

# Bidirectional Texture Function Modeling: A State of the Art Survey

Jiří Filip, *Member, IEEE*, and Michal Haindl, *Senior Member, IEEE*

**Abstract**—An ever-growing number of real-world computer vision applications require classification, segmentation, retrieval, or realistic rendering of genuine materials. However, the appearance of real materials dramatically changes with illumination and viewing variations. Thus, the only reliable representation of material visual properties requires capturing of its reflectance in as wide range of light and camera position combinations as possible. This is a principle of the recent most advanced texture representation, the Bidirectional Texture Function (BTF). Multispectral BTF is a seven-dimensional function that depends on view and illumination directions as well as on planar texture coordinates. BTF is typically obtained by measurement of thousands of images covering many combinations of illumination and viewing angles. However, the large size of such measurements has prohibited their practical exploitation in any sensible application until recently. During the last few years, the first BTF measurement, compression, modeling, and rendering methods have emerged. In this paper, we categorize, critically survey, and psychophysically compare such approaches, which were published in this newly arising and important computer vision and graphics area.

**Index Terms**—BTF, BRDF, 3D texture, surface texture, texture measurement, texture analysis, texture synthesis, texture modeling, data compression, psychophysical study, light transport.

## 1 INTRODUCTION

ROBUST visual classification, segmentation, retrieval, or view/illumination invariant methods dealing with images of textured natural materials, as well as augmented reality applications creating virtual objects in rendered scenes with real material surface optical properties, require realistic physically correct textures. Such texture representation considerably depends on the view and illumination directions and can be efficiently and most accurately obtained in the form of rough surface textures represented by Bidirectional Texture Function. Additionally, applications of this advanced texture representation allow accurate photo-realistic material appearance approximation for such complex tasks as visual safety simulations or interior design in automotive/airspace industry (Fig. 2), architecture, or dermatology [8] among others.

The first attempt to formally specify real material reflectance was by Nicodemus et al. [76], who introduced a novel nomenclature for the *Bidirectional Reflectance Distribution Function* (BRDF), although its importance has long been recognized by artists and scientists such as Galileo [78]. A four-dimensional BRDF was formalized in [76] as a specific case of eight-dimensional *Bidirectional Scattering Surface Reflectance Distribution Function* (BSSRDF), restricted to opaque materials. Multispectral BRDF is a 5D function describing how a sample's color reflectance depending on illumination and viewing directions. Two

principal properties of BRDF are view and illumination direction *reciprocity* and *energy conservation*. To represent the spatial dependencies in surface texture, a BRDF can be extended to the six-dimensional *Spatially Varying BRDF* (SVBRDF), i.e., a set of surface points with mutually independent BRDFs. However, the two mentioned properties impose restrictions on SVBRDF use, mostly for representation of near-flat and opaque materials.

Twenty years later, Dana et al. [10] proposed a more general representation of sample structure geometry and its light transport properties [54], in the form of *Bidirectional Texture Function* (BTF), which is applicable to most real-world surfaces. Multispectral BTF is a seven-dimensional function, which considers measurement dependency on color spectrum, planar material position, as well as its dependence on illumination and viewing angles:

$$BTF(r, \theta_i, \phi_i, \theta_v, \phi_v), \quad (1)$$

where the multi-index  $r = [r_1, r_2, r_3]$  specifies planar horizontal and vertical position in material sample image,  $r_3$  is the spectral index, and  $\theta, \phi$  are elevation and azimuthal angles of the illumination and view direction vector (see Fig. 3). The BTF measurements comprise a whole hemisphere of light and camera positions in observed material sample coordinates according to the selected quantization steps (see Fig. 4).

The variability of the material sample appearance in registered and rectified BTF images is illustrated in Fig. 5.

Rough textures provide ample information about local light field structure as well as the surface relief. Effects presented in rough textures such as self-occlusions, self-shadowing, interreflection, or subsurface scattering are preserved in BTF measurements (Fig. 1). A downside of using original measurements is their enormous storage size since an average sample takes several gigabytes.

Methods exist for interactive editing of measured BTF [47], which enable us to change materials properties by

• The authors are with the Pattern Recognition Department, Institute of Information Theory and Automation of the AS CR, Pod vodarenskou vezi 4, 182 08 Praha 8, Czech Republic. E-mail: {jiri.filip, haindl}@utia.cz.

Manuscript received 25 July 2007; revised 7 Apr. 2008; accepted 29 Sept. 2008; published online 9 Oct. 2008.

Recommended for acceptance by Y. Sato.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-07-0452.

Digital Object Identifier no. 10.1109/TPAMI.2008.246.

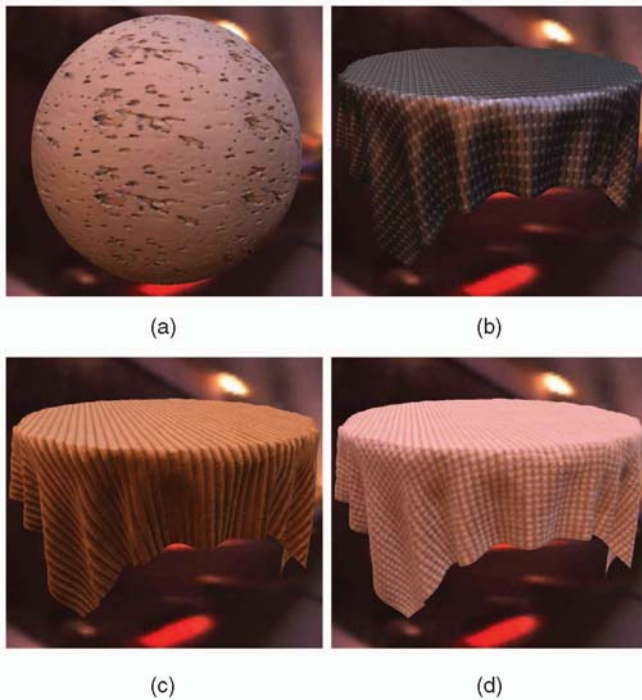


Fig. 1. Examples of measured BTF samples ((a) ceiling panel, (b) aluminum profile, and (c), (d) two fabrics) rendered on objects and illuminated by environment lighting.

several physically nonplausible operators. However, a fast BTF synthesis method with substantial compression is essential for many applications requiring accurate real-time rendering of these data using graphics hardware. In addition, the original BTF measurements cannot be used in any practical application due to missing necessary measurements from all arbitrary vantage points under arbitrary illumination and due to their small size. Thus, a seamless spatial enlargement (modeling) method of this, otherwise, huge BTF data is inevitable.

**Contribution of the paper.** The main contribution is to provide the first thorough state-of-the-art overview of BTF measurement, modeling, and compression methods published so far, while selected methods are mutually compared in several aspects. The only survey dealing with some parts of the complex BTF acquisition and modeling process is [72]. This survey provides an elegant, brief overview of principles of BTF measurement, compression, and visualization methods and explains issues related to a

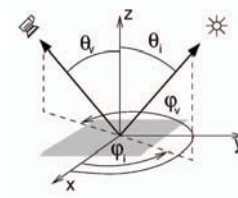


Fig. 3. Relationship between illumination and viewing angles within sample coordinate system.

whole BTF processing pipeline from acquisition to optimal BTF rendering on graphics hardware. Although the survey provides useful insight into the field of BTF acquisition, compression, and rendering, it lacks rigorous side by side comparison of individual BTF measurement setups and compression methods and only touches on BTF synthesis. Additionally, the mentioned survey paper does not comprehensively mention all published methods in the field of BTF acquisition, compression, and modeling, while our survey does. Furthermore, a number of novel BTF acquisition, compression, and synthesis methods have appeared since publication of the said survey [72].

In this paper, we focus mainly on thorough comparison and categorization of the BTF measurement systems and synthesis methods. We put the emphasis on modeling of BTF data and rigorous comparison of the selected compression and synthesis techniques. While, in [72], the parameters of the compared methods were selected in order to give subjectively nice results and to fit within current graphics hardware limits, we have performed a psychophysical experiment using several BTF samples. Our experiment determined BTF sample-dependent parameters of the selected tested methods providing results visually indiscernible from the original BTF rendering. The proposed psychophysical testing allowed us to prepare a fair comparison of the selected BTF compression and modeling techniques in terms of analysis and synthesis speed, compression ratio, etc.

**BTF applications in computer vision.** BTF data are the most advanced and accurate digital representation of a real-world material visual properties to date and their analysis provides abundant information about the measured material that cannot, for the majority, be attained using any alternative visual measurements or representations, e.g., image-based relighting, bump/displacement mapping, spatially varying BRDFs, etc.



Fig. 2. Examples of leather, fabrics, aluminum, and lacquered wood BTF rendering in Mercedes C Class interior using [25] method (3D model courtesy of DaimlerChrysler).

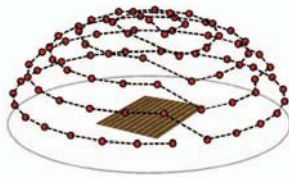


Fig. 4. An example of light or camera trajectory above the sample during measurement [80].

The nature of BTF data allows their straightforward exploitation for design and testing of illumination [38], [85] and view-invariant features and algorithms in numerous robust texture classification [5], segmentation, and retrieval applications. Other image processing problems, such as image restoration, aging modeling, face recognition, security, 3D object recognition, content-based image retrieval [85], and many other tasks can and should benefit from BTF comprehensive information. An example of usefulness of BTF data is a study of cast shadows by material structure in [79] and the analysis of material dimensionality in [82]. Moreover, a recent psychophysical studies of these data in [22] and [21], together with a study present in this paper, has shown that analysis of different BTF samples can help us to understand human perception of different real-world materials. For all of the above-mentioned tasks, a reliable and compact representation of massive BTF data is needed. Such a representation should allow fast reconstruction and modeling of BTF data, which is the aim of this paper. By modeling we understand BTF synthesis from its parameters of arbitrary size, without visible repetitions or other distortions, visually similar to original data.

**Paper organization.** Section 2 surveys principles and properties of BTF measurements systems. Different ways of representing measured BTF data and categorization of published methods are explained in Section 3. Section 4 summarizes BTF compression techniques, while the subsequent Section 5 deals with more general methods allowing simultaneous compression and enlargement. Modeling quality criteria are the subject described in Section 6. Selected methods, i.e., those that are described in dedicated numbered paragraphs in Sections 4 and 5, are further compared and tested thoroughly with psychophysical experiment in Section 7 and Section 8 concludes the paper.

## 2 BTF MEASUREMENT

Since accurate and reliable BTF acquisition is not a trivial task, only a few BTF measurement systems exist up to now [10], [39], [50], [70], [75], [80], [89]. However, their number increases every year with respect to the growing demands for photo-realistic virtual representations of real-world materials. These systems are (similar to BRDF measurement systems) based on light source, video/still camera, and material sample. The main difference between individual BTF measurement systems is in the type of measurement setup allowing four degrees of freedom for camera/light and the type of measurement sensor (CCD, video, etc.). In some systems, the camera is moving and the light is fixed [10], [73], [80], while in others, e.g., [50], it is just the opposite. There are also systems, where both camera and light source stay fixed [39], [70]. The main requirement on BTF measurements is

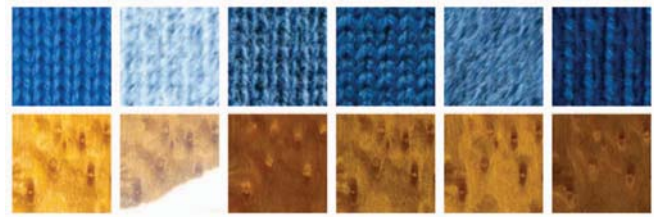


Fig. 5. Examples of significant material appearance change for varying illumination/view directions, as captured by BTF for knitted wool and lacquered wood.

accurate image rectification, i.e., aligning of texture normal with view vector, mutual registration of single BTF measurements for different viewpoints, and sample visual constancy during measurement. The registration accuracy strongly depends on positioning errors of the light/camera used while the visual constancy depends on stability of material properties during a long measurement time when exposed to an intensive light source. BTF, if appropriately measured from real material samples, offers enough information about material properties, such as anisotropy, masking, or self-shadowing.

Pioneering work in BTF acquisition has been done by Dana et al. [11], who measured a large set of various materials with measurement setup based on fixed light source and moving camera and material sample position. The resulting CURET BTF database has a relatively sparse angular resolution. Although individual images are not rectified to frontal view position, the authors provided image coordinates to allow their further rectification.

Some of material measurements from the CURET database were further extended in KTH TIPS database [40]. The main purpose of the authors was to provide variations of scale in addition to pose and illumination. Such a feature is not available in any other BTF database discussed below. Each measured material is sampled in three illuminations, three viewing directions, and nine scales. Measurement was performed by still camera and ordinary desk light. Authors provide no rectification marks in the data images, so the database is mainly focused on material classification applications. A slight variation of this database is the database KTH-TIPS2 [5], introducing additional ambient lighting.

The BTF measurement system developed by Koudelka et al. [50] uses fixed video camera observing material sample positioned in a computer-controlled pan/tilt head. The sample is illuminated by an LED array mounted on a robotic arm. The system offers an impressive angular resolution and rigorous image registration. However, the spatial resolution of the resulting images is rather small, which can negatively impact many BTF modeling methods.

A BTF measurement system based on extended setup of Dana was developed by Sattler et al. [80]. The main change is having the camera on a half-circle rail remote-controlled positioning system. The setup provides rectified measurements of reasonable angular and spatial resolutions. Later hardware upgrade and improvement of postprocessing algorithms in this setup suppressed registration errors and enabled an even higher spatial resolution. The data sets from this setup were used in our experiments for comparison of several BTF compression and modeling methods in Section 7.

The interesting idea of BTF measurement was presented by Han and Perlin [39]. Their system consists of a triangular tapered tube made of mirrors presenting kaleidoscopic replications of the material surface positioned under the tube. A fixed camera observes the kaleidoscopic image, where individual triangular subimages correspond to a surface observed from different viewpoints. The sample is illuminated by a digital projector illuminating individual triangles (i.e., generating illumination positions) in the kaleidoscopic image in a shared optical path with the camera, using a beam splitter. The advantage of this inexpensive system is subpixel BTF images registration. However, the spatial resolution is limited by the camera resolution.

A dermatology BTF database, the Rutgers Skin Texture Database [8], contains various skin diseases taken from the illumination and camera-controlled positions. They use two measurement setups: a light arc (quartz halogen or fiber optic illuminator) and camera mounted on a manually articulated arm on a tripod. Data sets have either three viewing positions and 10 illumination positions or four and eight corresponding positions. The Light Stage measuring system [15] was designed for similar application. It consists of a two-axis rotation system in which a small set of viewpoints is combined with a dense sampling of incidental illumination.

A similar technique [26] for BTF measurement involves a set of 5 cameras and a set of 6 lights each affixed to its own large motorized arc surrounding the object on a turntable. Additionally, a range scanner is used to capture 3D geometric data of the measured objects, so the measured samples need not to be planar. This system provides a relatively coarse angular sampling.

Another BTF setup presented by Dana and Wang [12] is based on a parabolic mirror in whose focal point is placed the surface of the observed material sample. Illumination is introduced by a coherent beam spatially positioned into the mirror by means of a moving aperture, which enables convenient automated control for illumination. The sample surface point reflectance for varying viewing directions is observed as the image captured by a camera sharing the same optical path with the illumination beam by means of a beam splitter. This setup provides very dense angular and spatial resolutions, but has limited maximal elevation angles and maximal sample height variations; it also requires long measurement times due to planar translations used to scan the surface of the sample.

A novel measurement system of the University of Bonn [70] uses a dense array of digital still cameras uniformly mounted on a hemispherical structure. Built-in flash lights of the cameras are used as light sources. The system enables subpixel registration of measured images by predefined image transformations of individual fixed cameras and provides high angular and spatial resolutions.

A system of similar topology, KULETH, was presented in [73]. The system is based on a half-hemispherical chassis containing a spatially uniform array of illumination sources. The material sample is placed on a turn table and observed by a camera being positioned using a tilt arm. Resulting BTF data sets have a very high angular and moderate spatial resolution.

BTF rendering that incorporates underlying geometry modeling, using a mesostructure distance function, is proposed by Wang et al. [90]. The method enables fast

rendering of mesostructure silhouette in graphics hardware. The setup for simultaneous measurement of distance function and BTF is presented as well.

Finally, the acquisition system [75] uses a number of planar patches of the material pasted onto square backing boards with known dimensions, which are then positioned to form a pyramid-like target. This setup provides sparsely sampled BTF measurements of 13 unaligned views from a variable manual camera position and the light direction is sampled by moving a hand-held electronic flash. The entire BTF space is interpolated from these sparsely sampled measurements by means of histogram fitting and interpolation of steerable pyramid parameters and pixel distributions. This system introduces large interpolation errors and requires manual marking of image positions.

The surveyed BTF acquisition systems can be divided into two categories: systems whose authors enable a wide research community to use some of the measured BTFs are more detailedly described in Table 1, while the parameters of the others systems are shown in Table 2.

The optimal BTF measurement setup design is a tricky task, heavily dependent on the required accuracy and the target application of the resulting BTF data. The highest illumination and view positioning accuracy requires avoidance of as many moving parts in the setup as possible. If this cannot be achieved completely [70], [39], a simple shift and rotation elements [12], [73], which are convenient for easy calibration and error compensation, should be preferred instead of complicated and imprecise robotic arms [11], [50], [80].

Data-consistency-critical applications can benefit from nonuniform sampling strategies. Such systems should apply more dense sampling in the areas of expected interest, e.g., near-specular reflection, etc. This approach should avoid missing specular peaks, etc., due to improper angular quantization steps. The resulting correct BTF data can be resampled to a uniform quantization by a global interpolation algorithm in a postprocessing step if required. A disadvantage of this approach is a necessity to use moving elements in the setup due to a variable quantization step, which is dependent on proximity of view and specular directions. An interesting case of a continual sampling of view and illumination directions is shown in [12].

For low-budget applications requiring capture of a reliable look-and-feel of the material without excessive accuracy demands, such as Web presentation of materials, etc., an approximate acquisition setups using only sparse BTF sampling might be sufficient [5], [40], [75].

As the rectification and registration of individual images is one of the main sources of error in BTF data, attention should be paid to design of proper, unambiguous ground-true registration marks accompanying the measured material sample [50]. Idealized errorless moving parts or immovable measurement setups can adopt a predefined rectification transformation for each view direction, without the need of an additional registration procedure [70].

It should also be noted that the larger the sample to be measured is, the farther away the light and camera should be placed to avoid a change of corresponding illumination and viewing angles over the sample span. Thus, the maximum required size of material samples should be considered prior to the setup design. Similarly, a maximum height of the measured materials should also be considered

TABLE 1  
Comparison of Publicly Available BTF Databases

parameter	public BTF databases					
	CUReT99 [11]	Yale03 [50]	Bonn03 [80]	Bonn03 ext.	KTH TIPS [40]	KTH TIPS2 [5]
No. of publicly available BTF samples	61	~17	6	4	10	11
Raw BTF images resolution [pixels]	640×480	480×360	3032×2008	4500×3000	1280×960	1280×960
Rectified images resolution [pixels]	400×300	192×192	256×256	800×800	≤200×200	≤200×200
Number of view/illum. positions/scales	max.205/55/1	90/120/1	81/81/1	81/81/1	3/3/9	3/4/9
Number of BTF images / material	205	10 800	6561	6561	81	72 or 108
Max. elevation $\theta_i / \theta_v$	85 <sup>0</sup> / 85 <sup>0</sup>	80 <sup>0</sup> / 75 <sup>0</sup>	75 <sup>0</sup> / 75 <sup>0</sup>	75 <sup>0</sup> / 75 <sup>0</sup>	45 <sup>0</sup> / 22.5 <sup>0</sup>	45 <sup>0</sup> / 22.5 <sup>0</sup>
Material sample size [cm]	10×12	< 10 <sup>2</sup>	10×10	10×10	n/a	n/a
Size of rectified BTF dataset in PNG	~100 MB	~700 MB	~700 MB	~5 GB	~7 MB	~7 MB
Rectification accuracy [pixels]	n/a	n/a	~5	~2	no rect.	no rect.
Camera(s) type	video	video	still	still	still	still
Moving [Sample/Camera/Light](DOF)	S(5),C(1)	S(2),L(4)	S(5),C(1)	S(5),C(1)	C(2),L(2)	C(2),L(2)
Raw / rectified data publicly available	yes / no+	yes / yes	no / yes	no / yes	yes / yes	yes / yes
BTF measurement time [hours]	n/a	~ 10	~ 14	~ 14	n/a	n/a
HDR samples	-	-	4	-	-	n/a

TABLE 2  
Comparison of Different BTF Measurement Systems Parameters

parameter	Other BTF measurement systems				
	NewYork03 [39]	Rutgers04 [12]	Bonn05 [70]	KULETH05 [73]	MIT06 [75]
Raw BTF images resolution [pixels]	2048×1536	n/a (principle)	2048×1536	800×600	n/a
Rectified images resolution [pixels]	~200×200	~200×200	1024×1024	460×460	~512×512
Number of view/illum. positions	22-79/22-79	continuous	151/151	264/169	13/13-100
Number of BTF images / material	484 - 6241	continuous	22801	44616	1300
Max. elevation $\theta_i / \theta_v$	76 <sup>0</sup> / 76 <sup>0</sup>	23-37 <sup>0</sup> /23-37 <sup>0</sup>	n/a / n/a	90 <sup>0</sup> / 90 <sup>0</sup>	60 <sup>0</sup> / 60 <sup>0</sup>
Material sample size [cm]	5.8×5.8	1.1×0.8	~10×10	n/a	n/a
Rectification accuracy [pixels]	subpixel	subpixel	subpixel	n/a	n/a
Camera(s) type	still	video	still (151)	video	still
Moving [Sample/Camera/Light](DOF)	none	mirror(2),L-aperture(2)	none	S(1),C(1)	L(3)
BTF measurement time [hours]	n/a	~ 1	~ 1	n/a	~ 1

when choosing the measurement setup since there can be principal limitation connected with some methods [12]. A type of the acquisition sensor also influences the results. While current video cameras allow fast response [12], [73], they cannot deliver resolution and color representation as well as still cameras can [70], [80].

Individual BTF measurements typically suffer from mutual registration problems. Even relatively well-rectified and registered data [80] measured with a moving camera contain registration errors between individual view directions, caused by inaccurate material sample position, self-occlusion, etc. A technique to avoid self-occlusion errors is to employ a separate compression/modeling step for each BTF subset comprehending all images obtained for a fixed view position. Such a BTF slice for a view direction  $\omega_v$  is a 5D function called *Fixed View Reflectance Field*  $\mathcal{R}_v(r_1, r_2, r_3, \theta_i, \phi_i)$ , which describes the radiance of the surface point  $r = (r_1, r_2, r_3)$ , where  $r_1, r_2$  are planar coordinates on a sample and  $r_3$  is the actual spectral band.

We used the BTF measurements of Bonn University [4] as input BTF data for all methods being tested in this paper. Six different BTF materials were used to test individual methods. Each data set comprises 81 viewing positions  $n_v$  and 81 illumination positions  $n_i$  (see Fig. 6) resulting in 6,561 images. Spatial resolution of the rectified original measurements was  $M \times N = 800 \times 800$  pixels.

### 3 DATA REPRESENTATION AND METHODS CATEGORIZATION

The selection of proper representation of BTF data suitable to intended application or modeling method prior to any processing may significantly influence their final performance. Measured BTF data can be either represented as rectified original measurements (Fig. 6 left) or in the form of pixelwise BRDF (Fig. 6 right), i.e.,  $ABRDF_r(\theta_i, \phi_i, \theta_v, \phi_v)$ . This BRDF is often called *apparent* because it can violate any of the two basic BRDF properties, i.e., view and illumination direction reciprocity and energy conservation. This behavior can be caused by shadowing, occlusions, subsurface scattering, and other complex effects occurring in the material structure.

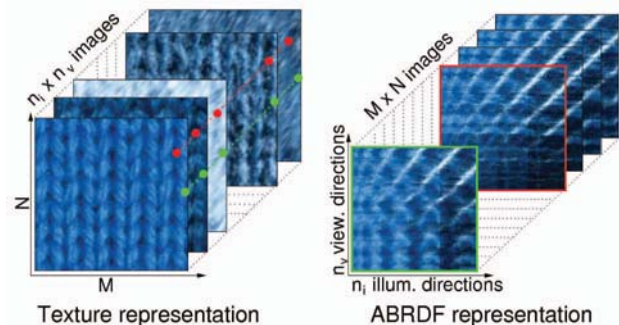


Fig. 6. Two BTF representations illustrated on [80] measurements.

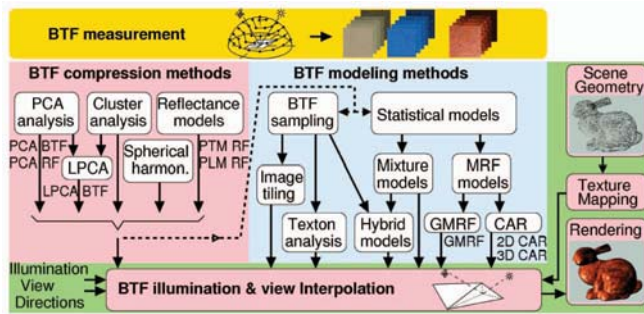


Fig. 7. BTF processing scheme with basic taxonomy of compression and modeling methods.

The first representation enables using methods based on analysis and synthesis of whole planar texture than can be extended to cope with texture appearance or its corresponding parameters change dependently on illumination and viewing conditions. To this category belong sampling-based approaches (Section 5.1) or probabilistic models (Section 5.3).

The second representation (ABRDF) describes in each image dependency of single pixel on illumination/view direction. Here, individual images describe the variance of light/view-dependent reflectance over the measured surface texture. This arrangement produces specularities with lower variance in the images and allows more reliable pixel-to-pixel comparison of images than the previous arrangement, where the cast shadows and variable occlusion effects have to be taken into account prior to any direct comparison. This representation allows us to employ a variety of BRDF-based models (Section 4.1).

On the other hand, linear factorization approaches (e.g., PCA, spherical harmonics) and other general statistical methods can be used, regardless of the BTF representation (Section 4.2).

Surveyed methods using either representation can principally be categorized into compression and modeling approaches, based on their inherent or absent spatial enlargement property. While the compression methods cannot enlarge any BTF measurements by themselves and just create more or less computationally and visually efficient parametrization of the original data, the modeling methods allow unconstrained seamless spatial BTF enlargement to any required size. Apart from this fundamental utilization feature, they automatically, and often significantly, compress BTF measurements.

A basic overview of BTF compression and modeling methods and their mutual relation in the BTF processing pipeline is shown in Fig. 7. Their principles, advantages, and shortcomings are explained in the following sections.

## 4 COMPRESSION METHODS

In contrast to other static planar texture representations, BTF is high-dimensional and massive. To render BTFs on graphics hardware, their compact representation is needed. The best currently publicly available raw BTF samples [80] take up about 5 GB of storage space per material sample and their size can be even greater when saved in high-dynamic range (HDR) data format. Thus, a BTF database even for simple VR scenes can easily reach an enormous data space

range of hundreds of gigabytes; even then, these samples cannot be used in any practical applications due to their small planar size.

Hence, some compression and seamless enlargement (modeling) method of these huge BTF data sets is inevitable. Such a method should provide compact parametric representation and preserve main visual features of the original BTF, while enabling its fast rendering taking advantage of contemporary graphics hardware.

Several methods were published for BTF compression based either on reflectance models, pixelwise BRDF models, or using an approach based on standard principle component analysis (PCA). However, none of these methods enables also texture synthesis (seamless texture enlargement) without additional extension, e.g., with the aid of tiling, spatial clustering, etc. The BTF compression models compared in this section are described in detail in their corresponding sections.

### 4.1 BTF Compression Based on Pixelwise BRDF

The first group of BTF compression methods represents BTF by means of pixelwise analytical BRDF models. McAllister et al. [65] represented the ABRDF of each pixel in BTF using the Lafortune reflectance model [52]. A similar approach, which consists of additional lookup table scaling reflectance lobes and handling shadowing and masking, was published by Daubert et al. [13]. Spatial inconsistency of individual pixels in BTF for different view directions led to separate modeling of individual views (so-called view reflectance fields  $\mathcal{R}_v$ ) in BTF. Malzbender et al. [63] represented each pixel for a given reflectance field of BTF by means of a polynomial.

Homomorphic factorization [66], similar to singular value decomposition (SVD), decomposes pixelwise ABRDF into several factors of lower dimensionality; each factor is dependent on a different interpolated geometric parameter. Compared to SVD, this technique generates a factorization with only positive factors, enables control over smoothness of the result, and works well with scattered, sparse data without a separate resampling and interpolation algorithm. Efficient multiple-term BTF approximation was suggested by Suykens et al. in [83]. This model decomposes ABRDF of each pixel into a product of three or more two-dimensional positive factors using a technique called chained matrix factorization. This technique uses a sequence of matrix decompositions, each in a different parametrization, allowing us to obtain the multiple factor approximation. This decomposition enables easier factor computation than homomorphic factorization [66] and its factors have lower dynamic range, so their quantization into 8 bits for real-time rendering is much safer. A novel technique for BTF representation was proposed by Ma et al. [60]. Their approach is based on fitting the Phong model to pixelwise ABRDF. The model's parameters are then averaged and the difference between original data and results of the Phong model, so-called spatial-varying residual function, is approximated by a delta function whose parameters are obtained from a system of linear equations. This approach allows good approximation quality and interactive BTF rendering frame rates.

Meseth et al. [67] represented BTF by several pixelwise Lafortune lobes for fixed viewing direction. Due to the expensive nonlinear fitting of its parameters, the number of

Lafortune lobes is practically limited to three lobes. The lobes are only used for luminance values fitting, which modulates an albedo map of individual color channels. This arrangement reduces the number of parameters to be stored, but simultaneously deteriorates approximation accuracy. In [23], only one lobe is used per color channel. The obtained results are then corrected by means of polynomials representing histogram matching functions between original and restored images.

In [62], a BTF compression method is introduced that separates geometric information from the reflectance data combining a layered volumetric model of material structure and the Lafortune reflectance model. The pixelwise surface normal vector, reflectance model, and light attenuation parameters are computed for individual layers separately. An advantage of the method is a high compression ratio and easy interpolation of BTF data, the number of layers and height of the material have to be set explicitly.

A method for intuitive editing of SVBRDF, i.e., tolerable BTF approximation for flat and opaque materials, was presented in [53]. This method is based on BRDF decomposition into a compact tree structure and allows editing of both reflectance properties specified by decomposed BRDF and spatial distribution of individual BRDFs over the material surface. Advanced interactive editing of SVBRDF was presented in [77], based on the number of user-defined editing constraints that are smoothly propagated to the entire data set performing similar editing effects in areas of similar appearance. An SVBRDF model based on pixelwise Ward reflectance model effectively handling directional, anisotropic reflections of subsurface fibers to preserve an appearance of wooden materials is proposed in [64]. These methods are limited only to flat and opaque materials that can be represented by means of SVBRDF and cannot be used for realistic representation of any real-world materials.

An approximation of BTF data by means of a shading map indexed by a Phong-type BRDF model is presented in [48]. The shading map is acquired as a set of material images for a fixed viewing direction and a changing elevation of illumination direction. During rendering, for a given illumination and viewing direction, the BRDF model is evaluated and, from the shading map, an image of the most similar average value is used as a pixel value for a given planar position. The authors presented also the shading map compression based on power functions representing individual illumination-dependent pixels. This technique provides reasonable results for small-scale structured and isotropic materials, but cannot reliably represent the masking effects caused by a rough material structure.

#### 4.1.1 Polynomial Texture Maps (PTM RF)

In the Polynomial Texture Maps approach [63], the BTF images corresponding to a fixed view direction are approximated by means of per-pixel polynomials. This method models illumination dependence of individual pixels using the following pixelwise biquadratic formula

$$\begin{aligned} \mathcal{R}_v(r, i) \approx & a_0(r)u_x^2 + a_1(r)u_y^2 + a_2(r)u_xu_y \\ & + a_3(r)u_x + a_4(r)u_y + a_5(r), \end{aligned} \quad (2)$$

where  $u_x, u_y$  are projections of the normalized light vector into the local coordinate system  $r = (x, y)$ . The set of  $n_i$  pixels

is considered as reflectance data, where  $i = 1, \dots, n_i$  is the illumination position index and  $v$  is the actual view position index  $v = 1, \dots, n_v$ . The  $n_p = 6$  polynomial coefficients  $a_0 - a_5$  are fitted in each pixel by means of SVD.

This method enables very fast rendering. However, it assumes that the modeled surfaces are either diffuse or their specular contribution had been separated in the previous preprocessing step. This separation can be quite difficult for reflectance fields obtained as a BTF slice. For such a reflectance field, the method exhibits considerable errors mainly for high grazing angles as shown in [67]. For BTF rendering, this method requires six parametric images to be stored per reflectance field  $\mathcal{R}_v$  and color channel.

#### 4.1.2 Polynomial Extension of Lafortune Reflectance Model (PLM RF)

Single surface reflectance field for a given reflectance field can be per-pixel modeled using the generalization of the one-lobe Lafortune model (LM) [52]:

$$Y_v(r, i) \approx \rho_v(r)[a_{v,1}(r)u_1 + a_{v,2}(r)u_2 + a_{v,3}(r)u_3]^{n_v(r)}, \quad (3)$$

where  $\omega_i(\theta_i, \phi_i) = [u_1, u_2, u_3]^T$  is a unit vector pointing to light and parameterized by the illumination elevation and azimuthal angles  $[\theta_i, \phi_i]$ , respectively, (see Fig. 3). For every planar position and spectral channel in BTF, the model parameters  $(\rho, a_1, a_2, a_3, n)$  are estimated using  $t = 2$  iterations of the Levenberg-Marquardt nonlinear optimization algorithm, whose performance strongly depends on chosen initial values. Unfortunately, reflectance values that are clearly and completely wrong result from the one-lobe LM model for certain combinations of illumination and viewing angles. The polynomial extension of one-lobe Lafortune model (3) (PLM RF) is proposed in [23], [24], which leads to the following formula:

$$\mathcal{R}_v(r, i) \approx \sum_{j=1}^{n_p} a_{v,i,j} Y_v(r, i)^j, \quad (4)$$

where  $a_{v,i,j}$  are polynomial parameters specifying the mapping function between cumulative histogram values of image  $\hat{Y}_{i,v}$  synthesized from one-lobe LMs parameters and the original BTF image;  $(n_p - 1)$  is a rank of this polynomial. For BTF rendering, this method requires  $n_p = 5$  parametric images to be stored per  $\mathcal{R}_v$  and a color channel with an additional 15 polynomial coefficients per BTF image.

## 4.2 BTF Compression Based on Linear Factorization Methods

The second group of BTF compression methods is based on linear basis decomposition methods such as PCA or spherical harmonics.

Koudelka et al. [50] ordered individual BTF images into vectors forming a matrix. The corresponding symmetric matrix was created and subsequently decomposed using SVD. The authors preserved 150 main eigenimages for a satisfactory BTF reconstruction. Vasilescu and Terzopoulos [86] decomposed the BTF space, ordered into a 3D tensor, by means of multimodal SVD. This method enables controllable BTF compression separately in viewing and illumination axes and demonstrates better performance than the previous approach using same number of components. Wang et al. [87] further extended this idea.

Instead of using a 3D texel-illumination-view tensor, it stores BTF data directly in a 4D form, i.e., also preserving spatial relationships in individual BTF images. This helps to significantly decrease the reconstruction error while maintaining the same level of compression as in the previous approach. Although, these methods enable realistic BTF rendering, they are not suitable for a fast BTF rendering application since they require the user to compute linear combinations of high number of eigencomponents. A much faster approach, applying SVD only on images of separate view reflectance fields, was presented by Sattler et al. [80].

Another method [42] uses blockwise PCA for scene illumination dependency coding. The coding is performed in  $Y-C_r-C_b$  color space and the resulting eigenimages are further compressed using a combination of cosine transformation and quantization techniques.

The method described in [93] compresses pixelwise illumination and view-dependent data by means of spherical harmonics using up to 25 coefficients. The coefficient planes are further coded using a discrete wavelet transformation and the method exploits  $Y-C_r-C_b$  color space, which allows an even higher color compression. The authors report better visual results and compression ratios on image and video data than with standard compression methods. A very similar approach, applying a radial basis functions instead of spherical harmonics for pixelwise compression was introduced in [56].

Ma et al. [61] presented a method (similar to [59]) for level-of-details representation of BTF aimed at real-time rendering. This method is based on BTF data decomposition by means of a Laplacian pyramid. BRDF vectors corresponding to BTF at a fixed planar position at individual pyramid levels are further approximated by PCA. The method enables significant BTF compression and real-time rendering. The authors computed PCA for individual reflectance fields instead of the whole BTF data space. This approach resulted in 16 eigenimages per one view position, which can easily be interpolated by means of graphics hardware. Müller et al. [71] exploited a vector quantization of BTF data space and each resulting cluster was represented by a local PCA model. Some of these compression methods are compared in [69].

An approach to generating a full BTF from its sparse sampling based on a clustering of underlying surface geometry was presented by Wang and Dana [88]. This technique estimates a set of geometric texon patches from example surfaces. These patches are then used for geometry synthesis of arbitrary view and illumination conditions and the result is blended with results of the eigenanalysis method. The method correctly preserves casted shadows in surface mesostructure, but cannot enlarge original BTF data.

While the above-mentioned methods do not solve the BTF synthesis problem, these methods are all capable of compressing the measured BTF space.

#### 4.2.1 Reflectance Field Factorization (PCA RF)

Reflectance Field Factorization [80] is based on computation of no more than  $n_c$  principal components per individual reflectance field instead of the whole BTF space. Individual images corresponding to reflectance field  $\mathcal{R}_v$  are used as  $\mathbf{A}$  matrix input vectors. From matrix  $\mathbf{A}\mathbf{A}^T$  of size  $n_i \times n_i$ , the eigenimages  $E_{v,k}$  are computed by means of SVD for each  $\mathcal{R}_v$  together with the corresponding weights  $\alpha_{v,k}$  and

mean image  $\mu$ . The reconstruction formula for a reflectance field is

$$\mathcal{R}_v(r, i) \approx \sum_{k=1}^{n_c} \alpha_{v,k}(i) E_{v,k}(r) + \mu(r). \quad (5)$$

For the following tests, the number of components  $n_c$  for individual samples was estimated by the psychophysical experiment, so  $n_c + 1$  parametric planes have to be stored per  $\mathcal{R}_v$ .

#### 4.2.2 BTF Space Global Factorization (PCA BTF)

In a PCA-based BTF factorization approach, Koudelka et al. [50] arranged individual color pixels of BTF images of size  $M \times N$  in vectors forming matrix  $\mathbf{A}$  of size  $3MN \times n_v n_i$ . The principal components are the eigenvectors  $E_k$  of the symmetric matrix  $\mathbf{A}\mathbf{A}^T$  of size  $n_i n_v \times n_i n_v$ . However, the  $\mathbf{A}\mathbf{A}^T$  computational time for larger BTF images can be unacceptable unless using advanced incremental approximate techniques. Computing the eigenvectors for spatially nonhomogeneous materials (large samples) often takes several days. BTF reconstruction is similar to a previous method stated by the following equation:

$$BTF(r, i, v) \approx \sum_{k=1}^{n_c} \alpha_k(i, v) E_k(r) + \mu(r). \quad (6)$$

To obtain satisfactory BTF approximation results, the number of preserved eigenimages  $n_c$  was again set by the psychophysical experiment. The entire BTF space is thus represented by  $n_c + 1$  parametric planes.

#### 4.2.3 BTF Space Local Factorization (LPCA BTF)

A BTF compression method well suited to contemporary graphics hardware was presented by Müller et al. in [71]. This method exploits the fact that high-dimensional data sets, in this case BTF, show a locally linear behavior. The authors propose a BTF compression algorithm based on a combination of iterative vector quantization and local PCA computed in individual clusters in BTF data. The BTF space is iteratively divided into clusters using modified K-means algorithm in the planar BTF space ( $t$  denotes the number of iterations). The squared eigenimage reconstruction error is used as a distance measure in the clustering process. Each cluster is represented by means of local PCA in the form of several eigenvectors dependent on illumination and viewing position. The described BTF factorization can be stated as

$$BTF(r, i, v) \approx \sum_{k=1}^{n_c} \alpha_{m(r),k}(r) E_{m(r),k}(i, v) + \mu_{m(r)}, \quad (7)$$

where  $m(r)$  is a cluster index lookup table given by planar coordinates  $r = (x, y)$ ,  $n_c$  is the number of preserved principal components representing each cluster,  $\alpha_k$  are PCA weights,  $E_k$  are saved eigenvectors, and  $\mu_{m(r)}$  is the mean vector for the given cluster  $m(r)$ . The entire BTF reconstruction together with the illumination and view interpolation can be implemented in graphics hardware, which enables fast BTF rendering. This method provides high BTF compression while ensuring high reconstruction quality and rendering speed [69]. For the following tests, the number of clusters  $c$  and number of components per each cluster on  $n_c$  were set by the psychophysical



experiment. For whole BTF space representation,  $c$  cluster index images are stored together with  $n_c + 1$  eigenvectors of size  $n_i n_v$  and  $n_c$  coefficient matrices of size  $n_c \times \dim c_i$  for each cluster  $i$ .

## 5 MODELING METHODS

BTF modeling methods allow seamless enlargement of BTF measurements to any size required by an application as well as the reconstruction/estimation of unmeasured parts of the BTF space. These methods can be divided into three major groups: sampling-based, reflectance-model-based, and adaptive-probabilistic-model-based methods.

### 5.1 Sampling Methods

Sampling methods, which are characteristic for computer graphics applications, are based either on simple texture repetition with edge blending or on more or less sophisticated image tiling methods [6], [18], [51], [81], [35] and some of them are suitable for [55] or can be adapted to BTF synthesis, e.g., [16], [35], [81]. The most successful sampling approaches [14], [18], [19], [41], [94] rely on sophisticated sampling from real texture measurements, which have to be stored in the texture database. The paper by Dong and Chantler [16] presents a survey of several sampling-based BTF synthesis approaches. Based on the amount of copied data, the sampling approaches can be divided into the per-pixel nonparametric sampling [19], [84], [91], [97] and the patch-based sampling [35], [36], [49], [58], [99], [95]. Given a randomly selected starting block of texture in the image, they propagate out from it selecting new texture blocks. For each new block in the image, all neighboring blocks that have already been generated are checked and the example image (or images) is searched for similar textures. The  $n$  best such matches are found and then the corresponding new texture patch is randomly chosen from among them. The methods [18], [19], [91] all vary in the way the blocks are represented, how similarity is determined, and how the search is performed.

A method similar to [89], combining a sparse set of BTF measurements according to an enlarged material range map using the [19] algorithm to generate dense BTF data was developed by Liu et al. [58]. It starts with BTF sample range map estimation using the shape-from-shading method. The enlarged range map is used to guide a blockwise sampling from BTF measurements. The authors tested the method performance on CURET data [10] only. This method is slow, overlapping blocks can potentially generate visible seams, mutual separation of analytical and synthesis parts is not possible, and its data compression is negligible.

A modification of this method similar to [71] appeared in [59]. This method exploits technique of 3D textons, i.e., the smallest repeatable texture elements, introduced in [57]. Only these textons are then approximated using local PCA and, finally, used for surface modeling.

The pyramid matching synthesis [41] was generalized [75] for sparsely sampled BTF data, but the visual quality of synthesis results restricts this method to textures without strong spatial characteristics.

The algorithm [84] performs BTF synthesis based on surface textons, which extract essential information from the sample BTF to facilitate the synthesis. A 3D texton set is constructed using the [57] method (BTF space clustering)

and single BTF pixels are assigned texton labels. The paper uses a general search strategy, called the k-coherent search, for constructing a neighbor candidate set. The method is extremely slow and was tested only on low-resolution CURET data [10]. Another sampling-based BTF synthesis method was published by Neubeck et al. [74]. The authors apply smart copy-and-paste smooth texture synthesis to BTF synthesis. The sampling is restricted to similar neighborhoods by introducing a reasonable subset of possible candidates (using the Ashikhmin's candidate search [1]) from the example image. This algorithm is iterative and slow, it is restricted to small size neighborhoods, it might blur the resulting texture, and analysis and synthesis cannot be separated from each other.

A generalization of the image quilting method [18] for BTF data PCA-compressed spherical harmonics expansion was presented in [49]. This method maintains all disadvantages of the original image quilting method, most of all in its slowness due to unseparated analytical and synthetic parts. The image quilting method was also used in an interactive application [99] allowing the user to paint BTF patches onto the surface such that the painted patches seamlessly integrate with the background patterns. This allows introduction of imperfections and other irregular features into the BTF surface. However, this method is extremely slow; it needs 20 minutes for synthesis of a small texture.

The BTF roller synthesis method [35], [36] is based on the fully automatic detection of one or several optimal double toroidal BTF patches per fixed view angle. These BTF patches are seamlessly repeated during the synthesis step. While the method allows only moderate texture compression, it is extremely fast due to complete separation of the analytical step of the algorithm from the texture synthesis part, which has negligible computation complexity. The method is easily implementable in graphical hardware for the purpose of real-time rendering of any type of static textures.

In [55], BTF tiling method based on Wang tiles [6] is proposed. The method cuts the tiles in spherical harmonics BTF representation and allows real-time rendering on an arbitrary surface. The method also allows users to interactively edit the created BTF tiles.

All of these methods are based on some sort of original spatial sampling of texture data or its pixelwise parameters and the best of them produce very realistic synthetic textures. However, these methods require storage of the original or transformed measurements (often thousands of images corresponding to measured combination of viewing and illumination angles of the original target texture sample), they often produce visible seams, some of them are computationally demanding, and cannot generate textures unseen by the algorithm. Obviously, all texture sampling techniques described in this section may be principally applied for spatial extension of BTF data or their parametric representation; however, their computational costs may vary significantly and only a few of them can perform texture rendering or relighting in real time.

### 5.2 Spatial Enlargement of BTF Reflectance Models

BTF reflectance models are pixelwise generalizations of BRDF compression models, and as such they represent a compact representation/compression of BTF measurements only. However, they can possibly be extended with the aid

of a parametric space modeling method to allow BTF spatial enlargement.

A BTF synthesis approach based on combination of image tiling and a pixelwise reflectance model was introduced in [95]. This approach involves BTF compression based on polynomial texture maps [63]. Estimated resulting parametric images containing polynomial coefficients are subsequently enlarged by means of the Efros image quilting algorithm [18].

In [16], a survey of several BTF compression approaches is presented. The authors have tested an image-based relighting method [17] based on BTF image reconstruction from several known BTF images according to Lambertian reflectance function, overdetermined photometric stereo based on SVD of 36 images, polynomial texture maps [63], and, finally, PCA analysis of all BTF images. BTF enlargement in all of these methods is accomplished again by means of the tiling algorithm [18].

The polynomially extended Lafortune reflectance model (PLM RF) [23], [24] was completed with the tiling method [81] applied to its parametric planes, which enables arbitrary and high-quality enlargement of BTF measurements.

### 5.3 Probabilistic Models

Texture synthesis based on probabilistic models [2], [3], [27], [28], [34], [37], [46], [100] requires no trifling multidimensional models (from 3D for static color textures up to 7D for static BTFs). If such an  $n$ D texture space can be factorized, then these data can be modeled using a set of lower dimensional (e.g.,  $(n - 1)$ D) random field models, but in any case, such models are uncommon and suffer from several unsolved theoretical problems, which have to be circumvented.

Unfortunately, real data space can be decorrelated only approximately; hence, the independent spectral component modeling approach causes a loss of image information. Alternative full  $n$ D models allow unrestricted spatial-spectral correlation modeling, but their main drawback is a large amount of parameters to be estimated and, in the case of Markov random field (MRF) models, also the necessity of estimating all of these parameters simultaneously. Model-based methods published so far are mostly too difficult to implement in current graphics hardware.

Gaussian mixtures (or their neural networks equivalent, Radial Basis Function) were used for monospectral texture synthesis [98]. Although they are able to model nonlinear spatial interactions, their parameter estimation and synthesis require computationally demanding numerical methods—the EM algorithm and Markov Chain Monte Carlo methods. Discrete distribution mixtures of product components applied to color texture synthesis (with straightforward generalization to BTF) were proposed in [27]. The texture synthesis is based on an easy computation of arbitrary conditional distributions from the model, however, the model requires a large training data set, powerful computing resources, and its data compression is much lower than that of the subsequent models.

Methods based on different Markov random fields [31], [29], [32], [30], combine an estimated range map with synthetic multiscale smooth texture. These methods (except [32]) estimate a BTF texture's range map followed by the spectral and spatial factorization of selected BTF texture images. Due to the stochastic nature of MRF models, they

do not reproduce well regular or near-regular structures in BTF samples; hence, this regular information was introduced into them by means of combination of synthesized spectral data with a relighted range map. The range map is estimated using the overdetermined photometric stereo from mutually aligned BTF images. The overall BTF texture visual appearance during changes of viewing and illumination conditions is simulated using either bump or displacement mapping technique. The next step of these methods is BTF illumination/view  $(\theta_i, \phi_i/\theta_v, \phi_v)$  space segmentation into  $c$  subspace images (the closest BTF images to cluster centers) using the K-means algorithm. Eigenanalysis of BTF data has shown that  $c = 20$  is sufficient to represent its reflectance correctly for most of the samples. The color cumulative histograms of individual BTF images, in perceptually uniform CIE Lab color space, are used as the data features. These subspace images are then spectrally [29], [30] and spatially [29], [30], [32] decomposed into band-limited monospectral factors, which are independently modeled by their dedicated 2D ([29], [30]) or 3D MRF ([32]) models.

All statistics in the models are solved analytically in the form of robust and numerically efficient Bayesian estimators resulting in a very compact set of parameters. Single band-limited factors (monospectral or multispectral) are subsequently synthesized using this compact parametric set and interpolated into fine resolution, smooth texture images. Finally, the required visual appearance of BTF is created by combining both multispectral and range information in a bump mapping or a displacement mapping filter of the rendering hardware.

#### 5.3.1 Gaussian Markov Random Field (GMRF) Model

This method [29] models the BTF subspace images by a set of dedicated 2D GMRF models and performs spectral decorrelation of individual subspace images using Karhunen-Loeve (KL) transformation. The resulting monospectral factors are further spatially decomposed by means of a Gaussian-Laplacian (GL) pyramid with  $p$  levels. Individual sub-band factors are analyzed using a GMRF model, which can be expressed as a stationary noncausal correlated noise-driven 2D autoregressive (AR) process on image grid:

$$Y_r = \gamma X_r + e_r, \quad (8)$$

where  $\gamma$  is the parameter vector,  $X_r$  is the corresponding data vector  $Y_{r-s}$  containing data from a symmetric contextual neighborhood (CN) of dimensionality  $n_p$ , and  $e_r$  is a random variable with zero mean and a constant but unknown variance  $\sigma^2$ . If individual pixel values in CN are assumed to be conditionally independent, the parameters  $\gamma$  and  $\sigma^2$  can be approximated analytically. The toroidal image lattice is assumed to enable fast subspace factor synthesis from model parameters using inverse fast Fourier transformation (FFT). In the remaining part of subspace image synthesis, the monospectral factors are obtained by the GL pyramid collapse and inverse KL transformation whose matrix has to be stored together with GMRF model parameters. The analysis and synthesis of BTF data space using this method is very fast, however, use of FFT somewhat restricts this method's hardware implementation.

### 5.3.2 2D Causal Autoregressive (2D CAR) Model

This method [30], [31] shares a similar processing pipeline as the GMRF model. However, the method uses 2D CAR model, which can be described as a stationary causal uncorrelated noise-driven 2D AR process:

$$Y_r = \gamma X_r + e_r. \quad (9)$$

Although the meaning of the above notation is the same as in the previous GMRF model, all parameters can be estimated without simplifying approximations,  $e_r$  is contrary to (8) mutually uncorrelated and CN is restricted to either causal or unilateral, i.e., all support pixel values are known with respect to movement on the image grid. Contrary to the previous model, the parameters  $\gamma$  and  $\sigma^2$  can be precisely estimated analytically and the synthesis is extremely fast by means of subsequent application of (9) on the image grid while using estimated parameters  $\gamma$  and a white noise generator with variance  $\sigma^2$ . The remaining parts of the synthesis, i.e., spectral and spatial factorization are the same as in the GRMF model.

### 5.3.3 3D Causal Autoregressive (3D CAR) Model

This MRF-based BTF subspace modeling method [32] avoids spectral decorrelation errors due to approximate BTF spectral space decorrelation. The 3D CAR model is able to represent all spectral correlations between individual subspace images. Thus, the method starts directly with building of the GL pyramid. The model can be expressed as a stationary causal uncorrelated noise-driven 3D AR process:

$$Y_r = \Theta X_r + E_r, \quad (10)$$

where the CN is restricted to be causal or unilateral,  $\Theta$  is the parameter matrix, and  $E_r$  is a Gaussian white noise vector with zero mean and a constant but unknown covariance matrix  $\Sigma$ .

The parameters  $\Theta$  and  $\Sigma$  are estimated analytically and the synthesis is, for an arbitrary image size, again performed by subsequent application of (10) on sub-band images' grid. The synthesized subspace images are obtained by interpolation of GL pyramid levels. The synthesis using this model is very fast. However, the simultaneous interpolation of all  $3 \times c$  subspace planes is more time-consuming and reduces the speed of fast hardware implementation.

Methods of Markov random field type are based on the estimated model in contrast to methods of prevailing intelligent sampling type, and as such they can only approximate realism of the original measurement. However, they offer an unbeatable data compression ratio (tens of parameters per texture only), easy simulation of even previously not measured BTF images, and fast seamless synthesis of any texture size.

## 5.4 Hybrid Methods

A hybrid method of color texture modeling based on Gaussian distribution mixtures (GM) was proposed [34] with the aim to combine advantages of both approaches (sampling and probabilistic modeling) to basic texture modeling. The hybrid model can be either used to directly synthesize color textures or to control sophisticated sampling from the original measurement data. In the latter option, the method can be viewed as a statistically controlled

sampling. It allows high visual quality of synthetic textures while requiring storage of only small patches of the original measurements or even only Gaussian-mixture parameters in the direct modeling version.

A generalization of the Gaussian distribution mixture-based method to BTF modeling is discussed in [33]. This method estimates local statistical properties of the mono-spectral version of a fixed view target BTF texture in the form of GM of product components. The synthesized texture is obtained by means of a stepwise prediction of the whole fixed view BTF texture subspace. In order to achieve an authentic BTF texture and avoid possible loss of high-frequency spatial details, optimally chosen pieces of the original BTF measurements are chosen in the synthesis phase. Thus, this BTF modeling method can be viewed as a statistically controlled sampling. This method allows moderate texture compression, high visual quality, synthesis of arbitrary large seamless texture, and fast synthesis, but its drawback is time-consuming analysis and difficult GPU implementation. An important aspect of the proposed approach is its possible extension to multispectral or mutually registered BTF texture images.

The next method [25] performs BTF data clustering in a spatial domain. Individual clusters (ABRDFs) are stored and their spatial mapping index/image is enlarged to an arbitrary size by means of 2D CAR synthesis of pixelwise normal vectors estimated using photometric stereo. This technique allows real-time BTF rendering and compression of about 1:300.

## 6 MODELING QUALITY CRITERIA

Verification of BTF data modeling quality is a difficult and still unsolved problem due to the lack of existing mathematical criteria capable of approximating the human eye's perception of textures. Modeling methods directly approximating single pixels in their original location (reflectance models without enlargement, PCA-based compression) can be verified using either similar criteria to those used in image restoration applications (e.g.,  $L_1, L_2$  norms) or using model of low-level human vision [9]. However, stochastic models do not produce an exact pixelwise copy of an original texture, but are intended to preserve the major statistical properties of the original BTF data. The quality of this representation depends on a chosen model type, its initial parameters, the support set shape and size, direction of the image lattice movement, etc. For this reason, any differential metrics based on pixelwise image comparison between original and estimated texture images do not make any sense. Unfortunately, no robust criterion for visual similarity exists. There have been several attempts at defining texture similarity metrics, e.g., the work of Julez [43], who suggested a similarity measure based on the second-order statistical moments. However, this promising method was questioned later by the same author in [44], [45] since many counterexamples have been shown, showing failures of the proposed similarity measure. Another method based on the same assumption but using third-order statistics was introduced in [96]. Although this method seems to be more robust, it can only decide whether two texture images are identical or not. This method does not provide any similarity measure. So it is

clear that we are still missing an approach providing an acceptable and applicable measure of texture similarity.

Currently, the only reliable way is to compare the overall visual similarity of two textures by independent observers in a psychophysical experiment. The first published psychophysical experiment using BTF data was conducted in [68], where the authors compared environmentally lit renderings of BTF [71], flat textures modulated by the Phong BRDF model, and photographs of a car interior scene. The image sets from these three techniques were the subject of a psychophysical study with the group of 22 participants. The authors concluded that most participants considered the BTF model as identical to the photographs while the BRDF representation scored worse. Another experiment with 11 subjects in [22] studied influence of various uniform BTF data resampling schemes on perceptual appearance of eight BTF samples. It has been shown that different materials require different sampling, generally downsampling of azimuthal angles  $\varphi$  should be preferred instead of elevation angles  $\theta$ , and that illumination direction may be sampled less densely than viewing direction. In [21] was introduced a psychophysically validated metric for automatic BTF sample size reduction based on vector quantization of BTF images controlled by their mean variance.

In the following section, we performed psychophysical experiment to determine optimal parameter settings of the relevant tested compression methods, to obtain visually indiscernible results.

## 7 SELECTED METHODS COMPARISON

We compared nine different BTF modeling methods. The categorization of the methods is shown in the overview scheme in Fig. 7 below the corresponding category blocks. The first method [81] provides tiling of the original BTF data. The next five methods are based on pixelwise modeling. The first three of them (Polynomial Texture Maps (PTM RF) in Section 4.1.1, Polynomial Extension of Lafortune Reflectance Model (PLM RF) in Section 4.1.2, and Reflectance Field Factorization (PCA RF) in Section 4.2.1) model BTF data for individual surface reflectance fields separately. The remaining two methods model the whole BTF space at once (BTF Space Global Factorization (PCA BTF) in Section 4.2.2 and BTF Space Local Factorization (LPCA BTF) in Section 4.2.3). The remaining group of three methods is based on probabilistic modeling (2D GMRF Model in Section 5.3.1, 2D CAR Model in Section 5.3.2, and 3D CAR Model in Section 5.3.3).

All of the above-described methods were compared to each other in terms of objective and subjective visual errors, storage requirements for their parametric representation, analysis and synthesis time, and computational complexity.

All of the surveyed methods were tested on the Bonn University BTF data set [80]. For considerable reduction of the size of parametric representation of the tested pixelwise methods and simultaneously for enabling seamless covering of arbitrarily large virtual objects, an image tiling approach was applied. The approach [81] finds suboptimal paths in the original data to cut the required set of arbitrarily contactable BTF tiles. The size of tiles  $n_r \times n_c$  (see Table 7) depends strongly on the type of the underlying materials' structure, regularity, etc. All of the pixelwise BTF

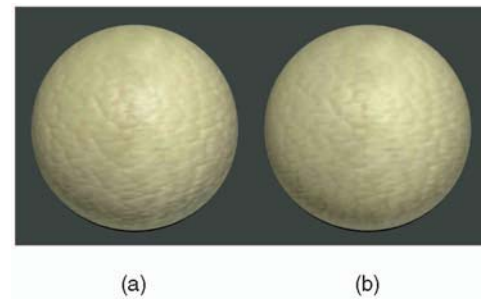


Fig. 8. Example of stimulus showing (a) original and (b) improperly parametrized sample *synthetic leather*.

models compared in this paper were further applied only on these BTF tiles. Six different BTF samples were tested: *knitted wool*, *fabric dark*, *fabric light*, *synthetic leather*, *leather*, and *lacquered wood* (see Fig. 11).

### 7.1 Psychophysical Experiment

For fair comparison of the pixelwise modeling methods, we performed psychophysical experiment. The goal of the experiment was to determine optimal methods' parameter settings in order to achieve a visual appearance indistinguishable from the original BTF measurements. As the first two methods (PTM RF and PLM RF) do not allow straightforward change of parameters, we were able to control visual appearance by changing the parameters only for the remaining PCA-based methods (PCA RF, PCA BTF, and LPCA BTF).

#### 7.1.1 Experimental Data

As experimental stimuli, we used pairs of static images of size  $800 \times 800$  pixels showing BTF rendered on a sphere for point-light positioned slightly above a camera. Each pair consisted of a rendering using the original BTF data set and one using its model in random order. For different models, we used different parameter quantization to obtain a subjectively similar range of visual degradation. The PCA RF method was used with the following numbers of principal components per each view direction: 2, 4, 6, 8, 10, and 12. For the PCA BTF method, the quantization of principal components representing the whole BTF was chosen as 10, 20, 30, 40, 50, and 60. And finally, for LPCA BTF, the same parameter per cluster was quantized to 5, 8, 11, 14, 17, and 20. Moreover, the number of clusters in LPCA BTF method was chosen according to the recommendation of the authors [71], i.e., 32 clusters per BTF size  $256 \times 256$  pixels. This number of clusters was recomputed for individual tested samples, respectively, depending on the tile size (i.e., *knitted wool* 3, *fabric dark* 2, *fabric light* 3, *synth. leather* 6, *leather* 8, and *lacquered wood* 19). In addition to these three methods, we also added to the experimental stimuli pairs containing renderings of the methods PTM RF, PLM RF, and original-to-original data. The described configuration resulted in 156 stimuli. The background of the stimuli was set to dark gray. An example stimulus is shown in Fig. 8.

#### 7.1.2 Participants

Twenty-two observers in two countries participated in the experiment. All were either postgraduate students or academic employees working on different research fields.

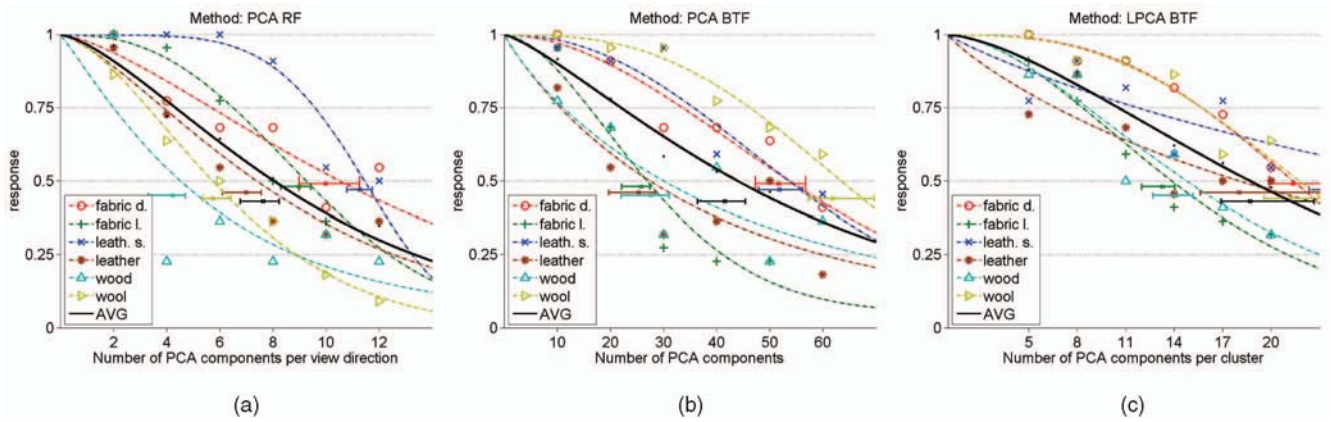


Fig. 9. Fitted psychometric functions to data obtained from the psychophysical experiment for six different BTF samples and three different compression methods (a) PCA RF, (b) PCA BTF, and (c) LPCA BTF.

All had normal or corrected to normal vision and all of them were naive with respect to the purpose and design of the experiment.

7.1.3 Experimental Procedure

Each participant was presented 156 stimuli in random order and asked a yes-no question: *Can you detect any difference in the texture covering the objects?* Participants were given as much time as they needed for their decision. There was a one-second pause between the stimuli and the average participant finished the whole experiment in 30 minutes. All stimuli were presented on calibrated 20.1" LCD displays NEC 2090UXi and NEC 2170Nx (60 Hz, resolution 1,600 × 1,200, color temperature 6,500 K, gamma 2.2, and luminance 120 cd/m<sup>2</sup>). The experiment was performed in controlled dim office lighting and participants were seated at 0.8 m from the display and each sphere in the stimulus occupied approximately 10 degree of their visual angle.

7.1.4 Fitting the Psychometric Data

When participants reported a difference between the rendered images, their response was assigned value of 1, and otherwise 0. By averaging the responses of all participants, we obtained psychometric data relating average response to variable parameter of BTF model. There are six such data sets (one for each tested sample) for each tested method (PCA RF, PCA BTF, LPCA BTF).

The obtained psychophysical data can be represented by psychometric function  $\psi(x)$  [92], which specifies the relationship between the underlying probability  $\psi$  of positive response and the stimulus intensity  $x$ :

$$\psi(x; \alpha, \beta, \gamma, \lambda) = \gamma + (1 - \gamma - \lambda)F(x; \alpha, \beta), \quad (11)$$

where  $F$  is a function with parameters  $(\alpha, \beta)$  fitting the data,  $\gamma$  specifies the guess rate (i.e., response to zero stimulus), and the  $\lambda$  miss rate (i.e., incorrect response for large stimulus).

Psychometric functions were fitted to the measured data using the *psignifit* package [92], based on bootstrap Monte Carlo resampling technique for confidence interval estimation of data fitting. As  $F$  we have used Weibull cumulative distribution, which is most commonly used in life data analysis due to its flexibility

$$F(x, \alpha, \beta) = 1 - \exp \left[ -\left(\frac{x}{\alpha}\right)^\beta \right], \quad (12)$$

for  $x \geq 0$ , where  $\beta > 0$  is the shape parameter and  $\alpha > 0$  is the scale parameter of the distribution.

The resulting fitted psychometric functions with original data points for all three tested methods are shown in Fig. 9. The graphs also include estimated fitting confidence intervals of individual functions at a response level 0.5. The function averaging the data over of all samples is shown as a solid black outline.

7.1.5 Results

To estimate the models' parameters giving a visual appearance indiscernible from original BTF renderings, we used the value of the parameter at which a difference between rendered images is detected by 50 percent of observers. Parameter value  $k$  can be estimated using

$$k_{p=0.5} = \alpha \sqrt[\beta]{\ln \left( \frac{1 - \gamma - \lambda}{1 - 0.5 - \lambda} \right)}, \quad (13)$$

where  $\alpha, \beta$  are estimated parameters of the Weibull distribution and  $\gamma$  and  $\lambda$  are estimated guess and miss rates, respectively. The estimated parameter values for all of the tested methods, samples and the average values are summarized in Table 3. These values should guarantee the same visual appearance of the renderings using the tested methods as those using original BTF data. These values for individual samples were used throughout the following section comparing efficiency of individual methods. The results in the table confirm the assumption that different BTF samples require dedicated settings of the tested

TABLE 3  
Estimated Numbers of PCA Components for Six Different ZBTF Samples with Their Average and Three Different Tested Compression Methods

method	Optimal No. of PCA components $k$						AVG
	knitted wool	fabric dark	fabric light	synth. leather	leath. lacq. wood		
PCA RF	6	10	9	11	7	4	8
PCA BTF	61	51	26	52	25	28	41
LPCA BTF	21	21	13	29	18	14	19

TABLE 4  
Observers Responses on PTM RF and PLM RF Methods  
and Guess Rates  $\gamma$  for All Tested BTF Samples

method	Response for stimuli $\langle 0, 1 \rangle$ and the guess rate $\gamma$						AVG
	knitted fabric		fabric		synth. leather	lacq. wood	
	wool	dark	light	leather			
PTM RF	0.86	1.00	1.00	0.95	0.95	1.00	0.96
PLM RF	0.27	0.73	0.68	0.91	0.91	0.68	0.70
$\gamma$	0.09	0.09	0.32	0.14	0.14	0.14	0.15

method to provide results visually indiscernible from the original data. This fact is justified by distinct underlying structure and surface roughness of the tested samples.

The remaining tested pixelwise methods (PTM RF, PLM RF) do not provide any dependent parameter, so only their average observers' responses for individual samples are shown in the first two lines of Table 4. The high values for PTM RF suggest its poor performance for all of the tested samples, while the values of PLM RF are also often above average values of the other tested PCA-based methods. The last row of Table 4 shows measured guess rates  $\gamma$  for individual samples and their averages. These values were obtained as incorrect responses to identical renderings, both using original data, and were used for initialization of the *psignifit* algorithm.

## 7.2 Computational and Visual Quality Comparison

Pixelwise computational comparison is possible only for methods that preserve pixelwise structure of the original BTF tiles. For this reason, a fair comparison of probabilistic model results is not possible to be achieved in this way. For all other methods, the pixelwise error between original and synthesized BTF images was computed using Mean

Average Error (MAE) in perceptually uniform CIE Lab color space. Comparison of MAE dependence on all 81 view directions for all tested pixelwise methods and six different material samples is illustrated in Fig. 10.

Comparison of averaged MAE values for all view directions, all tested pixelwise methods, and material samples is presented in Table 5. In the graphs and the table, we can see a considerable difference between PTM RF, PLM RF methods and the PCA-based methods, whose parameters were tuned specifically for each sample by means of the psychophysical experiment.

For subjective visual comparison, a 3D object was rendered using synthetic BTF data obtained by the individual tested methods. Such renderings are shown in Fig. 11 again for six different tested material samples. As expected, the visual performance of the tested PCA-based methods was quite similar due to sample-dedicated parameters set by the experiment. The PTM RF method apparently misses specular highlights and PLM RF slightly increases contrast, which is in accordance with Fig. 10 and Table 5.

## 7.3 Parametric Representation Size and Compression

The size of parametric representation of pixelwise BTF modeling methods depends on a number of stored parametric planes. These planes can represent coefficients of underlying models, i.e., they can be eigenimages, pixelwise polynomial, or reflectance model parameters. For more detailed information on parametric representation of tested methods, see their descriptions in Sections 4.1, 4.2, and 5.3.

Table 6 provides formulas for computation of the storage size of parametric representation for the tested methods. The compression ratio of these methods is obtained by dividing the storage size of *BTF tile* by the parameter

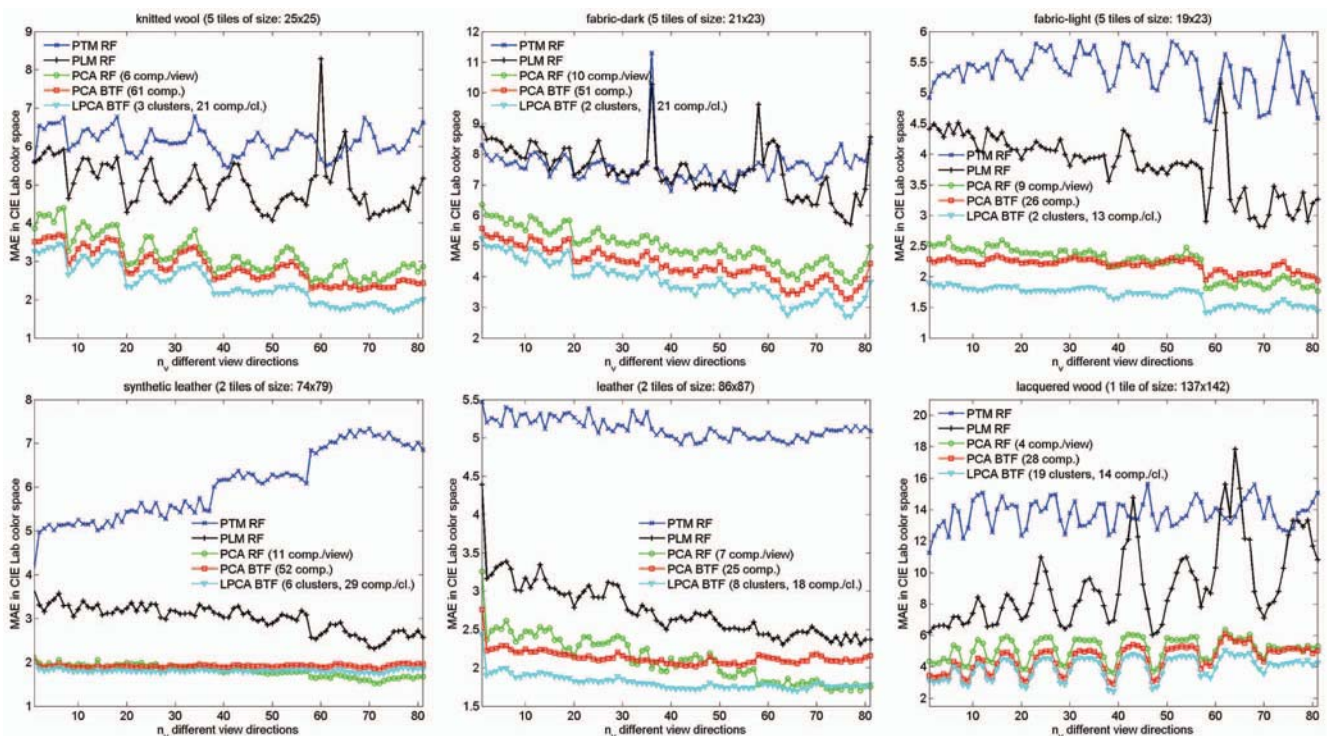


Fig. 10. The comparison of individual pixelwise BTF modeling methods for six different material samples in terms of MAE in CIE Lab color space dependent on viewing direction change (see Fig. 4) 0-the top, 81-the bottom of the hemisphere.

TABLE 5  
Mean Average BTF Reconstruction Error  
(in CIE Lab Color Space) of Tested Pixelwise Methods

method	Mean Average Error in CIE Lab for the tested samples					
	knitted wool	fabric dark	fabric light	synthetic leather	leather	lacquer. wood
PTM RF	6.13	7.60	5.35	6.04	5.12	13.74
PLM RF	5.03	7.48	3.84	3.01	2.75	9.21
PCA RF	3.16	4.98	2.23	1.81	2.11	5.14
PCA BTF	2.85	4.39	2.19	1.92	2.13	4.53
LPCA BTF	2.42	3.88	1.70	1.81	1.81	3.96

storage size of the respective method. Note that we assume all parameter values as floating-point numbers; hence, by means of their quantization, we can achieve even higher compression for most of the tested methods.

The overall comparison of parameters storage size and compression ratios of all nine tested methods for different

materials is shown in Table 7. The table summarizes parametric size and compression ratios of 10 BTF tiles and their parametric representation using the tested pixelwise methods. Note that these values are dependent on actual size of BTF tiles (the fourth row). The third line shows the compression obtained by direct cutting of BTF tiles from the original BTF data (800 × 800 pixels). The compression achieved by probabilistic methods was computed as a ratio of raw BTF data size and the respective fixed size of that method’s parametric representation. As expected, the best compression rates were obtained for smooth (or less rough) samples (e.g., wood and leathers), while the wool and fabrics, exhibiting more complex effects, reached lower values for the same visual quality. Note that the total compression of original BTF data achieved by combination of BTF tiling and one of the tested compression methods is obtained by multiplication of the two respective values.

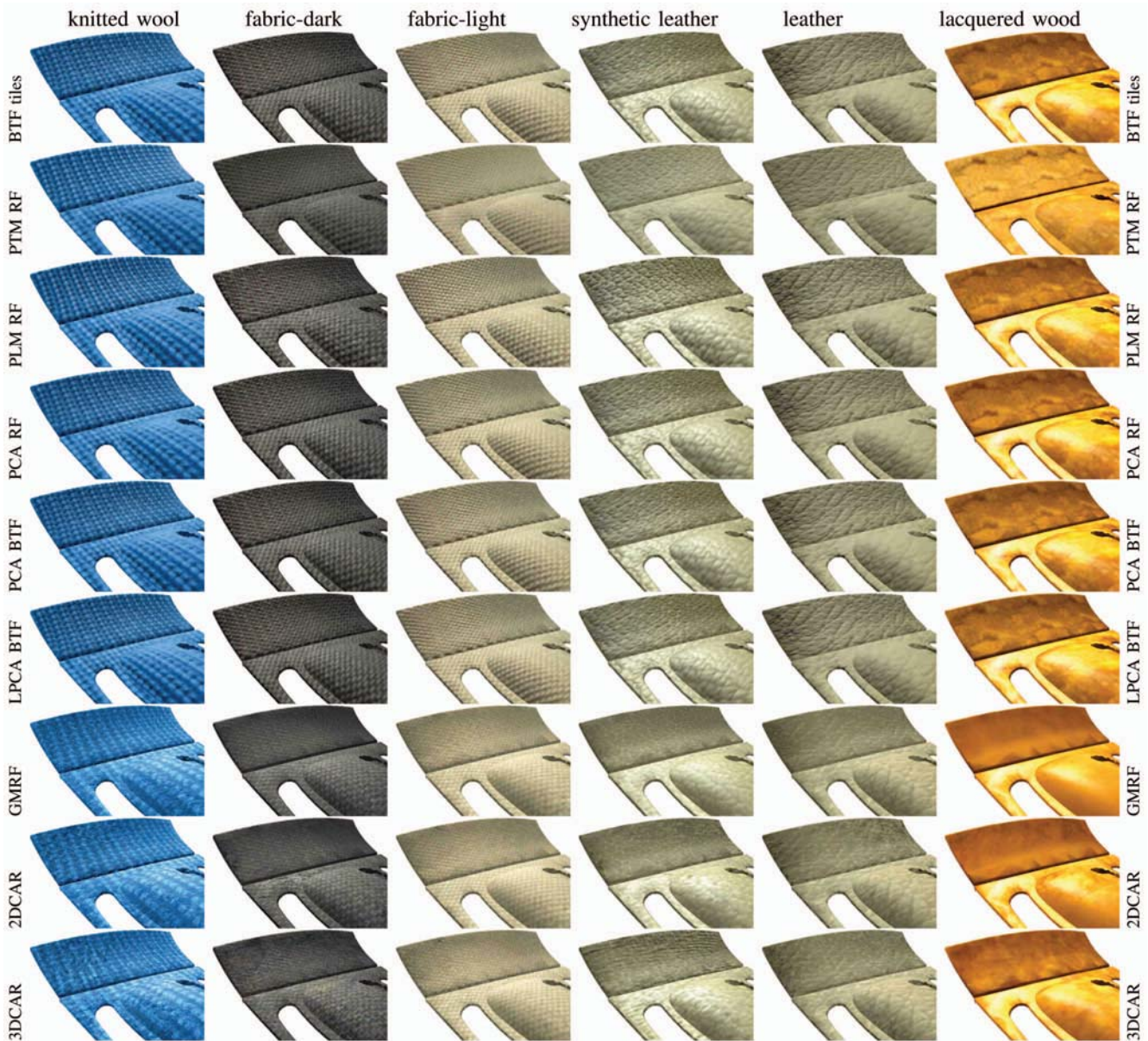


Fig. 11. BTF results of all eight compared methods mapped on a car gearbox console for six different tested materials. Light position: right-back.

TABLE 6  
Formulas Giving Size of Parametric Representations  
of the Tested Pixelwise Methods

<b>BTF tile</b>	$sn_r n_c n_i n_v$	where:
<b>PTM RF</b>	$6sn_r n_c n_v$	$s \dots$ no. of spectral channels;
<b>PLM RF</b>	$s(5n_r n_c + 5n_i)n_v$	$n_r \times n_c \dots$ spatial size of the sample;
<b>PCA RF</b>	$(sn_r n_c(k+1) + kn_i)n_v$	$n_i/n_v \dots$ no. of illum./view directions;
<b>PCA BTF</b>	$sn_r n_c(k+1) + kn_i n_v$	$k \dots$ no. of principal components;
<b>LPCA BTF</b>	$n_r n_c + scn_i n_v(k+1)$	$c \dots$ no. of clusters

Dependency of the tested PCA-based methods compression ratio on number of pixels in the analyzed BTF sample, illumination/view direction sampling quantization, and on the number of the preserved principal components is shown in Fig. 12. Note that for PCA-based methods, the parameters obtained from psychophysical experiment, averaged over all of the tested samples, are used (see last column of Table 3). From the first graph, it is obvious that for smaller BTF samples/tiles (less than  $\sim 170 \times 170$  pixels), the best compression can be achieved by PCA BTF, while for larger samples, the best suited method is LPCA BTF. On the other hand, the analysis of such a large BTF by means of this method can easily take several days. The second graph shows that by far the best compression with increasing angular quantization of illumination/view directions is provided by PCA BTF. When observing the last graph, we should again take into account the average number of components set by the psychophysical study (last column of Table 3).

It is obvious that the size of the parametric representation is correlated with the size of the original BTF (i.e., the size of BTF tiles in our case—see the fourth row of Table 7), so for bigger tiles, the view reflectance field-based models (PTM RF, PLM RF, PCA RF) easily reach several hundreds

of megabytes. This is due to storing the parametric planes for all view directions, i.e., reflectance fields. This huge data can be further considerably reduced when a certain parametric space quantization scheme is applied. Fig. 13 is an example of the lacquered wood BTF sample rendering using the PLM RF method without (a) and with quantization (b) using 256 parametric clusters per color channel. The visual differences are negligible while the size of parametric representation drops approximately 10 times. The pixelwise models represent original BTF tiles by means of a set of parametric tiles of some underlying model and these tiles are used for BTF data enlargement based on this tiling. A completely different approach is used for BTF models based on MRFs (GMRF, 2DCAR, 3DCAR), where only negligible statistics model parameters are stored in addition to tiled range and normal maps. The MRF models enable seamless synthesis from parameters in an arbitrary size, while only the range and normal map are enlarged using the tiling approach. Compare performance of the 2DCAR model on *l. wood* sample in Fig. 13c.

#### 7.4 Rendering Using Graphics Hardware

To speed up rendering of BTF data (i.e., its reconstruction from model parameters), the continually growing power and functionality of contemporary graphics hardware can be exploited. The reconstruction of BTF data from parameters of all of the tested pixelwise compression methods (PTM RF, PLM RF, PCA RF, PCA BTF, LPCA BTF) can be performed at interactive frame rates (i.e.,  $\sim 20$ -30 frames/s) when implemented in shaders of low-end programmable graphics processing units (GPUs) [20]. The same cannot easily be said about the remaining tested BTF modeling methods based on probabilistic MRF models (GMRF BTF, 2DCAR BTF, 3DCAR BTF). These methods require causal knowledge of spatially neighboring data during BTF subspaces synthesis, which is completely orthogonal to

TABLE 7  
Size of Parametric Representation and Compression Ratio of the Tested Methods Compared with Raw and Tiled Original BTF Data

method	BTF original and parametric representation storage size [MB] / compression ratio [1:x]						
	knitted wool	fabric dark	fabric light	synthetic leather	leather	lacquered wood	
raw BTF (PNG)	750.9 / 1.0	6928.7 / 1.0	6004.1 / 1.0	5186.4 / 1.0	5195.9 / 1.0	5205.5 / 1.0	
10 BTF tiles (PNG)	103.4 / 102.4	89.2 / 132.5	79.4 / 146.5	540.2 / 10.9	675.5 / 8.5	2462.5 / 3.3	
tile size [pixels]	25 × 25	21 × 23	19 × 23	74 × 79	86 × 87	137 × 142	
<b>PTM RF</b>	36.5 / 13.5	28.2 / 13.5	25.6 / 13.5	341.0 / 13.5	436.4 / 13.5	1134.6 / 13.5	
<b>PLM RF</b>	31.8 / 14.3	24.9 / 13.9	22.7 / 13.7	285.6 / 16.0	365.1 / 16.0	946.9 / 16.1	
<b>PCA RF</b>	81.0 / 11.2	76.7 / 7.0	45.1 / 7.7	688.3 / 6.7	585.5 / 10.1	924.7 / 16.2	
<b>PCA BTF</b>	20.3 / 23.8	16.2 / 23.2	8.0 / 41.8	49.6 / 90.6	29.3 / 197.0	73.4 / 204.1	
<b>LPCA BTF</b>	52.5 / 9.4	35.5 / 10.8	22.5 / 15.4	210.1 / 31.0	176.6 / 47.0	325.4 / 64.9	
<b>GMRF BTF</b>	0.12 / $0.610^4$	0.09 / $7.710^4$	0.06 / $10.010^4$	0.17 / $3.110^4$	0.18 / $2.910^4$	0.07 / $7.410^4$	
<b>2DCAR BTF</b>	0.09 / $0.810^4$	0.12 / $5.810^4$	0.09 / $6.710^4$	0.15 / $3.510^4$	0.20 / $2.610^4$	0.07 / $7.410^4$	
<b>3DCAR BTF</b>	0.75 / $0.110^4$	0.54 / $1.310^4$	0.44 / $1.410^4$	1.07 / $0.510^4$	0.35 / $1.510^4$	0.27 / $1.910^4$	

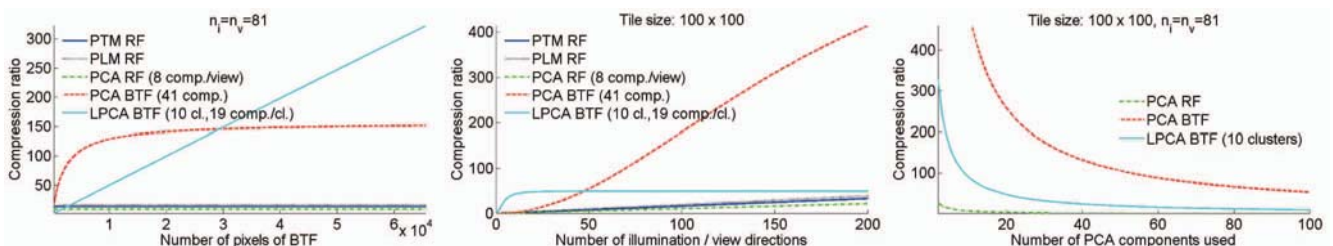


Fig. 12. The comparison of compression ratios dependent on BTF resolution, number of illumination/view directions, and preserved principal components, respectively, for the tested pixelwise compression methods.



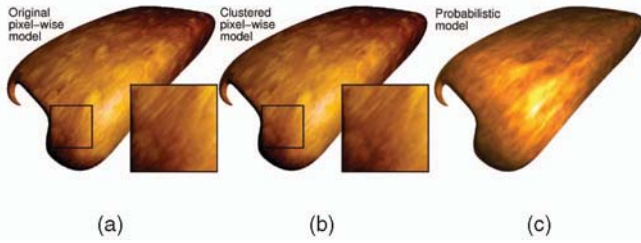


Fig. 13. Example of standard (BTF compression ratio  $\sim 1 : 10$ ) and clustered (BTF compression ratio  $\sim 1 : 100$ ) PLM RF model compared with probabilistic model 2D CAR (BTF compression ratio  $\sim 1 : 70,000$ ) for lacquered wood sample.

contemporary GPU hardware philosophy. This problem can partially be avoided either by using fragment buffer objects with rendering-to-texture techniques or by subsequent reading of previously synthesized pixels from a pixel buffer. However, such an operation can be time-consuming and the final computational time can be similar to standard CPU computation. On the other hand, this problem can also be circumvented in the near future with oncoming graphics hardware using faster memory chips.

In the BTF rendering stage for arbitrary illumination/view  $i/v$  directions, the methods PTM RF and PLM RF require interpolation only for  $v$  directions since arbitrary  $i$  directions can be passed as arguments of underlying functions. In contrast, all the other methods require simultaneous interpolation of both  $i$  and  $v$  directions. Such an interpolation for all renderings in this paper was performed by means of the three closest barycentric weights [7], computed separately for the three closest  $i$  and  $v$  directions resulting in nine interpolation weights. For each triangle of 3D object, nine synthesized BTF images are combined. Although the interpolation requires extra computational time, its weights can be precomputed, stored in a cube map, and rapidly accessed in shader programs of graphics hardware.

### 7.5 Speed Comparison

The speed of analysis and synthesis of individual methods was tested on a small BTF tile of resolution  $25 \times 25$  pixels. These tests were performed on CPU AMD Athlon 2.2 GHz, 3 GB RAM and the results are shown in Table 8.

All of the methods are supposed to be applicable in real-time rendering applications, so the corresponding synthesis has to be very fast, as shown in the third column of the table. For this reason, the time for synthesis of whole BTF space is more or less similar for all the methods. On the other hand, there are considerable differences in the analysis time (second column). The longest time is required by methods modeling all BTF data at once (PCA BTF, LPCA BTF), so for large BTF tiles representing less spatially homogeneous materials, the parameters computation can take many hours. The extremely long analysis time of PCA BTF method is caused mostly by computation of the data covariance matrix. However, when a much larger BTF tile is used, the longest computational times belong to LPCA BTF method having polynomial complexity with respect to the number of tile pixels  $n$ . The third column of Table 8 shows estimates of method complexity dependently on number of pixels  $n$  in original BTF tile. There are also other variables which affect computational complexity for some methods

TABLE 8  
Time Demands and Computational Complexity of Analysis and Synthesis Stages of the Tested Methods

method	CPU time [s] BTF $25 \times 25$ pix		approximate complexity of BTF analysis	operations for pixel synthesis		
	anal.	synth.		*	+	$x^y$
PTM RF	165	$\sim 1$	$\mathcal{O}(n_v n_i (n_p^3 + n_p^2) n)$	6	6	0
PLM RF	136	$\sim 1$	$\mathcal{O}(n_v n_i n_p^2 t n)$	7	9	1
PCA RF	10	$\sim 2$	$\mathcal{O}(n_v (n_i)^2 n)$	8	8	0
PCA BTF	3862	$\sim 8$	$\mathcal{O}((n_i n_v)^2 n)$	41	41	0
LPCA BTF	1098	$\sim 22$	$\mathcal{O}(\frac{1}{c} n_i n_v n^2)$	19	19	0
GMRF	600	$\sim 0.04$	-	-	-	-
2DCAR	600	$\sim 0.01$	-	-	-	-
3DCAR	1200	$\sim 0.02$	-	-	-	-

as  $n_i/n_v$ , i.e., number of illumination/view directions  $n_p$ , i.e., number of per-pixel parameters  $c$ , i.e., number of clusters and  $t$ , i.e., number of method iterations. Note that the complexity stated for individual methods can often be improved by means of various approximate methods. The last three columns of Table 8 describe numbers of basic floating-point operations (addition, subtraction, power) required by individual pixelwise methods for reconstruction of one pixel from its parameters for fixed illumination and viewing directions. Note that explicit values shown in this Table for PCA-based methods correspond to psychophysically set parameters averaged over all samples (the last column of Table 3).

### 7.6 Discussion

It is apparent from the previous section that different methods provide different performance, depending on various aspects. While the pixelwise-based methods (PLM RF, PCA RF, PCA BTF, and LPCA BTF) have generally good visual quality and can provide fast rendering, some of them, without additional quantization algorithm, have huge parameter storage requirements (PTM RF, PLM RF, PCA RF). The methods PCA BTF and LPCA BTF approximating whole BTF data space at once reach really long BTF analysis times, which are balanced by their good visual performance and relatively low size of parametric representation. However, all tested pixelwise methods alone only compress original BTF data, and thus, for real modeling, they have to be combined with BTF sampling-based algorithms. On the other hand, the MRF-based models (GMRF, 2DCAR, 3DCAR) enable seamless BTF synthesis of arbitrary size as well as synthesis of previously unmeasured BTF subspaces. Additionally, they provide us with unbeatable compression ratios unattainable by any pixelwise-based method. They provide excellent results for samples with relatively smooth surfaces and irregular random textures common in natural materials (see Fig. 13c) while their performance on considerably rough and translucent surfaces is not very convincing. Regardless of their visual performance, these models are ideal for BTF recognition or illumination invariant retrieval tasks as suggested in [38] due to their compact parametric representation. Mutual comparison of various properties of the compared methods is given in Table 9.

## 8 CONCLUSIONS

The BTF modeling approaches published so far can be categorized into two basic groups—compression and

TABLE 9  
Attributes Rough Comparison of the Implemented BTF Models†

Observed attribute	Original BTF Tiling	Pixel-Wise Models					MRF Models		
		PTM RF	PLM RF	PCA RF	PCA BTF	LPCA BTF	GMRF BTF	2D CAR BTF	3D CAR BTF
seamless enlargement	Yes-	-	-	-	-	-	Yes	Yes	Yes
compression ratio	*	*	*	*	*	*	****	****	****
regular samples representation	****	****	****	****	****	****	**	**	**
irregular samples representation	*	*	*	*	*	*	****	****	****
pixel-wise features representation	****	****	****	****	****	****	*	*	*
reflectance variations represent.	****	*	*	*	*	*	****	****	****
ease of GPU implementation	****	****	****	****	****	****	**	**	*
analysis speed	***	**	**	****	*	*	***	***	**
direct illumination interpolation	-	Yes	Yes	-	-	-	-	-	-
separated analysis and synthesis	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
unseen data modelling	-	No	No	No	No	No	Yes	Yes	Yes
block-wise processing	Yes	Yes	Yes	-	-	-	-	Yes	Yes

† the more stars the better the model is in that attribute

modeling methods. The modeling group can be further classified into sampling methods and random-field-based models. Finally, a hybrid combination of both basic approaches is possible as well. Our experience, similarly to other texture analytical tasks, shows that there is no ideal BTF modeling approach. Some pixelwise compression methods produce excellent visual quality but their compression ratio is only mild, while random-field-based models sometimes compromise visual quality but offer extreme BTF compression and very fast analysis as well as synthesis. Several models can easily be implemented in graphics hardware or paralleled. Some methods even allow us to model/interpolate previously unseen data (by modification of the corresponding parameters) or reconstruct parts of an unmeasured BTF space. The results of selected compression and modeling methods demonstrate their performance for six tested BTF samples. Furthermore, the performed psychophysical experiment showed that to obtain objectively the same visual performance, different BTF samples require different parametric settings of the tested methods. Finally, it has to be noted that there is no ideal universal BTF model and the most suitable one has to be chosen depending on the intended application (real-time, compact data representation, fast GPU implementation, visual quality, etc.) as well as on the specific material sample.

## ACKNOWLEDGMENTS

J. Filip was supported by the EC Marie Curie Intra-European Fellowship No 41358. He was with the Texture-Lab, School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, United Kingdom. This research was also supported by grants GACR 102/08/0593, 1ET400750407 and partially by the MŠMT grants 1M0572 and 2C06019. The authors would like to thank Bonn University for providing the BTF measurements, Daimler-Chrysler AG for 3D model of car interior, participants for taking part in the experiment, and reviewers for their valuable and inspiring comments.

## REFERENCES

- [1] M. Ashikhmin, "Synthesizing Natural Textures," *Proc. ACM Symp. Interactive 3D Graphics*, pp. 217-226, 2001.

- [2] J. Bennett and A. Khotanadz, "Multispectral Random Field Models for Synthesis and Analysis of Color Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 327-332, Mar. 1998.
- [3] J. Besag, "Spatial Interaction and the Statistical Analysis of Lattice Systems," *J. Royal Statistical Soc.*, vol. B-36, no. 2, pp. 192-236, Feb. 1974.
- [4] Bonn University BTF Database, <http://btf.cs.uni-bonn.de>, 2003.
- [5] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-Specific Material Categorisation," *Proc. Int'l Conf. Computer Vision*, pp. 1597-1604, 2005.
- [6] M. Cohen, J. Shade and, S.E.A. Hiller, "Wang Tiles for Image and Texture Generation," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 287-294, July 2003.
- [7] H.S.M. Coxeter, *Introduction to Geometry*. Wiley, 1969.
- [8] O. Cula, K. Dana, F. Murphy and, B. Rao, "Bidirectional Imaging and Modeling of Skin Texture," *IEEE Trans. Biomedical Eng.*, vol. 51, no. 12, pp. 2148-2159, Dec. 2004.
- [9] S. Daly, "The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity," *Digital Images and Human Vision*, pp. 179-206, 1993.
- [10] K.J. Dana, S.K. Nayar, B. van Ginneken, and J.J. Koenderink, "Reflectance and Texture of Real-World Surfaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 151-157, 1997.
- [11] K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink, "Reflectance and Texture of Real-World Surfaces," *ACM Trans. Graphics*, vol. 18, no. 1, pp. 1-34, 1999.
- [12] K. Dana and J. Wang, "Device for Convenient Measurement of Spatially Varying Bidirectional Reflectance," *J. Optical Soc. America*, vol. 21, no. 1, pp. 1-12, 2004.
- [13] K. Daubert, H.P.A. Lensch, W. Heidrich, and H.-P. Seidel, "Efficient Cloth Modeling and Rendering," *Proc. Conf. Rendering Techniques*, pp. 63-70, 2001.
- [14] J. De Bonet, "Multiresolution Sampling Procedure for Analysis and Synthesis of Textured Images," *Proc. ACM SIGGRAPH*, pp. 361-368, 1997.
- [15] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar, "Acquiring the Reflectance Field of A Human Face," *Proc. ACM SIGGRAPH*, pp. 145-156, July 2000.
- [16] J. Dong and M. Chantler, "Comparison of Five 3D Surface Texture Synthesis Methods," *Proc. Third Int'l Workshop Texture Analysis and Synthesis*, pp. 19-23, 2003.
- [17] J. Dong and M. Chantler, "Capture and Synthesis of 3D Surface Texture," *Int'l J. Computer Vision*, vol. 62, nos. 1/2, pp. 177-194, Apr. 2005.
- [18] A.A. Efros and W.T. Freeman, "Image Quilting for Texture Synthesis and Transfer," *Proc. ACM SIGGRAPH*, pp. 341-346, 2001.
- [19] A.A. Efros and T.K. Leung, "Texture Synthesis by Non-Parametric Sampling," *Proc. Int'l Conf. Computer Vision*, vol. 2, pp. 1033-1038, 1999.
- [20] R. Fernando, M. Harris, M. Wloka, and C. Zeller, "Programming Graphics Hardware," *Proc. Computer Graphics Forum*, Aug. 2004.
- [21] J. Filip, M. Chantler, P. Green, and M. Haindl, "A Psychophysically Validated Metric for Bidirectional Texture Data Reduction," *ACM Trans. Graphics*, vol. 27, no. 5, Dec. 2008.

- [22] J. Filip, M. Chantler, and M. Haindl, "On Optimal Resampling of View and Illumination Dependent Textures," *Proc. Fifth Symp. Applied Perception in Graphics and Visualization*, pp. 131-134, Aug. 2008.
- [23] J. Filip and M. Haindl, "Non-Linear Reflectance Model for Bidirectional Texture Function Synthesis," *Proc. 17th Int'l Assoc. for Pattern Recognition Int'l Conf. Pattern Recognition*, vol. 1, pp. 80-83, Aug. 2004.
- [24] J. Filip and M. Haindl, "Efficient Image-Based Bidirectional Texture Function Model," *Proc. Fourth Int'l Workshop Texture Analysis and Synthesis*, pp. 7-12, Oct. 2005.
- [25] J. Filip and M. Haindl, "BTF Modelling Using BRDF Texels," *Int'l J. Computer Math.*, vol. 84, no. 9, pp. 1267-1283, Sept. 2007.
- [26] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "Appearance Based Object Modeling Using Texture Database: Acquisition, Compression and Rendering," *Proc. 13th Eurographics Workshop Rendering*, pp. 257-266, 2002.
- [27] J. Grim and M. Haindl, "Texture Modelling by Discrete Distribution Mixtures," *Computational Statistics Data Analysis*, vol. 41, nos. 3/4, pp. 603-615, Jan. 2003.
- [28] M. Haindl, "Texture Synthesis," *CWI Quarterly*, vol. 4, no. 4, pp. 305-331, Dec. 1991.
- [29] M. Haindl and J. Filip, "Fast BTF Texture Modelling," *Proc. Third Int'l Workshop Texture Analysis and Synthesis*, Oct. 2003.
- [30] M. Haindl and J. Filip, "A Fast Probabilistic Bidirectional Texture Function Model," *Lecture Notes in Computer Science*, vol. 3212, pp. 298-305, Springer, 2004.
- [31] M. Haindl and J. Filip, "Extreme Compression and Modeling of Bidirectional Texture Function," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1859-1865, Oct. 2007.
- [32] M. Haindl, J. Filip, and M. Arnold, "BTF Image Space Utmost Compression and Modelling Method," *Proc. Seventh Int'l Assoc. for Pattern Recognition Int'l Conf. Pattern Recognition*, vol. 3, pp. 194-197, Aug. 2004.
- [33] M. Haindl, J. Grim, P. Pudil, and K.M., "A Hybrid BTF Model Based on Gaussian Mixtures," *Proc. Fourth Int'l Workshop Texture Analysis and Synthesis*, pp. 95-100, Oct. 2005.
- [34] M. Haindl, J. Grim, P. Somol, P. Pudil, and M. Kudo, "A Gaussian Mixture-Based Colour Texture Model," *Proc. 17th Int'l Assoc. for Pattern Recognition Int'l Conf. Pattern Recognition*, vol. 3, pp. 177-180, Aug. 2004.
- [35] M. Haindl and M. Hatka, "A Roller—Fast Sampling-Based Texture Synthesis Algorithm," *Proc. Int'l Conf. Central Europe Computer Graphics, Visualization and Computer Vision*, pp. 80-83, Jan. 2005.
- [36] M. Haindl and M. Hatka, "BTF Roller," *Proc. Fourth Int'l Workshop Texture Analysis and Synthesis*, pp. 89-94, Oct. 2005.
- [37] M. Haindl and V. Havlíček, "A Multiresolution Causal Colour Texture Model," *Advances in Pattern Recognition*, chap. 1, pp. 114-122, Springer-Verlag, Aug. 2000.
- [38] M. Haindl and P. Vácha, "Illumination Invariant Texture Retrieval," *Proc. 18th Int'l Conf. Pattern Recognition*, pp. 276-279, Aug. 2006.
- [39] J.Y. Han and K. Perlin, "Measuring Bidirectional Texture Reflectance with a Kaleidoscope," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 741-748, 2003.
- [40] E. Hayman, B. Caputo, M. Fritz, and J. Eklundh, "On the Significance of Real-World Conditions for Material Classification," *Proc. Eighth European Conf. Computer Vision*, pp. 253-266, 2004.
- [41] D. Heeger and J. Bergen, "Pyramid Based Texture Analysis/Synthesis," *Proc. ACM SIGGRAPH*, pp. 229-238, 1995.
- [42] P.-M. Ho, T.-T. Wong, and C.-S. Leung, "Compressing the Illumination-Adjustable Images with Principal Component Analysis," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 355-364, Mar. 2005.
- [43] B. Julesz, "Visual Pattern Discrimination," *IRE Trans. Information Theory*, vol. 8, no. 1, pp. 84-92, Feb. 1962.
- [44] B. Julesz, "Textons, the Elements of Texture Perception and Their Interactions," *Nature*, vol. 290, pp. 91-97, 1981.
- [45] B. Julesz, E. Gilbert, and J. Victor, "Visual Discrimination of Textures with Identical Third-Order Statistics," *Biological Cybernetics*, vol. 31, pp. 137-140, 1978.
- [46] R. Kashyap, "Analysis and Synthesis of Image Patterns by Spatial Interaction Models," *Progress in Pattern Recognition*, vol. 1, pp. 149-186, Elsevier, 1981.
- [47] J. Kautz, S. Boulos, and F. Durand, "Interactive Editing and Modelling of Bidirectional Texture Functions," *ACM Trans. Graphics*, vol. 26, no. 3, p. 53, 2007.
- [48] J. Kautz, M. Sattler, R. Sarlette, R. Klein, and H.-P. Seidel, "Decoupling BRDFs from Surface Mesostuctures," *Proc. Graphics Interface Conf. '04*, pp. 177-184, 2004.
- [49] H. Kawasaki, K.-D. Seo, Y. Ohsawa, and R. Furukawa, "Patch-Based BTF Synthesis for Real-Time Rendering," *Proc. IEEE Int'l Conf. Image Processing*, vol. 1, pp. 393-396, Sept. 2005.
- [50] M. Koudelka, S. Magda, P. Belhumeur, and D. Kriegman, "Acquisition, Compression, and Synthesis of Bidirectional Texture Functions," *Proc. Third Int'l Workshop Texture Analysis and Synthesis*, pp. 47-52, Oct. 2003.
- [51] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," *ACM Trans. Graphics*, vol. 22, no. 3 pp. 277-286, July 2003.
- [52] E.P. Lafortune, S.C. Foo, K.E. Torrance, and D.P. Greenberg, "Non-Linear Approximation of Reflectance Functions," *Computer Graphics*, vol. 31, pp. 117-126, 1997.
- [53] 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. Graphics*, vol. 25, no. 3, pp. 735-745, 2006.
- [54] J. Lehtinen, "A Framework for Precomputed and Captured Light Transport," *ACM Trans. Graphics*, vol. 26, no. 4, pp. 13:1-13:22, 2007.
- [55] C.-S. Leung, W.-M. Pang, C.-W. Fu, T.-T. Wong, and P.-A. Heng, "Tileable BTF," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 5, pp. 935-955, Sept./Oct. 2007.
- [56] C.-S. Leung, T.-T. Wong, P.-M. Lam, and K.-H. Choy, "An RBF-Based Compression Method for Image-Based Relighting," *IEEE Trans. Image Processing*, vol. 15, no. 4, pp. 1031-1041, Apr. 2006.
- [57] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials Using Three-Dimensional Textons," *Int'l J. Computer Vision*, vol. 43, no. 1, pp. 29-44, 2001.
- [58] X. Liu, Y. Yu, and H.Y. Shum, "Synthesizing Bidirectional Texture Functions for Real-World Surfaces," *Proc. ACM SIGGRAPH*, pp. 97-106, 2001.
- [59] X. Liu, J. Zhang, X. Tong, B. Guo, and H.-Y. Shum, "Synthesis and Rendering of Bidirectional Texture Functions on Arbitrary Surfaces," *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 3, pp. 278-289, May/June 2004.
- [60] W.-C. Ma, S.-H. Chao, B.-Y. Chen, C.-F. Chang, M. Ouhyoung, and T. Nishita, "An Efficient Representation of Complex Materials for Real-Time Rendering," *Proc. ACM Symp. Virtual Reality Software and Technology*, pp. 150-153, 2004.
- [61] 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," *Proc. 2005 Symp. Interactive 3D Graphics*, pp. 187-194, Apr. 2005.
- [62] S. Magda and D. Kriegman, "Reconstruction of Volumetric Surface Textures for Real-Time Rendering," *Proc. 17th Eurographics Workshop Rendering*, pp. 19-29, June 2006.
- [63] T. Malzbender, D. Gelb, and H. Wolters, "Polynomial Texture Maps," *Proc. ACM SIGGRAPH*, pp. 519-528, 2001.
- [64] S. Marschner, S. Westin, A. Arbree, and J. Moon, "Measuring and Modeling the Appearance of Finished Wood," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 727-734, 2005.
- [65] D.K. McAllister, A. Lastra, and W. Heidrich, "Efficient Rendering of Spatial Bi-Directional Reflectance Distribution Functions," *Graphics Hardware*, pp. 77-88, 2002.
- [66] M. McCool, J. Ang, and A. Ahmad, "Homomorphic Factorization of BRDFs for High-Performance Rendering," *Proc. ACM SIGGRAPH*, pp. 185-194, 2001.
- [67] J. Meseth, G. Müller, and R. Klein, "Preserving Realism in Real-Time Rendering of Bidirectional Texture Functions," *Proc. OpenSG Symp.*, pp. 89-96, Apr. 2003.
- [68] J. Meseth, G. Müller, R. Klein, F. Röder, and M. Arnold, "Verification of Rendering Quality from Measured BTFs," *Proc. Third Symp. Applied Perception in Graphics and Visualization*, vol. 153, pp. 127-134, July 2006.
- [69] J. Meseth, G. Müller, M. Sattler, and R. Klein, "BTF Rendering for Virtual Environments," *Proc. Conf. Virtual Concepts 2003*, pp. 356-363, Nov. 2003.

- [70] G. Müller, G. Bendels, and R. Klein, "Rapid Synchronous Acquisition of Geometry and BTF for Cultural Heritage Artefacts," *Proc. Sixth Int'l Symp. Virtual Reality, Archaeology and Cultural Heritage*, pp. 13-20, 2005.
- [71] G. Müller, J. Meseth, and R. Klein, "Compression and Real-Time Rendering of Measured BTFs Using Local PCA," *Proc. Conf. Vision, Modeling, and Visualisation*, pp. 271-280, Nov. 2003.
- [72] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein, "Acquisition, Synthesis and Rendering of Bidirectional Texture Functions," *Proc. Eurographics 2004 State of the Art Reports*, pp. 69-94, Sept. 2004.
- [73] A. Neubeck, A. Zalesny, and L. Gool, "3D Texture Reconstruction from Extensive BTF Data," *Proc. Fourth Int'l Workshop Texture Analysis and Synthesis*, pp. 13-18, Oct. 2005.
- [74] A. Neubeck, A. Zalesny, and L. Van Gool, "Viewpoint Consistent Texture Synthesis," *Proc. Conf. 3D Data Processing, Visualization and Transmission*, pp. 388-395, Sept. 2004.
- [75] A. Ngan and F. Durand, "Statistical Acquisition of Texture Appearance," *Proc. Eurographics Symp. Rendering '05*, pp. 31-40, Aug. 2006.
- [76] F. Nicodemus, R.J.C.J. Hsia, I. Ginsburg, and T. Limperis, "Geometrical Considerations and Nomenclature for Reflectance," NBS Monograph 160, Nat'l Bureau of Standards, US Dept. of Commerce, pp. 1-52, 1977.
- [77] F. Pellacini and J. Lawrence, "AppWand: Editing Measured Materials Using Appearance-Driven Optimization," *ACM Trans. Graphics*, vol. 26, no. 3, pp. 54:1-54:10, 2007.
- [78] S.C. Pont and J.J. Koenderink, "Bidirectional Texture Contrast Function," *Int'l J. Computer Vision*, vol. 62, nos. 1/2, pp. 17-34, 2005.
- [79] R. Ramamoorthi, M. Koudelka, and P. Belhumeur, "A Fourier Theory for Cast Shadows," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 288-295, Feb. 2005.
- [80] M. Sattler, R. Sarlette, and R. Klein, "Efficient and Realistic Visualization of Cloth," *Proc. Eurographics Symp. Rendering*, pp. 167-178, June 2003.
- [81] P. Somol and M. Haindl, "Novel Path Search Algorithm for Image Stitching and Advanced Texture Tiling," *Proc. 13th Int'l Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 155-162, 2005.
- [82] P. Suen and G. Healey, "The Analysis and Recognition of Real-World Textures in Three Dimensions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 491-503, May 2000.
- [83] F. Suykens, K. Berge, A. Lagae, and P. Dutré, "Interactive Rendering with Bidirectional Texture Functions," *Computer Graphics Forum*, vol. 22, no. 3, pp. 463-472, Sept. 2003.
- [84] X. Tong, J. Zhang, L. Liu, B. Wang, X. Guo, and H.-Y. Shum, "Synthesis of Bidirectional Texture Functions on Arbitrary Surfaces," *Proc. ACM SIGGRAPH*, pp. 665-672, 2002.
- [85] P. Vacha and M. Haindl, "Image Retrieval Measures Based on Illumination Invariant Textural MRF Features," *Proc. Sixth ACM Int'l Conf. Image and Video Retrieval*, pp. 448-454, 2007.
- [86] M. Vasilescu and D. Terzopoulos, "Tensor Textures: Multilinear Image-Based Rendering," *Proc. ACM SIGGRAPH*, vol. 23, no. 3 pp. 336-342, Aug. 2004.
- [87] H. Wang, G. Wu, L. Shi, Y. Yu, and N. Ahuja, "Out-of-Core Tensor Approximation of Multi-Dimensional Matrices of Visual Data," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 527-535, July 2005.
- [88] J. Wang and K. Dana, "Hybrid Textons: Modeling Surfaces with Reflectance and Geometry," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 372-378, 2004.
- [89] J. Wang and K. Dana, "Relief Texture from Specularities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 446-457, Mar. 2006.
- [90] J. Wang, X. Tong, J. Snyder, Y. Chen, B. Guo, and H.-Y. Shum, "Capturing and Rendering Geometry Details for BTF-Mapped Surfaces," *The Visual Computer*, vol. 21, nos. 8-10, pp. 559-568, 2005.
- [91] L. Wei and M. Levoy, "Texture Synthesis over Arbitrary Manifold Surfaces," *Proc. ACM SIGGRAPH*, pp. 355-360, 2001.
- [92] F. Wichmann and N. Hill, "The Psychometric Function: I. Fitting, Sampling, and Goodness of Fit," *Perception & Psychophysics*, vol. 63, no. 8, pp. 1293-1313, 2001.
- [93] T.-T. Wong and C.-S. Leung, "Compression of Illumination-Adjustable Images," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1107-1118, Nov. 2003.
- [94] Y. Xu, B. Guo, and H. Shum, "Chaos Mosaic: Fast and Memory Efficient Texture Synthesis," Technical Report MSR-TR-2000-32, Microsoft Research, 2000.
- [95] H. Yacov, T. Malzbender, and D. Gelb, "Synthesis of Reflectance Function Textures from Examples," *Proc. Third Int'l Workshop Texture Analysis and Synthesis*, pp. 47-52, Oct. 2003.
- [96] J. Yellott, "Implications of Triple Correlation Uniqueness for Texture Statistics and the Julesz Conjecture," *J. Optical Soc. Am.*, vol. 10, no. 5 pp. 777-793, May 1993.
- [97] S. Zelinka and M. Garland, "Interactive Texture Synthesis on Surfaces Using Jump Maps," *Proc. 14th Eurographics Workshop Rendering*, pp. 90-96, 2003.
- [98] J. Zhang, D. Wang, and G. Tran, "A Wavelet-Based Multi-resolution Statistical Model for Texture," *IEEE Trans. Image Processing*, vol. 7, no. 11, pp. 1621-1627, Nov. 1998.
- [99] K. Zhou, P. Du, L. Wang, Y. Matsushita, J. Shi, B. Guo, and H.-Y. Shum, "Decorating Surfaces with Bidirectional Texture Functions," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 5, pp. 519-528, Sept./Oct. 2005.
- [100] S. Zhu, X. Liu, and Y. Wu, "Exploring Texture Ensembles by Efficient Markov Chain Monte Carlo—Toward A "Trichromacy" Theory of Texture," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 554-569, June 2000.



**Jiří Filip** received the MSc degree in technical cybernetics and the PhD degree in artificial intelligence and biocybernetics from the Czech Technical University in Prague in 2002 and 2006, respectively. He is currently with the Pattern Recognition Department at the Institute of Information Theory and Automation of the AS CR, Praha, Czech Republic. He was a post-doctoral Marie Curie research fellow in the Texture Lab at the School of Mathematical and

Computer Sciences, Heriot-Watt University. Between 2002 and 2007, he was a researcher at the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic. His current research is focused on analysis, modeling, and human perception of high-dimensional texture data and video sequences. He is a member of the IEEE.



**Michal Haindl** received the graduate degree in control engineering from the Czech Technical University, Prague, in 1979, the PhD degree in technical cybernetics from the Czechoslovak Academy of Sciences in 1983, and the ScD (DrSc) degree from the Czech Technical University in 2001. From 1983 to 1990, he worked at the Institute of Information Theory and Automation of the Czechoslovak Academy of Sciences, Prague, on different adaptive control, image

processing, and pattern recognition problems. From 1990 to 1995, he was with the University of Newcastle; Rutherford Appleton Laboratory, Didcot; Center for Mathematics and Computer Science, Amsterdam; and Institute National de Recherche en Informatique et en Automatique, Rocquencourt, working on several image analysis and pattern recognition projects. In 1995, he rejoined the Institute of Information Theory and Automation, where he is the head of the Pattern Recognition Department. His current research interests are random fields applications in pattern recognition and image processing and automatic acquisition of virtual reality models. He is the author of about 230 research papers published in books, journals, and conference proceedings. He is a fellow of the IAPR and a senior member of the IEEE.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).