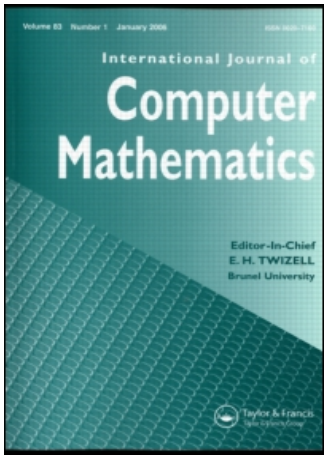

International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713455451

BTF modelling using BRDF texels

Jiří Filip [a]; Michal Haindl [a]
[a] Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague, Czech Republic

Online Publication Date: 01 September 2007
To cite this Article: Filip, Jiří and Haindl, Michal (2007) 'BTF modelling using BRDF texels', International Journal of Computer Mathematics, 84:9, 1267 - 1283
To link to this article: DOI: 10.1080/00207160701253802
URL: http://dx.doi.org/10.1080/00207160701253802

International Journal of Computer Mathematics
Vol. 84, No. 9, September 2007, 1267–1283

BTF modelling using BRDF texels

JIŘÍ FILIP* and MICHAL HAINDL

Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague, Czech Republic

(Received 20 September 2006; revised version received 08 December 2006; accepted 22 January 2007)

The highest fidelity representations of realistic real-world materials currently comprise Bidirectional Texture Functions (BTF). The BTF is a six-dimensional function depending on view and illumination directions as well as on planar texture coordinates. The huge size of such measurements, typically in the form of thousands of images covering all possible combinations of illumination and viewing angles, has prohibited their practical exploitation, and obviously some compression and modelling method of these enormous BTF data spaces is inevitable. The two proposed approaches combine BTF spatial clustering with cluster index modelling by means of efficient Markov random field models. The methods allow the generation of a seamless cluster index of arbitrary size to cover large virtual 3D object surfaces. Both methods represent original BTF data using a set of local spatially dependent Bidirectional Reflectance Distribution Function (BRDF) values which are combined according to the synthesized cluster index and illumination/viewing directions by means of two types of Markov random field models. BTF data compression using both methods is about 1:200 and their synthesis is very fast.

Keywords: Markov random fields; Image modelling; Pattern recognition; Texture synthesis; Virtual reality

AMS Subject Classifications: 68U10; 68U05

1. Introduction

Recent progress in graphics hardware computational power has finally enabled the fast and visually realistic rendering of virtual reality models that, until recently, was impossible. Such realistic models require, among other things, naturally looking textures covering virtual objects of the rendered scene. Applications of these advanced texture models in virtual reality systems now allow photo-realistic material appearance approximation for complex tasks such as visual safety simulations or interior design in the automotive/airspace industry or architecture.

For such advanced applications, standard textures lit by reflectance models alternatively combined with bump-mapping are not able to offer correct and realistic reproduction of the material appearance. This is caused by the inherited complexity of many materials, the rough structure of which produces visual effects such as self-shadowing, masking, inter-reflection or

*Corresponding author. Email: jiri.filip@utia.cz

International Journal of Computer Mathematics
ISSN 0020-7160 print/ISSN 1029-0265 online
http://www.tandf.co.uk/journals
DOI: 10.1080/00207160701253802

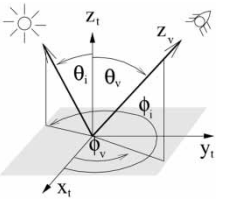

Figure 1. Relationship between illumination and viewing angle within the texture coordinate system.

subsurface scattering. One way to capture these materials' attributes is to use a much more complex representation of the rough or 3D texture called the Bidirectional Texture Function (BTF). Different from four-dimensional Bidirectional Reflectance Distribution Function (BRDF), BTF is a six-dimensional reflectance function depending on view and illumination directions as well as on planar texture coordinates, as illustrated in figure 1. This function is typically acquired in the form of several thousands of images covering varying light and camera directions, as shown in figure 2. However, the huge size of the measured BTF data prevents their use in any useful application; therefore the introduction of a fast compression and modelling method for BTF data is inevitable.

The majority of the results in the BTF area mainly deal with compression. They are either based on the eigen-analysis of the BTF data space [1–4] or on applications of pixel-wise reflectance models [5–8]. Although these methods can provide reasonable compression ratios (1/20–1/200) and visual quality, their main drawback is that they do not allow arbitrary size BTF synthesis, i.e. BTF texture enlargement.

To solve this problem, BTF modelling methods are necessary. Such methods are not only capable of enlarging the BTF space as required, but they simultaneously imply a compression capability as well. Unfortunately, only a few BTF enlargement approaches have been published. The majority of the available methods are based either on simple texture repetition with edge blending or on more or less sophisticated image tiling methods [9–13], and some of them can also be adapted for BTF synthesis (e.g., [14, 15]).

Finally, a group of probabilistic BTF models has recently been proposed [16, 17]. These methods allow unlimited texture enlargement, BTF restoration, huge BTF space compression and even the modelling of previously unseen BTF data. They are based on rough BTF segmentation in a space of illumination and viewing directions. The individual cluster representatives are the BTF images closest to cluster centres, which are combined with an estimated range-map in a bump-mapping filter for the required illumination and viewing angles. Although these

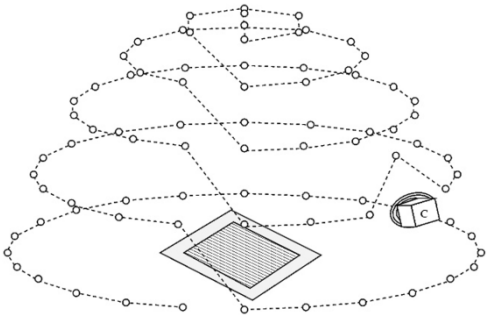

Figure 2. Illumination directions ($i = 1, \ldots, 81$) in the BTF data used. The viewing directions ($v = 1, \ldots, 81$) are the same.

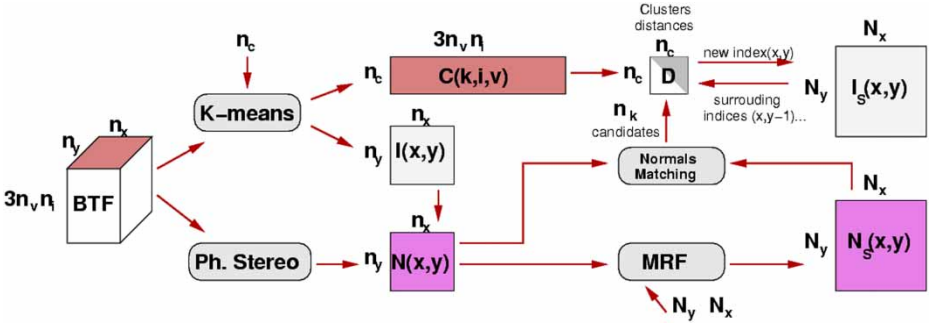

Figure 3. The overall scheme of the proposed BTF enlargement method.

methods achieve hugely impressive compression ratios, they sometimes compromise visual quality for certain materials.

In this paper we present a novel BTF model enabling seamless enlargement as well as the compression of BTF data. The overall scheme of the proposed model is illustrated in figure 3.

The method starts with normal-map estimation of the underlying material surface using photometric stereo. The estimated normal-map **N** is enlarged to the required size using a probabilistic MRF model. In the following step the original BTF data are clustered in the planar space. The results are cluster representatives **C** and cluster index **I**, which is used for new cluster index $\mathbf{I}_S$ generation up to the required size of the synthesized normal-map $\mathbf{N}_S$. This enlargement exploits the matching between the estimated **N** and synthesized $\mathbf{N}_S$ normal-maps and BRDFs (stored in **C**) at neighbouring planar locations.

The paper is organized as follows. Spatial BTF data segmentation is described in section 2 and surface geometry estimation (normal-map) is described in section 3. Surface geometry synthesis using the MRF model is the subject of section 4, while the final BTF data enlargement step is described in section 5. The subsequent sections show the results of the proposed model, discuss its properties and conclude the paper.

2. BTF space segmentation

The BTF data employed in this study were obtained from Bonn University [18]. We used BTFs of two different types of lacquered wood. Each data set comprises 81 viewing positions n_v and 81 illumination positions n_i (see figure 2), resulting in 6561 images with resolution 800 × 800 (the measurements after registration and rectification). To reduce the computational demands of the following BTF clustering step, an image tiling approach was applied. The method [12] finds sub-optimal paths in the original data to cut the required set of connectable BTF tiles. In our experiments, only one BTF tile per material was used.

The input to our algorithm is a seamless BTF tile in the form of n_i n_v illumination/view-dependent images of size $n_x \times n_y$. A vector of BTF values for a fixed planar position corresponds to the local BRDF and is denoted BRDF in this paper. In the first preprocessing step, all BTF images were converted into a CIE Lab perceptually uniform colour space and only data from luminance channel L were used in the data vector. The following K-means clustering was performed in the $n_x \times n_y$ planar space corresponding to individual pixels of BTF. Each pixel represents L-channel's BRDF of surface geometry at planar location (x, y).

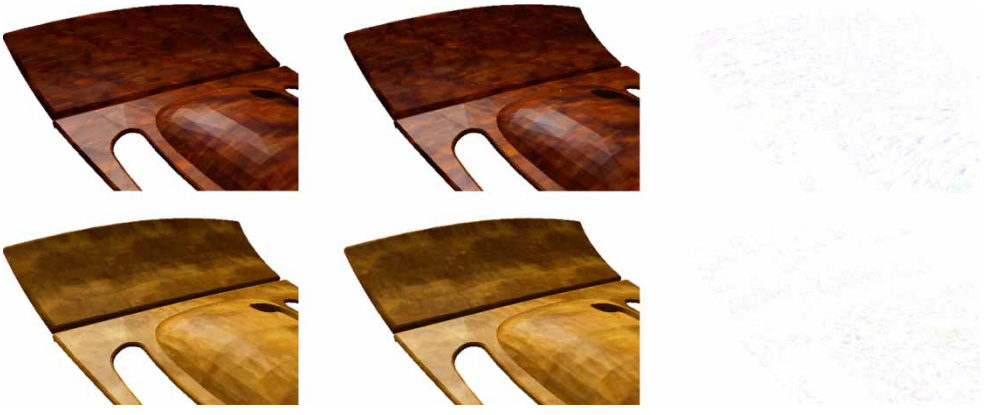

Figure 4. Results of the proposed BTF data clustering method ($n_c = 256$ clusters) mapped onto a 3D object (first column) in comparison with the original BTF tile mapping (second column) for two kinds of lacquered wood. Difference images (more visible in the colour electronic version) are depicted in the third column.

The clustering distance function is

$$d(x, y, k) = \sum_{v=1}^{n_v} \sum_{i=1}^{n_i} |\mathbf{B}(i, v, x, y) - \mathbf{C}(k, i, v)| \cos\theta_v, \tag{1}$$

where $\mathbf{B}(i, v, x, y)$ is the corresponding BTF value, $\mathbf{C}(k, i, v)$ are cluster centres and $i = 1, \ldots, n_i$ and $v = 1, \ldots, n_v$ are illumination and viewing directions of the original BTF data (see figure 1), respectively. The view elevation angle cosine accommodates the shortening of the surface emitting area. Clustering results in the index array $\mathbf{I}(x, y) \in 1, \ldots, n_c$ and the set of n_c cluster representatives $\mathbf{C}(k, i, v)$ of size $n_c \times 3n_i n_v$ corresponding to the closest colour BRDFs to cluster centres. Note that the individual colour BRDFs representing cluster centres $\mathbf{C}$ correspond to the representative set of material locations bearing the most distinct appearance over the BTF tile. The results of the proposed BTF clustering ($n_c = 256$) mapped onto the 3D object in comparison with the original BTF data mapping are shown in figure 4.

3. Surface geometry from the BTF

In order to find a smooth spatial representation of the cluster index $\mathbf{I}$ for further enlargement by means of the MRF model we use a normal-map describing the geometry of the original material surface. For this the standard photometric stereo technique [19] was applied. This approach is advantageous since the BTF data comprise a number of images with fixed viewpoint and a variety of defined illumination source directions. As we have many more than the necessary three different light positions we used over-determined photometric stereo. All directions to light sources are ordered in the rows of matrix $\mathbf{L}$ and the corresponding pixel intensities for different illumination directions are ordered in the vector $\mathbf{E}(x, y)$ as follows:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{1x} & \mathbf{L}_{1y} & \mathbf{L}_{1z} \\ \vdots & \vdots & \vdots \\ \mathbf{L}_{nx} & \mathbf{L}_{ny} & \mathbf{L}_{nz} \end{bmatrix}, \quad \mathbf{E}(x, y) = \begin{bmatrix} \mathbf{E}_1(x, y) \\ \vdots \\ \mathbf{E}_n(x, y) \end{bmatrix}. \tag{2}$$

The intensity equation can be written

$$\mathbf{E}(x, y) = \rho \mathbf{L}\, \mathbf{N}(x, y),$$

Figure 5. Example of five original BTF images taken from different illumination directions and the corresponding estimated normal-map for lacquered wood material.

where ρ is the surface albedo which was set to constant value $\rho = 1$. Then the surface normal-map $\mathbf{N}$ of the BTF tile at each pixel can be computed by means of least-squares fitting

$$\mathbf{N}(x, y) = \frac{(\mathbf{L}^{\mathrm{T}}\mathbf{L})^{-1}\mathbf{L}^{\mathrm{T}}\mathbf{E}(x, y)}{||(\mathbf{L}^{\mathrm{T}}\mathbf{L})^{-1}\mathbf{L}^{\mathrm{T}}\mathbf{E}(x, y)||}. \tag{3}$$

Note that the correct normal-map estimate using photometric stereo requires a Lambertian surface, i.e. a surface reflecting the light equally in all directions independent of illumination and viewing direction. This is rarely the case when BTF data are used as the input images, thus the estimated normal-map is mostly estimated incorrectly. Another principal problem is the normal-map estimation of materials with a smooth surface, e.g. lacquered wood, as shown in figure 5. However, in our method we do not need correctly estimated normals in their physical sense, but we exploit the estimated normal-map as some kind of illumination invariant, smooth, unique representation of the spatial BTF structure which can be further enlarged using the technique proposed in section 4. For this reason we can afford to use photometric stereo even for normal-map estimation of strongly non-Lambertian smooth surfaces.

The alternative approach is to use laser- or structured-light-based range scanners. However, a laser scanner is a costly device and does not allow satisfactory measurement of textile materials due to the laser beam scattering in the material structure. The ability to measure fine material structure by a structured light scanner is also limited due to the shadows cast by the structure elements, etc.

4. Probabilistic normal-map modelling

A normal-map enlargement in the scope of this paper is based on two different Markov random field (MRF) models. The simultaneous modelling of such a multichannel image, i.e. normal-map, generally requires three-dimensional models. If a 3D data space can be factorized, then these data can be modelled using a set of lower-dimensional 2D random field models, otherwise it is necessary to use a 3D random field model. Although full 3D models allow unrestricted spatial-channel correlation modelling, their main drawback is the larger number of parameters that have to be estimated and, consequently, the slightly more time-demanding analysis and synthesis.

In this paper, both approaches to multichannel modelling are compared. The first is a three-dimensional causal autoregressive model (3D CAR) and the second is its two-dimensional version (2D CAR).

Contrary to the models presented in this paper the alternative non-causal MRF models do not have any particular restriction on the shape of the contextual neighbourhood (CN); however, their computation is very slow since most non-causal MRF models require the use of iterative Monte Carlo methods. On the other hand, causal models restrict the CN shape to be either causal or unilateral, i.e. during computation the CN takes into account only the known or already computed image pixels. This enables the use of some exceptional models for which

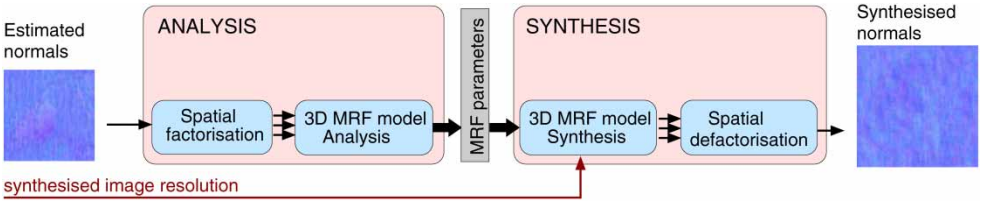

Figure 6. The overall 3D CAR smooth texture model scheme.

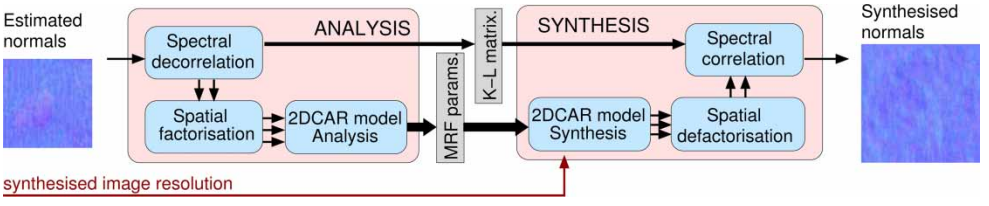

Figure 7. The overall 2D CAR smooth texture model scheme.

fast analytical parameter estimation is known, as well as fast model synthesis, i.e. CAR MRF. Generally, MRF models are generative, so they do not require storage of any samples of the original normal-map. The number of model parameters depends only on the number of input data channels and on the size of the model's contextual neighbourhood.

The whole normal-map enlargement process can be split into two major parts, as illustrated in figures 6 and 7. The first is a simultaneous *analysis* of all subspace images assuming the underlying MRF model. The second part is the fast subspace image *synthesis* of arbitrary resolution based on the MRF model parameters estimated in the previous analytical step.

In the case of both tested models (3D CAR, 2D CAR) the input normal-map is decomposed into a multi-resolution grid and the data of each resolution are independently modelled by their dedicated MRF model. This enables efficient modelling of all visual features of the subspace images. The multi-resolution grid is created by means of a Gaussian–Laplacian pyramid, as described in detail in section 4.1.

The results of the analytical part are several MRF models with different parameters corresponding to different synthetic results. Since there is no suitable similarity measure available with which to compare the visual quality of the colour textures, we choose the optimal model structure according to a subjective visual comparison of the input normal-map synthesized results.

4.1 *Spatial factorization*

Input normal-map $\bar{Y}_\bullet$ (the notation $\bullet$ indicates all possible values of the corresponding index) is decomposed into a multi-resolution grid and the data of each resolution are independently modelled by their dedicated CAR model. Each one generates a single spatial frequency band of the normal-map. The analysed normal-map is decomposed into multiple resolution factors using a Laplacian pyramid and the intermediary Gaussian pyramid $\ddot{Y}_\bullet^{(k)}$, which is a sequence of images in which each one is a low-pass down-sampled version of its predecessor. The Gaussian pyramid for reduction factor n is

$$\ddot{Y}_r^{(k)} = \downarrow_r^n (\ddot{Y}_{\bullet,i}^{(k-1)} \otimes w), \quad k = 1, 2, \ldots, \tag{4}$$

where $\ddot{Y}_{\bullet}^{(0)} = \bar{Y}_{\bullet}$, $\downarrow^n$ denotes down-sampling with reduction factor n, $\otimes$ is the convolution operation and r is the multi-index, having three components $r = \{r_1, r_2, r_3\}$. The first component is a row index, the second is a column index and the third is a channel index. The convolution mask based on weighting function (FIR generating kernel) w is assumed to execute separability, normalization, symmetry and equal contribution constraints. The FIR equation is then

$$\ddot{Y}_r^{(k)} = \sum_{i,j=-l}^{l} \hat{w}_i \hat{w}_j \ddot{Y}_{2r+(i,j)}^{(k-1)}.$$

The Laplacian pyramid $\dot{Y}_r^{(k)}$ contains band-pass components and provides a good approximation to the Laplacian of the Gaussian kernel. It can be constructed by differencing single Gaussian pyramid layers:

$$\dot{Y}_r^{(k)} = \ddot{Y}_r^{(k)} - \uparrow_r^n (\ddot{Y}_{\bullet}^{(k+1)}), \quad k = 0, 1, \ldots, \tag{5}$$

where $\uparrow^n$ is up-sampling with expanding factor n.

4.2 *3D causal auto-regressive MRF model*

The overall scheme of the 3D CAR model [20, 21] is depicted in figure 6. As input to the model, normal-maps of size $N \times M = 122 \times 125$ for *wood01* and $N \times M = 137 \times 142$ for *wood02* were estimated using the technique described in section 3.

4.2.1 3D CAR factor model. The normal-map in the previous spatial factorization step (section 4.1) was decomposed into a multi-resolution grid and the data of each resolution was modelled independently by a Gaussian noise driven 3D CAR model that enables simultaneous modelling of all normal components.

Let the normal-map Y be indexed on a finite rectangular three-dimensional $N \times M \times 3$ underlying lattice I, where $N \times M$ is the image size. Let us denote a simplified multi-index r as having three components $r = \{r_1, r_2, r_3\}$. The first component is a row index, the second is a column index and the third is a normal vector index. I_r specifies the shape of the contextual neighbourhood (CN) around the actual index $r = \{r_1, r_2, r_3\}$. The causality condition is satisfied when all data needed for CN are known (relative to the chosen direction of scanning of the image index lattice). From this causal CN the data are arranged in vector $X_r = [Y_{r-s}^{\mathrm{T}} : \forall s \in I_r^c]^{\mathrm{T}}$.

The (CAR) random field is a family of random variables with a joint probability density on the set of all possible realizations Y of the $M \times N \times 3$ lattice I, subject to the following condition:

$$p(Y \mid \Gamma, \Sigma^{-1}) = (2\pi)^{-3(MN-1)/2} |\Sigma^{-1}|^{(MN-1)/2} \times \exp\left\{-\frac{1}{2} \operatorname{tr}\left\{\Sigma^{-1} \begin{pmatrix} -I \\ \Gamma^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}} \tilde{V}_{MN-1} \begin{pmatrix} -I \\ \Gamma^{\mathrm{T}} \end{pmatrix}\right\}\right\}, \tag{6}$$

where I is the identity matrix, Γ is the parameter matrix, Σ is the covariance matrix of Gaussian white noise and

$$\tilde{V}_{r-1} = \begin{pmatrix} \tilde{V}_{YY(r-1)} & \tilde{V}_{XY(r-1)}^{\mathrm{T}} \\ \tilde{V}_{XY(r-1)} & \tilde{V}_{XX(r-1)} \end{pmatrix}. \tag{7}$$

The notation used in (7) is

$$\tilde{V}_{XX(r-1)} = \sum_{k=1}^{r-1} X_k X_k^{\mathrm{T}},$$
$$\tilde{V}_{XY(r-1)} = \sum_{k=1}^{r-1} X_k Y_k^{\mathrm{T}},$$
$$\tilde{V}_{YY(r-1)} = \sum_{k=1}^{r-1} Y_k Y_k^{\mathrm{T}}. \tag{8}$$

The simplified notation $r, r-1, \ldots$ denotes the multi-channel process position in I, i.e. $r = \{r_1, r_2, r_3\}$, where $r-1$ is the location immediately preceding $\{r_1, r_2, r_3\}$, etc. The direction of movement on the underlying image sub-lattice is common rows scanning. The data from the model history obtained during adaptation are denoted $Y^{(r-1)}$.

The 3D CAR model can be expressed as a stationary causal uncorrelated noise driven 3D autoregressive process:

$$Y_r = \Gamma X_r + \mathbf{e}_r, \tag{9}$$

where $\Gamma = [A_1, \ldots, A_\eta]$ is the $3 \times 3\eta$ parameter matrix, $\eta = \mathrm{card}(I_r^c)$, I_r^c is a causal CN, $\mathbf{e}_r$ is a Gaussian white noise vector with zero mean and a constant but unknown covariance matrix Σ and X_r is the corresponding data vector obtained from a model causal CN from already synthesized data Y_{r-s}.

4.2.2 Parameter estimation. There are two matrices, the parameter matrix $\hat{\Gamma}_r$ and the noise covariance matrix $\hat{\Sigma}_r$, to update in each step, i.e. the CN shift on the image lattice. Owing to the model causality and the normal-Wishart parameter, prior single CAR model parameters (10) and (11) can be estimated analytically [21]. The parameter matrix estimate is

$$\hat{\Gamma}_{r-1}^{\mathrm{T}} = V_{XX(r-1)}^{-1} V_{XY(r-1)}, \tag{10}$$

and the covariance matrix estimate is

$$\hat{\Sigma}_{r-1} = \frac{\lambda_{(r-1)}}{\beta(r)}, \tag{11}$$

where

$$\lambda_{(r-1)} = V_{YY(r-1)} - V_{XY(r-1)}^{\mathrm{T}} V_{XX(r-1)}^{-1} V_{XY(r-1)}, \tag{12}$$
$$V_{XX(r-1)} = \tilde{V}_{XX(r-1)} + V_{XX(0)},$$
$$V_{XY(r-1)} = \tilde{V}_{XY(r-1)} + V_{XY(0)},$$
$$V_{YY(r-1)} = \tilde{V}_{YY(r-1)} + V_{YY(0)}, \tag{13}$$

and matrices $V_{XX(0)}$, $V_{XY(0)}$ and $V_{YY(0)}$ are the corresponding matrices from the normal-Wishart parameter prior. Estimates (10) and (11) can also be evaluated recursively if necessary. $\beta(r) = \beta(0) + r - 1$ represents the number of model movements on the image plane ($\beta(0) > 1$).

4.3 *2D causal auto-regressive model*

The overall scheme of the 2D causal auto-regressive MRF model is depicted in figure 7. Since the model is two dimensional, only the input normal-map has to be factorized by means of the channel decorrelation step, as explained in the following section.

4.3.1 Channel decorrelation. The estimated input normal-map includes per-pixel normals normalized to unit vectors so the correlation of individual normal channels corresponding to the x, y and z coordinates of the normal vector is present. To remove this correlation, channel factorization using the Karhunen–Loeve transformation is utilized. This approach transforms the original centred normal-map data space $\tilde{Y}$ defined on the rectangular $M \times N$ finite lattice I into a new data space with K–L coordinate axes $\bar{Y}$. These new basis vectors are the eigenvectors of the second-order statistical moments matrix

$$\Phi = E\{\tilde{Y}_r \tilde{Y}_r^{\mathrm{T}}\}, \tag{14}$$

where the multi-index r has two components $r = [r_1, r_2]$, the first being the row and the second the column index. The projection of random vector $\tilde{Y}_r$ onto the K–L coordinate system uses the transformation matrix

$$\mathbf{T} = [u_1^{\mathrm{T}}, u_2^{\mathrm{T}}, u_3^{\mathrm{T}}]^{\mathrm{T}}, \tag{15}$$

which has single rows u_j that are eigenvectors of the matrix Φ. The components of the transformed vector,

$$\bar{Y}_r = \mathbf{T}\tilde{Y}_r, \tag{16}$$

are mutually uncorrelated and if we assume that they are also Gaussian, then they are independent, thus each transformed mono-channel factor can be modelled independently of the remaining channel factors.

4.3.2 2D CAR factor model. Channel factorization (15) of the normal-map into individual mono-channel factors allows us to use a simpler 2D CAR model [16]. These single orthogonal factors of the normal-map are further decomposed into a multi-resolution grid and the data of each resolution are independently modelled by their dedicated independent Gaussian noise driven autoregressive random field model (CAR) as follows.

The causal autoregressive random field (CAR) is a family of random variables with a joint probability density on the set of all possible realizations Y of the $M \times N$ lattice I, subject to the condition

$$p(Y|\gamma, \sigma^{-2}) = (2\pi\sigma^2)^{-(MN-1)/2} \exp\left\{\frac{-1}{2}\mathrm{tr}\left\{\sigma^{-2}\begin{pmatrix}-\alpha\\ \gamma^{\mathrm{T}}\end{pmatrix}^{\mathrm{T}} \tilde{V}_{MN-1}\begin{pmatrix}-\alpha\\ \gamma^{\mathrm{T}}\end{pmatrix}\right\}\right\}, \tag{17}$$

where α is a unit vector, γ is the parameter vector, σ is the variance of the Gaussian white noise, and notation similar to the 3D CAR model is used,

$$\tilde{V}_{r-1} = \begin{pmatrix}\tilde{V}_{YY(r-1)} & \tilde{V}_{XY(r-1)}^{\mathrm{T}}\\ \tilde{V}_{XY(r-1)} & \tilde{V}_{XX(r-1)}\end{pmatrix},$$

where $\tilde{V}_{XX(r-1)}$, $\tilde{V}_{XY(r-1)}$ and $\tilde{V}_{YY(r-1)}$ are the matrices defined in (8).

The 2D CAR model can be expressed as a stationary causal uncorrelated noise driven 2D autoregressive process:

$$Y_r = \gamma X_r + e_r, \tag{18}$$

where

$$\gamma = [a_1, \ldots, a_\eta] \tag{19}$$

is the parameter vector, I_r^c is a causal neighbourhood with $\eta = \text{card}(I_r^c)$, e_r is a white Gaussian noise with zero mean and a constant but unknown variance σ^2, and X_r is the corresponding data vector.

4.3.3 Parameter estimation. Analogously to the 3D CAR model there are two parameters, the parameter vector $\hat{\gamma}_r$ and the noise covariance $\hat{\sigma}_r$, to update in each step.

Parameter estimation of a 2D CAR model can be performed analytically using the maximum likelihood, the least-square or Bayesian methods. The Bayesian parameter estimations of the causal AR model with the normal-gamma parameter prior which maximize the posterior density are [21]

$$\hat{\gamma}_{r-1}^{\mathrm{T}} = V_{XX(r-1)}^{-1} V_{XY(r-1)} \tag{20}$$

and

$$\hat{\sigma}_{r-1}^2 = \frac{\lambda_{(r-1)}}{\beta(r)}, \tag{21}$$

where

$$\lambda_{(r-1)} = V_{YY(r-1)} - V_{XY(r-1)}^{\mathrm{T}} V_{XX(r-1)}^{-1} V_{XY(r-1)}, \tag{22}$$

$$\begin{aligned} V_{XX(r-1)} &= \tilde{V}_{XX(r-1)} + V_{XX(0)}, \\ V_{XY(r-1)} &= \tilde{V}_{XY(r-1)} + V_{XY(0)}, \\ V_{YY(r-1)} &= \tilde{V}_{YY(r-1)} + V_{YY(0)}, \end{aligned} \tag{23}$$

and sub-matrices $V_{XX(0)}$, $V_{XY(0)}$ and $V_{YY(0)}$ are from the normal-gamma parameter prior. Estimates (20) and (21) can also be evaluated recursively and $\beta(r) = \beta(0) + r - 1\,(\beta(0) > 1)$.

4.4 *Normal-map synthesis*

The CAR model synthesis is very simple and the Markov random field can be generated directly from the model equation (9) (respectively (18)) with respect to the CN data vector X_r and the estimated parameter matrix $\hat{\Theta}_r$ (respectively $\hat{\gamma}_r$) using a multivariate/scalar Gaussian white-noise generator. The fine-resolution normal-map is obtained from the pyramid collapse procedure, which is the inverse process to spatial factorization (4) and (5) described in section 4.1. The last step, inverse channel correlation, is needed only for the 2D CAR model. The resulting synthesized normal texture is then obtained from the set of synthesized mono-channel images using the inverse K–L transformation according to the equation

$$\tilde{Y}_r = \mathbf{T}^{-1} \bar{Y}_r. \tag{24}$$

This transformation matrix $\mathbf{T}^{-1}$ has to be stored together with the estimated 2D CAR model parameters. A comparison of the synthesized normal-maps using the $\mathbf{N}_S$ 3D CAR model with their originals $\mathbf{N}$ is illustrated in the first column of figure 9.

5. New cluster index synthesis

New cluster index $\mathbf{I}_S$ is obtained by row-wise scanning of synthesized normal-map $\mathbf{N}_S$, as depicted in figure 8. For each normal in $\mathbf{N}_S$ the n_k closest normal from normal-map $\mathbf{N}$ of the original BTF tile is determined with respect to the Euclidean metric between two unit vectors.

However, this approach alone is unsatisfactory because it allows ambiguous normal assignment owing to the material surface. For instance, a normal vector pointing straight upwards can represent either a peak or a valley on the surface. Thus, if a new index is created based only on normal matching, the resulting enlarged BTF images are very noisy, and the synthesized structure of the normal-map is considerably suppressed. To improve the spatial continuity of the generated new cluster index we used information on the surface height, occlusion and the masking of surface points, which is included in colour BRDFs of individual stored clusters $\mathbf{C}$. Individual cluster indices corresponding to candidate normal k from $\mathbf{N}$ are obtained from the same (x, y) location from $\mathbf{I}$, as is the spatial location of the normal k. From the obtained n_k normal candidates from the original index $\mathbf{I}$, the optimal one k^* is chosen that minimizes the distance D between the candidate's BRDF and the BRDFs of its surrounding pixels at locations $(x, y-1)$ and $(x-1, y)$ from the causal neighbourhood in $\mathbf{I}_S$

$$k^* = \arg \min_{k=1\ldots n_c} (\mathbf{D}(\mathbf{I}(x_k, y_k), \mathbf{I}_S(x, y-1)) + \mathbf{D}(\mathbf{I}(x_k, y_k), \mathbf{I}_S(x-1, y))). \tag{25}$$

To speed up this process the mutual distance between each couple of n_c clusters is precomputed (26) and stored in the form of a $n_c \times n_c$ matrix $\mathbf{D}$:

$$\mathbf{D}(a, b) = \sum_{v=1}^{n_v} \sum_{i=1}^{n_i} |\mathbf{C}(a, i, v) - \mathbf{C}(b, i, v)| \cos\theta_v. \tag{26}$$

The (x_{k^*}, y_{k^*}) position in new index $\mathbf{I}_S$ is obtained by means of $\mathbf{I}_S(x, y) = \mathbf{I}(x_{k^*}, y_{k^*})$ using the cluster indices from the original index $\mathbf{I}$. The proposed matching scheme incorporates effects such as masking and occlusions and, together with normal matching, it enables the reliable and perceptually correct spatial ordering of individual clusters in the new enlarged index $\mathbf{I}_S$. Additionally, this ordering enforces the continuity constraint by placement of similar BRDFs into neighbouring positions in generated cluster index $\mathbf{I}_S$.

For BTF rendering from the proposed model the cluster representatives $\mathbf{C}$ and synthesized cluster index $\mathbf{I}_S$ have to be stored, which enables a compression ratio of approximately $1/200$ (for $n_c = 256$). The required BTF value is obtained as

$$\mathrm{BTF}(x, y, i, v) = \mathbf{C}(\mathbf{I}_S(x, y), i, v). \tag{27}$$

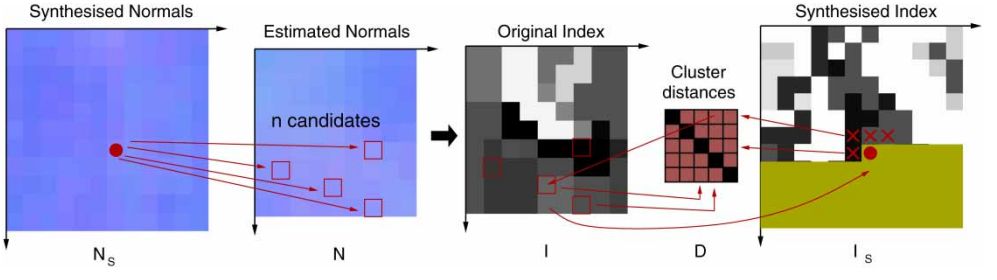

Figure 8. The new cluster index generation scheme.

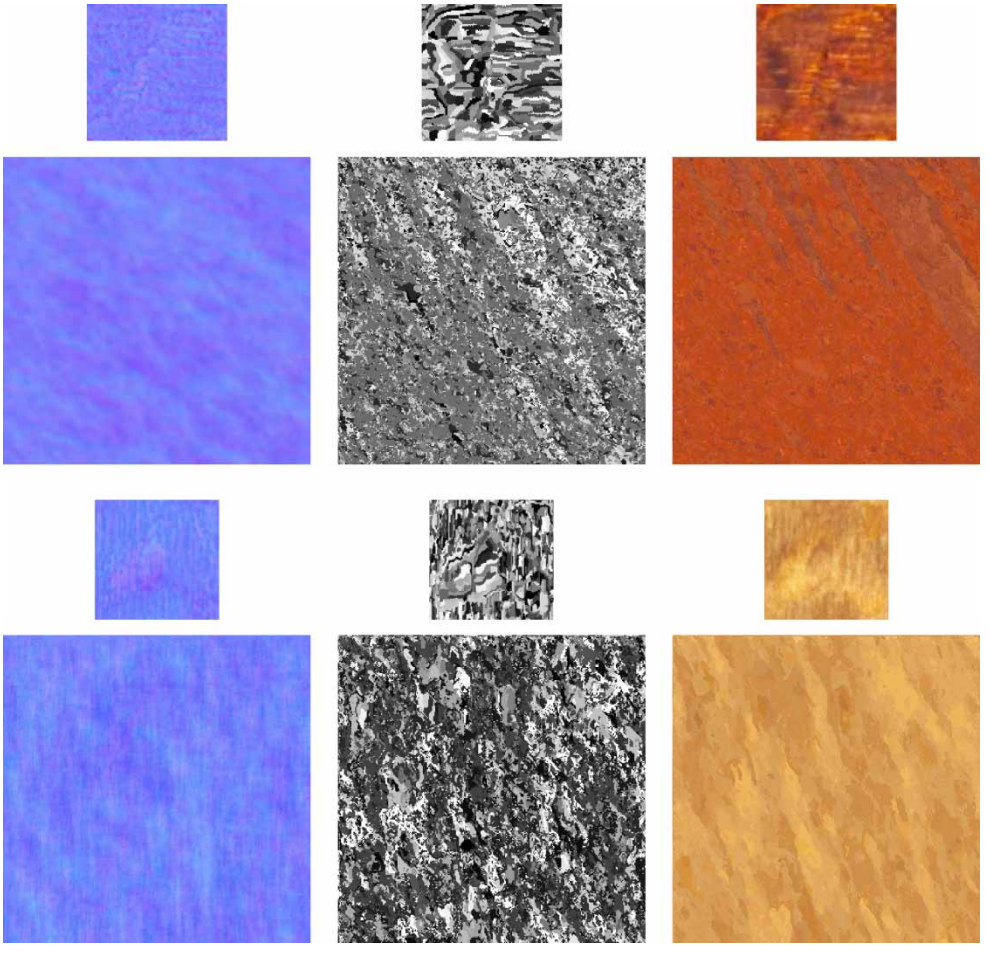

Figure 9. Individual sub-results of the BTF enlargement process for two kinds of lacquered wood (*wood01* (top) and *wood02* (bottom)) using the 3D CAR smooth texture model. Odd rows: estimated normal-map $\mathbf{N}$, cluster index $\mathbf{I}$ and restored BTF image for angles $\theta_i = 15°$, $\phi_i = 0°$, $\theta_v = 0°$ and $\phi_v = 0°$. Even rows: synthesized normal-map $\mathbf{N}_S$, corresponding synthesized cluster index $\mathbf{I}_S$ and enlarged BTF image.

Examples of synthesized cluster indices $\mathbf{I}_S$ as well as corresponding BTF images for both tested materials compared with their original counterparts are shown in the second and third column of figure 9.

6. Results

The proposed methods were applied to the BTF synthesis of two different types of smooth lacquered wood. The original BTF tile of *wood01* has size 122 × 125 and *wood02* size 137 × 142 pixels. The size of the synthesized normal-maps and subsequent index arrays in our experiments was, for both these materials, 1024 × 1024 pixels.

An example of a single planar BTF image enlarged by the proposed method is shown in the third column of figure 9. A comparison of the enlarged BTF data mapped onto part of a car gearbox console and car seat with one BTF tile mapping is shown for both materials in figures 10 and 11. The first column illustrates the disturbing effect of one BTF tile repetition,

Figure 10. Results of the proposed BTF data enlargement method using both the 2D and 3D CAR models (the third and fourth columns) mapped onto the console of a car gearbox in comparison with BTF tiling using one tile (the first column) and two tiles (the second column) for two kinds of lacquered wood. For all test images the number of clusters was fixed at $n_c = 256$.

which persists even when two different BTF tiles are used, as is shown in the second column. This effect will be more apparent for large virtual objects whose size is much bigger in comparison with the structure details of the given BTF material, as is shown in figure 11. The third and fourth columns of both figures show the performance of the proposed BTF enlargement technique, i.e. 2D CAR and 3D CAR models, respectively. The proposed method produces an efficient non-repetitive spatial representation of the wooden structure. Both tested models provide comparable results. The smoothness of the synthesized images depends mainly on the number of normal candidates n_k and the shape of the causal neighbourhood used for new index synthesis. These parameters have to be set carefully to avoid the presence of high variance noise in the synthesized cluster index. For this reason a better or additional smoothing constraint during index synthesis will be the main subject of our future research.

The time demands of the analytical part of the proposed method are not too important since BTF segmentation, normal-map estimation, synthesis and, finally, estimated and synthesized normal matching are off-line tasks. The most time-consuming part of the method is BTF tile clustering, which takes approximately 1 h when using $n_c = 256$ clusters for a BTF tile of *wood02*, while the remaining analytical steps are much faster, depending on the size of the original and required normal-map.

The synthesis of new cluster index $\mathbf{I}_S$ of size 1024×1024 for a BTF tile of *wood02* takes approximately 15 min. All experiments were performed on an AMD Athlon personal computer, 1.9 GHz, 2 GB RAM. When the cluster index is synthesized to a size that enables us to cover a given virtual object, the rendering of the BTF data using the corresponding clustered representation is very fast. Each pixel of the synthesized BTF image for the required illumination and view direction is obtained by finding a RGB value in the cluster centre, looked up by indexing in the synthesized cluster index (27). The BTF image for an arbitrary combination of illumination and viewing angles was obtained by means of original BTF measurement direction interpolation based on barycentric coordinates [22]. All processing was implemented directly in graphics hardware, yielding an interactive rendering frame rate of about 20 frames/s. Two screen shots of our BTF rendering software showing the proposed model of lacquered wood BTF enlargement applied to the upper part of the gearbox console in a car interior are depicted in figure 12.

The compression ratio of the proposed method for 256 clusters is approximately 1/200. For example, the original size of 10 *wood02* BTF tiles is 2.4 GB, while the parametric representation of the proposed model occupies only 9.3 MB. Despite providing such a compression ratio the proposed method preserves the original reflectance dependent on the illuminating and

Figure 11. Results of the proposed BTF data enlargement method using both the 2D and 3D CAR models (the third and fourth columns) mapped onto a car seat in comparison with BTF tiling using one tile (the first column) and two tiles (the second column) for two kinds of lacquered wood. For all test images the number of clusters was fixed at $n_c = 256$.

Figure 12. A screen-shot from a UTIA real-time BTF rendering application of the car interior using the proposed BTF enlargement technique (3D CAR) applied to lacquered wood materials on the upper console of the gearbox.

viewing directions for individual cluster centres (BRDFs). This BRDF can subsequently be approximated by means of standard empirical [23–26] or physical [27–31] reflectance models to achieve even higher BTF compression ratios at the cost of a slightly higher computational load required for BRDF cluster centre synthesis.

7. Summary and conclusions

This paper proposes two new techniques for seamless BTF data enlargement. The method strictly separates the analytical (possibly off-line) part from the fast (possibly real-time) synthesis part of the whole modelling process. BTF clustering allows us to arbitrarily trade-off the compression ratio and visual quality based on application requirements. The method shows the best performance for spatially random, i.e. non-regular, types of BTFs such as the tested lacquered wood or leather, etc. The method enables fast seamless BTF data enlargement of arbitrary size with minimal additional storage requirements since the number of clusters is fixed.

Acknowledgements

This research was supported by EC project No. FP6-507752 MUSCLE and partially by grants Nos. A2075302 and 1ET400750407 of the Grant Agency of the Academy of Sciences, Czech Republic, and MSMT project No. 1M0572, 2C06019. BTF measurements courtesy of Bonn University. Mercedes Class-C 3D model courtesy of DaimlerChrysler.

References

[1] Koudelka, M., Magda, S., Belhumeur, P. and Kriegman, D., 2003, Acquisition, compression, and synthesis of bidirectional texture functions. Paper presented at the 3rd International Workshop on Texture Analysis and Synthesis (Texture 2003).

[2] Vasilescu, M. and Terzopoulos, D., 2004, TensorTextures: multilinear image-based rendering. In: *ACM SIGGRAPH 2004* (New York: ACM Press), Vol. 23, pp. 336–342.

[3] Sattler, M., Sarlette, R. and Klein, R., 2003, Efficient and realistic visualization of cloth. Paper presented at the Eurographics Symposium on Rendering.

[4] Müller, G., Meseth, J. and Klein, R., 2003, Compression and real-time rendering of measured BTFs using local PCA. In: *Vision, Modeling and Visualisation 2003*.

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

[6] Malzbender, T., Gelb, D. and Wolters, H., 2001, Polynomial texture maps. In: *ACM SIGGRAPH 2001* (New York: ACM Press), pp. 519–528.

[7] Meseth, J., Müller, G. and Klein, R., 2003, Preserving realism in real-time rendering of bidirectional texture functions. In: *OpenSG Symposium 2003* (Switzerland: Eurographics Association), pp. 89–96.

[8] Filip, J. and Haindl, M., 2005, Efficient image based bidirectional texture function model. In: M. Chantler and O. Drbohlav (Eds) *Texture 2005*: *Proceedings of the 4th Internatinal Workshop on Texture Analysis and Synthesis*, Heriot-Watt University, Edinburgh, pp. 7–12.

[9] Efros, A.A. and Freeman, W.T., 2001, Image quilting for texture synthesis and transfer. In: E. Fiume (Ed) *ACM SIGGRAPH 2001* (New York: ACM Press), pp. 341–346.

[10] Cohen, M., Shade, J., Hiller, S. and Deussen, O., 2003, Wang tiles for image and texture generation. In: *ACM SIGGRAPH 2003* (New York: ACM Press), Vol. 22, pp. 287–294.

[11] Kwatra, V., Schödl, A., Essa, I. and Bobick, A., 2003, Graphcut textures: image and video synthesis using graph cuts. In: *ACM SIGGRAPH 2003* (New York: ACM Press), Vol. 22, pp. 277–286.

[12] Somol, P. and Haindl, M., 2005, Novel path search algorithm for image stitching and advanced texture tiling. Paper presented at the 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG05, University of West Bohemia, Plzen.

[13] Haindl, M. and Hatka, M., 2005, A roller-fast sampling-based texture synthesis algorithm. Paper presented at the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG05, University of West Bohemia, Plzen.

[14] Dong, J. and Chantler, M., 2003, Comparison of five 3D surface texture synthesis methods. Paper presented at the 3rd International Workshop on Texture Analysis and Synthesis, Texture 2003.

[15] Haindl, M. and Hatka, M., 2005, BTF roller. In: M. Chantler and O. Drbohlav (Eds) *Texture 2005*: *Proceedings of the 4th Internatinal Workshop on Texture Analysis and Synthesis*, Edinburgh, Heriot-Watt University, pp. 89–94.
[16] Haindl, M. and Filip, J., 2004, A fast probabilistic bidirectional texture function model. In: *Proceedings of the International Conference on Image Analysis and Recognition* (Lecture Notes in Computer Science 3212) (Berlin: Springer), Vol. 2, pp. 298–305.
[17] Haindl, M., Filip, J. and Arnold, M., 2004, BTF image space utmost compression and modelling method. In: *Proceedings of the 17th International Conference on Pattern Recognition* (IEEE Computer Society Press), Vol. 3, pp. 194–198.
[18] Universität Bonn, 2003. BTD Database Bonn. Available online at: http://btf.cs.uni-bonn.de/.
[19] Woodham, R., 1981, Analysing images of curved surface. *Artificial Intelligence*, **17**, 117–140.
[20] Haindl, M., 1991, Texture synthesis. *CWI Quarterly*, **4**, 305–331.
[21] Haindl, M. and Šimberová, S., 1992, A multispectral image line reconstruction method. In: *Theory & Applications of Image Analysis* (Singapore: World Scientific), pp. 306–315.
[22] Coxeter, H.S.M., 1969, *Introduction to Geometry* (New York: Wiley).
[23] Phong, B.T., 1975, Illumination for computer generated images. *Communications of the ACM*, **18**, 311–317.
[24] Blinn, J., 1977, Models of light reflection for computer synthesized pictures. In: *Computer Graphics Proceedings*, Annual Conference Series, ACM SIGGRAPH 1977 (New York: ACM Press), pp. 192–198.
[25] Banks, D., 1994, Illumination in diverse codimensions. In: *ACM SIGGRAPH 1994* (New York: ACM Press), pp. 327–334.
[26] Ashikhmin, M. and Shirley, P., 2000, An anisotropic phong light reflection model. *Journal of Graphics Tools*, **5**, 25–32.
[27] Torrance, K. and Sparrow, E., 1967, Theory for off-specular reflection from rough surfaces. *Journal of the Optical Society of America*, **57**, 1105–1114.
[28] Cook, R. and Torrance, K., 1981, A reflectance model for computer graphics. In: *ACM SIGGRAPH 1981* (New York: ACM Press), Vol. 15, pp. 307–316.
[29] He, X., Torrance, K., Sillion, F. and Greenberg, D., 1991, A comprehensive physical model for light reflection. *Computer Graphics*, **25**.
[30] Ward, G., 1991, Measuring and modeling anisotropic reflection. *Computer Graphics*, **26**.
[31] Lafortune, E.P., Foo, S.C., Torrance, K.E. and Greenberg, D.P., 1997, Non-linear approximation of reflectance functions. *Computer Graphics*, **31**, 117–126.