A PROBABILISTIC APPROACH TO ROUGH TEXTURE COMPRESSION AND RENDERING

Jiří Filip *Jiří Grim* *Michal Haindl*

Institute of Information Theory and Automation of the ASCR, Prague

ABSTRACT

Rough textures describe a general visual appearance of real-world materials with regard to view and illumination directions. As the massive size and dimensionality of such representations is a main limitation of their broader use, efficient parameterization and compression methods are needed. Our method is based on estimating the joint probability density of the included nine spatial, directional, and spectral variables in the form of a Gaussian mixture of product components. Our reflectance prediction formula can be expressed analytically as a simple continuous function of input variables and allows fast analytic evaluation for arbitrary spatial and directional values without need for a lengthy interpolation from a finite grid of angular measurements. This method achieves high compression ratio increasing linearly with texture spatial resolution.

Index Terms— rough texture, BTF, compression, mixture model

1. INTRODUCTION

Real-world natural or man-made materials change, due to rough and structured material surface, their appearance with respect to actual viewing and illumination conditions. A proper visualization, classification, or retrieval of such rough materials with respect to viewing and illumination conditions is a challenging task [1]. Although representations exist that can comprehend a material's appearance variability, due to the massive size of the corresponding high-dimensional data set, it is not practical to use them. Probably the most often, but still not widely, used is a *bidirectional texture function* (BTF), introduced by Dana et al [2]. BTF is a seven-dimensional function representing the appearance of a material sample's surface for variable illumination $\omega_i(\theta_i, \varphi_i)$ and view $\omega_v(\theta_v, \varphi_v)$ directions (see Fig. 1-a), resulting in a function $BTF(\lambda, x, y, \theta_i, \varphi_i, \theta_v, \varphi_v)$, where θ and φ are elevation and azimuthal angles, respectively, λ is the spectral channel, and (x, y) is the spatial location on the texture.

The behavior of each view- and illumination-dependent BTF pixel at position x, y can be approximated as a five-dimensional *bidirectional reflectance distribution function* (BRDF) [4] $BRDF_{x,y}(\lambda, \theta_i, \varphi_i, \theta_v, \varphi_v)$. A typical size of BTF is several gigabytes. Hence, it has been approximated by various compression and modeling approaches so far [5]. Most of them were focused on creating a highly compact parametric representation retaining as much of the original visual fidelity as possible.

This work has been supported by Czech Science Foundation grant 103/11/0335 and EC FP7 Marie Curie ERG 239294.

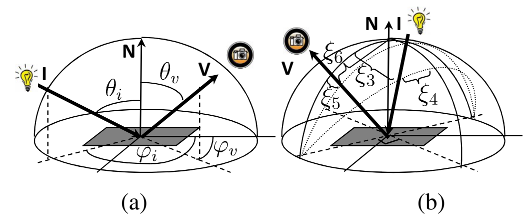

Fig. 1. Parameterization of illumination and view directions within the rough texture: (a) standard parameterization using spherical angles, (b) onion-slice parameterization [3].

In this paper we attempt to create an analytic BTF model based on a very compact set of parameters. Unlike most of the previous approaches, we want to design a model in ours that would allow fast, ideally graphics hardware supported, analytic evaluation for arbitrary input data variables without need for a lengthy interpolation from a predefined grid of angular measurements during the appearance rendering.

2. RELATED WORK

A number of rough texture compression and modeling techniques exist [5].

Factorization and BRDF Fitting Approaches – They are either based on BTF data linear factorization, clustering, pixel-wise modeling by analytical BRDF models, or by their combinations. A parametric representation of most of them is not fully analytic and thus the model evaluation requires a time-consuming interpolation for non-measured illumination and viewing directions in individual pixels on the basis of the compressed data known for the measured directions. The factorization approaches are most often based on PCA [6, 7] or tensors [8, 9, 10]. The clustering approaches usually accomplish factorization into a predefined set of clusters [3, 11]. Analytical pixel-wise reflectance BRDF models have also been used, but due to inherited limitations of BRDF (e.g., reciprocity) [12] they have had to be further extended [13, 14]. Approaches combining reflectance fitting with estimated meso-structure geometry have also been presented [15, 16]. Although the techniques mentioned above are able to reproduce material's appearance in high visual quality and reconstruction speed, they are due to storing some sort of pixel-wise parametric information limited to compression ratios $\approx 1:2000$.

Probabilistic Models – The approaches that allow us to achieve of such compression ratios are based on probabilistic Markov random field (MRF) BTF models that are closely related to our method. Several different MRF models have been published in the past, based either on causal autoregressive models [17, 18] or on a Gaussian MRF model [19]. Due to the stochastic nature of the MRF models, they are less successful at reproducing regular or near-regular structures in BTF samples [5]. Hence these methods combine an estimated range map with a synthetic multi-scale smooth texture.

The Gaussian mixture (GM) models have been applied to static texture synthesis. The method described in [20], [21] was based on a multivariate GM model of the local statistical texture properties in a moving contextual neighborhood.

In this paper we consider the problem of general modeling and rendering of rough textures in full complexity of viewing and lighting conditions – in a nine-dimensional space (i.e., spatial (2D), directional (4D), and color (3D) dependencies). We suggest a solution based on simultaneous modeling BTF data using the nine-dimensional GM model. Such a model, contrary to some of the previous approaches, allows full-color modeling of arbitrary materials as well as analytic evaluation from a compact parametric set. Therefore, the model offers a compression potential that outperforms most of the BTF compression factorization approaches published so far.

3. PROPOSED ROUGH TEXTURE MODEL

Method Overview – The proposed method starts with a mean BRDF computation by averaging view- and illumination-dependent reflectance across individual BTF images. Then these mean values are subtracted from individual BTF images and the resulting data are subject to the fitting using a Gaussian mixture model. The other remaining inputs to the model are the number of mixture components used M, and the number of iterations or minimal increment ε of the fitting quality evaluation function. Fitting of the Gaussian mixture is performed by means of the EM algorithm, resulting in a very compact parametric set. After the model's parameters are fitted, their pixel-wise reconstruction is combined with mean BRDF value to obtain final BTF.

Rough Texture Data Preprocessing – Let $\boldsymbol{\xi}$ be the nine-dimensional vector, where ξ_1, ξ_2 are the spatial pixel coordinates of the source rough texture; ξ_3, ξ_4 and ξ_5, ξ_6 define the information concerning view and illumination directions, respectively; and ξ_7, ξ_8, ξ_9 denote the color RGB values.

First, the input illumination and viewing directions are transformed from common spherical angles $(\theta_i, \varphi_i, \theta_v, \varphi_v)$ to "onion-cut" parameterization [3] $(\xi_3, \xi_4, \xi_5, \xi_6)$ (see Fig. 1), avoiding $0 \approx 2\pi$ discontinuity of azimuth angles in the spherical parameterization:

$$\xi_4 = \arcsin(\sin\theta_i \cdot \cos\varphi_i),\ \xi_3 = \arccos(\cos\theta_i / \cos\xi_4),$$

$$\xi_6 = \arcsin(\sin\theta_v \cdot \cos\varphi_v),\ \xi_5 = \arccos(\cos\theta_v / \cos\xi_6).$$

Finally, to simplify the prediction problem, we centralize the color reflectance values by subtracting the mean BRDF color values $\bar{\xi}$, i.e., obtaining $x_n = \xi_n - \bar{\xi}_n,\ \ n = 7, 8, 9$ and $x_n = \xi_n,\ \ n = 1 \dots 6$ forming $\mathcal{S} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(|\mathcal{S}|)}\}$.

Gaussian Texture Model – For the sake of predicting the output color values x_7, x_8, x_9 for any given input variables $x_1, \dots, x_6$, we approximate the joint multivariate density function of $\boldsymbol{x}$ in the form of Gaussian mixture

$$P(\boldsymbol{x}) = \sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m),\ \ \boldsymbol{x} \in R^9, \qquad (1)$$

Here $\mathcal{M} = \{1, 2, \dots, M\}$ is the index sets of components, $\mathcal{N} = \{1, 2, \dots, 9\}$ denotes the index sets of variables, w_m are probability weights and $F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m)$ denote the mixture components defined as products of univariate Gaussian densities [20], [21]:

$$F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m) = \prod_{n \in \mathcal{N}} f_n(x_n|\mu_{mn}, \sigma_{mn}), \qquad (2)$$

$$f_n(x_n|\mu_{mn}, \sigma_{mn}) = \frac{1}{\sqrt{2\pi}\sigma_{mn}} \exp\left\{-\frac{(x_n - \mu_{mn})^2}{2\sigma_{mn}^2}\right\}.$$

From the computational point of view the product components (2) avoid the risk of ill-conditioned covariance matrices and simplify the evaluation of marginal densities [cf. later Eq. (10)]. Note that the above-described GM model based on the product components does not imply independence of variables (i.e., it is not defined by marginal probability distributions alone). As there is no risk of over-fitting in the case of approximation problems like rough texture rendering, the chosen number of mixture components can be arbitrarily large. As it is well known, the weights of redundant components will be suppressed by the EM algorithm, and the computational time is the only practical limitation. Typically, we used ≈ 500 components, providing a reasonable trade-off between visual quality and computational complexity.

Parameter Estimation – We use the data set $\mathcal{S}$ introduced above to estimate the mixture model. The EM algorithm maximizes the corresponding log-likelihood function

$$L = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \log \left[\sum_{m \in \mathcal{M}} w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m) \right] \qquad (3)$$

by means of the well-known EM iteration equations [20]:

$$q(m|\boldsymbol{x}) = \frac{w_m F(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m)}{\sum_{j \in \mathcal{M}} w_j F(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)},\ \ \boldsymbol{x} \in \mathcal{S}, \qquad (4)$$

$$w_m' = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} q(m|\boldsymbol{x}),\ \ m \in \mathcal{M}, \qquad (5)$$

$$\mu'_{mn} = \frac{1}{\sum_{x \in \mathcal{S}} q(m|\boldsymbol{x})} \sum_{x \in \mathcal{S}} x_n q(m|\boldsymbol{x}), \;\; n \in \mathcal{N}, \quad (6)$$

$$(\sigma'_{mn})^2 = \frac{1}{\sum_{x \in \mathcal{S}} q(m|\boldsymbol{x})} \sum_{x \in \mathcal{S}} x_n^2 q(m|\boldsymbol{x}) - (\mu'_{mn})^2. \quad (7)$$

Here, the apostrophe denotes the new parameter values in each iteration.

Considering large sample size and a large number of components, we may expect numerous local maximums of the log-likelihood function (3) but, according to our experience, the corresponding mixture estimates are of comparable quality. The frequently discussed implementation points of EM algorithm are therefore less relevant: we choose the number of components in hundreds ($M \approx 10^2$) according to problem complexity and initialize the parameters randomly. After the mixture model is estimated, it is used for prediction of the functional values as we explain in the following section.

Reflectance Prediction from Parameters – Let us suppose that the first six input variables corresponding to pixel coordinates, illumination and view directions are known. Denoting $\boldsymbol{x}_\mathcal{I} = (x_1, x_2, \ldots, x_6) \in \mathcal{X}_\mathcal{I}$ the subvector of input variables, $\mathcal{I} = \{1, 2, \ldots, 6\} \subset \mathcal{N}$, we can estimate the output color reflectance values x_7, x_8, x_9 by means of the conditional densities

$$p_{n|\mathcal{I}}(x_n|\boldsymbol{x}_\mathcal{I}) = \frac{P_{n,\mathcal{I}}(x_n, \boldsymbol{x}_\mathcal{I})}{P_\mathcal{I}(\boldsymbol{x}_\mathcal{I})} = \sum_{m \in \mathcal{M}} W_m(\boldsymbol{x}_\mathcal{I}) f_n(x_n|\mu_{mn}, \sigma_{mn}). \quad (8)$$

Here
$$W_m(\boldsymbol{x}_\mathcal{I}) = \frac{w_m F(\boldsymbol{x}_\mathcal{I}|\boldsymbol{\mu}_m, \boldsymbol{\sigma}_m)}{\sum_{j \in \mathcal{M}} w_j F(\boldsymbol{x}_\mathcal{I}|\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j)} \quad (9)$$

are the conditional weights given $\boldsymbol{x}_\mathcal{I} \in \mathcal{X}_\mathcal{I}$ and

$$F(\boldsymbol{x}_\mathcal{I}|\mu_m, \sigma_m) = \prod_{n \in \mathcal{I}} f_n(x_n|\mu_{mn}, \sigma_{mn}) \quad (10)$$

denotes the marginal component functions corresponding to the subspace $\mathcal{X}_\mathcal{I}$. Note that the simple plug-in formula (8) is formally enabled by a simple evaluation of the marginal densities $P_{n,\mathcal{I}}(x_n, \boldsymbol{x}_\mathcal{I})$ and $P_\mathcal{I}(\boldsymbol{x}_\mathcal{I})$.

Equation (8) can be applied to predict the output color variables $x_n, n \in \mathcal{N} \setminus \mathcal{I}$, e.g., by computing the conditional expectations for indices $n = 7, 8, 9$

$$E\{x_n|\boldsymbol{x}_\mathcal{I}\} = \int x_n p_{n|\mathcal{I}}(x_n|\boldsymbol{x}_\mathcal{I}) dx_n = \sum_{m \in \mathcal{M}} W_m(\boldsymbol{x}_\mathcal{I}) \mu_{mn},$$

Note that the final estimated reflectance values $\hat{\xi}_7, \hat{\xi}_8, \hat{\xi}_9$, are obtained as a sum of the previously subtracted mean BRDF values and GM model contributions:

$$\hat{\xi}_n = E\{x_n|\boldsymbol{x}_\mathcal{I}\} + \bar{\xi}_n, \;\; n = 7, 8, 9. \quad (11)$$

As the proposed model is fully parametric, the rough texture values for any spatial and angular coordinates are obtained analytically from parameters of individual components. Contrary to MRF BTF models [17, 18, 19], the proposed reconstruction of individual pixels is completely independent and can be easily implemented directly in graphics hardware (GPU) for fast visualization purposes.

4. TESTING AND RESULTS

Test Datasets – We have used five data sets from the BTF Database Bonn[1] (aluminum profile, corduroy, dark and light fabrics, and knitted wool). Four of the tested material samples are fabrics, exhibiting challenging visual interactions between the light and the material's surface. These data have illumination and viewing directions ($n_i \times n_v = 81 \times 81$) producing uniform sampling of a hemisphere above a material sample. A spatial resolution of the data sets is 256×256 pixels; but for the sake of faster and more convenient data processing we cut BTF tiles [22] that can be freely repeated without visually disruptive seams. The mean BRDFs obtained from the tested samples by means of averaging these BTF pixels are shown in Fig. 2. The rows and columns in the BRDF images are indices of individually measured illumination- and viewing directions uniformly covering the hemisphere above the sample, by spiral movement starting at its pole.

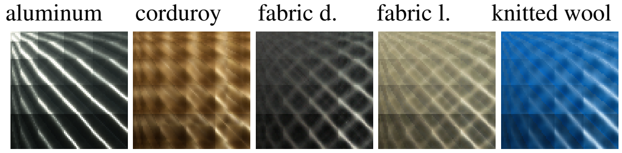

Fig. 2. The mean BRDF computed for the five tested materials (rows illumination directions, columns viewing directions). "Diamond"-like patterns correspond to material's anisotropic behavior for fixed elevation angles.

Experimental Results – The proposed parametric representation has broad application potential and its view- and illumination-dependent descriptive qualities are best illustrated in compressed data visualization. Therefore, we compared renderings from the original data (6561 images) with renderings obtained as reconstructions from the proposed model. Our implementation in the OpenGL evaluates the model at each pixel of the rough texture mapped on a triangulated 3D model. After conducting the practical experiments we set the numbers of components to $M = 600$. Note that M is related to spatial visual complexity of material structure rather than to pixel count. Results on the five BTF samples are shown in Fig. 3, where the first row (a) shows original data rendering and the second row (b) shows reconstruction by the model. Fig. 3 also shows tile sizes and compression ratios for the tested BTF tiles achieved by the proposed GM model's parametric representation. Note that the ratios include the model's parameters as well as mean BRDF data. Additionally, in Fig. 3-c we show differences between original and modeled images using RMSE, PSNR [dB] and SSIM metrics, as well as the difference image in the selected inset window. It is apparent that although the model tends to reliably match rough texture structure, in its current state it does not reproduce well all of the fine high-frequency details,

[1] http://btf.cs.uni-bonn.de/

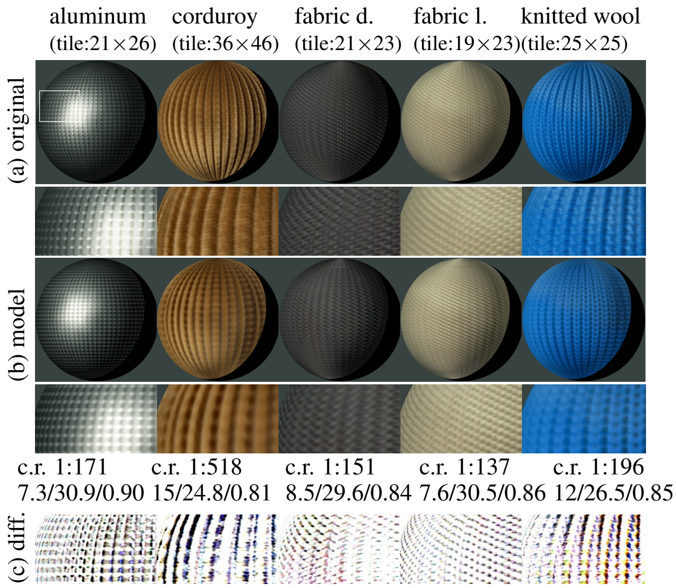

Fig. 3. The model's performance on five BTF samples – aluminum profile, corduroy, dark fabric, light fabric, and knitted wool. Individual rows show: (a) rendering of measured BTF data, (b) the proposed model (M=600), (c) 10 $\times$ difference images and averaged RMSE / PSNR[dB] / SSIM values.

as is visible, e.g., on the corduroy material. The current im-

Table 1. Performance of the LPCA method [11].

material	tilesize [pixels]	C.R.	RMSE/PSNR/SSIM
alu	21×26	1:18	1.9 / 42.5 / 0.99
corduroy	36×46	1:55	4.6 / 34.9 / 0.94
fabric dark	21×23	1:16	7.3 / 30.9 / 0.85
fabric light	19×23	1:10	2.0 / 42.2 / 0.98
knitted wool	25×25	1:21	3.3 / 37.9 / 0.95

plementation of our method has a supreme data-compression ratio, but its reconstruction accuracy is lower than PCA-based methods (see compression and visual quality reconstruction of the local PCA-based method [11] in Tab. 1); nevertheless, the achieved visual results are encouraging for further development. The compression ratio achieved for the *knitted wool* material (25×25 = 625 pixels) as compared with compression ratios achieved by previous approaches [5] are shown in Tab. 2.

Table 2. Compression ratios achieved by the GM model in comparison with previous approaches to the rough texture compression.

method (BTF tile:25×25 pix.)	C.R.
Gaussian mixture model	195.6
Polynomial texture maps (per-view) PTM RF	13.5
Polynomial Lafortune (per-view) PLM RF	14.3
PCA factorization (per-view) PCA RF	11.2
PCA factorization (entire data) PCA BTF	23.8
Local PCA clustering (entire data) LPCA BTF	21.4
Probabilistic GMRF model (tiled range-map)	600.0
Probabilistic 2D CAR model (tiled range-map)	800.0
Probabilistic 3D CAR model (tiled range-map)	100.0

The graph in Fig. 4 shows different methods' compression ratios dependency on number of BTF pixels. Although this dependency is linear in the Local PCA approach [11], which is also true for the proposed model, the compression ratios of our model are far higher given the same pixel count. Note, that the compression ratios of GMRF, 2D CAR, and 3D CAR methods have this dependency linear as well beyond the size of tiled height-map. Although their compression ratio is higher their visual quality is compromise due to height simulation of regular structures, that cannot handle non-local effects such as inter-reflections, translucency, or scattering.

Note that 500 components (M) require us to store M $\times$ 9 (variables) $\times$ 2 (μ and σ) = 9000 floating point values, in addition to the mean BRDF image represented by 81×81×3 = 6561×3 byte values, which in the case of a modest size of BTF data ($256 \times 256 \times 81 \times 81 \times 3 = 1.3 \cdot 10^9$) represent a BTF sample compression ratio of 1:45 323 (for HDR 32-bits/channel) and 1:22 989 (for LDR 8-bits/channel).

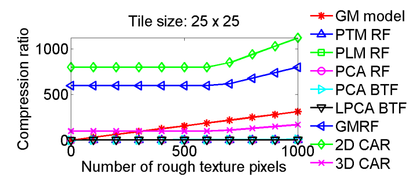

Fig. 4. Dependency of compression ratios of the compared methods on size of the rough texture.

The model's parameter fitting for BTF of size 64×64 using 600 components typically takes 5 hours in 40 iterations of the EM algorithm on Intel Xeon 2.7GHz using a single core and non-optimized C++ implementation. Image rendering for resolution of 800×800 in model using 600 components takes an average of 30 seconds using non-optimized CPU implementation. These times can be further substantially reduced by the EM algorithm implementation on multiple cores or a GPU.

5. CONCLUSIONS

This paper outlines a novel method for high-dimensional rough texture parameterization and compression. The method starts with data normalization and their modeling with the aid of a Gaussian mixtures model. This model allows full-color BTF reconstruction based on analytic evaluation for any combination of spatial and directional variables. Moreover, the size of the model's parametric representation is very compact, and its compression ratio is linearly dependent on a rough texture resolution. We are not aware of any parametric BTF model that would provide such a compact parametric representation together with the similar visual performance as the model proposed here. Although the model's visualizations do not convey all fine details in a material's structure yet, its main visual features are preserved accurately. In our future work we will attempt to improve implementation and input data processing steps to enhance speed and quality of the model's fitting and apply them to adaptive data measurement.

6. REFERENCES

[1] J. Filip and R. Vávra, "Fast method of sparse acquisition and reconstruction of view and illumination dependent datasets," *Computers & Graphics*, vol. 37, no. 5, pp. 376–388, August 2013.

[2] K.J. Dana, B. van Ginneken, S.K. Nayar, and J.J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, 1999.

[3] V. Havran, J. Filip, and K. Myszkowski, "Bidirectional texture function compression based on the multilevel vector quantization," *Computer Graphics Forum*, vol. 29, no. 1, pp. 175–190, 2010.

[4] F.E. Nicodemus, J.C. Richmond, J.J. Hsia, I.W. Ginsburg, and T. Limperis, "Geometrical considerations and nomenclature for reflectance," *NBS Monograph 160*, pp. 1–52, 1977.

[5] J. Filip and M. Haindl, "Bidirectional texture function modeling: A state of the art survey," *IEEE TPAMI*, vol. 31, no. 11, pp. 1921–1940, 2009.

[6] M.L. Koudelka, S. Magda, P.N. Belhumeur, and D.J. Kriegman, "Acquisition, compression, and synthesis of bidirectional texture functions," in *Texture 2003*, 2003, pp. 47–52.

[7] Mirko Sattler, Ralf Sarlette, and Reinhard Klein, "Efficient and realistic visualization of cloth," in *Eurographics workshop on Rendering*, 2003, pp. 167–177.

[8] M.A.O. Vasilescu and D. Terzopoulos, "TensorTextures: Multilinear image-based rendering," *ACM SIGGRAPH 2004*, vol. 23, no. 3, pp. 336–342, August 2004.

[9] H. Wang, G. Wu, L. Shi, Y. Yu, and N. Ahuja, "Out-of-core tensor approximation of multi-dimensional matrices of visual data," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 527–535, July 2005.

[10] Roland Ruiters, Christopher Schwartz, and Reinhard Klein, "Data driven surface reflectance from sparse and irregular samples," *Computer Graphics Forum (Proc. of Eurographics)*, vol. 31, no. 2, pp. 315–324, May 2012.

[11] G. Müller, J. Meseth, and R. Klein, "Compression and real-time rendering of measured BTFs using local PCA," in *Vision, Modeling and Visualsation*, 2003, pp. 271–280.

[12] D. K. McAllister, A. Lastra, and W. Heidrich, "Efficient rendering of spatial bi-directional reflectance distribution functions," *Graphics Hardware*, pp. 77–88, 2002.

[13] J. Filip and M. Haindl, "Non-linear reflectance model for Bidirectional Texture Function synthesis," in *ICPR 2004*, August 2004, pp. 80–83.

[14] Hongzhi Wu, Julie Dorsey, and Holly Rushmeier, "A sparse parametric mixture model for BTF compression, editing and rendering," *Computer Graphics Forum*, vol. 30, no. 2, pp. 465–473, 2011.

[15] N. Menzel and M. Guthe, "g-BRDFs: An intuitive and editable BTF representation," *Computer Graphics Forum*, vol. 28, no. 8, pp. 2189–2200, 2009.

[16] Sebastian Magda and David Kriegman, "Reconstruction of volumetric surface textures for real-time rendering," in *Eurographics Workshop on Rendering*, June 2006, pp. 19–29.

[17] M Haindl and J Filip, "Extreme compression and modeling of bidirectional texture function," *IEEE TPAMI*, vol. 29, no. 10, pp. 1859–1865, 2007.

[18] M. Haindl, J. Filip, and M. Arnold, "BTF image space utmost compression and modelling method," in *ICPR 2004*, August 2004, vol. 3, pp. 194–197.

[19] M. Haindl and J. Filip, "Fast BTF texture modelling," in *Texture 2003*, 2003, pp. 47–52.

[20] M. Haindl, J. Grim, P. Somol, P. Pudil, and M. Kudo, "A Gaussian mixture-based colour texture model," in *ICPR 2004*, August 2004, vol. III, pp. 177–180.

[21] M. Haindl, J. Grim, P. Pudil, and Kudo M., "A hybrid BTF model based on gaussian mixtures," in *Texture 2005*, October 2005, pp. 95–100.

[22] P. Somol and M. Haindl, "Novel path search algorithm for image stitching and advanced texture tiling," in *WSCG 2005*, 2005, pp. 155–162.