Modeling Bidirectional Texture Functions with Multivariate Spherical Radial Basis Functions

Yu-Ting Tsai, Kuei-Li Fang, Wen-Chieh Lin, *Member, IEEE,* and Zen-Chung Shih, *Member, IEEE*

Abstract—This paper presents a novel parametric representation for bidirectional texture functions. Our method mainly relies on two original techniques, namely multivariate *spherical radial basis functions* (SRBFs) and optimized parameterization. Firstly, since the surface appearance of a real-world object is frequently a mixed effect of different physical factors, the proposed sum-of-products model based on multivariate SRBFs especially provides an intrinsic and efficient representation for heterogenous materials. Secondly, optimized parameterization particularly aims at overcoming the major disadvantage of traditional fixed parameterization. By using a parametric model to account for variable transformations, the parameterization process can be tightly integrated with multivariate SRBFs into a unified framework. Finally, a hierarchical fitting algorithm for bidirectional texture functions is developed to exploit spatial coherence and reduce computational cost. Our experimental results further reveal that the proposed representation can easily achieve high-quality approximation and real-time rendering performance.

Index Terms—Reflectance and shading models, bidirectional texture functions, parameterization, spherical radial basis functions.

1 Introduction

Real-world surface reflectance, micro-scale appearance, and realistic lighting effects are too complicated to be described with simple analytic models. State-of-the-art data-driven rendering algorithms thus synthesize high-quality images from precomputed or measured reflectance data. Over the last decades, there have been tremendous advances in this field. For example, image-based rendering methods [1], [2], [3] generate virtual images from novel view directions by interpolating pre-captured images. Since they assume no specific reflectance characteristics of object surfaces, appearance of real-world objects can be faithfully rendered.

Moreover, the pioneering work by Dana et al. [4] further introduced the *bidirectional texture function* (BTF) to model spatially-varying reflectance distributions over a 2D surface. A BTF is a 6D function that combines textures and *bidirectional reflectance distribution functions* (BRDFs) to account for the appearance of a 2D surface under various illumination and view conditions. Therefore, images rendered from measured BTFs can realistically exhibit complex lighting effects and detailed mesostructures of real-world objects, including the microgeometry of rough surfaces, self-shadows, and multiple light scattering. In addition to rendering applications, BTFs provide realistic texture models for computer vision applications, such as segmentation, robust visual classification, retrieval or illumination/view invariant methods dealing with images of textured natural materials [5]

Nevertheless, a compact and efficient representation for BTFs remains challenging in practice. The enormous amount of BTF data frequently becomes the performance bottleneck at run-time and prohibits further analysis in computer vision and graphics applications. This challenge thus has stimulated the recent development of sophisticated approximation algorithms for large-scale surface appearance data [6], [7], [8], [9], [10], [11], [12]. In addition to the challenge of dealing with tremendous data size, a BTF data set is a mixed effect of various types of physical factors. This high-dimensional nature is so complicated that simple analytic models often fail to describe the *multivariate* behavior of a BTF.

In this paper, we introduce a novel functional representation to solve the tremendous data size and complex behavior problems in BTF modeling. The complex behaviors of a reflectance function are described as a weighted sum of the products of several *univariate* basis functions, which form a multivariate representation. 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 several univariate SRBFs[1] [13]. Although the optimization process of such a general model may be difficult, our experimental results demonstrate that a fast and practical implementation is feasible even for large-scale appearance data sets such as BTFs.

To obtain a compact representation, it is also well-known that transforming the parameters of a reflectance function into another parametric space, which we refer to as *parameterization*, can improve approximation efficiency [9], [14], [15], [16]. However, previous articles have considered only fixed transformation functions, little

- *Y.-T. Tsai is with the Department of Computer Science and Engineering, Yuan Ze University, 135 Yuan-Tung Road, Chung-Li City, Taoyuan, Taiwan 320, R.O.C. E-mail: yttsai@saturn.yzu.edu.tw.*
- *K.-L. Fang, W.-C. Lin, and Z.-C. Shih are with the Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu, Taiwan 300, R.O.C. E-mail: zean.fang@gmail.com, wclin@cs.nctu.edu.tw, and zcshih@cs.nctu.edu.tw.*

[1] Throughout this paper, the univariate SRBF is referred to as the original SRBF that was introduced by Tsai and Shih [13].

attention has been paid to a data-dependent method [17], [18]. In this paper, we further propose to learn a set of optimized parameterization functions for a given reflectance data set. By using a parametric representation to model the transformation functions, the parameterization process can be tightly integrated into our multivariate optimization framework. Previous fixed transformation methods, such as the half-way vector, thus become special cases in this general framework.

It should be noted that the multivariate SRBF representation and optimized parameterization only focus on BRDF modeling. For spatially-varying materials like BTFs, we adopt the apparent BRDF representation [5], [10], [19], [20]. Since this representation describes a BTF as a set of texelwise BRDFs, we can apply the proposed model to separately approximate the reflectance data of each texel. However, directly optimizing the model parameters of each texel is time-consuming. We thus further propose a hierarchical fitting algorithm to exploit spatial coherence in a BTF and reduce the computational cost.

In summary, this paper makes the following contributions:

- A compact functional representation based on a linear combination of multivariate SRBFs is introduced to efficiently model the complex behaviors of measured reflectance fields. Since our representation is a series of continuous functions, no additional interpolation or filtering techniques are required for rendering reflectance functions from novel illumination and view directions at run-time.
- An automatic parameterization framework is proposed to learn the best parameter transformation function from a given reflectance data set and a given form of transformation with unknown parameters. It can seamlessly cooperate with our multivariate representation to improve the approximation efficiency for reflectance functions.
- A hierarchical fitting algorithm for BTFs is presented to exploit spatial coherence and accelerate the approximation process. It is particularly suitable for multi-resolution analysis and data-driven rendering applications due to the inherent mipmap pyramid construction.
- The overall result of this paper is a compact and hardware-friendly representation for BTFs, which can be easily implemented on modern *graphics processing units* (GPUs).

The remainder of this paper is organized as follows. First, the literature on parameterization and approximation methods for surface appearance models is reviewed in Section 2. We then describe the main ideas of this paper by introducing the multivariate SRBF representation in Section 3 and the optimized parameterization framework in Section 4. A hierarchical fitting algorithm for BTFs and other practical implementation details, such as the initial guess of the multivariate SRBF representation, parameter optimization process, and run-time rendering, are respectively presented in Section 5 and Section 6. Finally, we demonstrate and discuss the experimental results in Section 7, and conclude this paper in Section 8 to shed some lights on future research directions.

2 Related Work

In this section, we first briefly review some previous parameterization methods for reflectance functions (Section 2.1). We also summarize three main categories of modern approximation algorithms for reflectance fields: functional linear models (Section 2.2), non-parametric models (Section 2.3), and probabilistic models (Section 2.4). Due to limited paper length, our review mostly concentrates on BTFs. For a comprehensive survey on BTF modeling in computer vision and graphics, interested readers may further refer to [5].

2.1 Parameterization of Reflectance Functions

Half-way [15] and reflected vector [21] parameterizations for BRDFs have been shown to be effective in modeling highly specular materials. Stark et al. [22] also proposed several physically interpretable parameterizations for isotropic BRDFs. Although their method naturally forms a barycentric coordinate system that contains some geometric information, it does not provide data-dependent parameterizations for different real-world BRDFs. Namely, the parameterizations proposed in [22] are all fixed. In recent years, various fixed parameterizations have been applied to approximate spatially-varying surface appearance [9], [16], further demonstrating their promising potentials. In general, parameterization is beneficial to reduce the dimensionality of reflectance functions, which leads to a compact and low-dimensional representation for surface appearance. It also greatly increases the data coherence that can be exploited by various approximation algorithms.

Nevertheless, previous parameterization techniques are limited to fixed transformation functions. It is unknown which parameter transformation would perform the best for the reflectance data at hand. Although Cole [17] introduced an automatic and data-dependent parameterization method for BRDFs, this approach is limited to linear transformations. By contrast, we propose to learn the best parameterization functions from data, within a certain non-linear functional form, by introducing additional parameters into our appearance representation. The proposed method thus successfully combines parameterization and reflectance model estimation in a unified framework to fill the gap between these two problems that have been solved separately in previous methods.

2.2 Functional Linear Models

Functional linear analysis of materials has received great attention in real-time rendering applications due to its compactness and efficiency. The main concept is to expand a reflectance function as a linear combination of simple basis functions. In this category, choosing an appropriate basis is one of the major research issues as it significantly influences image quality and rendering performance. Previous approaches have modeled (spatially-varying) reflectance fields using parametric kernels [7], [23], [24], [25], polynomials [26], [27], radial basis functions [12], [16], [28], [29], spherical harmonics [21], [30], [31], and wavelets [32]. In general, kernel and radial basis functions provide efficient run-time rendering performance and high image quality for all-frequency materials. However, they usually require computationally expensive non-linear optimization for parameter estimation, which becomes even worse and impractical when modeling materials with spatially-varying reflectance.

In this paper, we propose to describe a reflectance function as a weighted sum of the products of several univariate basis functions, and take a step further to search for data-dependent parameterization for illumination and view variations. This general weighted sum-of-products representation not only models the complex multivariate behavior of reflectance functions, but also includes various popular parametric reflection models as its subset. Moreover, under the proposed hierarchical optimization framework, our experimental results show that the time-consuming non-linear parameter fitting process for BTFs can be robustly pre-conditioned and accelerated by a bottom-up approach, leading to a multi-resolution representation and an efficient off-line algorithm.

2.3 Non-Parametric Models

Non-parametric models can be regarded as functional models that do not have pre-defined forms of basis functions. In this category, an appropriate basis is learnt from data for an accurate representation, rather than prior information specified by researchers. The most popular approaches in computer vision and graphics include clustering and dimensionality reduction techniques, such as variants of principal component analysis [6], [14], [19], [33], [34], [35], [36], [37], [38], matrix factorization [9], [18], [39], [40], [41], tensor approximation [11], [42], and vector quantization [43], [44].

Although non-parametric methods are data-driven models that yield accurate and flexible representations, the amount of compressed data is still cumbersome when compared to other categories of approximation algorithms. For BTFs, it is also difficult to achieve real-time performance for run-time analysis and rendering in computer vision and graphics applications. Additionally, special interpolation or estimation techniques are required to synthesize surface appearance from novel illumination and view directions that are not sampled in raw data. By contrast, our algorithm provides not only a higher compression ratio but also a real-time rendering rate with comparable image quality. Furthermore, novel view and illumination directions can be easily handled by our continuous multivariate model and parameterization, and spatial mipmap texture filtering for run-time rendering is fully supported and inherent in our hierarchical appearance representation.

2.4 Probabilistic Models

In this category, spatial correlations among appearance data are described with probability density functions so that similar appearance data can be synthesized from estimated parameters of distributions and noise maps [45], [46]. Recently, Haindl and Filip [8] further proposed a multi-scale probabilistic BTF model based on the casual autoregressive random field and combined range maps to enhance the surface roughness of rendered objects.

Although probabilistic models can achieve a high compression ratio, their main goal is efficient and seamless BTF synthesis, not an optimal reconstruction of the original BTF data. Additionally, the run-time rendering process is slow and currently not GPU-friendly. By contrast, our algorithm can be easily implemented on modern GPUs and provides a better tradeoff among compression ratio, image quality, and rendering performance.

3 Multivariate SRBFs

3.1 Mathematical Formulation

Let ω and ξ denote two points on the unit sphere $\mathbb{S}^m$ in $\mathbb{R}^{m+1}$. A univariate *spherical radial basis function* (SRBF) [13] on $\mathbb{S}^m$ is defined as a function $G(\cos\phi) = G(\omega \cdot \xi)$ that depends on the geodesic distance ϕ between ω and ξ. A popular example of univariate SRBFs is the univariate Gaussian SRBF kernel[2]:

$$G(\cos\phi|\lambda) = G(\omega \cdot \xi|\lambda) = e^{\lambda(\omega\cdot\xi)-\lambda}, \tag{1}$$

where $\lambda \in \mathbb{R}$ represents the bandwidth parameter that controls the concentration of a univariate SRBF, and ξ is also known as the SRBF center. A univariate spherical function $F(\omega) \in \mathbb{R}$ thus can be approximated with a linear combination of J univariate SRBFs as

$$F(\omega) \approx \sum_{j=1}^{J} \beta_j G(\omega \cdot \xi_j|\lambda_j), \tag{2}$$

where $\beta_j \in \mathbb{R}$ denotes the basis coefficient of the j-th univariate SRBF.

Nevertheless, there are two problems when applying Eq. (2) to model a reflectance function. First, the appearance of real-world materials is an effect of various physical factors. Whether these factors are visible or hidden, the observed reflectance distribution is often a function of at least two different variables, e.g. illumination and view directions. However, Eq. (2) is a univariate model

[2] It is easy to verify that the *normalized* univariate Gaussian SRBF on $\mathbb{S}^m$ is equivalent to the von Mises-Fisher distribution. This suggests that various techniques developed for von Mises-Fisher distributions can be applied to univariate Gaussian SRBFs with only minor modifications.

that only takes a single direction on $\mathbb{S}^m$ into account. This suggests that a multivariate representation may be more favorable to describe the complex behaviors of a reflectance function. Second, even though we can apply Eq. (2) to respectively model the reflection distribution for each illumination/view direction, the outcome is a discrete and often non-compact representation. It is nontrivial to generalize this representation to estimate the distributions from novel illumination/view directions.

To represent the appearance of a reflectance field under different physical conditions, we can construct a multivariate SRBF from the product of several symmetric univariate SRBFs. For complex or heterogenous materials, multiple multivariate SRBFs can be further linearly mixed to derive a general weighted sum-of-products model. More formally, let $\Omega = \{\omega_n\}_{n=1}^N$ and $\Xi = \{\xi_n\}_{n=1}^N$ denote two N-element point sets, with ω_n and ξ_n on the unit sphere $\mathbb{S}^{m_n}$ in $\mathbb{R}^{m_n+1}$. We define a multivariate SRBF on the Cartesian product space $\mathbb{S}^{m_1} \times \cdots \times \mathbb{S}^{m_N}$ as

$$G\big(\Omega|\Xi\big) = G\big(\omega_1, \ldots, \omega_N | \xi_1, \ldots, \xi_N\big) = \prod_{n=1}^{N} G(\omega_n \cdot \xi_n), \quad (3)$$

and the multivariate Gaussian SRBF kernel thus corresponds to

$$G\big(\Omega|\Xi, \Lambda\big) = e^{\sum_{n=1}^{N} \left(\lambda_n (\omega_n \cdot \xi_n) - \lambda_n\right)}, \quad (4)$$

where $\Lambda = \{\lambda_n\}_{n=1}^N$ is the set of bandwidth parameters of the involved univariate SRBFs, and Ξ is also called the SRBF center set. Similarly, an N-variate function $F(\Omega) \in \mathbb{R}$, with each variable defined on $\mathbb{S}^{m_n}$, can be approximated by a weighted sum-of-products representation:

$$F(\Omega) \approx \sum_{j=1}^{J} \beta_j G\big(\Omega|\Xi_j, \Lambda_j\big) = \sum_{j=1}^{J} \beta_j \prod_{n=1}^{N} G\big(\omega_n \cdot \xi_{j,n} | \lambda_{j,n}\big), \quad (5)$$

where $\Xi_j = \{\xi_{j,n}\}_{n=1}^N$ and $\Lambda_j = \{\lambda_{j,n}\}_{n=1}^N$ are respectively the center set and the bandwidth set of the j-th multivariate SRBF.

3.2 Example

Consider a BRDF $\rho(\omega_l, \omega_v) \in \mathbb{R}$, where ω_l and ω_v respectively denote the illumination and view directions on $\mathbb{S}^2$. Based on Eq. (5), we can approximate $\rho(\omega_l, \omega_v)$ as

$$\rho(\omega_l, \omega_v) \approx \sum_{j=1}^{J} \beta_j G\big(\omega_l \cdot \xi_{j,\omega_l} | \lambda_{j,\omega_l}\big) G\big(\omega_v \cdot \xi_{j,\omega_v} | \lambda_{j,\omega_v}\big). \quad (6)$$

Note that Eq. (6) is similar to many factorization-based representations for BRDFs, e.g., principal component analysis [14] and non-negative matrix factorization [47], but our multivariate representation, like other parametric models, is more compact and it is also more intuitive to interpret or edit the derived parameters. This relation to matrix factorization methods also suggests the potential of applying our multivariate SRBF representation to approximate reflectance functions. As we will present in Section 4, Eq. (6) can be further extended into a more general model than previous methods when combined with optimized parameterization.

4 Optimized Parameterization

4.1 Mathematical Formulation

Previous articles have reported that fixed parameterization for a reflectance function, such as the half-way and difference vectors [15], can significantly influence the performance of approximation algorithms. However, since a pre-defined parameterization method often relies on certain assumptions of material properties, it may be inadequate to handle various real-world reflectance data. For example, the half-way parameterization tends to align the specular peak of a reflectance function, but the shadowing and masking effects of micro-facets are ignored. This situation will become even worse for a real-world BTF, since it is usually measured over a rough surface with complex meso-structures and light scattering properties.

To overcome the disadvantages of fixed parameterization, we propose to learn a set of optimized transformation functions for a given reflectance data set. Since our goal is to obtain a compact reflectance representation, we choose to model the transformation functions using parametric equations. This particularly allows the parameterization process to be tightly integrated into our multivariate SRBF framework. Although the derived optimal solution is constrained to a certain functional form, our experimental results show that even a linear combination of the parameters of a reflectance function, followed by projection onto the unit sphere, can be more effective than previous fixed parameterization approaches. Finding the *truly* optimal parameterization using non-parametric models thus is left as our future work.

More formally, let $\psi(\Omega|\Theta) \in \mathbb{S}^m$ be a transformation function that depends on a given set of parameterization coefficients $\Theta = \{\theta_i \in \mathbb{R}\}_{i=1}^{I_\Theta}$, where I_Θ denotes the total number of parameterization coefficients in Θ and is specified by users. We would like to find an optimal solution to Θ so that a multivariate spherical function $F(\Omega) \in \mathbb{R}$ can be efficiently approximated by transforming it into another univariate spherical function $F_p\big(\psi(\Omega|\Theta)\big) \in \mathbb{R}$ that is more suitable for univariate SRBF expansions (Eq. (2)):

$$F(\Omega) = F_p\big(\psi(\Omega|\Theta)\big) \approx \sum_{j=1}^{J} \beta_j G\big(\psi(\Omega|\Theta) \cdot \xi_j | \lambda_j\big). \quad (7)$$

From Eq. (7), it is also intuitive to extend the same concept to transform $F(\Omega)$ into an N_p-variate spherical function $F_p(\Psi) \in \mathbb{R}$ as

$$\begin{aligned} F(\Omega) &= F_p(\Psi) = F_p\big(\psi_1(\Omega|\Theta_1), \ldots, \psi_{N_p}(\Omega|\Theta_{N_p})\big) \\ &\approx \hat{F}_p(\Psi) = \sum_{j=1}^{J} \beta_j \prod_{n=1}^{N_p} G\big(\psi_n(\Omega|\Theta_n) \cdot \xi_{j,n} | \lambda_{j,n}\big), \quad (8) \end{aligned}$$

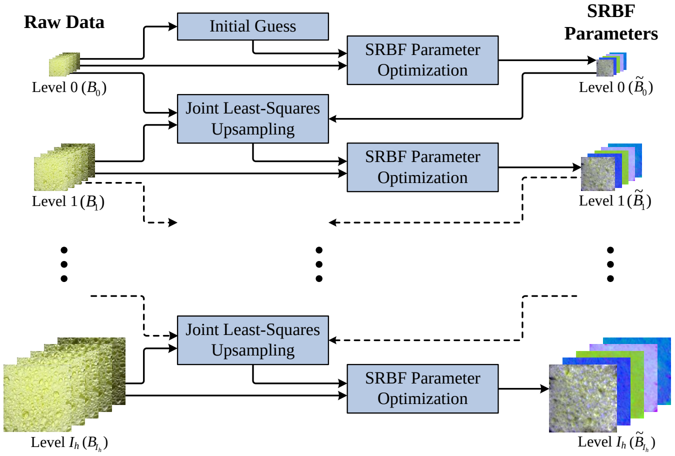

Raw Data
Initial Guess
SRBF Parameters
SRBF Parameter Optimization
Level 0 (B_0)
Level 0 ($\tilde{B}_0$)
Joint Least-Squares Upsampling
SRBF Parameter Optimization
Level 1 (B_1)
Level 1 ($\tilde{B}_1$)
Joint Least-Squares Upsampling
SRBF Parameter Optimization
Level I_h (B_{I_h})
Level I_h ($\tilde{B}_{I_h}$)

Fig. 1. Approximate a BTF using the proposed hierarchical fitting algorithm.

where $\Psi = \{\psi_n(\Omega|\Theta_n)\}_{n=1}^{N_p}$ is a set of N_p transformation functions, $\Theta_n = \{\theta_{i,n}\}_{i=1}^{I_{\Theta_n}}$ specifies the parameterization coefficient set with I_{Θ_n} elements for the n-th transformation function, and $\hat{F}_p(\Psi)$ denotes the approximate multivariate SRBF representation of $F_p(\Psi)$. Note that the number of variables of $F_p(\Psi)$, namely N_p, is not necessarily identical to that of $F(\Omega)$, but rather can be specified by users. This flexibility particularly allows our representation to accurately model various complex behaviors of a real-world reflectance function.

In summary, we combine the proposed multivariate SRBF representation and optimized parameterization to derive the parameterized multivariate SRBF representation (Eq. (8)), and solve its parameters by minimizing the following objective function:

$$E = E_{err} + E_{addl}, \tag{9}$$

where

$$E_{err} = \int_{\mathbb{S}^{m_1}} \cdots \int_{\mathbb{S}^{m_N}} \left(F(\Omega) - \hat{F}_p(\Psi) \right)^2 d\omega_1 \cdots d\omega_N \tag{10}$$

is the expected squared error between $F(\Omega)$ and $\hat{F}_p(\Psi)$, and E_{addl} denotes the additional energy terms that should also be minimized for a robust and satisfying solution. Fore more details about E_{addl} and the practical algorithm for solving Eq. (9), please refer to Sections 5 and 6.

4.2 Example

We again take the BRDF $\rho(\omega_l, \omega_v)$ for an example. Based on Eq. (8), if we choose a trivariate SRBF representation ($N_p = 3$), Eq. (6) can be expressed as

$$\rho(\omega_l, \omega_v) \approx \sum_{j=1}^{J} \beta_j \prod_{n=1}^{3} G\big(\psi_n(\omega_l, \omega_v|\Theta_n) \cdot \xi_{j,n}|\lambda_{j,n}\big), \tag{11}$$

while each transformation function can be modeled with the normalization of a linear combination of ω_l and ω_v as follows:

$$\psi_n(\omega_l, \omega_v|\Theta_n) = \frac{\theta_{1,n}\omega_l + \theta_{2,n}\omega_v}{\|\theta_{1,n}\omega_l + \theta_{2,n}\omega_v\|_2}, \tag{12}$$

where $\theta_{1,n}$ and $\theta_{2,n}$ are the parameterization coefficients of the n-th transformation function, and $\|\cdot\|_2$ denotes the ℓ^2 norm of a vector. Note that when $J = 1$, Eq. (11) is similar to the mathematical formulation of homomorphic factorization [40], but our multivariate representation allows a linear combination of multiple multivariate functions to be used to model heterogeneous materials, which is particularly important for representing BTF data sets. Moreover, Eq. (12) was inspired by the fact that many common parameterizations, such as the halfway, illumination, and view vectors, are its special cases. It is also noteworthy that these three parameterizations were employed in the implementation of homomorphic factorization [40], which further implies the practical effectiveness of our model.

5 Hierarchical Fitting Algorithm

In previous two sections, we have introduced the multivariate SRBF representation (Section 3) and optimized parameterization (Section 4) to model a single reflectance function. To extend our method to model a BTF, we represent the BTF as a set of texelwise BRDFs and respectively approximate the reflectance data of each texel. However, this brute-force approach is actually time-consuming even with our GPU-based implementation (Section 6.2). Similar to the multi-resolution reflectance framework in [29], we present a hierarchical fitting algorithm to reduce the computational cost and preserve spatial coherence, while simultaneously constructing the mipmap pyramid for high-quality rendering on GPUs at run-time.

5.1 Overview

Our hierarchical fitting algorithm operates on a given BTF pyramid $\{B_i\}_{i=0}^{I_h}$ and an initial guess of each texel at the coarsest level. As illustrated in Fig. 1, our algorithm consists of a sequence of upsampling and optimization stages from the coarsest level B_0 to the finest level B_{I_h}. For a pyramid level $i > 0$, the upsampling stage (Section 5.2) derives the initial solution of each texel at level i from $\tilde{B}_{i-1}$, where $\tilde{B}_{i-1}$ denotes the optimized results of level $i-1$. Instead of using traditional image interpolation techniques, such as bi-cubic or Lanczos filtering, we propose a joint least-squares upsampling algorithm by exploiting the relation between B_i and B_{i-1} to assist the resampling of $\tilde{B}_{i-1}$.

After that, the optimization stage (Section 5.3) updates the initial solution of each texel at level i based on our parameterized multivariate SRBF representation. To take advantage of hardware texture filtering during run-time rendering, we introduce additional spatial smoothness energy terms in the objective function, which will constrain the parameter coherence of adjacent texels. The

TABLE 1
The hierarchical SRBF fitting algorithm.

```
Procedure: HierarchyOptimize($\{B_i\}_{i=0}^{I_h}$, $\tilde{B}_0, J, N_p$)
Input: BTF pyramid $\{B_i\}_{i=0}^{I_h}$, number of SRBFs $J$, number of variate $N_p$ for parameterization, and initial guess $\tilde{B}_0$.
Output: Optimized parameters of each level $\{\tilde{B}_i\}_{i=0}^{I_h}$.
for each texel $\mathbf{t}$ at level 0 do
  $\tilde{B}_0(\mathbf{t}) \leftarrow$ Optimize($B_0(\mathbf{t}), \tilde{B}_0(\mathbf{t}), J, N_p$)
end for
for $i \leftarrow 1$ to $I_h$ do
  Upsample $\tilde{B}_{i-1}$ to obtain $\tilde{B}_i$
  repeat
    for each texel $\mathbf{t}$ at level $i$ do
      $\tilde{B}_i(\mathbf{t}) \leftarrow$ Optimize($B_i(\mathbf{t}), \tilde{B}_i(\mathbf{t}), J, N_p$)
    end for
  until convergence
end for

Function: Optimize($F, \tilde{F}, J, N_p$)
Input: Reflectance function $F$, $J$, $N_p$, and initial guess $\tilde{F} = \left\{ \{\beta_j, \Xi_j, \Lambda_j\}_{j=1}^{J}, \{\Theta_n\}_{n=1}^{N_p} \right\}$.
Output: $\{\beta_j, \Xi_j, \Lambda_j\}_{j=1}^{J}$ and $\{\Theta_n\}_{n=1}^{N_p}$.
repeat
  Update SRBF coefficient set $\{\beta_j\}_{j=1}^{J}$
  Update SRBF center set $\{\Xi_j\}_{j=1}^{J}$
  Update SRBF bandwidth set $\{\Lambda_j\}_{j=1}^{J}$
  Update parameterization coefficient set $\{\Theta_n\}_{n=1}^{N_p}$
until convergence
Update all parameters to obtain a locally optimal solution
```

procedure ”HierarchyOptimize” in Table 1 summarizes the pseudo-code of the overall fitting process. Note that finding an appropriate initial guess for the coarsest level is non-trivial. We postpone the discussion of this issue to Section 6.1. For implementation details about the procedure ”Optimize” in Table 1, please refer to Section 6.2.

5.2 Upsampling Stage

Given the optimized parameters of pyramid level $i-1$, namely $\tilde{B}_{i-1}$, an appropriate initial solution of each texel at level i is derived in the upsampling stage. This initial solution significantly influences the quality and computational cost for approximating B_i. However, since some details of B_i may be lost when downsampled to B_{i-1} during BTF pyramid construction, traditional image interpolation techniques are inadequate for a high-quality upsampling from $\tilde{B}_{i-1}$. Our key observation is that because both B_i and B_{i-1} are available in this stage, their relation can be employed to ’jointly’ derive the initial guess of $\tilde{B}_i$ from $\tilde{B}_{i-1}$. This relies on the assumptions that the relation between $\tilde{B}_i$ and $\tilde{B}_{i-1}$ is similar to that between B_i and B_{i-1}, and $\tilde{B}_{i-1}$ approximates B_{i-1} with low reconstruction errors.

Specifically, a BTF texel $\mathbf{t}$ in B_i can be approximated with a linear combination of a set of texels in B_{i-1}:

$$B_i(\mathbf{t}) \approx \sum_{\mathbf{t}' \in \mathcal{N}_{i-1}(\mathbf{t})} w_{\mathbf{t}'} B_{i-1}(\mathbf{t}'), \tag{13}$$

where $\mathcal{N}_{i-1}(\mathbf{t})$ represents the set of participating texels in B_{i-1} for $\mathbf{t}$, and $w_{\mathbf{t}'}$ denotes the blending weight of a texel $\mathbf{t}' \in \mathcal{N}_{i-1}(\mathbf{t})$. In current implementation, we heuristically determine $\mathcal{N}_{i-1}(\mathbf{t})$ as the neighboring texels of $B_{i-1}(\mathbf{t})$ within a user-defined window. Since both B_i and B_{i-1} are known in this stage, the unknown blending weights thus can be derived by solving an unconstrained linear least-squares problem, and then applied to compute the initial solution of $\tilde{B}_i$ as

$$\tilde{B}_i(\mathbf{t}) = \sum_{\mathbf{t}' \in \mathcal{N}_{i-1}(\mathbf{t})} w_{\mathbf{t}'} \tilde{B}_{i-1}(\mathbf{t}'). \tag{14}$$

It should be noted that reconstructing the interpolated parameters of participating texels may not exactly correspond to interpolating their reconstructed values. For example, since the center and bandwidth sets are related to the exponents of multivariate Gaussian SRBFs, Eq. (14) will linearly blend these two sets of different texels in the logarithmic space, which is definitely not equivalent to the weighted summation of the Gaussian SRBFs of each texel. However, they would be close to each other if the spatial variations in the parameters of participating texels are smooth. Since the model parameters of level $i-1$ were already updated with spatial smoothness energy terms in the optimization stage of previous iteration, we have found that the proposed joint least-squares upsampling algorithm works very well in practice.

5.3 Optimization Stage

In this stage, the initial guess of each texel at pyramid level i is individually updated to obtain a locally optimal solution that minimizes the objective function in Eq. (9). Similar to the mixture model of [29], [48], we introduce an additional smoothness energy term E_{addl} in Eq. (9) to guarantee spatial coherence in the derived parameters. Previous articles [29], [48] proposed to align only centers of Gaussian/spherical functions, but we have found that other model parameters should also be appropriately aligned in our experiments. There are three main reasons for this. First, the alignment can be regarded as a regularization process that avoids overfitting. Second, the smoothness energy terms particularly allow much faster convergence for the optimization process. Third, linear interpolation, instead of nonlinear filtering [29], [48], on model parameters for efficient run-time performance (Section 6.3) can be employed if all the model parameters were appropriately aligned. This would cause some slight loss of high-frequency features in the fitted and rendered results, but one may consider it as a tradeoff between run-time performance and image quality.

In this paper, E_{addl} is thus defined as follows:

$$E_{addl} = \mu_\beta E_\beta + \mu_\xi E_\xi + \mu_\lambda E_\lambda + \mu_{\xi,\lambda} E_{\xi,\lambda} + \mu_\theta E_\theta, \quad (15)$$

$$E_\beta = \sum_{\mathbf{t}' \in \mathcal{N}_i'(\mathbf{t})} \sum_{j=1}^{J} \left(\beta_j(\mathbf{t}) - \beta_j(\mathbf{t}')\right)^2, \quad (16)$$

$$E_\xi = \sum_{\mathbf{t}' \in \mathcal{N}_i'(\mathbf{t})} \sum_{j=1}^{J} \sum_{n=1}^{N} \left(1 - \xi_{j,n}(\mathbf{t}) \cdot \xi_{j,n}(\mathbf{t}')\right), \quad (17)$$

$$E_\lambda = \sum_{\mathbf{t}' \in \mathcal{N}_i'(\mathbf{t})} \sum_{j=1}^{J} \sum_{n=1}^{N} \left(\lambda_{j,n}(\mathbf{t}) - \lambda_{j,n}(\mathbf{t}')\right)^2, \quad (18)$$

$$E_{\xi,\lambda} = \sum_{\mathbf{t}' \in \mathcal{N}_i'(\mathbf{t})} \sum_{j=1}^{J} \sum_{n=1}^{N} \left\| \lambda_{j,n}(\mathbf{t}) \xi_{j,n}(\mathbf{t}) - \lambda_{j,n}(\mathbf{t}') \xi_{j,n}(\mathbf{t}') \right\|_2^2, \quad (19)$$

$$E_\theta = \sum_{\mathbf{t}' \in \mathcal{N}_i'(\mathbf{t})} \sum_{n=1}^{N_p} \sum_{j=1}^{I_{\Theta_n}} \left(\theta_{j,n}(\mathbf{t}) - \theta_{j,n}(\mathbf{t}')\right)^2, \quad (20)$$

where E_β, E_ξ, E_λ, $E_{\xi,\lambda}$, and E_θ are respectively the smoothness energy terms of basis coefficients, centers, bandwidths, and parameterization coefficients, $\mathcal{N}_i'(\mathbf{t})$ denotes the set of participating texels at levels i and $i{-}1$ for $\mathbf{t}$, and μ_β, μ_ξ, μ_λ, $\mu_{\xi,\lambda}$, and μ_θ are respectively the user-defined weights for E_β, E_ξ, E_λ, $E_{\xi,\lambda}$, and E_θ.

In this way, the smoothness energy terms in Eq. (15) will guide the model parameters of $\mathbf{t}$ to approach those of $\mathcal{N}_i'(\mathbf{t})$. Specifically, E_β, E_λ, $E_{\xi,\lambda}$, and E_θ will penalize large squared errors between the basis coefficients, bandwidths, and parameterization coefficients of $\mathbf{t}$ and $\mathcal{N}_i'(\mathbf{t})$, while E_ξ will minimize the cosine of angular differences between the centers of $\mathbf{t}$ and $\mathcal{N}_i'(\mathbf{t})$.

Note that Eq. (19) is specially designed for multivariate Gaussian SRBFs as their centers and bandwidths are highly coupled with each other. Moreover, the SRBF parameters in Eq. (15)–(20) should depend on the level index i and texel $\mathbf{t}$, but we drop them for notational simplicity. Similar to $\mathcal{N}_{i-1}(\mathbf{t})$ in the upsampling stage, $\mathcal{N}_i'(\mathbf{t})$ is defined as the 'valid' neighboring texels of $\mathbf{t}$ at levels i and $i{-}1$ within a user-defined window, while a valid texel is referred to as the texel whose model parameters have ever been optimized. Since a change in the model parameters of $\mathbf{t}$ will influence those of $\mathcal{N}_i'(\mathbf{t})$, the above process is repeated until the parameters of each spatial location at level i converge or a user-defined maximum number of passes is reached.

6 Implementation Details

6.1 Initial Guess

Since the approximation quality of the proposed SRBF representation with optimized parameterization (Eq. (8)) significantly depends on the initial guess of model parameters, we propose a heuristic technique to determine an effective initial guess that reduces approximation errors and computational cost. For the initial guess of parameterization coefficients, we have found that previous fixed parameterizations generally provide an appropriate starting point if they are special cases of the adopted transformation functions. Take Eq. (12) for an example, the initial guess of the first three parameterization coefficient sets can be explicitly set to the halfway, illumination, and view parameterizations, while the remainders are randomly generated.

Once the initial values of parameterization coefficients are determined, the initial guess of basis coefficients, center sets, and bandwidth sets can be estimated by treating a multivariate reflectance function as multiple univariate functions, and iteratively processes one variable of the reflectance function at a time. Specifically, the key idea is that if we collect all the reflectance data of a single variable (say ω_n), the resulting data set will be the observations of a univariate spherical function. Therefore, we can separately apply the scattered univariate SRBF representation [13] to approximate the observations of each univariate function, but additionally constrain that the representations for different data sets should employ the same center and bandwidth sets. After carefully examining the derived parameters, it is obvious that the basis coefficients form the observations of another multivariate spherical function without dependence on ω_n. The above process thus can be repeatedly performed to remove a single variable at each iteration until all model parameters are obtained.

Note that one can always process the variables of a multivariate function in an arbitrary order. It is also feasible to find the optimal order for a small number of variables by a brute-force approach. However, we do not consider this issue in current implementation. Developing an efficient technique for determining the optimal order is left as a possible research direction in the future.

6.2 Optimization Process

In our current implementation, we apply the L-BFGS-B solver [49], [50] to optimize the parameters of the proposed model. Instead of solving all the model parameters at the same time, we employ an iterative alternating least-squares method (the procedure ”Optimize” in Table 1) that updates only one set of parameters at each step, while leaving the others unchanged. This scheme often yields better results, since the four sets of model parameters, including coefficient, center, bandwidth, and parameterization coefficient sets, are highly coupled with each other.

During each iteration, the gradient computation is performed on GPUs using NVIDIA CUDA [51]. The computed results are then transferred from GPUs to the host memory for the L-BFGS-B solver to update model parameters on CPUs. Since the gradient computation is one of the main performance bottlenecks in the optimization process, we have found that this approach can reduce the computation time by a factor of 2 to 5.

TABLE 2

Statistics of the proposed model and tensor approximation for multi-resolution BTF approximation. T1 and T2 denote different settings of tensor approximation for comparison under similar compression ratio and rendering speed.

Material	Carpet				Hole				Impalla				Sponge				Wool			
Illum. directions	120				51				81				120				81			
View directions	90				51				81				90				81			
Spatial resolution	128×128				128×128				128×128				128×128				128×128			
Raw data (GB)	2.64				0.64				1.6				2.64				1.6			
Method	T1	T2	Fix.	Opt.	T1	T2	Fix.	Opt.	T1	T2	Fix.	Opt.	T1	T2	Fix.	Opt.	T1	T2	Fix.	Opt.
SRBFs: J	-	-	16	12	-	-	24	24	-	-	12	12	-	-	8	8	-	-	12	12
Variates: N_p	-	-	3	4	-	-	3	3	-	-	3	3	-	-	3	3	-	-	3	3
Reduced illum.	16	16	-	-	24	20	-	-	12	12	-	-	12	10	-	-	12	12	-	-
Reduced view	12	4	-	-	12	8	-	-	12	8	-	-	8	4	-	-	12	8	-	-
Weight: μ_β	-	-	5.0×10^{-4}	7.5×10^{-5}	-	-	5.0×10^{-4}	7.5×10^{-4}	-	-	2.5×10^{-5}	1.0×10^{-7}	-	-	1.0×10^{-7}	1.0×10^{-7}	-	-	7.5×10^{-6}	7.5×10^{-6}
Weight: μ_ξ	-	-	8.0×10^{-3}	2.0×10^{-3}	-	-	7.5×10^{-3}	3.0×10^{-3}	-	-	6.5×10^{-3}	2.8×10^{-3}	-	-	1.0×10^{-3}	7.0×10^{-4}	-	-	2.0×10^{-3}	2.0×10^{-3}
Weight: μ_λ	-	-	3.8×10^{-3}	1.0×10^{-3}	-	-	4.5×10^{-3}	2.5×10^{-3}	-	-	2.5×10^{-3}	1.0×10^{-3}	-	-	1.0×10^{-4}	4.8×10^{-5}	-	-	1.5×10^{-3}	1.5×10^{-3}
Weight: $\mu_{\xi,\lambda}$	-	-	0.0	0.0	-	-	2.5×10^{-5}	1.0×10^{-5}	-	-	1.0×10^{-5}	0.0	-	-	0.0	0.0	-	-	0.0	0.0
Weight: μ_θ	-	-	5.0×10^{-4}	5.0×10^{-4}	-	-	1.0×10^{-3}	3.0×10^{-3}	-	-	7.5×10^{-4}	7.5×10^{-4}	-	-	5.0×10^{-7}	5.0×10^{-7}	-	-	4.0×10^{-3}	4.0×10^{-3}
Comp. data (MB)	8.01	2.68	8.0	7.83	12.01	10.01	12.0	12.25	6.01	4.01	6.0	6.25	4.01	1.67	4.0	4.25	6.01	4.01	6.0	6.25
SE ratio (%)	2.67	4.3	4.01	3.62	4.64	6.56	5.53	4.95	2.8	3.4	3.18	2.73	0.57	0.82	0.86	0.81	1.36	1.84	1.77	1.76
Comp. time (hr.)	0.09	0.07	5.81	13.96	0.12	0.1	7.1	8.83	0.14	0.12	5.13	8.09	0.1	0.1	3.32	5.21	0.05	0.04	7.01	12.15

TABLE 3

Rendering performance of the proposed model and tensor approximation for multi-resolution BTF approximation.

Model	Bunny					Bunny					Cloth					Bunny					Cloth				
Material	Carpet					Hole					Impalla					Sponge					Wool				
Vertices	36k					36k					30k					36k					30k				
Coord. texture	2048×2048					2048×2048					1024×1024					2048×2048					1024×1024				
Method	Raw	T1	T2	Fix.	Opt.	Raw	T1	T2	Fix.	Opt.	Raw	T1	T2	Fix.	Opt.	Raw	T1	T2	Fix.	Opt.	Raw	T1	T2	Fix.	Opt.
Total data (MB)	2700	31.5	25.17	24.0	23.83	650.3	38.5	34.5	28.0	28.25	1640.3	16.0	13.5	10.0	10.25	2700	25.0	21.92	20.0	20.25	1640.3	16.0	13.5	10.0	10.25
Frames per sec.	<0.01	69.86	128.7	125.42	124.15	<0.05	42.5	69.97	76.89	74.36	<0.02	104.21	134.09	137.97	135.21	<0.01	134.63	274.53	294.79	269.63	<0.02	94.82	119.84	122.18	116.42

6.3 Run-Time Rendering

The rendering process of approximated BTFs based on multivariate SRBFs is quite simple and intuitive. To utilize mipmap texture filtering on GPUs, we concatenate the SRBF parameters at each level into several two-dimensional texture arrays[3], with one (or more if necessary) texture array for one category of SRBF parameter sets[4]. For meso-structure synthesis, we apply appearance-space texture synthesis [52] on raw data to obtain the spatial coordinate texture S.

The rendering process thus consists of the following steps:

1) For current pixel $\mathbf{p}$, sample the synthesized texture S for the BTF spatial coordinates $\mathbf{t_p}$.
2) Sample the texture(s) for all the SRBF parameters that correspond to $\mathbf{t_p}$.
3) The shading color of pixel $\mathbf{p}$ is then given by performing the reconstruction according to the adopted multivariate SRBF representation.

For a pixel, note that we do not reconstruct the shading color of each participating texel and then perform mipmap filtering, but instead we filter the SRBF parameters of each participating texel first and reconstruct the final shading color. When the derived SRBF parameters of each BTF texel are smooth enough, this approach usually increases the rendering performance by a factor of about 6 for trilinear mipmap filtering without noticeable artifacts. In general, the performance gain strongly depends on the utilized filtering technique. The more sophisticated the filtering technique is, the more performance gain our approach can achieve.

7 Results and Discussion

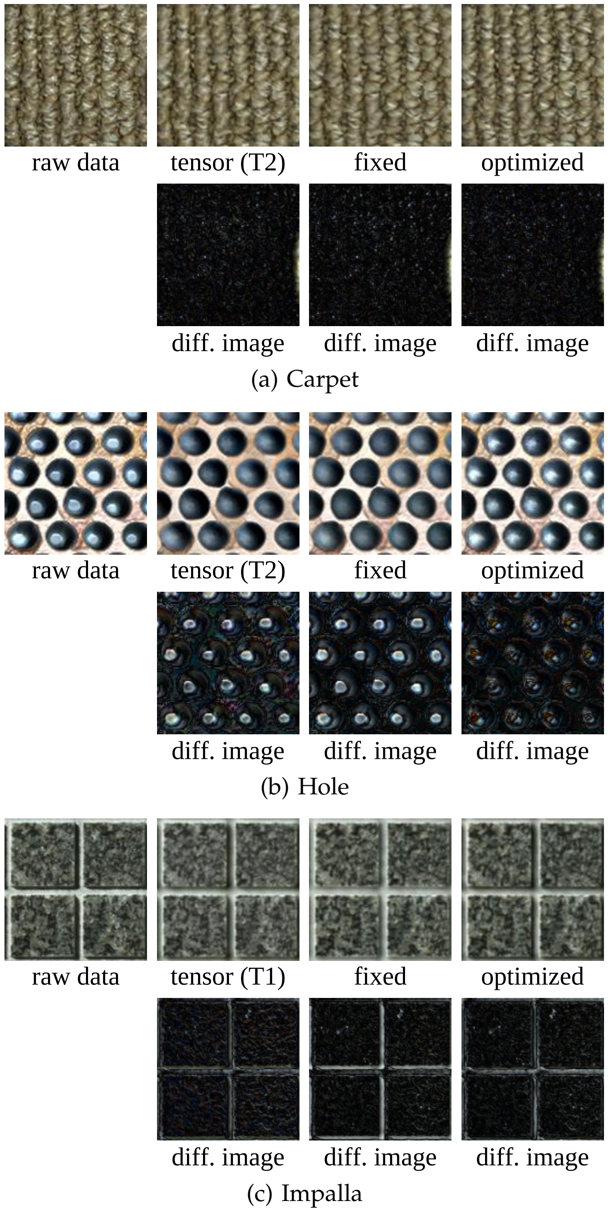
7.1 Experimental Results

The experiments of multivariate SRBF representation and optimized parameterization were conducted on a workstation with an Intel Core 2 Extreme QX9650 CPU, an NVIDIA GeForce GTX 280 graphics card, and 8 gigabytes of main memory. The measured BTFs were provided in courtesy of University of California at San Diego [19], University of Bonn [36], and Dr. Xin Tong. The multivariate Gaussian SRBFs (Eq. (4)) were adopted to represent BTFs. In general, we have found no significant difference between various types of SRBFs for approximating BTFs, but Gaussian SRBFs are locally compact basis functions and handle most common cases very well. Additionally, the SRBF center and bandwidth sets were constrained to be the same for red, green, and blue channels of a BTF texel since separately fitting the data of each channel only slightly reduces approximation errors, but increases computational cost and storage space by a factor of 2 or more.

Table 2 compares the statistics of the proposed model and tensor approximation for modeling hierarchical BTF

[3] If texture arrays are not supported in the graphics application programming interface, we may tile the mipmap of each parameter texture into one or more 'big' textures. However, one should be careful not to include texels out of the tile boundaries in the texture filtering. This can be achieved by clamping texture coordinates into an appropriate range before sampling.

[4] One may pack the SRBF center and bandwidth sets into one two-dimensional texture array to slightly reduce texture access time.



(a) Carpet

(b) Hole

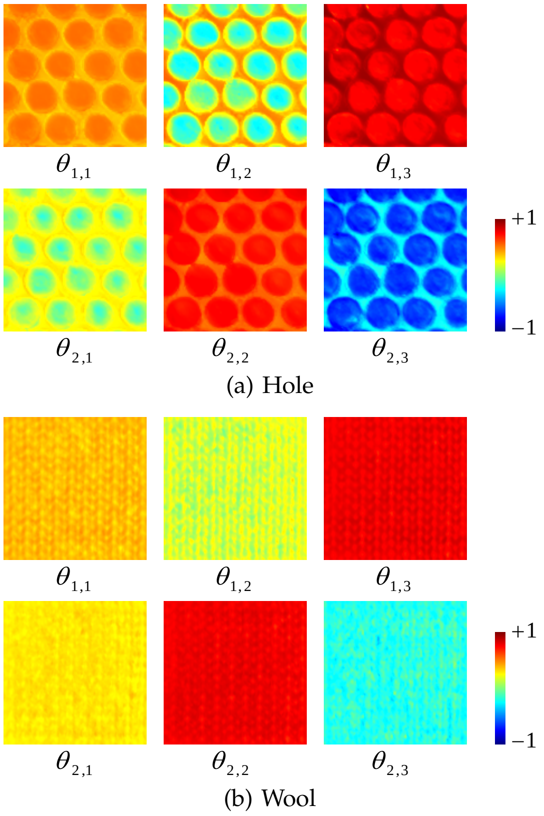
(c) Impalla

Fig. 2. Reconstructed BTF images of the proposed model and tensor approximation. From left to right: raw data; tensor approximation; fixed parameterization; optimized parameterization. From top to bottom in each sub-figure: reconstructed images; absolute difference images scaled by 2.

data sets, which includes the experimental results of *N-mode singular value decomposition* (N-SVD) [11], [42], [53], traditional fixed (Fix.), and optimized (Opt.) parameterization. In this table, all compressed data were stored as half-precision (16-bit) floating point numbers [54], and the quality of compressed data is measured by signal-to-mean squared error ratio ('SE ratio'). Moreover, T1 and T2 denote different parameter settings of tensor approximation used for comparison under similar compression ratio and rendering speed, respectively.

User-defined constants in the proposed model, such as the number of SRBFs (J), the number of variates (N_p), and the weights of smoothness energy terms, were manually tuned in the current implementation. We first conduct the experiment of a BTF at the coarsest level 0 with $J = 8$ and $N_p = 3$, and gradually increase J or N_p if the approximation error is large. Note that this only takes little time to determine J and N_p since there are only few (usually one) BTF texels at level 0. After that, the weights of smoothness energy terms are resolved and fine tuned in proportion to the ℓ^2 norm of the apparent BRDF data for a BTF texel. Typically, these weight values are in the interval from 10^{-5} to 10^{-2}. The final weight values of each BTF are also listed in Table 2.

In our experiments, at most three traditional param-



(a) Hole

(b) Wool

Fig. 3. Images of optimized parameterization coefficients.

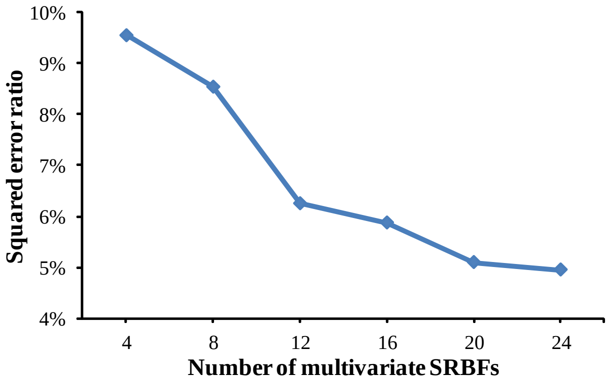


Fig. 4. Plot of the squared error ratio versus the number of SRBFs based on the proposed model for multi-resolution BTF approximation (*Hole*).

eterizations: half-way, illumination, and view directions were employed in the fixed parameterization. As for optimized parameterization, we instead utilized the parameterization function defined in Eq. (12). If the number of variates was more than 3, namely $N_p > 3$, the optimization stage at the coarsest level 0 was performed multiple times to account for the randomness of the initial guess. Then, only the SRBF parameters with the lowest approximation errors for BTF texels at level 0 were employed in the subsequent upsampling and optimization stages at higher levels.

Note that we skip the reduction of spatial resolution for the results of tensor approximation, since our multivariate SRBF representation can only handle directional variables. For fast BTF rendering based on tensor approximation, it is usually better not to reduce the spatial domain of a BTF. Therefore, we believe that this will not introduce an unfair comparison among these methods.

Figure 2 demonstrates the reconstructed BTF images of the proposed model and compare them with those of tensor approximation based on N-SVD. From this figure and Table 2, the proposed optimized parameterization generally outperforms the traditional fixed approach in terms of approximation errors and visual quality, especially when the BTF data sets exhibit complex mesostructures, specular reflectance, or sharp shadows. As shown in Figure 2(b), multivariate SRBFs tend to capture more sharp features in a BTF, such as specular highlights, which tensor approximation fails to preserve under similar rendering rate. In Section 7.2, we will further discuss the advantages and disadvantages of multivariate SRBF representation and tensor approximation in details.

Figure 3 demonstrates the images of optimized parameterization coefficients. In the experiments, we have found that parameterization coefficients tend to align with illumination and geometric features of a BTF, especially shadows, specular highlights, and uneven surfaces. For example, one can roughly perceive the distribution of surface normals of *Hole* and *Wool* from Figure 3. For diffuse-like BTFs such as *Wool*, surface geometry often dominates the spatial variation of parameterization coefficients, and the solution of parameterization coefficients is generally not far away from our initial guess (half-way, illumination and view directions) when the object surface is rather flat. As for BTFs with specular effects like *Hole*, illumination features are also very important to the spatial variation of parameterization coefficients. In highlight- and shadow-covered regions, parameterization coefficients generally changes more quickly. These interesting findings particularly open a connection between the results of our approach and surface normal estimation of a BTF, or even material/face recognition.

Figure 4 further plots the squared error ratio of the BTF *Hole* versus the number of SRBFs based on the proposed model. This figure particularly shows the scalability of the proposed multivariate SRBF representation. The approximation error of a BTF can be gradually reduced by increasing the number of SRBFs. Although we currently do not have a theoretical proof on how to decide the number of SRBFs, our framework allows one to incrementally increase the number of SRBFs by updating previous optimized results. From our experiments, we have found that the more lighting and geometric saliences exhibit in the BTF (for example, more rapidly spatially- or angularly-varying shadows and specular highlights, or rougher object surfaces), the larger number of SRBFs is needed to achieve a low approximation error, even for approximations without smoothness energy terms.

Table 3 and Figure 5 respectively present the runtime performance and rendered images of the proposed model and tensor approximation for various BTF data sets. In our experiments, the screen resolution and the number of directional light sources were respectively set to 640×480 and 2. In Table 3, we list the resolution of the synthesized coordinate texture of each BTF in the row ’Coord. texture’. Note that the statistics in the row ’Total data’ also include the amount of synthesized texture data. Figure 5 demonstrates that our approach achieves much faster rendering speed while maintaining comparable image quality compared to tensor approximation under similar compression ratio. For comparison under similar rendering speed, one can find that the optimized parameterization has smaller approximation error and compressed data size than the tensor approximation (T2 in Tables 2 and 3). In particular, when a BTF data set exhibits specular reflectance and sharp shadows, the optimized paramterization outperforms the fixed parameterization and tensor approximation (Figure 6). In general, all experimental results show that the proposed model can achieve a better tradeoff between rendering performance and image quality than tensor approximation, especially when rendering time is a critical issue at run-time.

Due to the access overhead of additional parameters, the rendering performance of optimized parameterization is slightly slower than that of fixed parameterization, but there are no significant differences between them. For a medium-size model, both methods can easily achieve real-time rendering rates at run-time. Moreover, the run-time performance of multivariate SRBF representation is typically faster than that of tensor approximation. This is owing to that tensor approximation needs extra computational overhead and auxiliary texture data for efficient interpolation at run-time. By contrast, the proposed multivariate SRBF representation is a continuous parametric model, hence no additional interpolation techniques for smooth transitions across different illumination (or view) directions are required.

In Figure 7, we also compare the effects of smoothness energy terms (Section 5.3) and hardware mipmap filtering acceleration. Figures 7(c) and 7(d) were respectively generated using the approximated results of the proposed model without/with smoothness energy terms. In general, including the smoothness energy terms in Eq. (15) will significantly reduce the computational cost

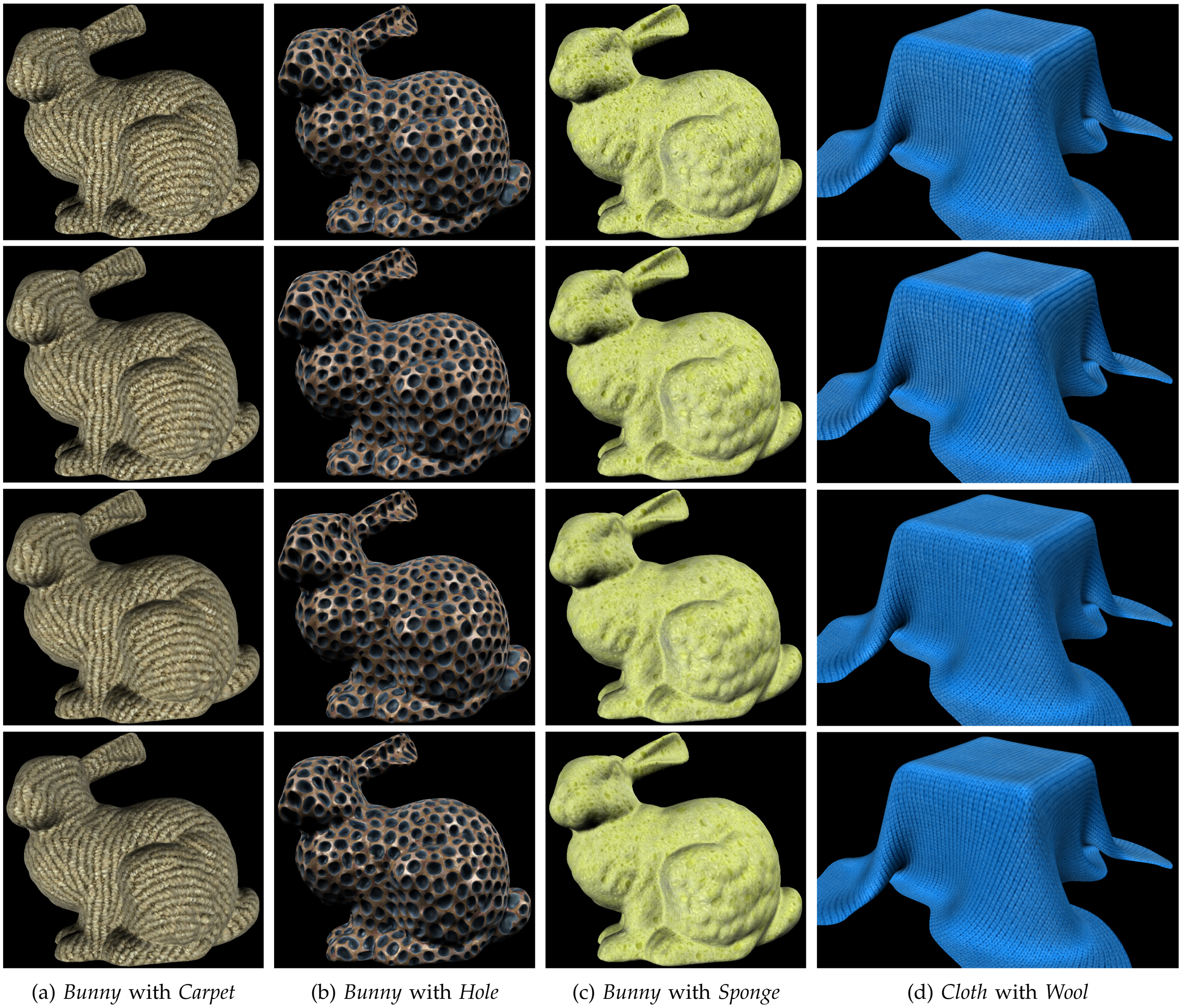

(a) *Bunny* with *Carpet* (b) *Bunny* with *Hole* (c) *Bunny* with *Sponge* (d) *Cloth* with *Wool*

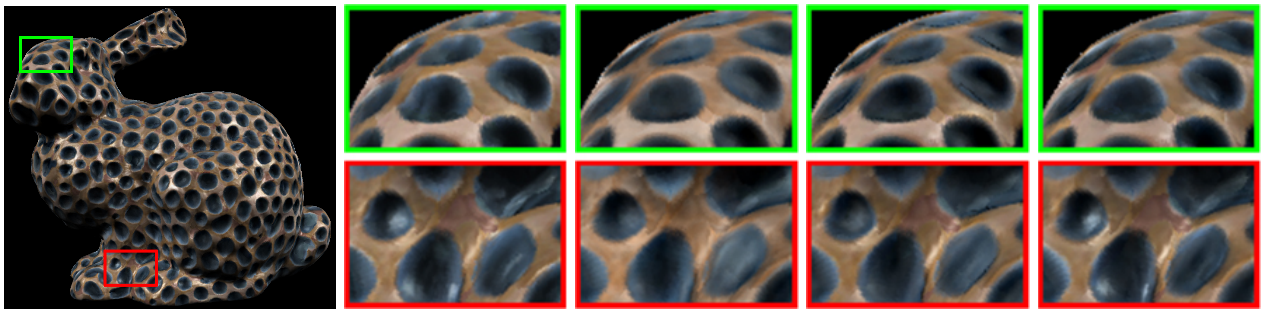
Fig. 5. Comparison of rendered images under similar compression ratio. From top to bottom: raw data; tensor approximation (T1); fixed parameterization; optimized parameterization. This figure illustrates that our approach can achieve similar rendering quality with much faster rendering speed under similar compression ratio. For the configurations of parameterizations and run-time rendering, please refer to Tables 2 and 3.

of the hierarchical fitting process while only slightly increasing the approximation error of a BTF.

Note that for Figures 7(c) and 7(d), we disabled the built-in trilinear mipmap filtering of GPUs, reconstructed the shading color of each participating texel, and then performed mipmap filtering in shaders. By contrast, Figure 7(e) was rendered by enabling hardware trilinear mipmap filtering to interpolate SRBF parameters first and then reconstructing the final shading color. This will greatly increase rendering performance without noticeable defects on image quality when the derived SRBF parameters of a BTF are smooth enough.

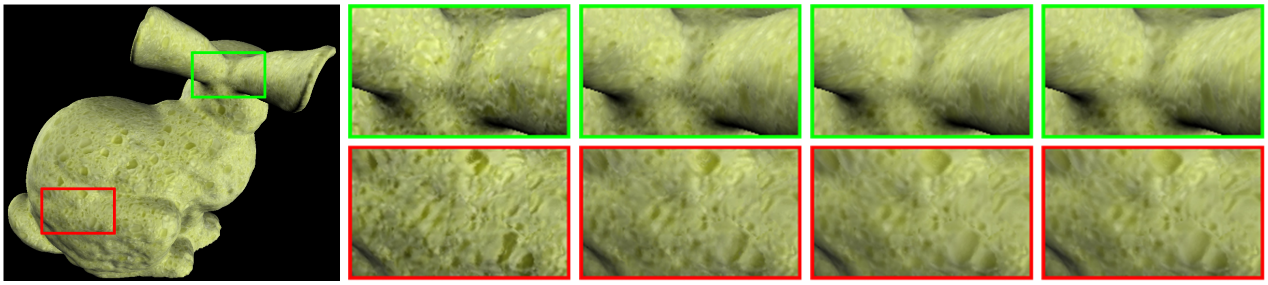
7.2 Discussion and Limitations

There are some previous articles that employed basis functions similar to SRBFs to approximate a reflectance function or BTF [12], [28], [29]. The most significant difference between our approach and previous approaches is that we propose *multivariate* SRBFs to achieve an accurate and efficient representation, while the basis functions adopted in previous work are *univariate*. This particularly allows us to multi-laterally combine various directional factors to describe a reflectance function, which was ignored in previous approaches. Moreover, Green et al. [28] proposed to parameterize a BRDF so that it can be efficiently represented using a mixture of isotropic Gaussians for light transport problems, but their fixed parameterizations may not be adequate to represent complex real-world reflectance functions. By contrast, our approach utilizes optimized parameterization that is data-dependent. Tan et al. [29] and Wang



(a) Optimized (b) Raw (c) Tensor (T2) (d) Fixed (e) Optimized

Fig. 6. Comparison of rendered images under similar rendering speed. (a) the whole rendered image of optimized parameterization. (b)-(e) are the enlarged images generated by different models. One can observe that the optimized parameterization result better preserves specular effects and sharp shadows. For the configurations of approximation methods and run-time rendering, please refer to Tables 2 and 3.



(a) (b) (c) 0.59%, 502.71 hr., 54.27 FPS (d) 0.81%, 5.21 hr., 54.43 FPS (e) 0.81%, 5.21 hr., 311.9 FPS

Fig. 7. Effects of smoothness energy and hardware texture filtering. (a) Whole rendered result using raw data; (b) raw data; (c) without smoothness; (d) with smoothness; (e) with smoothness and hardware texture filtering. The squared error ratio, compression time, and rendering performance of the approximated BTF are shown under each figure.

TABLE 4
Feature comparisons between multivariate SRBFs and tensor approximation for BTF modeling.

Method	Multivariate SRBFs	Tensor approximation
Assumption	Parametric	Non-parametric
Compression error	Moderate	Low
Compression ratio	Moderate*	High
Compression time	Time-consuming	Moderate
Formulation	Continuous	Discrete
Visual quality	High	High
Auxiliary data for rendering	No	Yes
Rendering performance	Fast	Moderate

$*$. Recall that multivariate SRBFs can only handle directional variables, but tensor approximation can reduce the dimensionality of all kinds of variables at the same time. We currently do not apply any other approximation methods to compress the spatial domain of SRBF parameters for a BTF.

et al. [12] also applied a mixture of isotropic/spherical Gaussians to represent the distribution of normals in the traditional micro-facet model. Specifically, their methods [12], [29] rely on physical assumptions, such as Fresnel and shadowing terms, to derive the simple normal distribution function for reflectance data fitting, which additionally needs to estimate surface normals for the BTF of a rough surface. By contrast, we do not assume any physical properties of the available BTF data, and implicitly incorporate the traditional physical terms into our model. In this way, a single multivariate SRBF can be regarded as data-dependently including/approximating many traditional physical properties in terms of different variates.

Table 4 compares the features of multivariate SRBFs and tensor approximation for BTFs. Comparisons of these two compression methods are similar to the traditional debates between parametric and non-parametric models in the statistics and machine learning communities. In general, multivariate SRBFs (parametric models) lead to more efficient rendering performance at run-time, while tensor approximation (non-parametric models) provides a more accurate representation for visual data sets.

An additional advantage of multivariate SRBFs is that approximating a hierarchical set of multi-resolution BTFs can be achieved using the proposed hierarchical fitting algorithm, while sophisticated run-time mipmap texture filtering can also be readily performed on GPUs by including smoothness energy terms in the objective function. Nevertheless, approximating a BTF with multivariate SRBFs is computationally expensive (typically several hours) due to the non-linear optimization process, even if our implementation already achieves considerable acceleration with GPUs. On the contrary, tensor approximation usually takes only tens of minutes to decompose a given BTF data set since only linear algebra operations are needed. Thus, there is always a trade-off between off-line and run-time costs for these two categories of compression methods.

For the proposed optimized parameterization, it is possible to adopt other parameterization functions other than Eq. (12). Any other parameterization functions can be easily integrated into the proposed model, as long as they will not lead to difficult gradient computation in the optimization process. For instance, one may formulate the parameterization functions as symmetric 3×3 matrix transformations [23] or linear transformations in the spherical coordinate system [17]. It may also be effective to first apply linear or non-linear projections to obtain some special parameterizations [15], [16], [18], [21], [22], and then linearly combine them to construct final parameterization functions. These can be considered as generalizations of previous traditional parameterizations. In the current implementation, we choose Eq. (12) simply because it includes popular fixed parameterizations (half-way, illumination, and view directions) as its subset and results in efficient parameter estimation.

There are also some disadvantages and limitations of the proposed model:

- As with many alternating optimization algorithms, the stability of estimated SRBF parameters is influenced by the initial guess and the alternating order.
- User-defined constants in the proposed model need to be manually tuned for different BTFs.
- The smoothness energy terms (Section 5.3) and linear interpolation on model parameters at runtime (Section 6.3) inevitably decrease approximation quality and result in some artifacts. It is expected that high-frequency features in BTFs will be slightly smoothed or even lost after approximation.

In current implementation, we propose a heuristic approach to determine a reasonable initial guess and ignore the effects of the alternating order. We have also found that bounding the values of SRBF parameters can increase the stability of estimated parameters. In our experience, it is recommended to bound coefficients in the interval $[-b_{max}, +b_{max}]$, where b_{max} is the maximum absolute value of input BTF data, bandwidths in the interval $[-32, +32]$, and parameterization coefficients in the interval $[-1, +1]$.

Moreover, when multivariate Gaussian SRBFs are adopted and rendering performance is not a major concern, one may instead align only SRBF centers and parameterization coefficients (Eq. (17) and Eq. (20)) to preserve more high-frequency features in BTFs. Nevertheless, the nonlinear filtering technique [48] should be applied to interpolate SRBF centers and bandwidths at run-time, while linear interpolation is employed for other model parameters. Note that this can improve image quality, but rather reduce run-time performance.

8 Conclusions and Future Work

In this paper, we have introduced a novel data representation for BTFs. Based on multivariate SRBFs, reflectance functions can be modeled in their intrinsic spherical domain to avoid artifacts that result from false boundaries, distortions, and unnecessary parameterization. Most existing methods were not originally developed to handle spherical functions and deficient in this important feature. Therefore, they have to model a spherical function in an inappropriate domain rather than the unit hypersphere. Moreover, while a sum-of-products model with multivariate SRBFs provides an intrinsic and efficient description of multivariate spherical functions, optimized parameterization overcomes the major disadvantage of traditional fixed parameterizations by learning data-dependent transformation functions. Experimental results reveal that multivariate SRBFs and optimized parameterization can be seamlessly integrated to obtain a practical solution of photorealistic BTF rendering at real-time rates. Finally, our approach for computing the optimal parameterization and SRBF coefficients of BTF data can be potentially applied to several computer vision problems, such as surface normal estimation, material classification, and object recognition when the appearances of materials or objects under various illumination or view directions are available.

In the current model, the proposed multivariate SRBF representation only employs one type of SRBFs. An obvious question is whether we can utilize more than one type of SRBFs to approximate observations on the unit hyper-sphere or not. The answer is certainly 'yes'. We may achieve this simply by weightedly summing different types of multivariate SRBFs. For a multivariate SRBF, it can even be constructed from the product of different types of univariate SRBFs. Nevertheless, the real question is whether this sophisticated approach will outperform the current SRBF representation or not? This may need more experiments to reach a final conclusion.

The proposed framework of optimized parameterization relies on a pre-defined parametric model. Hence, its optimality is built upon a specific functional form. Finding the *truly* optimal parameterizations for a given visual data set is still a challenging problem. We plan to investigate this issue using non-parametric models in the future. Moreover, the proposed optimized parameterization framework only supports directional variables, but real-world visual data sets may also depend on other types of physical factors. These non-directional factors should not be excluded from parameterization. For example, we may additionally take the spatial variables of a BTF into account for parameterization. In this way, spatially-varying characteristics of the BTF can be implicitly modeled in a unified framework.

Acknowledgments

The authors would like to thank the anonymous reviewers for profound comments and suggestions, Dr. Xin

Tong for providing BTF data, and Jia-Yin Ji for preparing the model *Cloth*. This work was supported in part by the National Science Council of Taiwan under Grant No. NSC96-2221-E-009-152-MY3 and NSC99-2628-E-009-178.

References

[1] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The Lumigraph," in *Proc. of SIGGRAPH '96*, 1996, pp. 43–54.

[2] M. Levoy and P. Hanrahan, "Light Field Rendering," in *Proc. of SIGGRAPH '96*, 1996, pp. 31–42.

[3] S. M. Seitz and C. R. Dyer, "View Morphing," in *Proc. of SIGGRAPH '96*, 1996, pp. 21–30.

[4] K. J. Dana, B. Van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and Texture of Real-World Surfaces," *ACM Trans. Graph.*, vol. 18, no. 1, pp. 1–34, 1999.

[5] J. Filip and M. Haindl, "Bidirectional Texture Function Modeling: A State of the Art Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 1921–1940, 2009.

[6] O. G. Cula and K. J. Dana, "Compact Representation of Bidirectional Texture Functions," in *Proc. of CVPR '01*, 2001, pp. 1041–1047.

[7] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz, "Shape and Spatially-Varying BRDFs from Photometric Stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1060–1071, 2010.

[8] M. Haindl and J. Filip, "Extreme Compression and Modeling of Bidirectional Texture Function," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1859–1865, 2007.

[9] 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 Trans. Graph.*, vol. 25, no. 3, pp. 735–745, 2006.

[10] G. Müller, R. Sarlette, and R. Klein, "Data-driven Local Coordinate Systems for Image-Based Rendering," *Comput. Graph. Forum*, vol. 25, no. 3, pp. 369–378, 2006.

[11] M. A. O. Vasilescu and D. Terzopoulos, "TensorTextures: Multilinear Image-Based Rendering," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 336–342, 2004.

[12] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo, "All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance," *ACM Trans. Graph.*, vol. 28, no. 5, 2009.

[13] Y.-T. Tsai and Z.-C. Shih, "All-Frequency Precomputed Radiance Transfer Using Spherical Radial Basis Functions and Clustered Tensor Approximation," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 967–976, 2006.

[14] J. Kautz and M. D. McCool, "Interactive Rendering with Arbitrary BRDFs Using Separable Approximations," in *Proc. of EGWR '99*, 1999, pp. 247–260.

[15] S. Rusinkiewicz, "A New Change of Variables for Efficient BRDF Representation," in *Proc. of EGWR '98*, 1998, pp. 11–22.

[16] T. Zickler, R. Ramamoorthi, S. Enrique, and P. N. Belhumeur, "Reflectance Sharing: Predicting Appearance from a Sparse Set of Images of a Known Shape," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1287–1302, 2006.

[17] F. H. Cole, "Automatic BRDF Factorization," Master's thesis, Harvard University, 2002.

[18] P. Peers, K. Vom Berge, W. Matusik, R. Ramamoorthi, J. Lawrence, S. Rusinkiewicz, and P. Dutré, "A Compact Factored Representation of Heterogeneous Subsurface Scattering," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 746–753, 2006.

[19] M. L. Koudelka, S. Magda, P. N. Belhumeur, and D. J. Kriegman, "Acquisition, Compression, and Synthesis of Bidirectional Texture Functions," in *Proc. of Texture '03*, 2003, pp. 59–64.

[20] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein, "Acquisition, Synthesis, and Rendering of Bidirectional Texture Functions," *Comput. Graph. Forum*, vol. 24, no. 1, pp. 83–109, 2005.

[21] R. Ramamoorthi and P. Hanrahan, "Frequency Space Environment Map Rendering," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 517–526, 2002.

[22] M. M. Stark, J. Arvo, and B. E. Smits, "Barycentric Parameterizations for Isotropic BRDFs," *IEEE Trans. Vis. Comput. Graph.*, vol. 11, no. 2, pp. 126–138, 2005.

[23] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg, "Non-Linear Approximation of Reflectance Functions," in *Proc. of SIGGRAPH '97*, 1997, pp. 117–126.

[24] D. K. McAllister, A. Lastra, and W. Heidrich, "Efficient Rendering of Spatial Bi-Directional Reflectance Distribution Functions," in *Proc. of Graphics Hardware '02*, 2002, pp. 79–88.

[25] G. J. Ward, "Measuring and Modeling Anisotropic Reflection," in *Proc. of SIGGRAPH '92*, 1992, pp. 265–272.

[26] T. Malzbender, D. Gelb, and H. J. Wolters, "Polynomial Texture Maps," in *Proc. of SIGGRAPH '01*, 2001, pp. 519–528.

[27] J. Meseth, G. Müller, and R. Klein, "Reflectance Field Based Real-Time, High-Quality Rendering of Bidirectional Texture Functions," *Computers & Graphics*, vol. 28, no. 1, pp. 105–112, 2004.

[28] P. Green, J. Kautz, and F. Durand, "Efficient Reflectance and Visibility Approximations for Environment Map Rendering," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 495–502, 2007.

[29] P. Tan, S. Lin, L. Quan, B. Guo, and H.-Y. Shum, "Multiresolution Reflectance Filtering," in *Proc. of EGSR '05*, 2005, pp. 111–116.

[30] P.-P. J. Sloan, X. Liu, H.-Y. Shum, and J. Snyder, "Bi-Scale Radiance Transfer," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 370–375, 2003.

[31] P.-P. J. Sloan, J. Kautz, and J. Snyder, "Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 527–536, 2002.

[32] W.-C. Ma, S.-H. Chao, Y.-T. Tseng, Y.-Y. Chuang, C.-F. Chang, B.-Y. Chen, and M. Ouhyoung, "Level-of-Detail Representation of Bidirectional Texture Functions for Real-Time Rendering," in *Proc. of I3D '05*, 2005, pp. 187–194.

[33] W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk, "Light Field Mapping: Efficient Representation and Hardware Rendering of Surface Light Fields," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 447–456, 2002.

[34] G. Müller, J. Meseth, and R. Klein, "Fast Environmental Lighting for Local-PCA Encoded BTFs," in *Proc. of CGI '04*, 2004, pp. 198–205.

[35] K. Nishino, Y. Sato, and K. Ikeuchi, "Eigen-Texture Method: Appearance Compression and Synthesis Based on a 3D Model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1257–1265, 2001.

[36] M. Sattler, R. Sarlette, and R. Klein, "Efficient and Realistic Visualization of Cloth," in *Proc. of EGSR '03*, 2003, pp. 167–178.

[37] P.-P. J. Sloan, J. Hall, J. C. Hart, and J. Snyder, "Clustered Principal Components for Precomputed Radiance Transfer," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 382–391, 2003.

[38] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle, "Surface Light Fields for 3D Photography," in *Proc. of SIGGRAPH '00*, 2000, pp. 287–296.

[39] L. Latta and A. Kolb, "Homomorphic Factorization of BRDF-Based Lighting Computation," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 509–516, 2002.

[40] M. D. McCool, J. Ang, and A. Ahmad, "Homomorphic Factorization of BRDFs for High-Performance Rendering," in *Proc. of SIGGRAPH '01*, 2001, pp. 171–178.

[41] F. Suykens, K. vom Berge, A. Lagae, and P. Dutré, "Interactive Rendering with Bidirectional Texture Functions," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 463–472, 2003.

[42] H. Wang, Q. Wu, L. Shi, Y. Yu, and N. Ahuja, "Out-of-Core Tensor Approximation of Multi-Dimensional Matrices of Visual Data," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 527–535, 2005.

[43] T. K.-H. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials Using Three-Dimensional Textons," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 29–44, 2001.

[44] J. Filip, M. J. Chantler, P. R. Green, and M. Haindl, "A Psychophysically Validated Metric for Bidirectional Texture Data Reduction," *ACM Trans. Graph.*, vol. 27, no. 5, 2008.

[45] M. Haindl and J. Filip, "Fast BTF Texture Modelling," in *Proc. of Texture '03*, 2003, pp. 47–52.

[46] M. Haindl, J. Grim, P. Pudil, and M. Kudo, "A Hybrid BTF Model Based on Gaussian Mixtures," in *Proc. of Texture '05*, 2005, pp. 95–100.

[47] J. Lawrence, S. Rusinkiewicz, and R. Ramamoorthi, "Efficient BRDF Importance Sampling Using a Factored Representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 496–505, 2004.

[48] C. Han, B. Sun, R. Ramamoorthi, and E. Grinspun, "Frequency Domain Normal Map Filtering," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[49] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A Limited Memory Algorithm for Bound Constrained Optimization," *SIAM J. Sci. Comput.*, vol. 16, no. 5, pp. 1190–1208, 1995.

[50] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization," *ACM Trans. Math. Softw.*, vol. 23, no. 4, pp. 550–560, 1997.

[51] NVIDIA, "NVIDIA CUDA: Compute Unified Device Architecture," 2011, http://www.nvidia.com/cuda/.

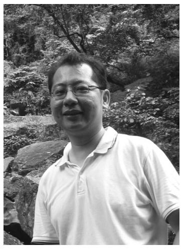
[52] S. Lefebvre and H. Hoppe, "Appearance-Space Texture Synthesis," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 541–548, 2006.

[53] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the Best Rank-1 and Rank-$(R_1, R_2, \ldots, R_n)$ Approximation of Higher-Order Tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1324–1342, 2000.

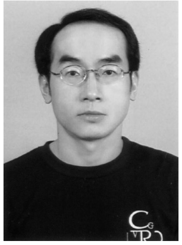
[54] ILM, *Technical Introduction to OpenEXR*, Industrial Light & Magic, 2006, http://www.openexr.com/documentation.html.

Yu-Ting Tsai received the B.S. and M.S. degrees in electronics engineering from National Chiao Tung University, Taiwan, in 2000 and 2002, respectively, and the Ph.D. degree in computer science from National Chiao Tung University, Taiwan, in 2009. Currently, he is an assistant professor in the Department of Computer Science and Engineering at Yuan Ze University. His research interests include computer graphics, computer vision, machine learning, and signal processing.

Kuei-Li Fang received the B.S. degree from National Dong Hua University in 2005 and the M.S. degree from National Chiao Tung University in 2008, respectively. He is interested in real-time rendering and game development. After graduated from National Chiao Tung University, he decided to devote himself to the game industry.



Wen-Chieh Lin received the BS and MS degrees in control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1994 and 1996, respectively, and the PhD degree in robotics from Carnegie Mellon University, Pittsburgh, in 2005. Since 2006, he has been with the Department of Computer Science and the Institute of Multimedia Engineering, National Chiao-Tung University, as an assistant professor. His current research interests include computer graphics, computer animation, and computer vision. He is a member of the IEEE and the ACM.



Zen-Chung Shih received his B.S. degree in Computer Science from Chung-Yuan Christian University in 1980, M.S. degree in 1982 and Ph.D. degree in 1985 in Computer Science from the National Tsing Hua University, Taiwan. Currently, he is a professor in the Department of Computer Science and Institute of Multimedia Engineering at the National Chiao Tung University in Hsinchu. He is also members of IEEE and ACM. His current research interests include procedural texture synthesis, non-photorealistic rendering, real-time rendering, and stylized rendering.