Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Design and Engineering 2 (2015) 1–15

www.elsevier.com/locate/jcde

Adaptive B-spline volume representation of measured BRDF data for photorealistic rendering

Hyungjun Park[a,*], Joo-Haeng Lee[b]

[a] *Department of Industrial Engineering, Chosun University, Gwangju 501-759, South Korea*
[b] *Electronics and Telecommunications Research Institute, Daejeon 305-350, South Korea*

Received 21 April 2014; received in revised form 20 July 2014; accepted 21 July 2014
Available online 6 December 2014

Abstract

Measured bidirectional reflectance distribution function (BRDF) data have been used to represent complex interaction between lights and surface materials for photorealistic rendering. However, their massive size makes it hard to adopt them in practical rendering applications. In this paper, we propose an adaptive method for B-spline volume representation of measured BRDF data. It basically performs approximate B-spline volume lofting, which decomposes the problem into three sub-problems of multiple B-spline curve fitting along *u*-, *v*-, and *w*-parametric directions. Especially, it makes the efficient use of knots in the multiple B-spline curve fitting and thereby accomplishes adaptive knot placement along each parametric direction of a resulting B-spline volume. The proposed method is quite useful to realize efficient data reduction while smoothing out the noises and keeping the overall features of BRDF data well. By applying the B-spline volume models of real materials for rendering, we show that the B-spline volume models are effective in preserving the features of material appearance and are suitable for representing BRDF data.

Keywords: B-spline volume; Measured BRDF; Approximate volume lofting; Adaptive knot placement; Data reduction

1. Introduction

Bidirectional reflectance distribution function (BRDF) is a fundamental radiometric concept that models directional properties of reflection between light and surface materials [1,2]. In computer graphics, BRDF has been widely adopted for realistic representation of material appearance. Although there have been extensive works on modeling surface reflectance, it is still an active research topic to develop BRDF models. Since the first introduction of Phong model [3], various BRDF models have been proposed either to overcome the shortcomings of the others or to achieve the different goals such as accuracy, computational efficiency, controllability, versatility, and compactness in data size. The good description of BRDF models can be found in [1–5].

Physically-based rendering aims to simulate the interaction of light and materials by more precisely considering various physical properties of optics to enhance the realism of synthetic images [1,2]. Recently, there have been strong requirements to simulate sophisticated material properties with great accuracy that cannot be easily achieved with existing modeling techniques of material appearance. Advances in measuring technology have made it possible to accurately acquire reflection properties of real materials using a gonioreflectometer [6–8]. Obviously, using measured BRDF data has been one of the alternatives to simple phenomenological models. For example, measured BRDF data are actively used for virtual prototyping in fashion, consumer electronics, and automotive industries to visualize the sophisticated real materials [9,10], and for VFX in the film industry to express delicate textile materials of digital doubles [11].

To be applied in such applications, the measured BRDF data should be stored into a simple tabular form or fit into a more sophisticated representation. As the measured BRDF data are

[*] Corresponding author. Tel.: +82 62 230 7039; fax: +82 62 230 7128.
E-mail address: hzpark@chosun.ac.kr (H. Park).
Peer review under responsibility of Society of CAD/CAM Engineers.

http://dx.doi.org/10.1016/j.jcde.2014.11.001
2288-4300/

usually huge and even noisy, it is much preferable to compute suitable mathematical models which approximate the data and smooth out the noise in the data. When the BRDF data is used in the tabular form without compression, their size causes a critical problem in production environment. For example, in case of the complete BRDF data set of a single material, its storage size varies from 33 to 750 megabytes, which can be a serious burden even to a commercial rendering system handling multiple sets of such BRDF data along with several gigabytes of geometry and texture data, not only in a memory but also over the network. Thus, it is very important to develop a compact representation scheme that preserves the overall features of the data faithfully.

A general approach is to approximate the measured BRDF data by one of the analytical models with a small set of control parameters [12]. However, as most of the analytical models are defined by nonlinear functions, it is required to employ nonlinear optimization algorithms for estimating the underlying parameters. Obviously this nonlinear optimization approach has several shortcomings. According to Ngan et al. [12], most of the analytical models obtained by non-linear least squares fitting could not express all the reflection features such as generalized diffuse, backscattering, off-specular, and Fresnel effects. For better results, it is required to increase the number of lobes or parameter complexity, which usually increase not only the difficulty in solving the non-linear optimization problem but also the computational burden of evaluating BRDF values from the fitted models.

In an effort to reduce the size of measured BRDF data while keeping an accurate representation of the data, many alternative approaches have been proposed. Shröder and Sweldens [13] used spherical wavelets to represent reflectance functions. Several research groups used polynomial based analytical models. Koenderink et al. [14] used Zernike polynomials for representing BRDF data, and Ozturk et al. [15] represented diffuse and glossy BRDF data by polynomial functions combined with the principal component analysis. In most cases of using polynomial based analytical models, a higher degree of polynomials is required to accurately fit some complex data. Several research groups including Lawrence et al. [16] have investigated techniques for factoring large BRDF datasets into more compact, manageable form. In all cases, the four dimensional (4D) anisotropic BRDF is factored into products of two dimensional (2D) functions that can be represented as texture maps. In most cases, the factorizations are not so accurate as the original measured BRDF data. Recently, Kurt and Cinsdikici [17] presented a new BRDF model which uses neural networks SOMs and MANs to represent measured BRDF data.

B-splines have been one of de facto industrial standards to represent curves and surfaces in the field of Computer Aided Geometric Design (CAGD) for more than a decade [18,19]. It is well-known that higher-degree polynomial models are inefficient to process and are numerically unstable to be fitted [18,19]. As B-splines are defined by piecewise polynomials, they can overcome the shortcomings of polynomial based models. Although many useful methods have been presented for B-spline curve or surface fitting to measured data, comparatively little attention has been given to how to apply B-spline volume fitting for rendering. Lee and Park [20,21] proposed a method for B-spline volume representation of isotropic measured BRDF data. Their method works well when the values of the given data changes smoothly (*i.e.*, for BRDF data of materials which are diffuse or moderately glossy), but it has difficulty when the data values change rapidly or have sharp peaks considered as not outliers but features (*i.e.*, for BRDF data of glossy or specular materials). In subsequent work in the form of short note [22], they coined the idea of adaptive placement of knots of resulting B-spline volumes as a way of alleviating the shortcoming.

In this paper, we solidify the idea coined in [22] and propose an adaptive method for B-spline volume representation of measured BRDF data. The method basically performs approximate B-spline volume lofting, which decomposes the problem into three sub-problems of multiple B-spline curve fitting along u-, v-, and w- parametric directions. Especially, it makes the efficient use of knots in the multiple B-spline curve fitting and accomplishes adaptive knot placement using dominant columns along each parametric direction of a B-spline volume. The resulting B-spline volume called B-spline volume BRDF (hereafter BVB) becomes a compact but accurate representation of the measured BRDF data. We show that the proposed method is quite useful to realize efficient data reduction while smoothing out the noises and keeping the overall features of BRDF data. By applying the BVB models of real materials for rendering, we also show that the BVB models are effective in preserving the features of material appearance and are suitable for representing the measured BRDF data coming from a wide range of materials (*i.e.*, not specific to diffuse and moderately glossy materials).

The rest of the paper is organized as follows. In Section 2, we explain some preliminaries related to BRDF. In Section 3, we present the adaptive method for BVB model fitting. In Section 4, we describe the application of BVB models to photorealistic rendering. In Section 5, we explain experimental results to show that BVB models are appropriate for BRDF representation and are useful to enhance rendering process. Finally, we conclude the paper with some remarks in Section 6.

2. Preliminaries Related to BRDF

2.1. Measured BRDF Data

For two unit vectors ω_i and ω_o denoting input (or incident) and output (or reflected) directions of light rays at a surface point p, BRDF is the differential ratio of reflected radiance along ω_o to the irradiance incident on the point p along ω_i [1,2]. At a point p in the spherical coordinate system, BRDF is defined as $f(\omega_o,\omega_i) = f(\theta_o, \phi_o, \theta_i, \phi_i)$ where $\omega_i = (\theta_i, \phi_i)$ and $\omega_o = (\theta_o, \phi_o)$. If BRDF can be defined as $f(\theta_o, \theta_i, \ \phi_d)$ where $\phi_d = |\phi_o - \phi_i|$, such BRDF is called isotropic. Otherwise, it is called anisotropic. As shown in Fig. 1, the domain of isotropic

BRDF is 3D rather than 4D. Examples of anisotropic materials include brushed metals and velvets. Most of ordinary materials are isotropic. In this paper, we deal with only isotropic BRDF.

Image-based measurement equipment is actively used due to its relatively simple configuration [6–8]. While a special light source rotates along the spherical sample of a target material, a digital camera captures the images of the sample at each light position. After post-processing, we can obtain an isotropic BRDF data set which is stored in a simple tabular form (hereafter TBF)

$$f_{i,j,k} = f(\theta_i = a_i, \theta_o = b_j, \phi_d = c_k) \tag{1}$$

where $(i,j,k) \in [0,..,i_{max}] \times [0,..,j_{max}] \times [0,..,k_{max}]$. The angle values a_i, b_j, c_k are defined as follows: $a_i = \frac{\pi}{2} i / i_{max}$, $b_j = \frac{\pi}{2} j / j_{max}$, $c_k = 2\pi k / k_{max}$. When RGB triple is used, each BRDF data element should be a vector of three color coordinates as follows: $\mathbf{f}_{i,j,k} = (f^r_{i,j,k}, f^g_{i,j,k}, f^b_{i,j,k})$. Thus, the measured BRDF data are given as follows:

$$\mathbf{F} = \{\mathbf{f}_{i,j,k} | i \in [0,..,i_{max}], j \in [0,..,j_{max}], k \in [0,..,k_{max}]\} \tag{2}$$

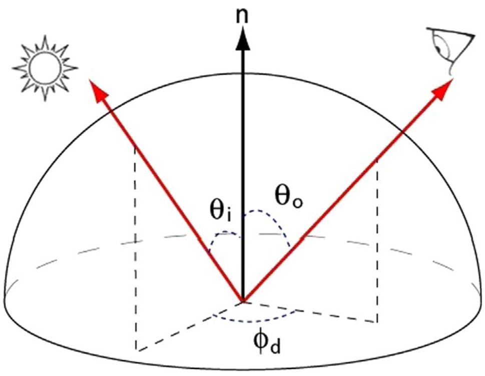

Fig. 1. Illustration of isotropic BRDF.

Fig. 2 shows the iso-meshes of measured BRDF data sets obtained from various materials. In the figure, the iso-meshes are displayed at specific incident angles in the spherical coordinate system.

When each color coordinate is encoded as four bytes and the measurement step is 1 degree for each dimension (*i.e.*, $i_{max} = j_{max} = 90$, $k_{max} = 360$), the size of measured BRDF data of each material reaches 70 megabytes. This is a very large size for both off-line and real-time renderers. Hence, we are interested in developing a new representation scheme that requires small storage while preserving the reflection features of the original material.

2.2. *Fitting analytical models to measured BRDF data*

Ngan et al. [12] evaluated several analytical models in terms of their ability of approximating measured BRDF data. Due to practical problems of non-linear fitting (such as dependency on initial guesses and convergence to local optima) and limited power of expression, it is difficult to find proper parameter values of each analytical model that preserves all the overall features of the BRDF data. Thus the fitted models often do not fit the data satisfactorily.

As the Lafortune model (hereafter LFT) [4] and the Cook–Torrance model (hereafter CKT) [5] have been widely used among analytical models, we use them as reference models to assess the usefulness of BVB models for data fitting and rendering. According to Ngan et al. [12], the CKT model

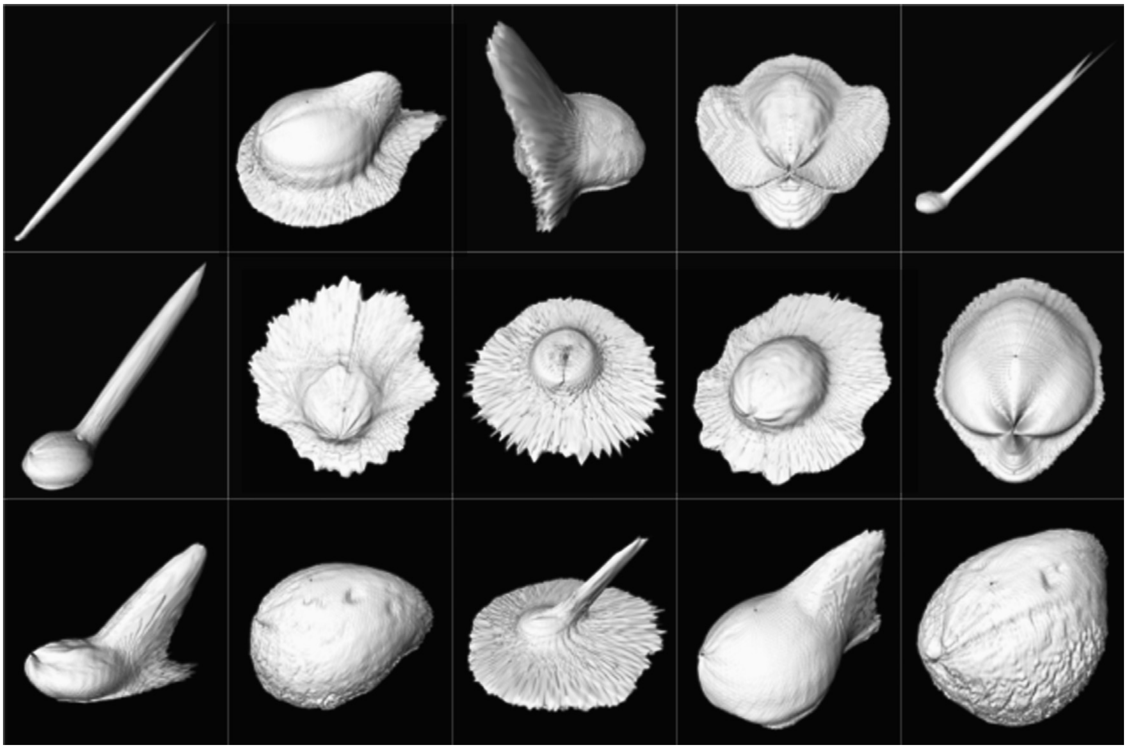

Fig. 2. Iso-meshes of measured BRDF data obtained from various materials.

outperformed the other analytical models, and the LFT model showed the average performance for fitting and rendering.

The CKT model, which originated from the field of physics, has been adopted for computer graphics, specifically to support more physically based properties of light reflection on the surface geometry [5]. The LFT model is a generalization of the Phong model, which is expressed as a sum of multiple specular lobes [4]. It is useful in representing relatively simple reflection features with flexibility of control, and it is efficient in evaluation. However, it is not suitable to represent materials with complex reflection features. Moreover, it fits the data poorly in the region of grazing angles where the light vector is nearly parallel to the tangent plane at each surface point.

3. Adaptive method for BVB model fitting

3.1. B-spline volume

A parametric B-spline curve of order p is defined as follows:

$$\mathbf{C}(t)=\sum_{i=0}^{n} N_{i,p}(t)\mathbf{b}_i \tag{3}$$

where $\mathbf{b}_i$ are control points, $N_{i,p}(t)$ are the normalized B-spline functions defined over the knot vector $\mathbf{T}=\{t_0,t_1,...,t_{n+p-1},t_{n+p}\}$, and the parameter domain is given as $t\in[t_{p-1},t_{n+1}]$. By directly extending B-spline curves to a trivariate case, we define a parametric tensor product B-spline volume $\mathbf{B}(u,v,w)$ of order (degree+1) p, q, r as follows:

$$\mathbf{B}(u,v,w)=\sum_{i=0}^{n_u}\sum_{j=0}^{n_v}\sum_{k=0}^{n_w} N_{i,p}(u)N_{j,q}(v)N_{k,r}(w)\mathbf{b}_{i,j,k} \tag{4}$$

where $\mathbf{b}_{i,j,k}$ are control points, and the parameter domains are given as $u\in[u_{p-1},u_{n_u+1}]$, $v\in[v_{q-1},v_{n_v+1}]$, and $w\in[w_{r-1},w_{n_w+1}]$. $N_{i,p}(u)$, $N_{j,q}(v)$, and $N_{k,r}(w)$ are the normalized B-spline functions of order p, q, r defined on knot vectors $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ in the u, v, w directions, respectively. The knot vectors $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ are given as $\mathbf{U}=\{u_0,u_1,..,u_{n_u+p}\}$, $\mathbf{V}=\{v_0,v_1,..,v_{n_v+q}\}$, and $\mathbf{W}=\{w_0,w_1,..,w_{n_w+r}\}$. In this work, we pursue a proper way of fitting a B-spline volume to the measured BRDF data $\mathbf{F}$ of a real material given in Eq. (2). The resulting B-spline volume becomes a BVB model.

3.2. Approximate B-spline volume lofting

We want each element $\mathbf{f}_{i,j,k}$ of the BRDF data $\mathbf{F}$ to be a B-spline volume element at specific parameter values $(u,v,w)=(\overline{u}_i,\overline{v}_j,\overline{w}_k)$, that is, $\mathbf{B}(\overline{u}_i,\overline{v}_j,\overline{w}_k)=\mathbf{f}_{i,j,k}$. Hence, the problem of constructing a B-spline volume $\mathbf{B}(u,v,w)$ from the data $\mathbf{F}$ can be solved by the least-squares B-spline volume fitting. When the parameter values $\overline{u}_i$, $\overline{v}_j$, $\overline{w}_k$ of the volumetric data $\mathbf{f}_{i,j,k}$, the orders p, q, r of B-spline functions, and the knot vectors $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ of a B-spline volume $\mathbf{B}(u,v,w)$ are given, the control points $\mathbf{b}_{i,j,k}$ can be determined by minimizing the following least-squares error

$$E(\mathbf{b}_{0,0,0},...,\mathbf{b}_{n_u,n_v,n_w})=\sum_{i=0}^{i_{max}}\sum_{j=0}^{j_{max}}\sum_{k=0}^{k_{max}}\|\mathbf{B}(\overline{u}_i,\overline{v}_j,\overline{w}_k)-\mathbf{f}_{i,j,k}\|^2. \tag{5}$$

According to the uniform angle steps in Eq. (1), the parameter values $\overline{u}_i$, $\overline{v}_j$, and $\overline{w}_k$ are equally spaced as $\overline{u}_i=i/i_{\max}$, $\overline{v}_j=j/j_{\max}$, and $\overline{w}_k=k/k_{\max}$, respectively. It is natural to make the parameters values linearly related to the angle values. The knots of the knot vectors $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ can be placed using the parameter values [18,19]. Since the objective functional in Eq. (5) is quadratic, this minimization problem leads to that of solving a system of linear equations. Without loss of generality, $i_{max}>n_u$, $j_{max}>n_v$, and $k_{max}>n_w$, so the least-squares minimization provides reasonably good solutions. However, as the size of the system matrix becomes very large (that is, $M\times M$ where $M=(n_u+1)\times(n_v+1)\times(n_w+1)$), the direct approach to solving such a large linear system should be avoided due to inefficiency in computation time and memory usage.

The proposed method for B-spline volume representation of BRDF data basically takes the scheme of approximate B-spline volume lofting, which is extended from approximate surface lofting based on the property of tensor products of B-splines [23,24]. In the approximate volume lofting, we construct a B-spline volume by successively applying multiple B-spline curve fitting to a set of polylines along each parametric (w-, v-, u-) direction. Consider that the equations $\mathbf{B}(\overline{u}_i,\overline{v}_j,\overline{w}_k)=\mathbf{f}_{i,j,k}$ are written as follows:

$$\begin{aligned}\mathbf{f}_{i,j,k}&=\mathbf{B}(\overline{u}_i,\overline{v}_j,\overline{w}_k)\\&=\sum_{c=0}^{n_w}N_{c,r}(\overline{w}_k)\left(\sum_{a=0}^{n_u}\sum_{b=0}^{n_v}N_{a,p}(\overline{u}_i)N_{b,q}(\overline{v}_j)\mathbf{b}_{a,b,c}\right)\end{aligned} \tag{6}$$

By replacing $\sum_{a=0}^{n_u}\sum_{b=0}^{n_v}N_{a,p}(\overline{u}_i)N_{b,q}(\overline{v}_j)\mathbf{b}_{a,b,c}$ by *intermediate* control points $\mathbf{g}_{i,j,c}$, we can simplify Eq. (6) as follows:

$$\mathbf{f}_{i,j,k}=\sum_{c=0}^{n_w}N_{c,r}(\overline{w}_k)\mathbf{g}_{i,j,c}=\mathbf{C}_{i,j}(\overline{w}_k)\ \text{ for }\ (i,j)\in[0,..,i_{max}]\times[0,..,j_{max}]. \tag{7}$$

Eq. (7) means multiple B-spline curve fitting to $(i_{max}+1)\times(j_{max}+1)$ polylines each of which consists of $(k_{max}+1)$ points passing along the w direction. We can obtain the control points $\mathbf{g}_{i,j,c}$ for $(i,j,c)\in[0,..,i_{max}]\times[0,..,j_{max}]\times[0,..,n_w]$ by solving the multiple B-spline curve fitting where n_w, r, $\overline{w}_k$, and $\mathbf{W}$ are given. Now the intermediate control points $\mathbf{g}_{i,j,c}$ are expressed as follows:

$$\mathbf{g}_{i,j,c}=\sum_{b=0}^{n_v}N_{b,q}(\overline{v}_j)\left(\sum_{a=0}^{n_u}N_{a,p}(\overline{u}_i)\mathbf{b}_{a,b,c}\right) \tag{8}$$

By replacing $\sum_{a=0}^{n_u}N_{a,p}(\overline{u}_i)\mathbf{b}_{a,b,c}$ by intermediate control points $\mathbf{h}_{i,b,c}$, we can simplify Eq. (8) as follows:

$$\mathbf{g}_{i,j,c}=\sum_{b=0}^{n_v}N_{b,q}(\overline{v}_j)\mathbf{h}_{i,b,c}\ \text{ for }\ (i,c)\in[0,..,i_{\max}]\times[0,..,n_w]. \tag{9}$$

Eq. (9) means multiple B-spline curve fitting to $(i_{max}+1)\times(n_w+1)$ polylines each of which consists of $j_{max}+1$ points passing along the v direction. Similarly, we can obtain the control points $\mathbf{h}_{i,b,c}$ for $(i,b,c) \in [0,..,i_{max}] \times [0,..,n_v] \times [0,..,n_w]$ by applying multiple B-spline curve fitting where n_v, q, $\bar{v}_j$, and $\mathbf{V}$ are given. The intermediate control points $\mathbf{h}_{i,b,c}$ are expressed as follows:

$$\mathbf{h}_{i,b,c} = \sum_{a=0}^{n_u} N_{a,p}(\bar{u}_i)\mathbf{b}_{a,b,c}$$

For

$$(b,c) \in [0,..,n_v] \times [0,..,n_w] \tag{10}$$

Eq. (10) means multiple B-spline curve fitting to $(n_v+1) \times (n_w+1)$ polylines each of which consists of $i_{max}+1$ points passing along the u direction. We can finally obtain the control points $\mathbf{b}_{a,b,c}$ for $(a,b,c) \in [0,..,n_u] \times [0,..,n_v] \times [0,..,n_w]$ by solving the multiple B-spline curve fitting where n_u, p, $\bar{u}_i$, and $\mathbf{U}$ are given.

The overall process of approximate B-spline volume lofting is depicted in Fig. 3 and its main steps are summarized as follows:

(1) From the input points $\mathbf{f}_{i,j,k}$, construct w-directional polylines, each of which has the same number of points $k_{max}+1$. Determine $\bar{w}_k$ and $\mathbf{W}$ along w-direction. Then, do w-directional multiple B-spline curve fitting by Eq. (7), which converts input points $\mathbf{f}_{i,j,k}$ to intermediate control points $\mathbf{g}_{i,j,c}$ for $(i,j,c) \in [0,..,i_{max}] \times [0,..,j_{max}] \times [0,..,n_w]$.

(2) From the intermediate points $\mathbf{g}_{i,j,c}$, construct v-directional polylines, each of which has the same number of points $j_{max}+1$. Determine $\bar{v}_j$ and $\mathbf{V}$ along v-direction. Then, do v-directional multiple B-spline curve fitting by Eq. (9), which converts the points $\mathbf{g}_{i,j,c}$ to intermediate control points $\mathbf{h}_{i,b,c}$ for $(i,b,c) \in [0,..,i_{\max}] \times [0,..,n_v] \times [0,..,n_w]$.

(3) From the intermediate points $\mathbf{h}_{i,b,c}$, construct u-directional polylines, each of which has the same number of points $i_{\max}+1$. Determine $\bar{u}_i$ and $\mathbf{U}$ along u-direction. Then, do u-directional multiple B-spline curve fitting by Eq. (10), which converts the points $\mathbf{h}_{i,b,c}$ to final control points $\mathbf{b}_{a,b,c}$ for $(a,b,c) \in [0,..,n_u] \times [0,..,n_v] \times [0,..,n_w]$.

(4) Construct a B-spline volume $\mathbf{B}(u,v,w)$ with the control points $\mathbf{b}_{a,b,c}$.

The control points $\mathbf{b}_{a,b,c}$ of the B-spline volume $\mathbf{B}(u,v,w)$ are obtained by applying multiple B-spline curve fitting three times, which converts points $\mathbf{f}_{i,j,k}$ to intermediate points $\mathbf{g}_{i,j,c}$ and $\mathbf{h}_{i,b,c}$, and finally to $\mathbf{b}_{a,b,c}$.

3.3. Multiple B-spline curve fitting with adaptive knot placement

The multiple B-spline curve fitting is a key ingredient of approximate B-spline volume lofting. Given a set of polylines, it constructs a set of *compatible* B-spline curves each of which fits the points of its corresponding polyline [23,24]. B-spline curves are called compatible when they have the same B-spline order, the same number of control points, and the same knot vector. In the multiple B-spline curve fitting derived from Eqs. (7), (9), or (10), the polylines have the same number of points.

Given m_r+1 polylines $\mathbf{P}_i = \{\mathbf{p}_{i,j} | j = 0,...,m_c\}$ for $i \in [0,..,m_r]$, consider how to construct B-spline curves of

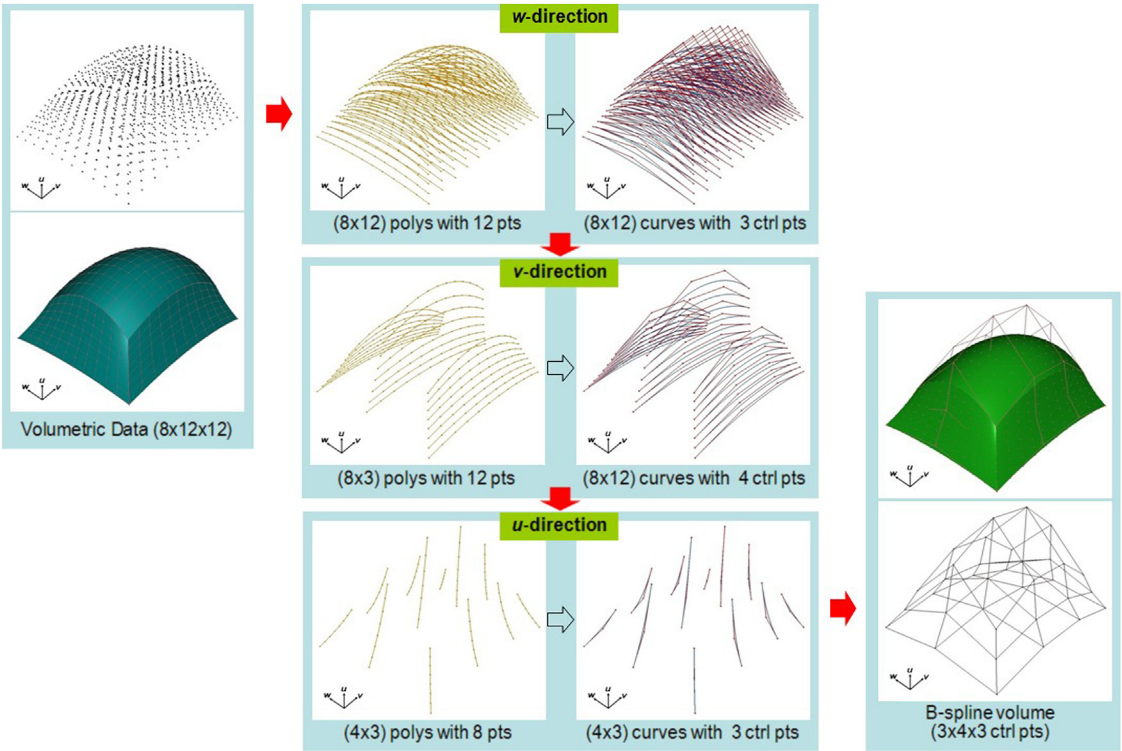

Fig. 3. Overall process of approximate B-spline volume lofting.

order p which are defined as $\mathbf{C}_i(t) = \sum_{k=0}^{n} N_{k,p}(t)\mathbf{b}_{i,k}$. As the polylines have the same number of points, all the points of each column $\mathbf{Q}_j = \{\mathbf{p}_{i,j} | i = 0, .., m_r\}$ are made to have the same parameter value $\bar{t}_j$, that is, $\mathbf{C}_i(\bar{t}_j) = \mathbf{p}_{i,j}$ for $i \in [0, .., m_r]$. The values $\bar{t}_j$ are equally spaced (*i.e.* $\bar{t}_j = j/m_c$) as mentioned in Section 3.2. The knots of a common knot vector $\mathbf{T} = \{t_0, t_1, .., t_{n+p}\}$ are usually chosen to reflect the distribution of the parameter values [18,19,23–26]. The control points $\mathbf{b}_{i,k}$ for $k \in [0, .., n]$ of the ith B-spline curve $\mathbf{C}_i(t)$ can be determined by minimizing the following least-squares errors:

$$E(\mathbf{b}_{i,0}, ..., \mathbf{b}_{i,n}) = \sum_{j=0}^{m_c} \|\mathbf{C}_i(\bar{t}_j) - \mathbf{p}_{i,j}\|^2 \tag{11}$$

where $i \in [0, .., m_r]$

Each minimization problem is equivalent to solving a linear system. Each resulting curve $\mathbf{C}_i(t)$ is a piecewise B-spline curve of order (degree+1) p with C^{p-2} continuity at interior knots [18,19]. It is easily found that the successive application of multiple B-spline curve fitting using Eq. (11) is much more efficient in terms of computation time and memory usage than the direct approach using Eq. (5).

Recall that m_c denote the number of points of each polyline and n the number of control points. The results of multiple curve fitting are affected by how to place the knots in Eq. (11). Several ways of placing the interior knots of the knot vector $\mathbf{T}$ have been suggested including the averaging technique for $m_c = n$, and the knot placement techniques (hereafter KTP) for $m_c > n$ [18,19]. When m_c is nearly greater than n ($|m_c - n|$ is small), it often generates undesirable results [23]. To avoid this, Piegl and Tiller [25] suggested another knot placement technique (hereafter NKTP).

Note that, in Lee and Park's work [20,21] on B-spline volume representation of BRDF data, knots are determined by the knot placement techniques KPT and NKTP. However, these knot placement techniques select the knots in a simple and trivial manner that each knot span contains almost the same number of parameter values, and that all interior knots are changed even though the number n increases by one, which makes it difficult to realize adaptive curve fitting and prevents compact and accurate B-spline volume representation of BRDF data when the data values change rapidly or have sharp peaks considered as not outliers but features (*i.e.*, for BRDF data of glossy or specular materials).

Recently, Park and Lee [26] presented a novel approach (hereafter DOM) to knot placement for B-spline curve fitting. After selecting dominant points from the given points, we can determine the interior knots by averaging the parameter values of the dominant points. This knot placement results in a stable system matrix which is not singular, and supports local modification realizing adaptive curve fitting that fewer knots are placed at flat regions but more at complex regions.

We extend the DOM approach for multiple B-spline curve fitting where a set of polylines are considered instead of a single polyline. Note that the points of each column $\mathbf{Q}_j = \{\mathbf{p}_{i,j} | i = 0, .., m_r\}$ have the same parameter value $\bar{t}_j$. In multiple B-spline curve fitting to a set of $(m_r + 1)$ polylines $\mathbf{P}_i = \{\mathbf{p}_{i,j} | j = 0, ..., m_c\}$, we select $(n+1)$ dominant columns $\mathbf{D}_k = \mathbf{Q}_{f(k)}$ from $(m_c + 1)$ columns $\mathbf{Q}_j$, determine a common knot vector $\mathbf{T}$ by averaging the parameter values $\bar{t}_j$ associated with the dominant columns $\mathbf{D}_k$ as follows:

$$t_{p+i-1} = \frac{1}{p-1} \sum_{k=i}^{i+p-2} \bar{t}_{f(k)} \text{ for } i = 1, .., n - p + 1 \tag{12}$$

where $f(k)$ is a monotonically increasing function that returns the index of the column corresponding to the kth dominant column (*i.e.* $\mathbf{Q}_{f(k)} = \mathbf{D}_k$). Then, with these knots, we obtain compatible B-spline curves $\mathbf{C}_i(t)$ by minimizing the least-squares errors in Eq. (11). Fig. 4 shows multiple B-spline curve fitting with and without adaptive knot placement.

To determine the dominants columns, we employ an adaptive refinement approach that places a new dominant column in the subset with the largest deviation at the current iteration. Note that the points $\mathbf{p}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ do not have any geometric meaning in Euclidean space because they come from BRDF data in Eq. (2). Thus, during the determination of the dominant columns, we transform the points $\mathbf{p}_{i,j}$ into $\hat{\mathbf{p}}_{i,j} = (\hat{x}_{i,j}, \hat{y}_{i,j}, \hat{z}_{i,j})$ whose coordinates are given as

$$\hat{x}_{i,j} = i/m_r, \hat{y}_{i,j} = \bar{t}_j,$$

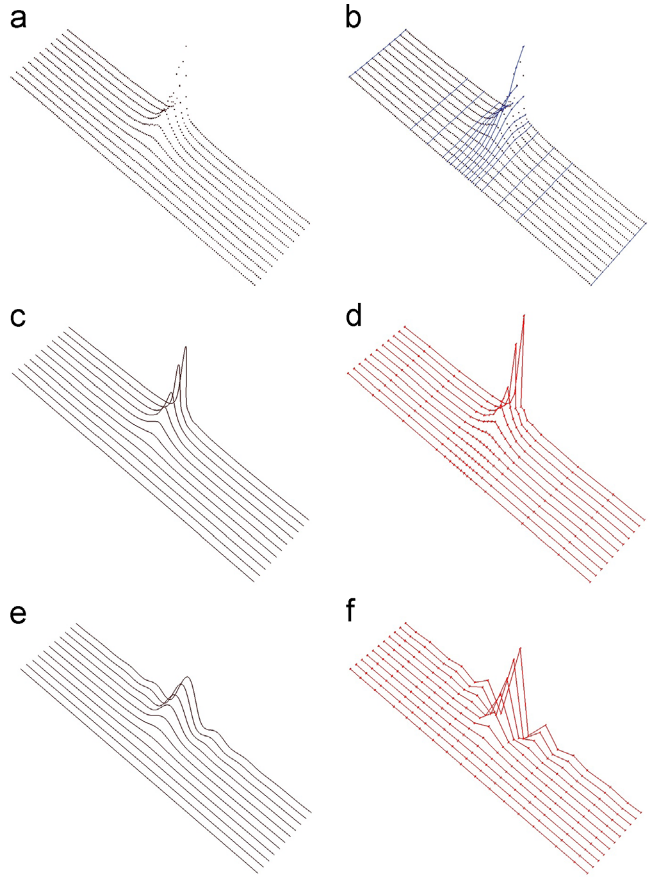

Fig. 4. Multiple B-spline curve fitting: (a) 11 polylines each of which consists of 151 points; (b) 17 dominant columns drawn in blue with the polylines; (c) compatible B-spline curves obtained using the dominant columns; (d) control polygons each of which consists of 17 control points; (e) compatible B-spline curves obtained by NKTP; and (f) control polygons (17 control points per each) obtained by NKTP.

$$\hat{z}_{i,j} = \sqrt{\left(\lambda_1 x_{i,j}\right)^2 + \left(\lambda_2 y_{i,j}\right)^2 + \left(\lambda_3 z_{i,j}\right)^2} \tag{13}$$

In this work, the same weight was set to each coordinate (*i.e.*, $\lambda_1 = \lambda_2 = \lambda_3 = 1$).

Now consider a subset of columns $\mathbf{S}^{s,e} = \{\mathbf{Q}_j | j = s, .., e\}$ in which there are only two dominant columns $\mathbf{Q}_s$ and $\mathbf{Q}_e$. Starting with an initial subset $\mathbf{S}^{0,m_c}$, we can obtain as many dominant columns as required by repeating the following steps: Among subsets of columns, find a subset $\mathbf{S}^{s,e}$ ($|e-s| > 1$) with the largest deviation, and choose a new dominant column $\mathbf{Q}_w$ ($s < w < e$) which minimizes the difference between the total arc lengths of $\mathbf{S}^{s,w}$ and $\mathbf{S}^{w,e}$. Then, divide the subset $\mathbf{S}^{s,e}$ into two subsets $\mathbf{S}^{s,w}$ and $\mathbf{S}^{w,e}$.

The deviation $D^{s,e}$ of a subset $\mathbf{S}^{s,e}$ can be defined in various ways. In this work, we define it as the maximum chord height $D^{s,e}_{\text{chd\_ max}} = \max^{e}_{j=s} \max^{m_r}_{i=0} \|\hat{\mathbf{p}}_{i,j} - \tilde{\mathbf{p}}_{i,j}\|$ or the average chord heights $D^{s,e}_{\text{chd\_avg}} = \frac{1}{(m_r+1)(e-s+1)} \sum^{e}_{j=s} \sum^{m_r}_{i=0} \|\hat{\mathbf{p}}_{i,j} - \tilde{\mathbf{p}}_{i,j}\|$ where $\tilde{\mathbf{p}}_{i,j}$ is the orthogonal projection of the $\hat{\mathbf{p}}_{i,j}$ onto the line segment $\overline{\hat{\mathbf{p}}_{i,s}\hat{\mathbf{p}}_{i,e}}$ (*i.e.*, chord height of the $\hat{\mathbf{p}}_{i,j}$ with respect to the line segment). The total arc length $L^{s,e}$ of the subset $\mathbf{S}^{s,e}$ is defined as $L^{s,e} = \sum^{m_r}_{i=0} \sum^{e-1}_{j=s} \|\hat{\mathbf{p}}_{i,j+1} - \hat{\mathbf{p}}_{i,j}\|$.

The maximum chord height tends to make dominant columns closer at sharp regions than the average chord height. For the BRDF data of diffuse or moderately glossy materials, as their polylines usually have smooth shape without sharp peaks, the maximum chord height is likely to be good to locate the dominant columns. For the BRDF data of glossy or specular materials, as their polylines usually have sharp peaks at small regions, the average chord height is enough to locate the dominant columns. We have experienced that the maximum chord height is usually good for the BRDF data sets of diffuse or moderately glossy materials, and that the average chord height is good for the BRDF data sets of glossy or specular materials.

As shown in Fig. 4, the multiple B-spline curve fitting using dominant columns has a notable tendency to generate fewer control points (correspondingly, curve segments) at flat regions but more at complex regions, which is very helpful to approximate BRDF data by a compact and accurate B-spline volume consisting of fewer control points. Note that the points in Fig. 4 are the ones transformed from an iso-mesh of the BRDF data.

3.4. Error-bounded BVB model fitting

When B-spline order (p, q, r), the number of control points (n_u, n_v, n_w), and the measured BRDF data of any real material are given, we can generate a BVB model to represent the BRDF data by applying the adaptive method of BVB model fitting. For practical applications, it is more useful to specify a tolerance in order to obtain a BVB model satisfying that the fitting error between the BVB model and the given BRDF data is smaller than the tolerance. The resulting BVB model is called *error-bounded*. In order to get a compact but accurate BVB model using the error-bounded BVB model fitting, it is important to reduce the number of knots (correspondingly control points) while keeping desired accuracy. As the problem of BVB model fitting is decomposed into sub-problems of multiple B-spline curve fitting, it can be completed with the successive application of error-bounded multiple B-spline curve fitting along each parametric direction. Thus, the given tolerance is equally divided into three, each for the error-bounded multiple B-spline curve fitting along its corresponding (u-, v-, or w-) parametric direction where the error is defined as $E = \max^{m_r}_{i=0} \max^{m_c}_{j=0} \|\mathbf{p}_{i,j} - \mathbf{C}_i(\bar{t}_j)\|$.

As it is not known in advance how many control points are required for the error-bounded multiple B-spline curve fitting, it is common to take an iterative (incremental or binary search-based) process which repeats fitting, checking deviation, and adjusting the number of knots (*i.e.*, the number of control points). In this work, we take the binary search algorithm [23,24] to complete the process since it is better in computation.

4. Application of BVB models to rendering

Given the BRDF data of any material, we can now generate a B-spline volume to represent the BRDF data by the proposed method for BVB model fitting. Then, we can use the B-spline volume (*i.e.* BVB model) as an alternative BRDF model for photorealistic rendering. Fig. 5 shows a conceptual framework of photorealistic rendering employed in this work.

It has three shaders for measured BRDF data, analytical BRDF models, and BVB models. They are integrated into PBRT renderer [2], which provides C++ API to extend its basic functionalities. Each shader is invoked with its associated BRDF model at run time when surfaces of interest should be rendered. The PBRT renderer supports several materials based on classic BRDF models such as perfect diffuse, Lafortune (LFT), Cook–Torrance (CKT). In our case, we added new C++ classes that inherit the abstract BRDF classes. For BVB models, we apply the proposed method for B-spline volume representation of the measured BRDF data of 100 materials obtained by Matusik et al. [7,8]. For analytical BRDF models, we use the results of Ngan et al. [12] where they conducted analytical BRDF model fitting to the measured BRDF data of the 100 materials.

For photorealistic rendering, we employed distribution ray tracing in PBRT [2]; we used direct lighting with 64 pixel samples and 64 light samples while applying single environmental lighting with a high dynamic range image and setting the maximum trace depth to 4. In order to decouple the effect of importance sampling [16] on the quality of rendering, we applied cosine-weighted uniform hemisphere sampling [2].

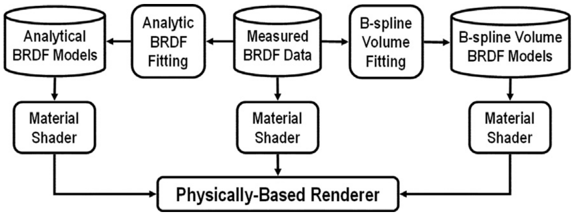

Fig. 5. Framework of rendering using three types of BRDF models.

All the rendered images were generated based on single threaded processing.

Figs. 6–8 show happy Buddha and sphere images which were generated by photorealistic rendering using four kinds of BRDF models (TBF, BVB, LFT, and CKT) for three materials (alum bronze, green metallic paint, and red plastic). Recall that TBF denotes a tabular form of measured BRDF data.

In this work, the size of rendered images is set to 256 × 512 and 512 × 512 pixels for happy Buddha and sphere scenes, respectively. For rendering with BVB models, we generated BVB models by error-bounded BVB model fitting with B-spline orders and tolerance rates specific to their associated BRDF data. Note that the images rendered using BVB models are very faithful to the images rendered using TBF models.

5. Experimental results

We implemented the adaptive BVB model fitting proposed in Section 3 and the photorealistic rendering described in Section 4. The implementation has been completed using C and C++ languages on an IBM compatible personal computer running MS Windows Vista with an Intel Xeon processor X5365. In this section, we show the experimental results of generating BVB models and of applying them to rendering in order to demonstrate that the BVB models are compact and faithful to represent the BRDF data for rendering applications. We also compare the BVB models with two types of analytical BRDF models (CKT and LFT) in accuracy of data fitting and quality of rendering.

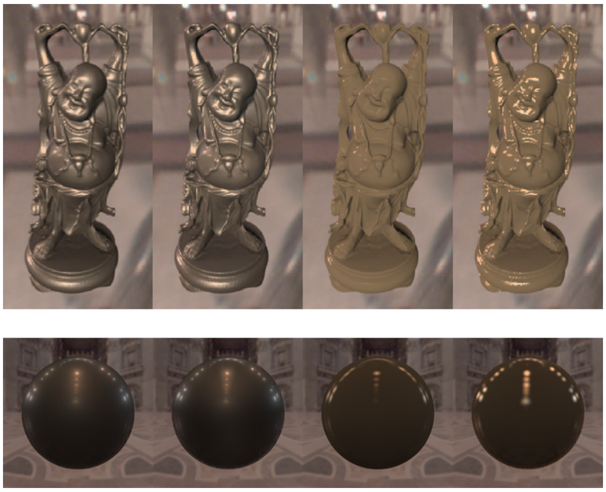

Fig. 6. Images rendered with BRDF models (from left to right, TBF, BVB, LFT, and CKT) of alum bronze.

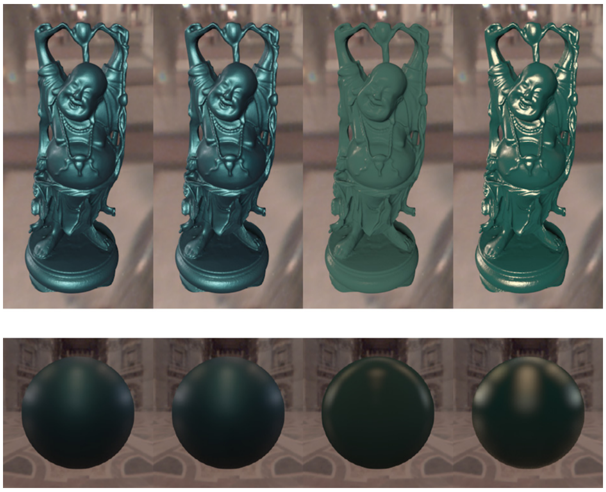

Fig. 7. Images rendered with BRDF models (from left to right, TBF, BVB, LFT, and CKT) of green metallic paint.

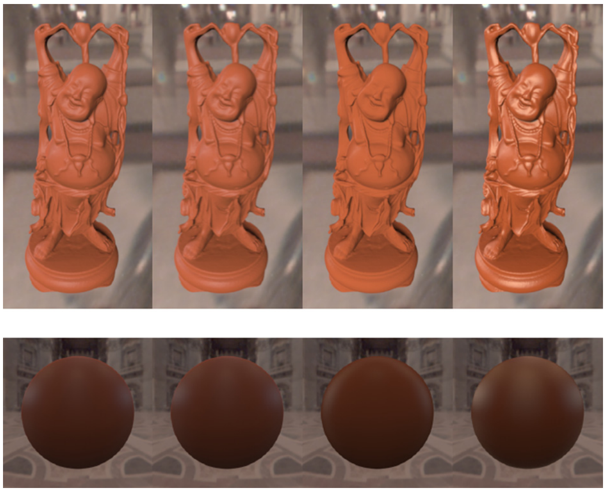

Fig. 8. Images rendered with BRDF models (from left to right, TBF, BVB, LFT, and CKT) of red plastic.

For test datasets, we use isotropic BRDF data of 100 real materials measured by Matusik et al. [7,8]. The size of each BRDF data is given as $(i_{max}, j_{max}, k_{max}) = (90, 90, 360)$. We found that all the BRDF data sets are more or less contaminated with noise and especially the data sets coming from specular materials contain sharp peaks considered as features.

5.1. Generation of BVB models and their application to rendering

For given B-spline orders (p, q, r), we applied the proposed method to obtain a BVB model of each material either by specifying the number of control points (n_u, n_v, n_w) or by specifying the tolerance. Then, we applied photorealistic rendering using the BVB model. To evaluate the quality of a BVB model, we computed the error between the BVB model and its corresponding BRDF data. In this work, we use the average fitting error AE which is defined as follows:

$$AE = \frac{1}{(i_{max}+1)(j_{max}+1)(k_{max}+1)} \sum_{\forall (i,j,k)} \|\mathbf{B}(\overline{u}_i, \overline{v}_j, \overline{w}_k) - \mathbf{f}_{i,j,k}\| \tag{14}$$

As simple statistical values of the BRDF data, we use the average and the largest values (AR, MR) of the BRDF data which are defined as follows:

$$AR = \frac{1}{(i_{max}+1)(j_{max}+1)(k_{max}+1)} \sum_{\forall (i,j,k)} \|\mathbf{f}_{i,j,k}\|,$$

$$MR = \max_{\forall (i,j,k)} \|\mathbf{f}_{i,j,k}\| \tag{15}$$

For the 100 materials used in this work, their *MR* values (0.4–8907.5) have great variation (*std*=1594.8), but their *AR* values (0.008–0.24) have small variation (*std*=0.056). We used the *MR* values to roughly classify the materials into three: diffuse, glossy specular, and highly specular materials. Diffuse materials mostly have small *MR* values (*i.e.* 0.76 for blue fabric and 3.49 for polyethylene), but highly specular materials have large *MR* values (*i.e.* 4533.6 for aluminum and 8907.5 for chrome). Glossy specular materials have in-between *MR* values (*i.e.* 31.4 for gold paint and 196.8 for purple paint).

To evaluate the quality of a BRDF model in view of rendering, we estimated the difference error between an image rendered using TBF (*i.e.* target image) and an image rendered using its fitted model (*i.e.* test image). For more consistent estimation, we made the two images have the same background image in TIFF format. Let $\mathbf{a}_{i,j}$ denote a pixel of a target image $\mathbf{a}$ and $\mathbf{b}_{i,j}$ a pixel of a test image $\mathbf{b}$. Each image size is $(x_{max}+1) \times (y_{max}+1)$, and the RGB value of each image pixel varies from 0 to 255. To assess the test image $\mathbf{b}$ with respect to the target image $\mathbf{a}$, we use the mean absolute error (*MAE*) defined as follows:

$$MAE = \frac{1}{(x_{max}+1)\bullet(y_{max}+1)} \sum_{\forall (i,j)} \|\mathbf{a}_{i,j} - \mathbf{b}_{i,j}\|. \tag{16}$$

BVB model fitting by specifying the number of control points is the basic tool for generation of a BVB model of each material. As an example of such BVB model generation, we constructed from the BRDF data of gold metallic paint ($AR = 0.092$, $MR = 4.074$) a set of B-spline volumes. Fig. 9 shows some graphical results of a BVB model where $(n_u+1, n_v+1, n_w+1) = (15, 15, 60)$ and $p = q = r = 4$. It is found that $AE = 0.003$ and $ME = 0.343$ for the BVB model. The first row of the figure shows three iso-meshes ($\theta_i = 30^\circ, 60\ ^\circ, 90^\circ$) in a (θ_o, ϕ_d) spherical coordinate system for the TBF model. The second row of the figure shows three corresponding iso-surfaces ($u = \frac{1}{3}, \frac{2}{3}, 1$) in a (v, w) spherical coordinate system for the BVB model. Note the BVB model can smooth out the noises and keep the overall feature of the data. Whereas the storage size of the BRDF data reaches to 70 megabytes in TBF (35 megabytes in halfway direction representation [7]), the size of the BVB model is about 320 kilobytes (0.46% of the input data).

Fig. 10 shows happy Buddha images rendered using four kinds of BRDF models for gold metallic paint. Fig. 10(a) is a target image obtained using the TBF model. Fig. 10(b)–(d) are test images obtained using BVB, LFT, and CKT models. The BVB model is the same one used in Fig. 9. Fig. 10(e)–(g) are the different images between the target image and the test images. In case of the difference image for the BVB model in Fig. 10(e), difference values are scaled up 20 times for easy recognition. The difference image errors (*MAE*) are 0.12, 7.55, and 11.82 for BVB, LFT, and CKT, respectively. Note that the BVB model is very faithful to be a good alternative of the measured BRDF data for rendering.

Fig. 11 shows the results of B-spline volume representation of the measured BRDF data of gold metallic paint where B-spline order is specified as $p = q = r = 2, 3, 4$, and the number of control points is given in the form of $(n_u+1, n_v+1, n_w+1) = (n, n, 4n)$. Fig. 11(a) shows the average fitting errors *AE*. Fig. 11 (b) shows image difference error (*MAE*) of the happy Buddha images rendered by using the resulting BVB models.

Note that, in Fig. 11, the errors *AE* and *MAE* usually decrease as the number of control points increases. This trend is similarly noticed in BVB model-based fitting and rendering for other materials. Obviously, the quality of fitted BVB models is positively correlated with the quality of images rendered using the BVB models. This implies that in order to make BVB models appropriate for better quality of rendering,

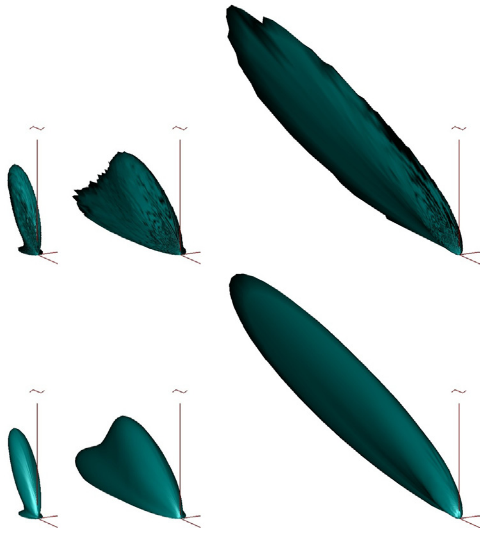

Fig. 9. Iso-meshes of a TBF model and a BVB model at three incident angles. (From left to right, θ_l=30°, 60°, and 90°).

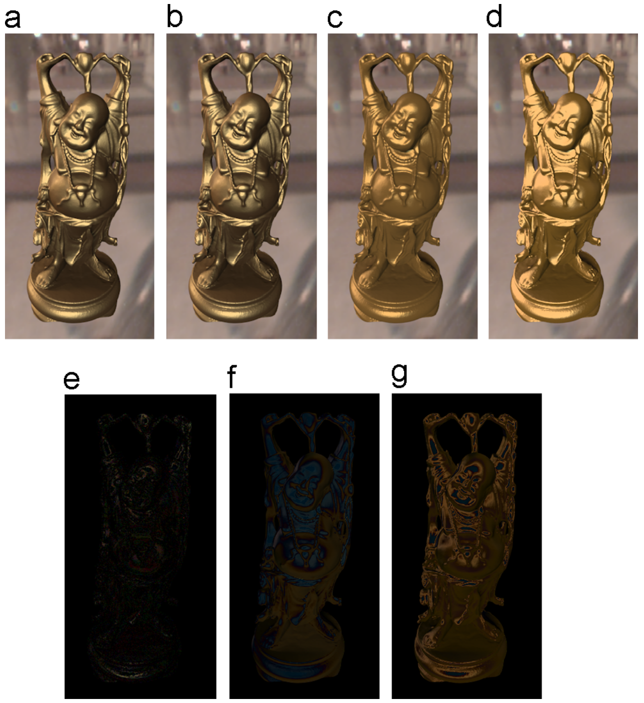

Fig. 10. Happy Buddha images rendered using BRDF models for gold metallic paint and their difference images.

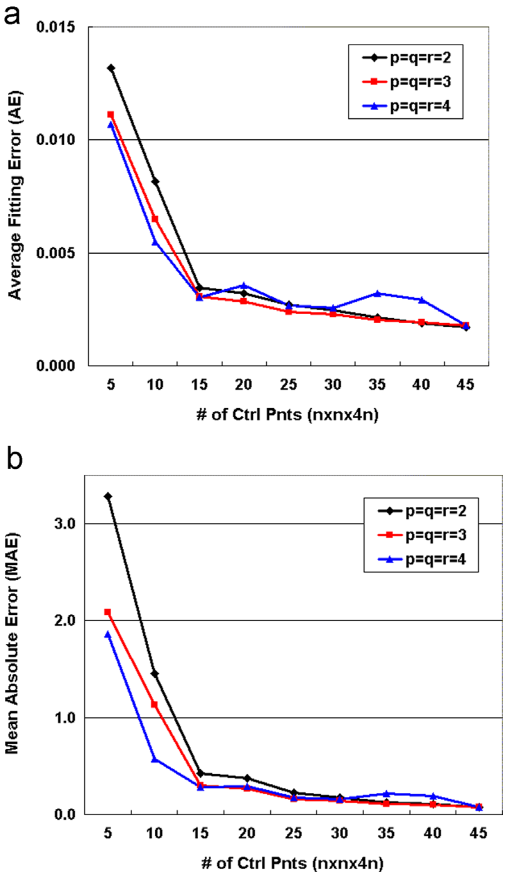

Fig. 11. Results of B-spline volume representation of the measured BRDF data of gold metallic paint: (a) plot of average fitting errors (AE) and (b) plot of image difference errors (MAE).

we need to reduce their fitting error as much as possible. However, there exists a trade-off between the fitting error and the number of control points of BVB models, so it is necessary to find a proper compromise between the fitting accuracy and the compactness of the BVB models. In addition, it is found from our extensive experiments that, for the same number of control points and the same B-spline order, the fitted BVB models of highly specular materials are likely to have larger fitting and rendering errors than those of the other materials. This means that the BRDF data of highly specular materials require more control points of BVB models to keep the same quality of fitting or rendering.

Specifying the tolerance is much easier to control the quality of BVB models than specifying the number (n_u, n_v, n_w) of control points since it is not known in advance how many control points are required. As an example of BVB model generation by specifying the tolerance (*i.e.* the error-bounded BVB model fitting), we constructed from the BRDF data of specular orange phenolic ($AR=0.133$, $MR=2357.369$) a set of B-spline volumes with different combinations of B-spline order and tolerance: B-spline order is specified as $p=q=r=2,3,4$, and the tolerance is given as $rate=0.50,\ 0.10,\ 0.05,\ 0.01,\ 0.005$. In this work, the tolerance is specified as $tol=MR\times rate$ where $0<rate<1$Note that specular orange phenolic is highly specular (its MR value is ranked 22th among 100).

Fig. 12 shows results of error-bounded B-spline volume representation of the measured BRDF data of specular orange phenolic. Fig. 12(a) shows the required number of control points of BVB models. Fig. 12(b) shows image difference error (*MAE*) occurring in sphere images rendered using the BVB models. There exists a trade-off between the tolerance and the number of control points (correspondingly, the rendering quality). As the tolerance decreases, the fitted BVB models require more control points while the quality of rendering usually increases. This trade-off is similarly noticed in BVB model-based fitting and rendering for other materials.

Fig. 13 shows the trade-off more clearly. Fig. 13(a) shows the target image rendered using the TBF model. Fig. 13(b)–(d) show three test images rendered using BVB models which are obtained with $p=q=r=2$ and $rate=\frac{1}{10},\frac{1}{20},\frac{1}{200}$. The required numbers of control points in Fig. 13(b), (c), and (d) are $(49,58,8)$, $(80,83,10)$, and $(90,90,12)$, respectively. Note that the BVB model obtained with the smaller tolerance represents the BRDF data more faithfully in rendering, but requires more control points.

Using the error-bounded BVB model fitting, we can find BVB models such that the rendering using them produces visually plausible results. This task requires an iterative

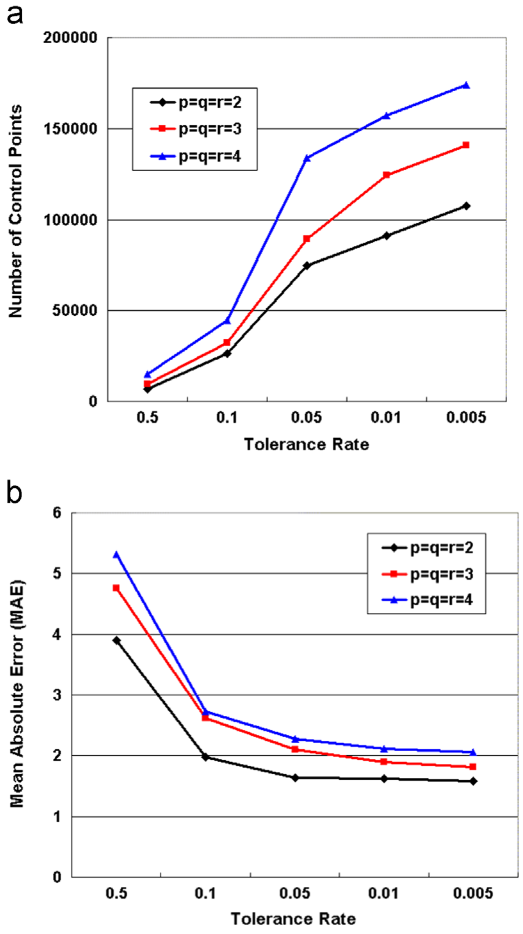

Fig. 12. Results of error-bounded B-spline volume representation of the measured BRDF data of specular orange phenolic: (a) the number of control points required and (b) image difference error (MAE) of sphere images rendered using BVB models.

process repeating the following steps: error-bounded BVB model fitting, BVB-based rendering, checking rendered images, and adjusting tolerances. From our practical experience, it is suggested to specify the tolerance rate as follows: $0.3 \leq rate \leq 0.05$ for diffuse materials ($MR < 50$), and $0.05 \leq rate \leq 0.0005$ for specular materials ($MR \geq 50$). It has been found by experience that low degree B-splines are effective in BVB model fitting to BRDF data for specular materials, and high degree for diffuse materials.

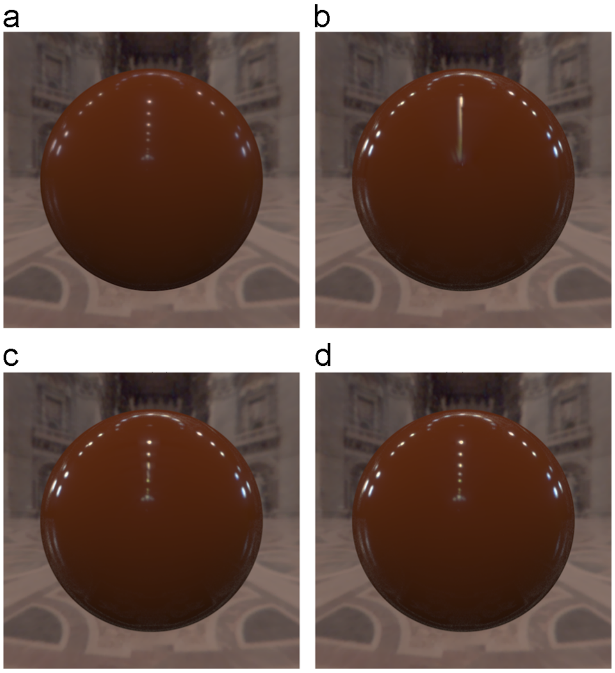

Fig. 13. Sphere images rendered using BRDF models for specular orange phenolic: (a) TBF; (b) BVB model ($rate = 0.1$,$(n_u, n_v, n_w) = (49, 58, 8)$); (c) BVB model ($rate = 0.05$,$(n_u, n_v, n_w) = (80, 83, 10)$); and (d) BVB model ($rate = 0.005$,$(n_u, n_v, n_w) = (90, 90, 12)$).

For all the 100 materials, we applied the iterative process to generate BVB models providing visually plausible rendering results, and investigated how many control points of the BVB models are required to represent the measured BRDF data sets. Fig. 14 shows the data size ratios of the resulting BVB models, and the tolerance rates with which the BVB models are fitted to the measured BRDF data. The data size ratio means the ratio of the number of control points of a resulting BVB model to the number of BRDF data points. The average ratio is 2.4%. The minimum ratio is 0.02% for blue-fabric with $(n_u, n_v, n_w) = (5, 5, 13)$, and the maximum ratio is 7.5% for chrome-steel with $(n_u, n_v, n_w) = (90, 90, 26)$. In the figure, BRDF materials were sorted in increasing order of their MR values. For most materials which are not highly specular, very low data size ratios (less than 1%) occur in BVB model generation. Even for highly specular materials, moderate data reduction ratios (2.5–7.5%) occur in BVB model generation.

From these results, we found that BVB models are quite suitable for representing the measured BRDF data since they are compact in model size and effective in preserving the features of material appearance. Note that 165 megabytes of memory is required to store all the 100 BVB models, whereas 7 gigabytes of memory is needed to store all the 100 BRDF data sets in the form of TBF. For a scene where a significant number of different BRDF data sets is used, it is very difficult to perform TBF-based rendering using a contemporary IBM-compatible personal computer because of excessive consumption of random access memory for the BRDF data sets, but it is easy to do with BVB-based rendering. Fig. 15 shows an image of a simple scene rendered using BVB models where 100 spheres of different materials are placed.

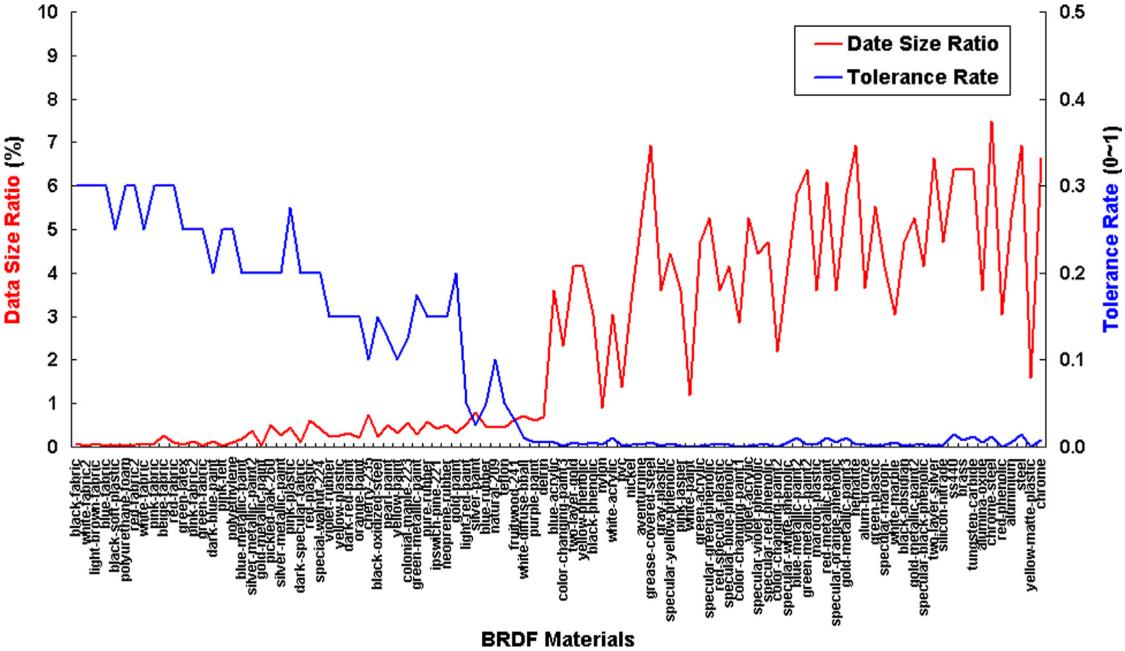

Fig. 14. Data size ratio required for BVB model fitting with visually plausible rendering results.

5.2. Comparison of BVB models with analytical models

We compared BVB models with two kinds of analytical BRDF models (CKT and LFT) in accuracy of data fitting and in quality of rendering. Recall that we used the results of Ngan et al. [12] to fit the analytical models to the measured BRDF data sets provided by Matusik et al. [7,8]. As the materials were named slightly differently by the two research groups, we used 86 identically named materials for comparison among BRDF models. For BVB models to be compared with CKT and LFT models, we employed the BVB models which are fitted with the tolerance rates shown in Fig. 14.

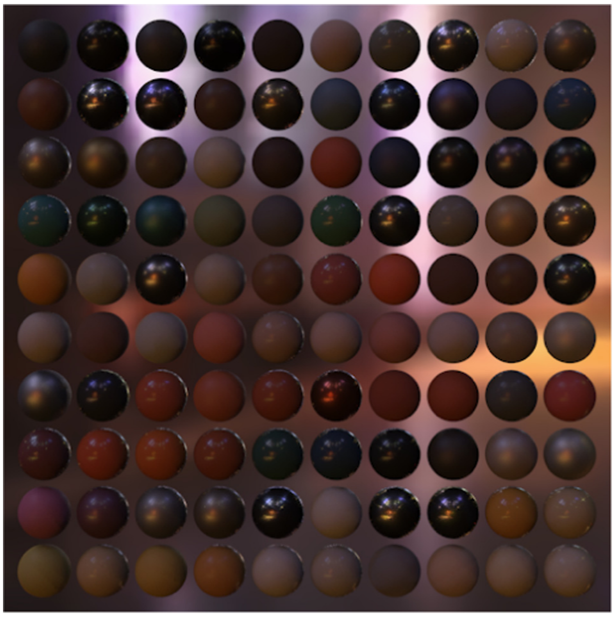

Fig. 15. Image of 100 spheres of different materials.

Fig. 16 shows the fitting errors AE of BRDF models (CKT, LFT, BVB) obtained by model fitting to the measured BRDF data of the 86 materials. In the figure, the errors were plotted in log scale and sorted in increasing order of MR value. Following the work of Ngan et al. [12], we omitted the BRDF values with incoming or outgoing angles larger than 80° to avoid possible outliers when computing the errors. Some BRDF models like CKT have extremely large errors when all the BRDF values are included, but the BVB models do not. Comparing the BRDF models in accuracy of data fitting, we found that the BVB models have much smaller fitting errors than the analytical models CKT and LFT.

This becomes more obvious when the sectional views of iso-meshes of the BRDF models are displayed. Fig. 17, for example, shows the sectional views ($\phi_d = 180°$) of iso-meshes ($\theta_i = 47°$) of three BRDF models generated from the measured BRDF data of nylon ($AR = 0.190$, $MR = 796.479$). Nylon is not specular, but special in that it has high peak values at graze angles. In the figure, cubic root was applied to scale down the data values. The BVB model fits the BRDF data very well while representing the overall features of the data. However, the CKT and LFT models have difficulty in representing the BRDF data.

Fig. 18 shows the plot of image difference errors (MAE) occurring in happy Buddha and sphere images rendered using BRDF models (TBF, BVB, LFT, CKT) for the 86 materials. In the figures, the errors were plotted in log scale and sorted in increasing order of MR value. Comparing the BRDF models in

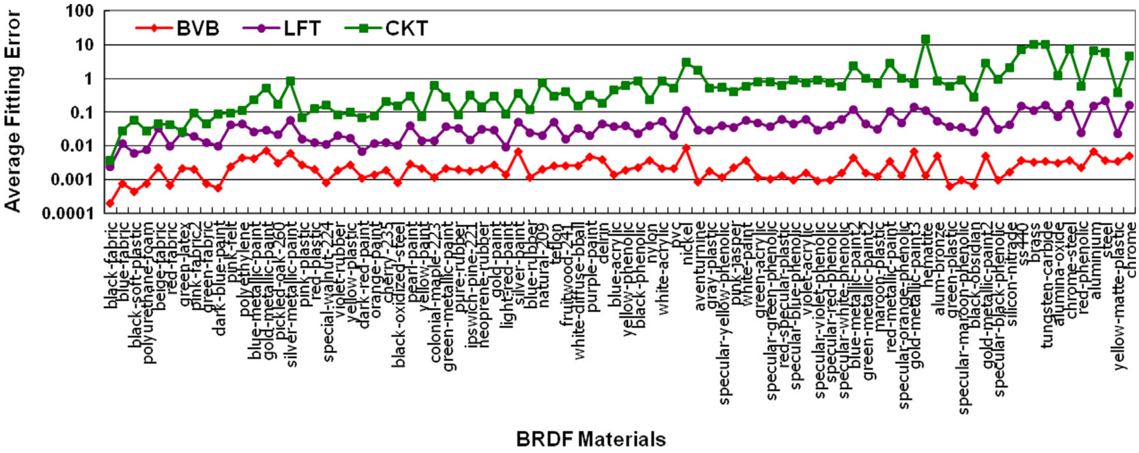

Fig. 16. Comparison of BRDF models CKT, LFT, BVB in average fitting errors.

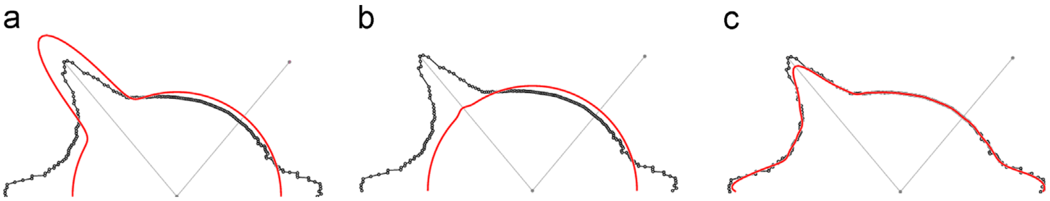

Fig. 17. Comparison of sectional views of iso-meshes of fitted BRDF models ($\theta_i = 47^\circ$, cubic root applied, nylon): (a) CKT; (b) LFT; and (c) BVB.

quality of rendering, we found that the BVB models have much smaller image difference errors than the CKT and LFT models. This becomes more obvious by referring to Figs. 6–10 that show the images rendered using four kinds of BRDF models (TBF, BVB, LFT, CKT). The images rendered using BVB models are much more faithful to the target images rendered using TBF models (*i.e.*, measured BRDF data), whereas the images rendered using LFT and CKT models are not in most cases.

From all the results mentioned above, we found that BVB models outperform the well-known analytical BRDF models (LFT and CKT) both in accuracy of data fitting and in quality of rendering.

6. Concluding remarks

In this paper, as one group of advocates asserting that B-spline models can be well suited for good representations of

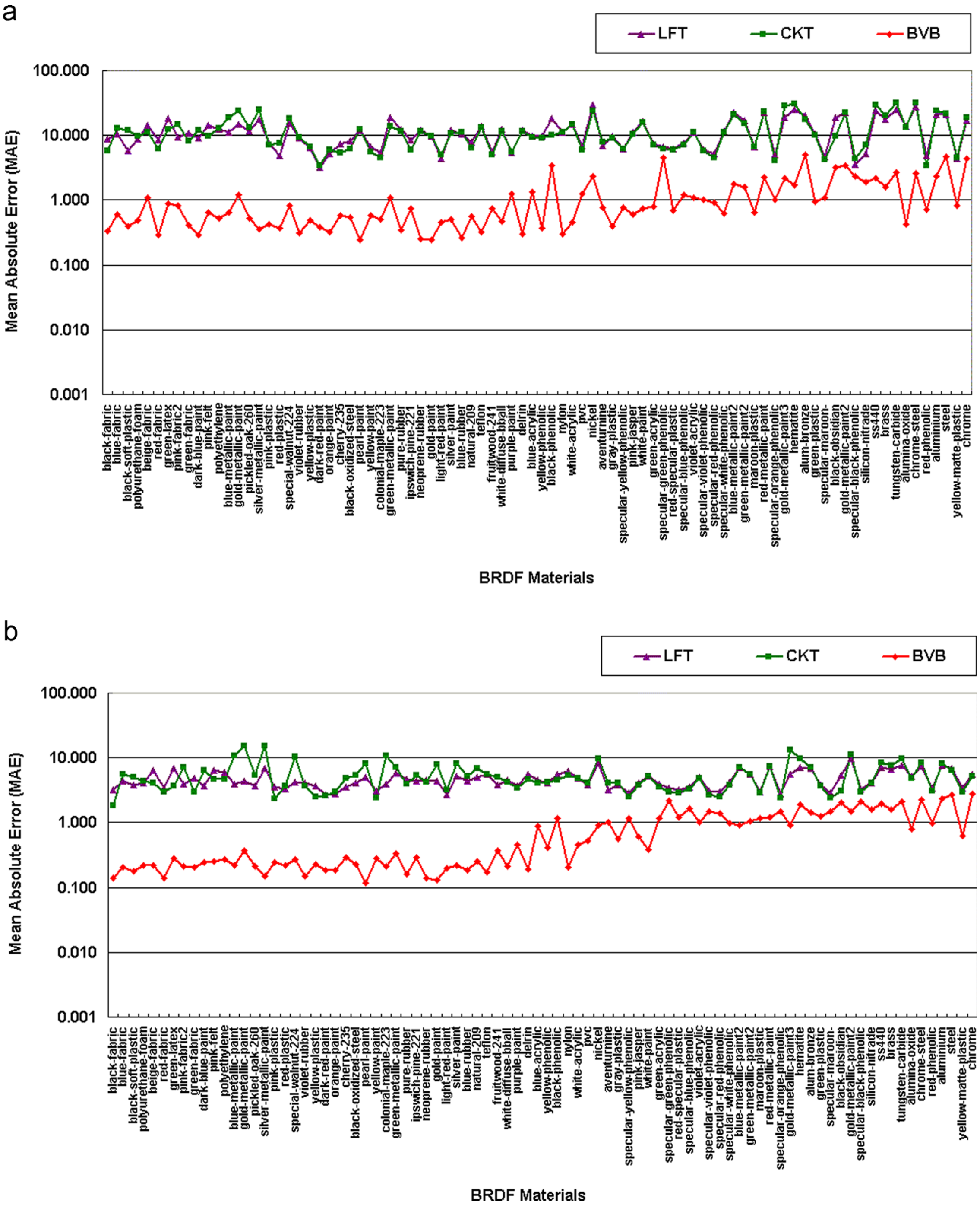

Fig. 18. Comparison of image difference errors (MAE) occurring in physically-based rendering using three BRDF models CKT, LFT, BVB: (a) happy Buddha images and (b) sphere images.

scientific data arising in computer graphics applications, we have proposed an adaptive method for B-spline volume representation (called BVB model) of BRDF data, and described the application of BVB models to photorealistic rendering. The proposed method basically performs approximate B-spline volume lofting, which decomposes the problem into three sub-problems of multiple B-spline curve fitting along *u*-, *v*-, and *w*- parametric directions. Especially, it makes the efficient use of knots in the multiple B-spline curve fitting and accomplishes adaptive knot placement along each parametric direction of a resulting B-spline volume. We have demonstrated with extensive experiments that BRDF data coming from a wide range of materials can be represented by BVB models which are compact in model size and faithful in preserving the features of material appearance.

In the Monte Carlo integration method [2], we need to provide a probability density function of BRDF for multiple importance sampling in ray tracing or incremental path generation of path tracing, which would further enhance the quality of rendered image with the same amount of computation. As a few analytical BRDF models including Phong [3] and Lafortune [4] support easy importance sampling, a reasonable approach to importance sampling for measured BRDF data has been to approximate the BRDF data by one of such analytical BRDF models. However, this also confronts the shortcomings of solving nonlinear optimization problems, and requires additional computation to evaluate the analytic form of a probability density function. In case of using TBF directly, a probability density function can be stored as another tabular form. However, this consumes a quite amount of storage size [2]. It is notable that Lawrence et al. [16] properly exploited a model based on factorization of BRDF data which is convenient for importance sampling. In a similar way of representing measured BRDF data in the form of tensor product B-splines, it is possible to approximate or represent the probability density function using the same form of B-splines.

As we have implemented single-threaded programs to perform rendering, we found that the BVB model demands more computation for rendering than the others. For example, the average time to render happy Buddha images of 256 × 512 pixels is found to be 31, 34, 37, 50 min for LFT, CKT, TBF, BVB, respectively. The CKT model demands more computation than the LFT model. Although TBF-based rendering does not need any complicated computation, it takes more time than LFT or CKT-based rendering since it is a great computational burden to load BRDF data into memory and to read the data during a rendering process. BVB-based rendering usually demands more computation in evaluating model values using Eq. (4) than the others. Thus, it takes more rendering time although using BVB models of lower degree somewhat reduces the rendering time. However, BVB models support the computation of BRDF values by matrix multiplication, but analytical models do not allow it since they are defined by non-polynomial equations. This means that the computation of BRDF values with BVB models can be much enhanced by adopting, for example, single instruction multiple data (SIMD) parallel computing [27].

We expect to do more research on the following topics for future works. First, as BVB models are not good in computation to render with single thread programs, we plan to implement advanced SIMD parallel computing to enhance the computation of BRDF values with BVB models and correspondingly the overall computational performance of BVB-based rendering. Second, we will devise a method for representing the probability density function of BRDF in the form of B-splines. Third, in order to deal with 4D BRDF data which come from anisotropic materials, we will extend the proposed method to generate a multivariate B-spline model from multivariate vector-valued scientific data.

Conflict of interest statement

The authors have no conflict of interest.

References

[1] Dorsey J, Rushmeier H, Sillion F. *Digital Modeling of Material Appearances*, 1st ed., New York: Morgan Kaufmann; 2008.

[2] Pharr M, Humphreys G. *Physically Based Rendering: From Theory to Implementation*, 1st ed., New York: Morgan Kaufman; 2004.

[3] Phong BT. Illumination for computer generated pictures. *Communications of the ACM* 1975;**18**(6)311–7.

[4] Lafortune EPF, Foo SC, Torrance KE, Greenberg DP. Non-linear approximation of reflectance functions. In: Proceedings of ACM SIGGRAPH Computer Graphics; 1997.

[5] Cook RL, Torrance KE. A reflectance model for computer graphics. *ACM Transactions on Graphics* 1982;**1**(1)7–24.

[6] Rusinkiewicz S. A survey of BRDF representation for computer graphics. Technical report. Stanford University; 1997; ⟨http://www.cs.princeton.edu/~smr/cs348c97/surveypaper.html⟩.

[7] Matusik W, Pfister H, Brand M, McMillan L. Efficient isotropic BRDF measurement. In: Proceedings of the 14th Eurographics Workshop on Rendering; 2003.

[8] Matusik W, Pfister H, Brand M, McMillan L. A data-driven reflectance model. *ACM Transactions on Graphics* 2003;**22**(3)759–69.

[9] Optical Gauging Mechanism (OGM). ⟨http://www.dressingsim.com/DFL_en/product/OGM/⟩.

[10] Gunther J, Chen T, Goesele M, Wald I, Seidel HP. Efficient acquisition and realistic rendering of car paint. In: Proceedings of 10th International Fall Workshop on Vision, Modeling, and Visualization; 2005.

[11] Borshukov G. Measured BRDF in film production: realistic cloth appearance for "The Matrix Reloaded". In: Proceedings of ACM SIGGRAPH Sketches & Applications; 2003.

[12] Ngan A, Durand F, Matusik W. Experimental analysis of BRDF models. In: Proceedings of Eurographics Symposium on Rendering; 2005.

[13] Schroder P, Sweldens W. Spherical wavelets: efficiently representing functions on the sphere. In: Proceedings of ACM SIGGRAPH Computer Graphics; 1995.

[14] Koenderink JJ, van Doorn AJ, Stavridi M. Bidirectional reflection distribution function expressed in terms of surface scattering modes. In: Proceedings of the Fourth European Conference on Computer Vision; 1996.

[15] Ozturk A, Kurt M, Bilgili A, Gungor C. Linear approximation of bidirectional reflectance distribution functions. *Computers & Graphics* 2008;**32**(2)149–58.

[16] Lawrence J, Rusinkiewicz S, Ramamoorthi R. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics* 2004;**23**(3)496–505.

[17] Kurt M, Cinsdikici MG. Representing BRDFs using SOMs and MANs. *SIGGRAPH Computer Graphics* 2008;**42**(3)1–18.

[18] Hoschek J, Lasser D. *Fundamentals of Computer Aided Geometric Design*, 1st ed., Massachusetts: A K Peters; 1993.

[19] Piegl L, Tiller W. *The NURBS Book*, 2nd ed., New York: Springer; 1995.

[20] Lee JH, Park H. B-spline representation of measured BRDF for photorealistic rendering. In: Proceedings of the 3rd Korea–China Joint Conference on Geometric and Visual Computing (KCJC); 2007.

[21] Lee JH, Park H. B-spline volume BRDF representation and application in physically-based rendering. *Transactions of the Society of CAD/CAM Engineers* 2008;**13**(6)469–77.

[22] Lee JH, Park H. Note on B-spline volume representation of measured BRDF Data. In: Proceedings of 2010 Asian Conference on Design & Digital Engineering (ACDDE); 2010.

[23] Park H, Kim K, Lee SC. A method for approximate NURBS curve compatibility based on multiple curve refitting. *Computer-Aided Design* 2000;**32**(4)237–52.

[24] Park H. An approximate lofting approach for B-spline surface fitting to functional surfaces. *International Journal of Advanced Manufacturing Technology* 2001;**18**(7)474–82.

[25] Piegl L, Tiller W. Surface approximation to scanned data. *The Visual Computer* 2000;**16**(7)386–95.

[26] Park H, Lee JH. B-spline curve fitting based on adaptive curve refinement using dominant points. *Computer-Aided Design* 2007;**39**(6)439–51.

[27] Kurzak J, Alvaro W, Dongarra J. Optimizing matrix multiplication for a short-vector SIMD architecture – CELL processor. *Parallel Computing* 2009;**35**(3)138–50.