

Efficient Image-Based Bidirectional Texture Function Model

Jiří Filip Michal Haindl
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
182 08 Prague 8, Czech Republic
{filipj,haindl}@utia.cas.cz

Abstract

Recent advances in computer hardware and virtual modelling allow to respect view and illumination dependencies of natural surface materials. The corresponding texture representation in the form of Bidirectional Texture Function (BTF) enables significant improvements of virtual models realism at the expense of immense increase of material sample data space. Thus introduction of some fast compression, modelling and rendering method for BTF data is inevitable. In this paper we introduce a generalisation of our polynomial extension of the Lafortune model computed for every original BTF measurement pixel allowing seamless BTF texture enlargement. This nonlinear reflectance model is further extended using parameters clustering technique to achieve higher compression ratio. The presented method offers BTF modelling in excellent visual quality as was tested on variety of BTF measurements. The method gives BTF compression ratio $\sim 1:200$ as well as fast graphics hardware implementation.

1 Introduction

Realistic models of virtual reality require, among others, naturally looking textures covering virtual objects of a rendered scene. Applications of these advanced texture models allow photorealistic material appearance approximation for such complex tasks as visual safety simulations or interior design in automotive / airspace industry or architecture.

One of the early attempts to capture real material appearance was done recently by Dana et al. [2] in the form of *Bidirectional Texture Function* (BTF). BTF is a seven-dimensional function which accounts also for viewing and illumination measurement dependency on planar material position.

$$BTF(r_1, r_2, r_3, \theta_i, \phi_i, \theta_v, \phi_v) \quad (1)$$

where θ, ϕ are elevation and azimuthal angles of illumination and view direction vector (see Fig. 2), r_1, r_2 specify

planar horizontal and vertical position in material sample image and r_3 is the spectral index .



Figure 1. Two examples of car interior modelling. Images illustrate seven different materials approximated by means of proposed reflectance BTF model (PLM-C). Mercedes-Class C 3D model courtesy of DaimlerChrysler.

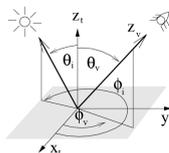


Figure 2. Relationship between illumination and viewing angles within texture coordinate system.

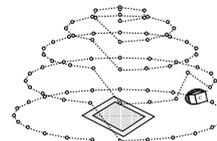


Figure 3. Light vector trajectory above the sample. The light starts at the top.

BTF measurement is very time consuming and such systems require high precision measuring setup hence only few such systems exist up to now [2, 8, 18]. BTF appropriately measured from real material samples offers enough information about material properties, e.g., anisotropy, masking or self-shadowing. In contrast to a regular 2D texture or even to BRDF, BTF is high-dimensional and involves large amounts of data. To render BTF on graphics hardware, its compact representation is needed. Thus BTF database even for simple VR scenes require enormous data space (TB). Some compression and modelling method of these huge datasets is inevitable. Such a method should provide compact parametric representation and preserve main visual fea-

tures of the original BTF, while enabling its fast rendering taking advantage of the contemporary graphics hardware.

The first group of BTF modelling methods represents BTF by means of pixel-wise analytical BRDF models. McAllister et al. [14] represented BRDF of each pixel in BTF using the Lafortune reflectance model (LM) [10]. An extension of this approach was published by Daubert et al. [3]. Spatial inconsistency of individual pixels in BTF for different view directions led into separate modelling of individual view positions. Malzbender et al. [13] represented each pixel in BTF by means of per-pixel polynomials. Meseth et al. [15] exploited pixel-wise LM computed for each view separately.

The second type of BTF compression methods is based on standard PCA statistical approach. Koudelka et al. [8] ordered individual BTF sub-images into vectors of matrix. The corresponding symmetric matrix was created and subsequently decomposed using SVD. Authors preserved 150 main eigen-images for satisfactory BTF reconstruction. Vasilescu et al. [20] decomposed BTF space, ordered into tensor, by means of multi-modal SVD. This method enables controllable BTF compression separately in viewing and illumination axes and demonstrates better performance with the same number of components as the previous approach. Even though, both methods enable realistic BTF rendering, they are not suitable for fast rendering application since they require to compute the linear combinations of high number of eigen-components. A much faster approach was presented by Sattler et al. [18]. Authors computed PCA for individual reflectance fields \mathcal{R}_v instead of the whole BTF dataspace. This approach resulted in 16 eigen-images per one view position, which can be easily interpolated by means of graphics hardware. Müller et al. [16] exploited a vector quantisation of BTF data-space and each resulted cluster was represented by a local PCA model. The method combining sparse set of BTF measurements according to enlarged material range-map was developed by Liu et al. [11]. The same author presented later in [12] another model similar to [16]. This method exploits technique of 3D textons, i.e., the smallest repeatable texture elements. Only these textons are then approximated using local PCA and finally used for a surface modelling. The main drawback of the above mentioned methods is that they do not allow BTF synthesis of arbitrary size, i.e. the texture enlargement. Finally a group of probabilistic BTF models was recently proposed [6], [7]. These methods allow unlimited texture enlargement, BTF texture restoration, huge BTF space compression and even modelling of unseen BTF data. However, such models are non trivial and they suffer with several unsolved mathematical problems.

In this paper we present a generalisation of our reflectance model [5] enabling BTF enlargement and significant improvement of BTF compression ratio.

This paper is organised as follows. The proposed BTF model is described in Section 2, the parameter space enlargement is described in Section 3 and the parametric space compression using the K-means clustering is subject of Section 4. Following sections shows results of the proposed model, discusses its properties and concludes the paper.

2 Reflectance BTF Model

The BTF measurements comprehend the whole hemisphere of light and camera positions in the observed material sample coordinates according to selected quantisation steps. We used the Bonn University BTF dataset [18]. Fig.3 illustrates directional illumination source course above the sample for fixed view position. This BTF dataset contains 6561 images per texture sample (81 view and 81 illumination positions). All images are rectified to head-on view position ($\theta_v = 0^\circ, \phi_v = 0^\circ$) and resampled to obtain normal-textures of size 800×800 pixels. Different view measurements suffer with registration errors even after the rectification process due to self-occlusion. Therefor the correct way how to model such a real measurements is to model each BTF subset comprehending all images obtained for a fixed view position. Such a BTF subspace for a view direction ω_v is a 5D function called *Surface 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.

Single surface reflectance fields (\mathcal{R}) (i.e., 81 images - n_i) for an actual view position v can be per-pixel modelled using n_l -lobe Lafortune model [10] described in the formula

$$Y_v(r, i) = \sum_{k=1}^{n_l} \rho_{k,v,r} (\omega_i^T \mathbf{D}_{k,v,r})^{n_{k,v,r}} \quad (2)$$

where $\omega_i(\theta_i, \phi_i) = [u_x, u_y, u_z]^T$ is a unit vector pointing to light and parametrised by the illumination elevation and azimuthal angles $[\theta_i, \phi_i]$ respectively (see Fig. 2). As a reflectance data the set of pixels $\mathcal{R}_v(r_1, r_2, r_3, \omega_i)$ is considered, where $i = 1, \dots, n_i$ is the illumination position index and v is the actual view position index ranging from 1 to n_v .

The representation using this model (2) is compact and memory efficient as each reflectance lobe is determined by means of only five parameters ρ, D_x, D_y, D_z, n . Pixel-wise one-lobe LM simplifies to

$$Y_{i,v}(r) = \rho_v(r) [D_{v,x}(r)u_x + D_{v,y}(r)u_y + D_{v,z}(r)u_z]^{n(r)} \quad (3)$$

For every planar position and spectral channel in BTF all the model parameters are estimated using the Levenberg-Marquardt non-linear optimisation algorithm [17].

Although the original Lafortune model requires to store only five parametric planes for each spectral channel per lobe, its reflectance accuracy is erroneous for some combinations of illumination and viewing angles. Even using

more than one lobe, which is very time consuming process, does not solve this problem. For this reason we generalised this model with additional fitting scheme based on histogram matching technique adopted and extended for BTF data and polynomial fitting as illustrated on the scheme in Fig. 4. At the beginning the image cumulative histograms

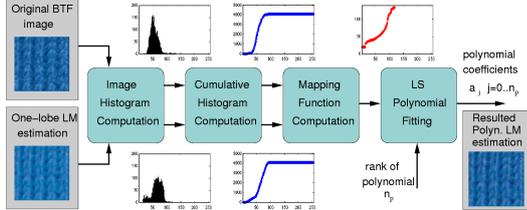


Figure 4. Procedure of polynomial coefficients computation.

in all spectral channels are computed for both original BTF image and its one-lobe LM estimation. These two cumulative histograms are inputs to the histogram matching algorithm giving mapping function from estimated image using one-lobe LM to the original BTF image. The resulted mapping function between both cumulative histograms is approximated by means of polynomial using a least squares fitting scheme to obtain polynomial coefficients $a_{r_3,v,i,j}$. These coefficients are computed and stored for individual colour channels of every BTF image.

The proposed polynomial extension of one-lobe LM (PLM) (3) using coefficients $a_{r_3,v,i,j}$ results in a novel model expressed by the following formula

$$\tilde{Y}_{i,v}(r) = \sum_{j=0}^{n_p} a_{r_3,v,i,j} Y_{i,v}(r)^j, \quad (4)$$

where $a_{r_3,v,i,j}$ are polynomial parameters specifying mapping function between histogram values of image $Y_{i,v}(r)$ synthesised from one-lobe LM's parameters and original BTF image and $(n_p - 1)$ is a rank of this polynomial. Satisfactory results were obtained already with $n_p = 5$. Thus additional fifteen float numbers have to be stored with each BTF image which are negligible to the size of LM parametric planes.

Besides, individual mapping functions can be efficiently interpolated with respect to illumination direction so during final rendering the BTF synthesis is obtained by means of barycentric interpolation of PLM results for three closest measured view directions v only.

3 Parametric Planes Enlargement

Reflectance models can only model previously measured BTF pixels. Thus some form of the model parameter planes

enlargement is inevitable when an object has to be covered by BTF. A simple seamless one parametric tile repetition can provide satisfactory solution for regular types of textures. Non-regular textures, such as skin or wood, require more elaborated enlargement approach such as random field based synthesis methods [7] or advanced sampling methods. There is a variety of image-based texture sampling methods published recently [4, 1]. In this paper we use the image tiling method based on the image stitching introduced in [19]. The idea of stitching is based on the minimum error boundary cut. The principle of the stitching procedure is demonstrated in Fig. 5. The minimum



Figure 5. Image stitching. The source image is cropped from the right along the minimum error path and placed over the target background image.

sub-optimal [19] error path is constructed to lead through the error map which represents the visual difference between source and target for each pixel of the overlapping region. This algorithm is used as a fast alternative to the slow optimal path search procedures (e.g., using the dynamical programming). The algorithm has linear complexity $O(kn)$ with n depicting the number of pixels and k the number of user-specified algorithm iterations. In contrary, the most effective optimal algorithm implementations achieve $O(n \log n)$. This method is a step-wise procedure that sequentially improves some actual solution and thus it can be stopped at any moment to yield a usable result. If the algorithm is unable to find a good path through the error map, resulting visible artifacts are diminished by the adaptive boundary blending of individual paths. The idea is to interpolate between the overlapped source region to the target with a locally adjusted intensity while utilising the minimum error path. Additional tiles can be created by making a copy of the template tile and subsequently covering its inner area by patches taken from different positions in the source texture image. BTF tiling is complex task as the stitch should appear consistent in all BTF planes. To decrease the computational complexity of such an extensive data processing we adopted a two-stage process. In the first stage we only determine the stitching and other parameters to be used later for actual tile creation. For this purpose only a few sample parametric images are taken (prepared in full size, i.e., 800×800 pixels) to represent different azimuthal and elevation view positions. The optimal stitching paths are found in this subset of parametric images. In the second stage the complete parametric BTF data are processed using the pre-computed stitching parameters. Once all tiles

become available, the final parameters of the proposed LM are computed based on the parametric tiles. This procedure saves considerable computational demands of Lafortune parameters estimation algorithm.

4 Compression of Parametric Planes

Using the BTF tiling approach of PLM parametric planes described above we were able to achieve the maximal compression ratio of a real BTF data about $\frac{1}{20}$ depending on the resolution of parametric tiles. This compression is insufficient because it still requires to store several hundreds megabytes of data per material. To reduce the storage space while maintaining the computational cost and keeping visual quality almost the same, the parameter clustering was incorporated into the model. The individual PLM parametric planes for each reflectance field are segmented and only cluster indices and model parameters corresponding to the individual clusters are saved for each RGB spectrum. The number of clusters for each spectrum is set to 256 to enable reproduction of 256 different grayscale levels. Thus the theoretical number of colour hues within this setup is 256^3 .

The whole PLM parameter segmentation procedure is performed for each reflectance field \mathcal{R}_v separately and it works as follows. At the beginning the K-means segmentation algorithm is employed using original pixels from all 81 images corresponding to actual \mathcal{R}_v as data features. The segmentation cannot be performed directly on model parameters as these individual parameters have strong non-linear impact on the restored pixel value and any general weights cannot be attached to them.

The K-means segmentation process is computationally very demanding and the segmentation of 10 parametric tiles of resolution 64×64 for all 81 reflectance fields takes several hours. To reduce the computational time we have decreased the size of feature vectors from 81 to approximately 20. To choose an appropriate subset of images bearing the most different information from the already selected images we used an algorithm based on the Kullback-Leiber distance [9] between histograms of individual 81 BTF images.

When the segmentation is finished we obtain cluster indices $I_v(r_1, r_2, r_3)$ for the individual colour spectra r_3 of each reflectance field \mathcal{R}_v . Cluster indices are stored in form of colour images of original parameter images resolution, i.e., in each colour channel we store the corresponding cluster index. An important product of segmentation is the table containing individual cluster centers $K_v(c)$ where c is the cluster index. For each cluster five PLM parameters are stored for individual colour channels.

The final synthesis is straightforward. The parameters ρ, D_X, D_Y, D_Z and n of the original model (3) are computed as

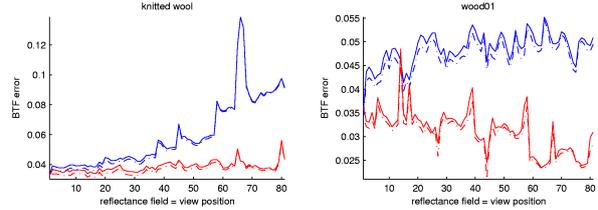


Figure 6. The mean average error (MAE) of clustered one-lobe lafortune model (LM-C – blue line) and its clustered polynomial extension (PLM-C – red line) compared with non-clustered variants LM and PLM (dash-dot lines) for all 81 reflectance fields of two BTFs: *wool* and *wood01*.

$$\begin{aligned} \rho(r)_v &= K_{v,1}(I_v(r)) & D(r)_{v,X} &= K_{v,2}(I_v(r)) \\ n(r)_v &= K_{v,5}(I_v(r)) & D(r)_{v,Y} &= K_{v,3}(I_v(r)) \\ & & D(r)_{v,Z} &= K_{v,4}(I_v(r)) \end{aligned} .$$

We refer to this clustered polynomial extension of the reflectance model as PLM-C in the following text. The synthesis based on the described approach is quite fast, requiring the look-up index tables only which can be implemented using standard OpenGL features.

Using this approach the storage size of model parameters reduces considerably since only one colour parametric look-up image and several cluster parameters have to be stored (check the columns 4,5 Tab. 2). The rendering speed for PLM-C is higher than for PLM since only 256 clusters (pixels) have to be computed for each spectral channel instead of five parametric planes.

Table 1. The MAE of the synthesised BTFs for one-lobe Lafortune model (LM), its polynomial extension (PLM) and clustered polynomial extension (PLM-C).

		Mean Average Error		
<i>material</i>		<i>LM</i>	<i>PLM</i>	<i>PLM-C</i>
	wool	0.058	0.037	0.038
	proposte	0.054	0.052	-
	fabric01	0.058	0.036	0.038
	fabric02	0.053	0.032	0.033
	foil01	0.067	0.021	0.023
	foil02	0.048	0.020	0.023
	leather02	0.032	0.018	0.021
	wood01	0.047	0.030	0.031
	wood02	0.058	0.035	0.038

5 Results

For the sake of the BTF results comparison the standard mean average pixel-wise error (MAE) between original data (Y) and estimated data (\hat{Y}) was used. Fig. 6 shows the error curves (MAE) for individual test materials. For each material the MAE is computed for all 81 view positions \mathcal{R}_v (depicted on x axis) of clustered one-lobe Lafortune model (LM-C, blue solid line) and its clustered polynomial extension (PLM-C, red solid line) are compared with the corre-

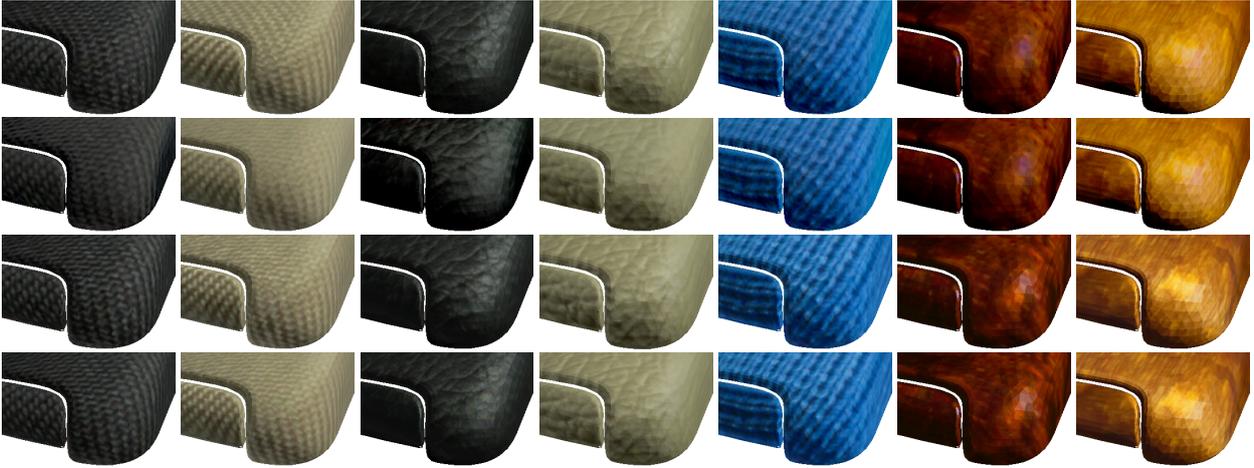


Figure 7. A part of a car armrest covered with BTFs. Tiled original BTF data (first row), results of one-lobe LM (second row), proposed one-lobe PLM (third row) and proposed one-lobe PLM-C (fourth row) for seven different materials: *fabric01*, *fabric02*, *foil01*, *foil02*, *knitted wool*, *wood01*, *wood02*.

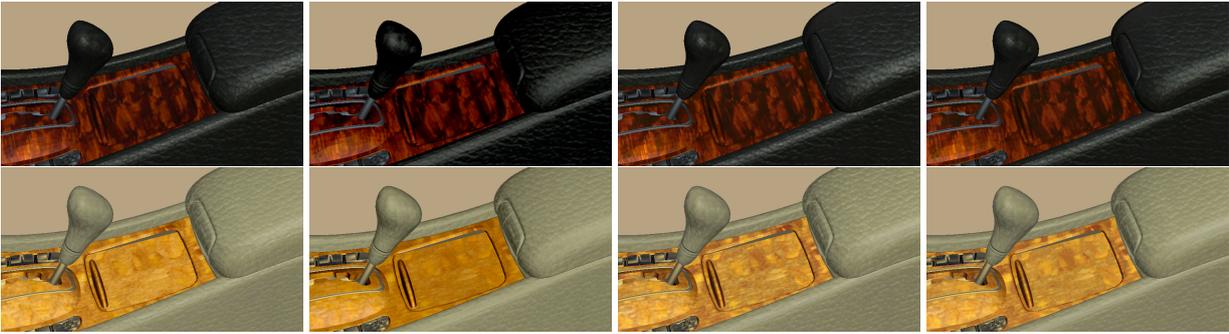


Figure 8. A part of car gearbox covered using four BTFs: *foil01*, *wood01* and *foil02*, *wood02*. The first column illustrates original tiled BTF data, the second column depicts approximation using one-lobe LM, the third column is result of proposed one-lobe PLM model, the fourth column shows result of proposed clustered PLM-C model.

Table 2. The storage size of the proposed PLM and PLM-C in comparison with size of the raw BTF data and their tiled representation.

<i>material</i>	storage size in MegaBytes				<i>tile size</i> [<i>pixels</i>]
	<i>raw BTF</i>	<i>10 BTF tiles</i>	<i>PLM</i>	<i>PLM-C</i>	
wool	733.3	103.4	33.5	4.3	25×25
fabric01	6766	87.1	24.9	2.9	21×23
fabric02	5863	77.5	24.1	4.0	19×23
foil01	5190	728.1	406.8	19.2	86×96
foil02	5065	527.5	296.7	13.8	74×79
leather02	5074	659.7	381.0	18.6	86×87
wood01	5330	1333.2	771.8	31.8	122×125
wood02	5083	2405.0	973.4	29.1	137×142

sponding non-clustered variants of LM and PLM (both depicted as dash-dot line). Individual reflectance fields are ordered according to camera position circular movement from

top to bottom of a hemisphere above the observed material as illustrated in Fig. 3. The overall MAE values of all tested materials were computed as averaged MAE of all reflectance fields and are shown in Tab. 1 in contrast to the corresponding values of non-clustered PLM. The MAE for PLM-C is slightly higher in comparison with PLM but this higher error is well counterbalanced by the model size. The number of parameters to be stored have been reduced using the proposed parameter clustering at least ten times in contrast to the non-clustered PLM as it is evident from the fifth column of Tab. 2. The storage size of one-lobe LM is almost the same as PLM one. The tile resolutions for individual materials are listed in this table as well. A part of a car armrest in the Fig. 7 is covered by seven tested BTFs approximated by means of the proposed models. The first row represents armrest covered by the original tiled BTF measurements, while the second row represents BTF data approximated using the one-lobe LM, the third

row shows results of the proposed one-lobe PLM and finally the fourth row shows result of the proposed clustered one-lobe PLM-C. According to graphs in Fig. 6 the images in the second row are dim and less contrast (materials *fabric02*, *foil02*) losing information in dark parts as it is apparent, e.g., for *foil01* material in comparison with the original BTF data in the first row. Similarly a part of car gearbox is covered by several distinct BTFs in Fig.8. One can observe obviously better preservation of visual quality while the memory and time requirements of proposed method are comparable to simple pixel-wise one-lobe Lafortune model. Fig. 1 depicts two distinct examples of car interior covered by seven different BTFs.

6 Conclusions

The proposed BTF modelling approach is based on the polynomial extension and the texture enlargement generalisation of the pixel-wise Lafortune reflectance model computed for individual spectral channel for every pixel. This model uses only one reflectance lobe while the remaining fitting is done by means of the polynomial extension of one-lobe Lafortune model. Using of one-lobe model considerably reduces the number of model parameters which have to be stored. Moreover, the memory requirements of introduced polynomial coefficients are negligible in comparison to Lafortune parameters. The proposed reflectance model has similar computational requirements as the pixel-wise one-lobe Lafortune model while using only few additional linear operations so it can be easily implemented in current advanced graphics hardware. To enlarge BTF textures to arbitrary size we apply our sampling based method to model parametric planes. The model's BTF compression ratio even more increased using parametric clustering which enables ratios $\sim \frac{1}{2 \cdot 10^2}$ whereas the computational requirements remain similar. The results of this model show its excellent visual performance for all tested BTFs even for materials with complicated underlying structure.

6.1 Acknowledgments

This research was supported by the EC projects no. IST-2001-34744 RealReflect, FP6-507752 MUSCLE, the GAAV grants No.A2075302, 1ET400750407 and partially by the MŠMT grant 1M6798555601 DAR.

References

- [1] M. Cohen, J. Shade, S. Hiller, and O. Deussen. Wang tiles for image and texture generation. In *ACM SIGGRAPH 2003*, *ACM Press*, volume 22, pages 287–294, New York, NY, USA, July 2003.
- [2] K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.
- [3] K. Daubert, H. P. A. Lensch, W. Heidrich, and H.-P. Seidel. Efficient cloth modeling and rendering. In *Proceedings of the 12th Eurographics Workshop on Rendering*, 2001.
- [4] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In E. Fiume, editor, *ACM SIGGRAPH 2001*, *ACM Press*, pages 341–346, 2001.
- [5] J. Filip and M. Haindl. Non-linear reflectance model for bidirectional texture function synthesis. In *Proceedings of 17th ICPR*, volume 1, pages 80–84. IEEE Computer Society Press, August 2004.
- [6] M. Haindl and J. Filip. A fast probabilistic bidirectional texture function model. In *Proceedings of ICIAR. (Lecture Notes in Computer Science. 3212)*, volume 2, pages 298–305, Berlin Heidelberg, September 2004. Springer-Verlag.
- [7] M. Haindl, J. Filip, and M. Arnold. BTF image space utmost compression and modelling method. In *Proceedings of 17th ICPR*, volume 3, pages 194–198. IEEE Computer Society Press, August 2004.
- [8] M. Koudelka, S. Magda, P. Belhumeur, and D. Kriegman. Acquisition, compression, and synthesis of bidirectional texture functions. In *Texture 2003*, pages 47–52, October 2003.
- [9] S. Kullback. *Information Theory and Statistics*. Dover Books, New York, USA, 1997.
- [10] E. P. Lafortune, S. C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. *Computer Graphics*, 31(Annual Conference Series):117–126, 1997.
- [11] X. Liu, Y. Yu, and H. Y. Shum. Synthesizing bidirectional texture functions for real-world surfaces. In *ACM SIGGRAPH 2001*, *ACM Press*, pages 97–106, 2001.
- [12] X. Liu, J. Zhang, X. Tong, B. Guo, and H.-Y. Shum. Synthesis and rendering of bidirectional texture functions on arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(3):278–289, June 2004.
- [13] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *ACM SIGGRAPH 2001*, *ACM Press*, pages 519–528. Eurographics Association, Switzerland, 2001.
- [14] D. K. McAllister, A. Lastra, and W. Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. *Graphics Hardware*, pages 77–88, 2002.
- [15] J. Meseth, G. Müller, and R. Klein. Preserving realism in real-time rendering of bidirectional texture functions. In *OpenSG Symposium 2003*, pages 89–96. Eurographics Association, Switzerland, April 2003.
- [16] G. Müller, J. Meseth, and R. Klein. Compression and real-time rendering of measured BTFs using local PCA. In *Proceedings of Vision, Modeling and Visualisation '03*, November 2003.
- [17] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.
- [18] M. Sattler, R. Sarlette, and R. Klein. Efficient and realistic visualization of cloth. In *Eurographics Symposium on Rendering 2003*, June 2003.
- [19] P. Somol and M. Haindl. Novel path search algorithm for image stitching and advanced texture tiling. In *Proceedings of 13-th WSCG05*, February 2005.
- [20] M. Vasilescu and D. Terzopoulos. TensorTextures: Multilinear image-based rendering. *ACM SIGGRAPH 2004*, *ACM Press*, 23(3):336–342, August 2004.