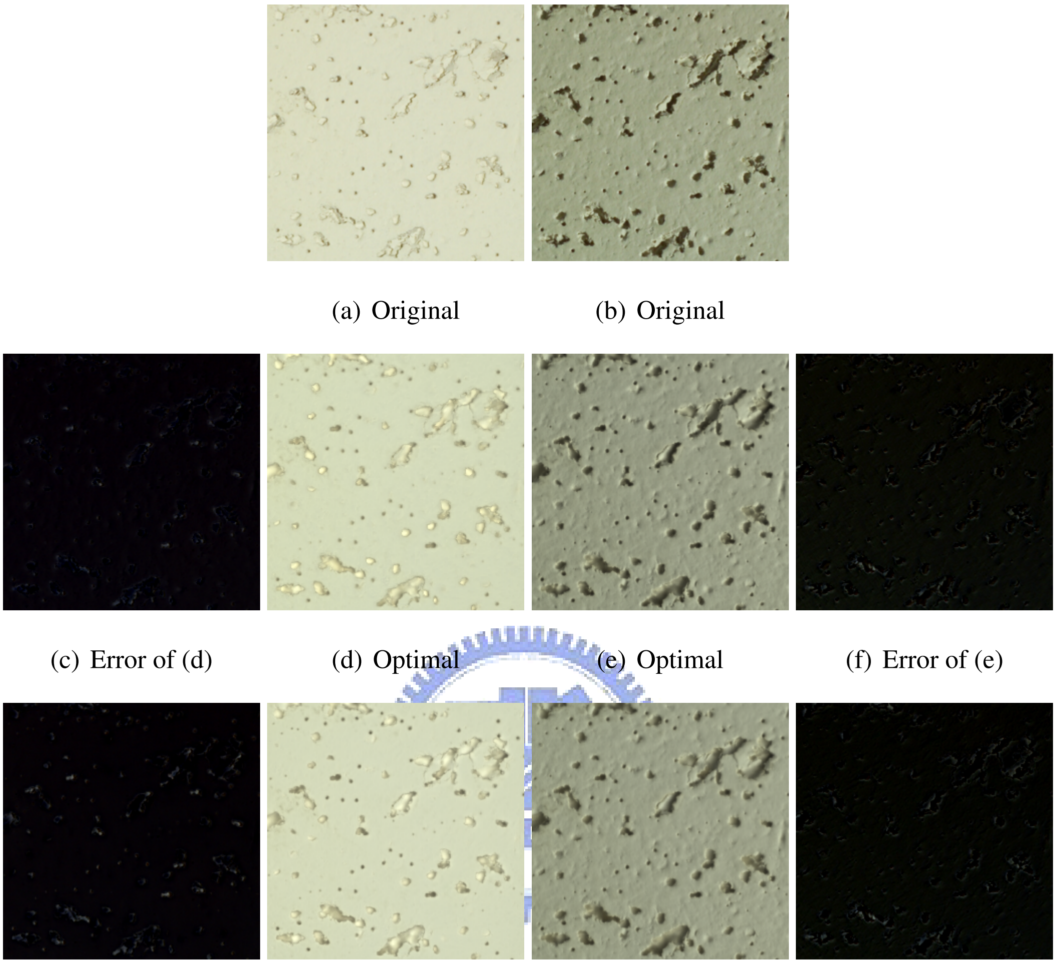


(a) Original (b) Original

(c) Error of (d) (d) Optimal (e) Optimal (f) Error of (e)

(g) Error of (h) (h) Half-way (i) Half-way (j) Error of (i)

Figure 4.4: Reconstructed results comparison of our optimal reparameterization and half-way reparameterization. (a)(b) are textures of corduroy data from Bonn BTF database. (d)(e) are reconstructed results using our optimal reparameterization. (h)(i) are reconstructed results using half-way reparameterization. (c)(f)(g)(j) are error images of (d)(e)(h)(i) compare to (a)(b)(a)(b) respectively. Twenty multivariate SRBF kernels ($J = 20$) and three multivariate modes ($K = 3$) are used in (d)(e)(h)(i). The RMS error of (d)(e)(h)(i) are 0.049926, 0.047594, 0.057197, 0.04814 respectively.

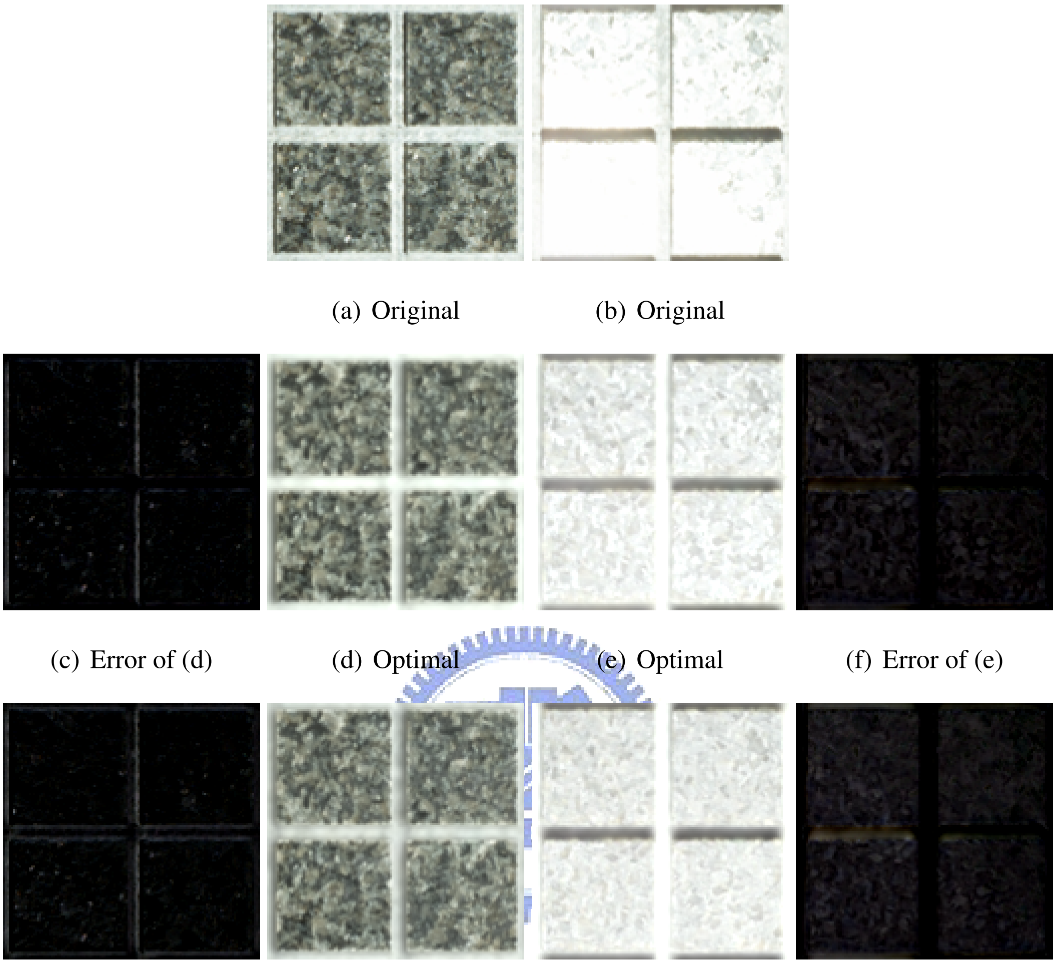


(a) Original (b) Original

(c) Error of (d) (d) Optimal (e) Optimal (f) Error of (e)

(g) Error of (h) (h) Half-way (i) Half-way (j) Error of (i)

Figure 4.5: Reconstructed results comparison of our optimal reparameterization and half-way reparameterization. (a)(b) are textures of ceiling data from Bonn BTF database. (d)(e) are reconstructed results using our optimal reparameterization. (h)(i) are reconstructed results using half-way reparameterization. (c)(f)(g)(j) are error images of (d)(e)(h)(i) compare to (a)(b)(a)(b) respectively. Twenty multivariate SRBF kernels($J = 20$) and three multivariate modes($K = 3$) are used in (d)(e)(h)(i). The RMS error of (d)(e)(h)(i) are 0.029616, 0.040331, 0.032102, 0.042219 respectively.



(a) Original

(b) Original

(c) Error of (d)

(d) Optimal

(e) Optimal

(f) Error of (e)

(g) Error of (h)

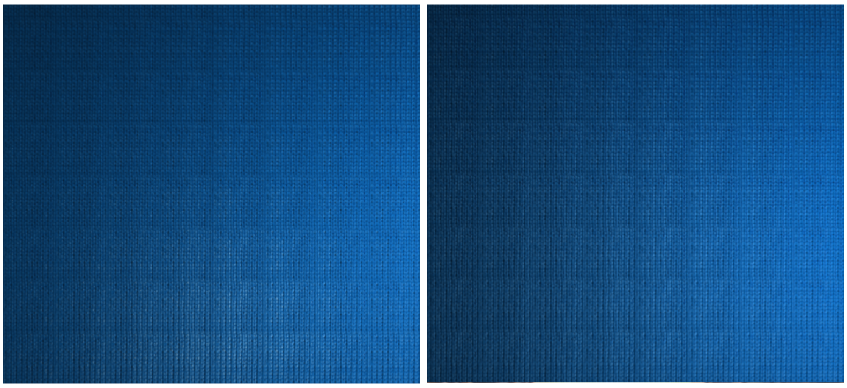
(h) Half-way

(i) Half-way

(j) Error of (i)

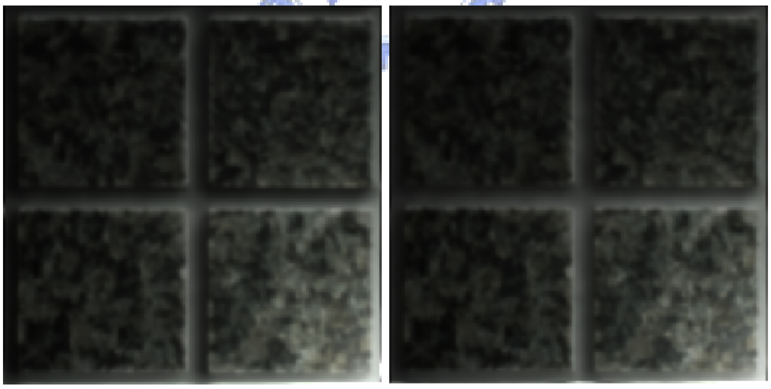
Figure 4.6: Reconstructed results comparison of our optimal reparameterization and half-way reparameterization. (a)(b) are textures of Impalla data from Bonn BTF database. (d)(e) are reconstructed results using our optimal reparameterization. (h)(i) are reconstructed results using half-way reparameterization. (c)(f)(g)(j) are error images of (d)(e)(h)(i) compare to (a)(b)(a)(b) respectively. Twenty multivariate SRBF kernels($J = 20$) and three multivariate modes($K = 3$) are used in (d)(e)(h)(i). The RMS error of (d)(e)(h)(i) are 0.060246, 0.064558, 0.063839, 0.081254 respectively.

The rendering results comparison using optimal reparameterization and half way reparameterization are shown in Figure 4.7 and Figure 4.8. Our optimal reparameterization keep the part of self shadow more than half way reparameterization, and the details are preserved better using our optimal reparameterization.



(a) Optimal (b) Half-way

Figure 4.7: Rendering results comparison of wool using optimal reparameterization and half way reparameterization. Twenty multivariate SRBF kernels($J = 20$) and three multivariate modes($K = 3$) are used.

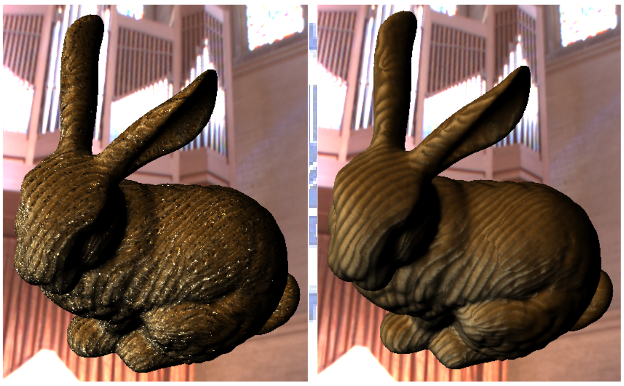


(a) Optimal (b) Half-way

Figure 4.8: Rendering results comparison of impalla using optimal reparameterization and half way reparameterization. Twenty multivariate SRBF kernels($J = 20$) and three multivariate modes($K = 3$) are used.

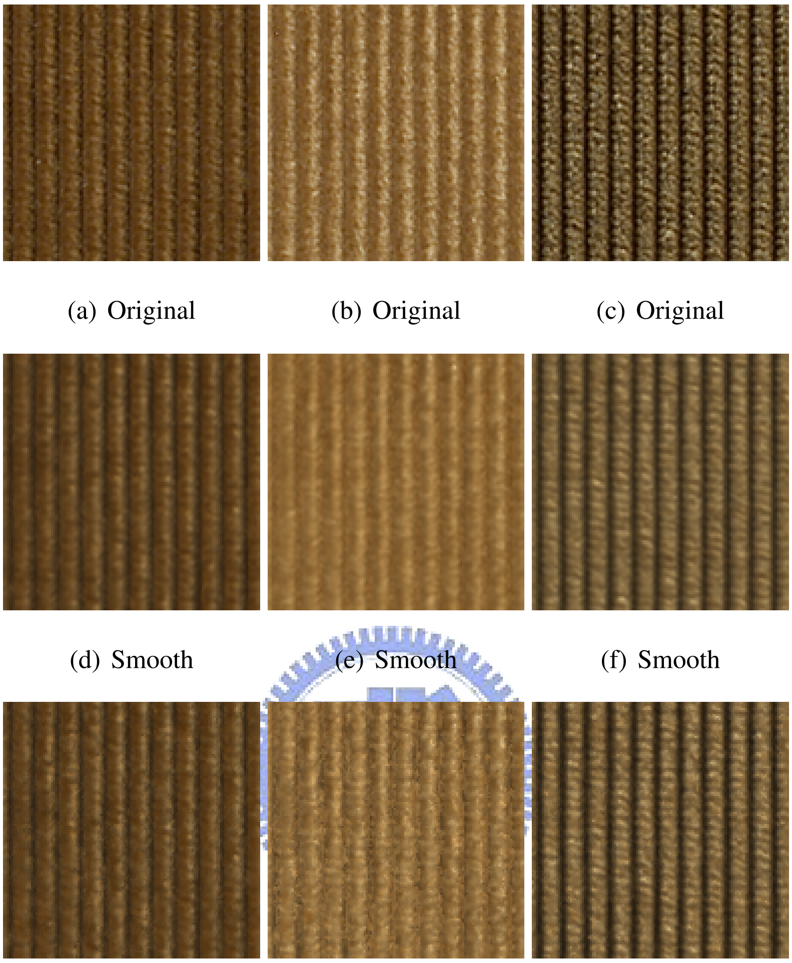
4.4 Smooth terms vs. No Smooth terms

We set smooth terms to improve the quality and performance in rendering. Figure 4.10 shows some results of reconstruction of our BTF model with and without smooth terms. As we can see, details are better preserved than those without smooth terms, but if we do not set smooth terms, we can not use texture filter from hardware support to smooth coefficients ϖ of our BTF model in spatial domain, so we must reconstruct nearby pixels which is the bottleneck of performance and smooth it in color domain to render one pixel. Figure 4.9 shows the rendering results not using smooth terms and using smooth terms with texture filter from hardware support.



(a) No Smooth (b) Smooth

Figure 4.9: Rendering results comparison of corduroy without smooth terms and with smooth terms. Twenty multivariate SRBF kernels($J = 20$) and three multivariate modes($K = 3$) are used.



(a) Original (b) Original (c) Original

(d) Smooth (e) Smooth (f) Smooth

(g) Without smooth (h) Without smooth (i) Without smooth

Figure 4.10: Reconstructed results comparison of our optimal parameter BTF model with smooth terms and without smooth terms. (a)(b)(c) are textures of corduroy data from Bonn BTF database. (d)(e)(f) are reconstructed results with smooth terms. (g)(h)(i) are reconstructed results without smooth terms. Twenty multivariate SRBF kernels ($J = 20$) and three multivariate modes ($K = 3$) are used in (c)(d)(e)(f)(g)(h)(i).

4.5 Time Consumption and Compression Ratio

The modeling time of our optimal reparameterization is 1.5 times bigger than half-way reparameterization, and it takes less time to model BTF data which is more diffuse, such as wool and ceiling.

BTF Data	Elapsed Time	J	K	Reparameterization	Min. Mipmap Level
Corduroy	20 hour 25 min 6 sec	20	3	Half-way	1
Corduroy	29 hour 30 min 55 sec	20	3	Optimal	1
Wool	10 hour 6 min 24 sec	20	3	Half-way	1
Wool	16 hour 3 min 29. sec	20	3	Optimal	1
Ceiling	13 hour 12 min 5 sec	20	3	Half-way	0
Ceiling	20 hour 42 min 23 sec	20	3	Optimal	0
Impalla	12 hour 3 min 54 sec	20	3	Half-way	1
Impalla	17 hour 44 min 0.7 sec	20	3	Optimal	1

Table 4.1: Elapsed time of BTF modeling. J and K denote number of multivariate SRBF kernels and number of multivariate modes.

We store Coefficients ϖ of our BTF model ϖ in each level and each pixel with double, the data size is dependent on number of multivariate SRBF kernels and the number of levels. We find that resolution 128*128 is enough to keep great quality in rendering, so we use level 1 with resolution 128*128 as our minimum mipmap level. We use twenty multivariate SRBF kernels in most cases, so the compression ratio is about 1/6 compare to BTF data from Bonn.

4.6 Rendering Results

Figure 4.11 and Figure 4.12and Figure 4.13 Figure 4.14 are rendering results of wool and corduroy and ceiling BTF data. All these data are modeled using our optimal reparameterization, wool and corduroy data are modeled with twenty BTF kernels and three multivariate

BTF Data	Data Size	Lighting and Viewing Directions	Resolution
Corduroy	327MB	81*81	256*256
Wool	346MB	81*81	256*256
Ceiling	2.05GB	81*81	800*800
Impalla	314MB	81*81	256*256

Table 4.2: Data size of data in BTF database Bonn. All textures are stored with JPEG format

Data Size	J	K	Min. Mipmap Level
52.1MB	20	3	1
26.1MB	10	3	1
204MB	20	3	0
104MB	10	3	0

Table 4.3: Data size of our BTF modeling. J and K denote number of multivariate SRBF kernels and number of multivariate modes.

modes, ceiling data are modeled with ten BTF kernels and three multivariate modes.

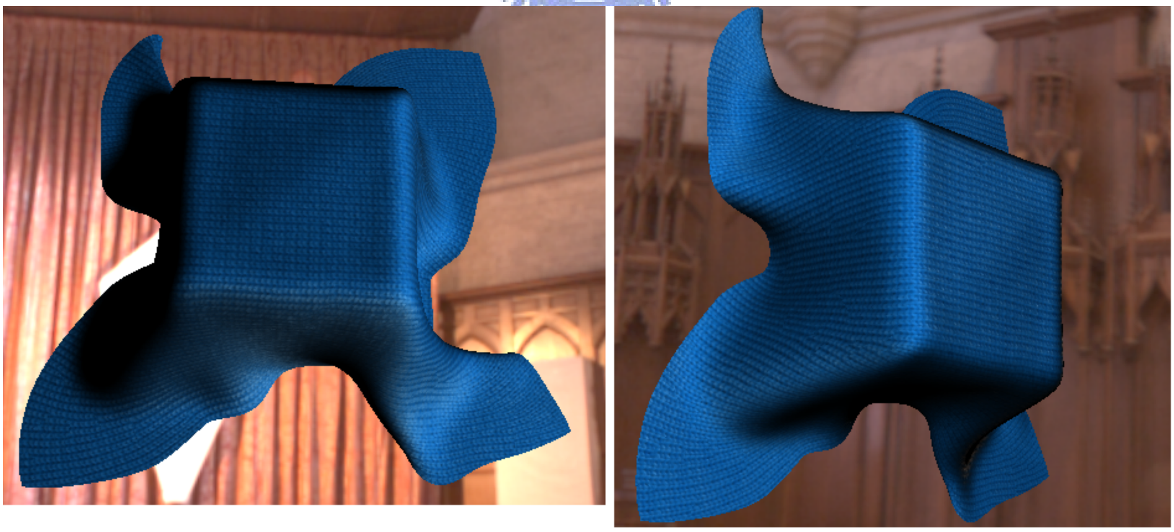


Figure 4.11: Rendering result of wool with optimal reparameterization. (J=20,K=3)

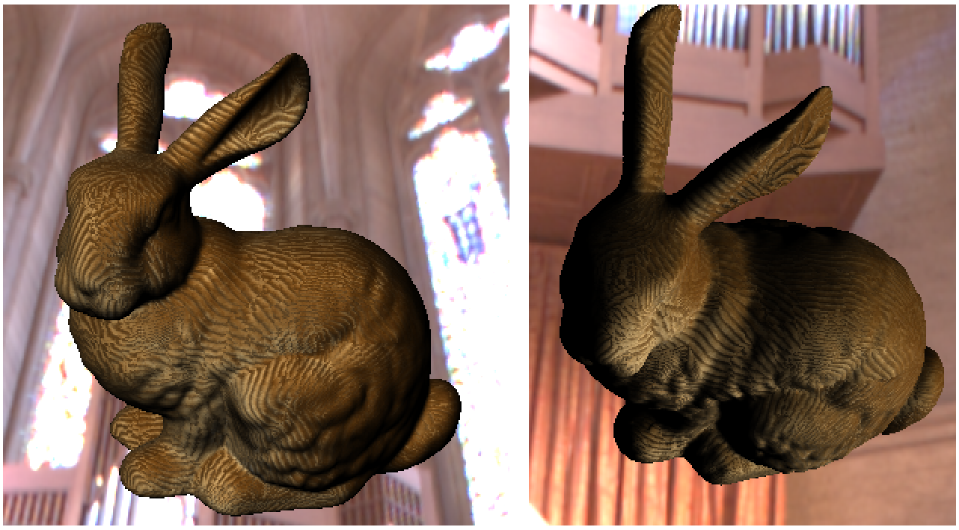


Figure 4.12: Rendering result of corduroy with optimal reparameterization. (J=20,K=3)

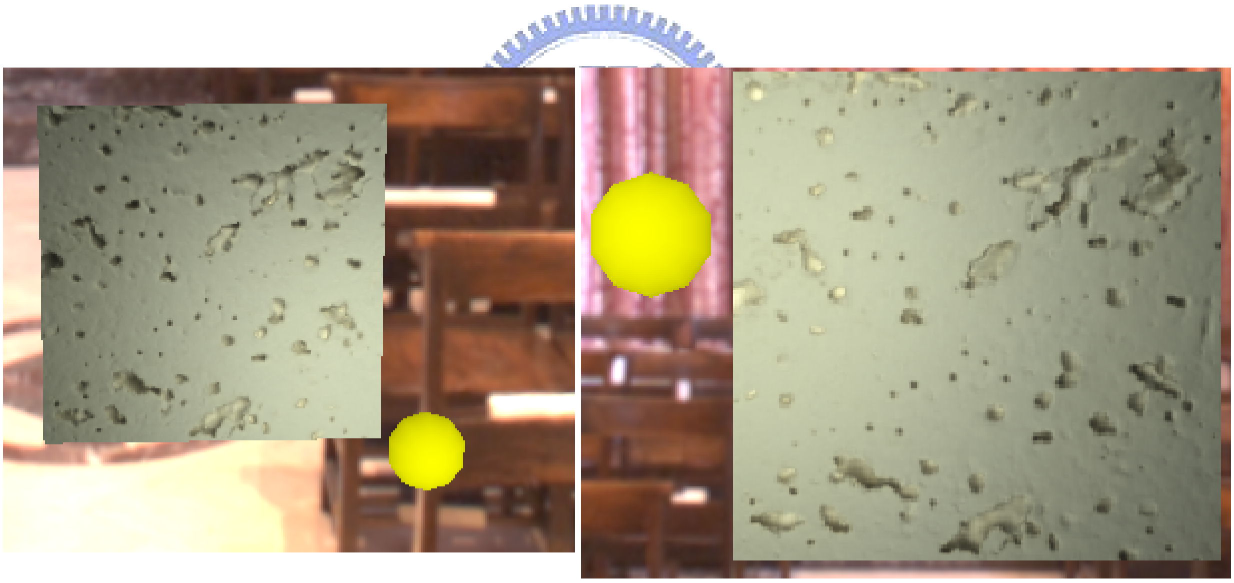


Figure 4.13: Rendering result of ceiling with optimal reparameterization. (J=10,K=3)

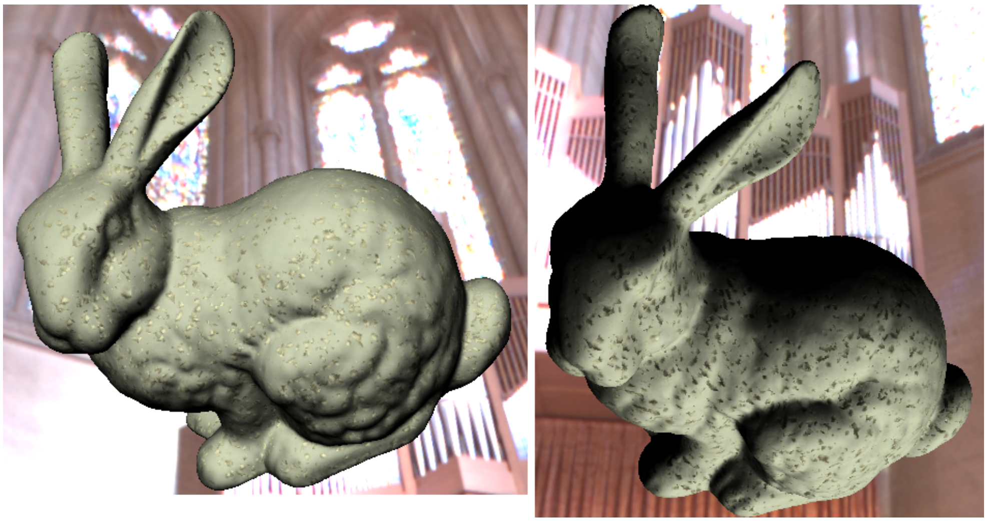


Figure 4.14: Rendering result of ceiling with optimal reparameterization. (J=10,K=3)

The FPS of our BTF rendering is shown in Table 4.4. Evaluation time for each pixel follows the number of multivariate SRBF kernels direct proportion relation, so the FPS using 20 multivariate SRBF kernels is about half of using 10 multivariate SRBF kernels.

Model	triangle	FPS	J	K
Plane	800	50.23	10	3
Plane	800	26.10	20	3
Bunny	69630	18.77	10	3
Bunny	69630	9.42	20	3
cloth	59168	11.63	10	3
cloth	59168	6.15	20	3

Table 4.4: FPS of rendering. J and K denote number of multivariate SRBF kernels and number of multivariate modes.

CHAPTER 5

Conclusion and Future Work

In this thesis, we use a data-driven parametric representation in viewing and lighting space to model the BTF data. Unlike fixed reparameterization, we can define reparameterization function and optimize their parameters to get the optimal reparameterization function. This optimal reparameterization can make BTF modeling more accurately.

To use spatial coherence efficiently, we construct mipmaps of BTF data and use mipmap hierarchy in our optimization process. We can speed up BTF modeling and keep the properties of mipmap which improves the performance and quality in rendering.

EM algorithm are used to speed up the heave work of optimization. We use EM algorithm to find a proper initial guess, so our optimization process works much faster and still remain great quality of results.

Our experiments has shown the improvement of our multivariate SRBF kernel and our optimal reparameterization method compared to univariate SRBF kernel and half-way reparameterization. Our BTF model is efficient both in modeling and rendering, our BTF model use only about one sixths than original BTF data and we can render our BTF model in real time.

Currently, there are some artifacts caused by mipmap and smooth terms. Fitting using multiresolution hierarchy can speed up our framework a lot, but we assume corresponding textels in different mipmap levels are similar. When the corresponding textels are totally different, it will induce large error, so how to solve this problem is the most important task in the future.

We smooth our BTF model in spatial dimension by adding difference value of nearby(5*5 square) into objective function. When the data differs a lot in this square, our smooth terms will cause bad influence, it increase error ratio in represent result. In the future, robust error matrix will be applied into our method to fix this problem.

Bibliography

[1] M. Alex, O. Vasilescu, and D. Terzopoulos. Tensortextures: Multilinear image-based rendering. *ACM Transactions on Graphics*, pages 336–342, 2004.

[2] W. chun Ma, S. hsiang Chao1, Y. ting Tseng, Y. yu Chuang, C. fa Chang, B. yu Chen, and M. Ouhyoung. Level-of-detail representation of bidirectional texture functions for real-time rendering. In *Proceedings of I3D 2005*, pages 187–194, 2005.

[3] O. G. Cula and K. J. Dana. Compact representation of bidirectional texture functions. In *Proceedings of CVPR 2001*, pages 1041–1047, 2001.

[4] K. J. Dana, B. V. Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. In *Proceedings of ACM SIGGRAPH 1999*, volume 18, pages 1–34, 1999.

[5] W. R. Davis. *Cornelius Lanczos Collected Published Papers With Commentaries*. North Carolina State University, 1999.

[6] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and spatially-varying brdfs from photometric stereo. In *Proceedings of ICCV 2005*, pages 341–348, 2005.

[7] J. Kautz and M. D. Mccool. Interactive rendering with arbitrary brdfs using separable approximations. In *Proceedings of EGWR 1999*, pages 247–260, 1999.

[8] M. L. Koudelka, S. Magda, P. N. Belhumeur, and D. J. Kriegman. Acquisition, compression, and synthesis of bidirectional texture functions. In *Proceedings of Texture 2003*, pages 59–64, 2003.

[9] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *Proceedings of ACM SIGGRAPH 1997*, pages 117–126, 1997.

[10] D. Lathauwer, B. D. Moor, and J. Vandewalle. On the best rank-1 and rank-(r1,r2,. . .,rn) approximation of higher-order tensors. *SIAM J Matrix Anal Appl*, 21:1324–1342, 2000.

[11] J. Lawrence, A. Ben-artzi, C. Decoro, W. Matusik, H. Pfister, R. Ramamoorthi, and S. Rusinkiewicz. Inverse shade trees for non-parametric material representation and editing. *ACM Transactions on Graphics*, 25:735–745, 2006.

[12] S. Lefebvre and H. Hoppe. Appearance-space texture synthesis. In *Proceedings of ACM SIGGRAPH 2006*, pages 541–548, 2006.

[13] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Proceedings of Texture 2003*, 43(1):29–44, 2001.

[14] M. D. McCool, J. Ang, and A. Ahmad. Homomorphic factorization of brdfs for high-performance rendering. In *Proceedings of ACM SIGGRAPH 2001*, pages 171–178, 2001.

[15] F. J. Narcowich and J. D. Ward. Nonstationary wavelets on the m-sphere for scattered data. *Applied and Computational Harmonic Analysis*, 3(4):324–336, 1996.

[16] R. Ramamoorthi and P. Hanrahan. Frequency space environment map rendering. In *Proceedings of ACM SIGGRAPH 2002*, pages 517–526, 2002.

[17] S. M. Rusinkiewicz. A new change of variables for efficient brdf representation. In *Eurographics Workshop on Rendering*, pages 11–22, 1998.

[18] M. Sattler, R. Sarlette, and R. Klein. Efficient and realistic visualization of cloth. In *Proceedings of Eurographics Symposium on Rendering*, pages 167–178, 2003.

[19] Y. ting Tsai and Z. chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transactions on Graphics*, 25:967–976, 2006.

[20] H. Wang, Q. Wu, L. Shi, Y. Yu, and N. Ahuja. Out-of-core tensor approximation of multidimensional matrices of visual data. In *Proceedings of ACM SIGGRAPH 2005*, pages 527–535, 2005.

[21] T. Zickler, S. Enrique, R. Ramamoorthi, and P. Belhumeur. Reflectance sharing: Image-based rendering from a sparse set of images. In *Proceedings of Eurographics Symposium on Rendering*, pages 253–264, 2005.