

# Extreme Compression and Modeling of Bidirectional Texture Function

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

## Abstract

The recent advanced representation for realistic real-world materials in virtual reality applications is the Bidirectional Texture Function (BTF) which describes rough texture appearance for varying illumination and viewing conditions. Such a function can be represented by thousands of measurements (images) per material sample. The resulting BTF size excludes its direct rendering in graphical applications and some compression of these huge BTF data spaces is obviously inevitable. In this paper we present a novel, fast probabilistic model-based algorithm for realistic BTF modeling allowing an extreme compression with the possibility of a fast hardware implementation. Its ultimate aim is to create a visual impression of the same material without a pixel-wise correspondence to the original measurements. The analytical step of the algorithm starts with a BTF space segmentation and a range map estimation by photometric stereo of the BTF surface, followed by the spectral and spatial factorization of selected sub-space color texture images. Single mono-spectral band-limited factors are independently modeled by their dedicated spatial probabilistic model. During rendering, the sub-space images of arbitrary size are synthesized and both color (possibly multi-spectral) and range information is combined in a bump-mapping filter according to the view and illumination directions. The presented model offers a huge BTF compression ratio unattainable by any alternative sampling-based BTF synthesis method. Simultaneously this model can be used to reconstruct missing parts of the BTF measurement space.

## Index Terms

Rough texture, 3D texture, BTF, texture synthesis, texture modeling, data compression.

M. Haindl and J. Filip are with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, 182 08 Prague, Czech Republic. Phone +420266052350, Fax:+4202844683031, Email: haindl@utia.cz, EDICS: MOD-SRCE,SDR-IREP,COD-LSYI

## I. INTRODUCTION

A visualization of physically correct virtual models cannot be accomplished without natural-like color textures covering virtual or augmented reality (VR) scene objects. These textures can be either smooth or rough. The rough textures (coarse surface textures) rarely follow Lambert's law due to shadowing, masking, interreflections and some other effects, thus their reflectance is illumination and view angle dependent. The first attempt to capture real material appearance was done by Nicodemus et al. [24] who introduced the *Bidirectional Reflectance Distribution Function* (BRDF). BRDF is a 4D function which describes the relation between incident irradiance and outgoing radiance. Twenty years later, Dana et al. [5] generalized BRDF to represent also spatial dependencies in the form of *Bidirectional Texture Function* (BTF). BTF is a six-dimensional function  $BTF(\theta_i, \phi_i, \theta_v, \phi_v, \tilde{r})$  where  $\theta$  and  $\phi$  are elevation and azimuthal angles of illumination and view direction vector, respectively (see. Fig. 1). The vector  $\tilde{r}$  specifies the planar horizontal ( $r_1$ ) and vertical ( $r_2$ ) position in a material sample image, as well as the spectral index ( $r_3$ ). The BTF measurements comprise the whole hemisphere of light and view directions in the observed material sample coordinates according to selected quantization steps (Fig. 2). Effects presented in rough textures such as occlusions, self shadowing, inter-reflection or subsurface scattering are preserved. Materials represented using BTF appearing

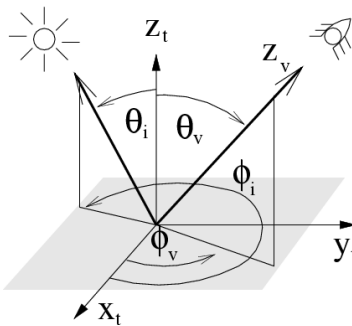


Fig. 1. Relationship between illumination and viewing angles within texture coordinate system.

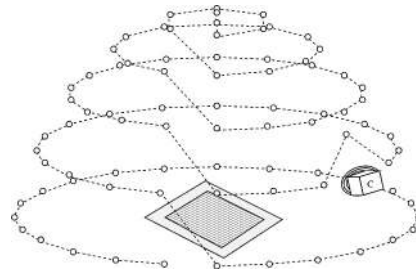


Fig. 2. Light vector trajectory above the sample. The movement of illumination starts at the top.

in virtual scene models can be either digitized natural textures or textures synthesized from an appropriate mathematical model. The sampling solution becomes unfeasible for most rough textures which require for every texture the storage of thousands of different illumination and

view angle samples. Moreover a usual VR scene requires storage of terabytes of BTF data (a car interior - 20 materials  $\approx 0,5$  TB), which is far beyond the limits of any current real time hardware. Thus, the development of an efficient BTF compression and modeling method is obviously inevitable. The BTF synthesis is a relatively new research area, so there is only a limited number of methods available unlike in the research area of smooth texture synthesis. The BTF modeling approaches can be basically divided into two major categories.

### *BTF Compression Methods*

The methods included in this category compress BTF data only into some kind of pixel-wise BTF parametric representation, but they do not solve the BTF seamless enlargement (synthesis) problem. They can be further divided into two subgroups. The first sub-group represents BTF by means of pixel-wise analytical reflectance models. The pioneering work [20] represented the reflectance of each pixel in BTF using the Lafortune reflectance model [16]. The spatial inconsistency of individual pixels in BTF due to different view directions led to the separate modeling of BTF images that correspond to one view direction only [19]. Here, the individual BTF pixels were represented by means of polynomials. Pixel-wise generalization of the Lafortune model computed for each view position separately was introduced in [8], [21]. These methods are very fast, however, they only enable moderate compression of the original BTF measurements. The second sub-group of the BTF compression methods is based on the standard principle component analysis (PCA) statistical approach. In [15], all BTF sub-images were decomposed using singular value decomposition (SVD) and at least 150 main eigenimages were kept for a satisfactory BTF reconstruction. The compression method [27] decomposes the BTF space, ordered into a tensor, by means of multi-modal SVD. Even though both methods enable realistic BTF rendering, they are not suitable for fast BTF applications since they require the computation of linear combinations of a large number of eigencomponents. Much faster approaches were presented in [22], [25]. The authors in [25] computed 16 principal components for individual BTF images with the same view position instead of the whole BTF dataspace. A local PCA model was used in [22]. This group of methods offers a high visual quality, but mild compression ratio.

### *BTF Synthesis Methods*

Up to now there have been only a very few BTF synthesis methods available enabling BTF seamless enlargement to an arbitrary size. Unfortunately, these methods are often too slow to enable real-time BTF rendering. BTF synthesis methods can be divided into two groups - the sampling and the modeling methods. The basic difference between the two groups is the kind of data to be stored. Sampling approaches require storage of the original BTF measurements or the corresponding pixel-wise parameterization. Modeling approaches on the other hand call for storage of some highly efficient statistics.

The prevailing BTF sampling [3], [6], [7], [31] is based on reproducing BTF data from the original BTF measurements or their pixel-wise parametric representation to generate a corresponding larger BTF texture space. A method combining a sparse set of BTF measurements according to enlarged material range-map was suggested in [17]. Another BTF model [18] exploits the technique of 3D textons, i.e. the smallest repeatable texture elements. Only these textons are then approximated using local PCA and finally used for surface modeling.

Some of the sampling methods are very time-consuming and they reach only a low compression ratio ( $\approx \frac{1}{10} - \frac{1}{1000}$  depending on sample size). These disadvantages of the contemporary BTF sampling models can be remedied by probabilistic BTF modeling. The probabilistic models are generative, so that no form of the original BTF measurements needs to be stored but only very restricted model statistics. The resulting synthetic textures are far more flexible, extremely compressed (only a few parameters have to be stored), they may be evaluated directly in procedural form and can fill an infinite texture space without visible discontinuities. Several successful smooth texture modeling approaches were published, e.g., [1], [14] and some survey articles are available [9]. Furthermore, a few probabilistic BTF models based on the underlying Markov random field (MRF) texture models have been introduced recently. A probabilistic BTF model based on the Gaussian MRF underlying texture model was introduced in [10]. Another approach using a 3D causal auto-regressive texture model (3D CAR) was proposed in [12]. Both of these models approximate a regular rough structure dependent on illumination position by means of displacement / bump-mapping filter and they require the analysis and synthesis of all BTF images. Even if the texture synthesis using these models is relatively fast, the simultaneous synthesis of all BTF images is computationally demanding. Moreover, even when modeling a

subset of BTF images, the GMRF model requires a time consuming FFT transformation for the texture synthesis, and the 3D CAR model requires the estimation of a large number of model parameters. These features limit a direct implementation of both probabilistic models in the graphics processing unit (GPU).

To overcome these shortcomings, we have introduced a novel BTF modeling approach [11] based on BTF clustering into subspace images. These subspace images are independently modeled by a set of fast causal auto-regressive models (2D CAR). The rough structure is introduced to the model by means of GPU implemented bump-mapping. This paper shows a real application of the BTF model on parts of a car interior that are covered with synthesized BTF samples of different real-world materials. Furthermore it discusses the possibility of full hardware implementation and compares properties of the method with several different BTF modeling approaches. The rest of the paper is organized as follows: The following section outlines the proposed probabilistic BTF model. It contains subsections about BTF space segmentation, range-map estimation and about multi-scale color probabilistic model-based texture synthesis. We also discuss here the underlying 2D CAR model parameter estimation and synthesis solutions. Section III. describes the GPU implementation of the method. The test results on different BTF samples are reported in Section IV., followed by conclusions in Section V.

## II. PROBABILISTIC BTF MODEL

The cornerstone of our novel BTF compression and modeling method is the replacement of a huge number of original BTF measurements by their efficient parametric estimates derived from an underlying set of spatial probabilistic models. Single processing steps of the method are indicated in Fig.3. The whole method can be divided into an analytical part (dashed box), which is usually solved off-line, and an on-line synthesis part, which has to be numerically as efficient as possible. The *analytical part* consists of a BTF space segmentation into several subspaces, range map estimation and parameters estimation of a multi-scale multidimensional probabilistic texture model. The *synthesis part* consists of three functional parts - range map enlargement, the multi-scale multidimensional probabilistic model synthesis and the bump-mapping filter (a compromise for a displacement mapping due to GPU). The filter combines the enlarged range map with synthetic multi-scale smooth textures into the resulting synthetic BTF space.

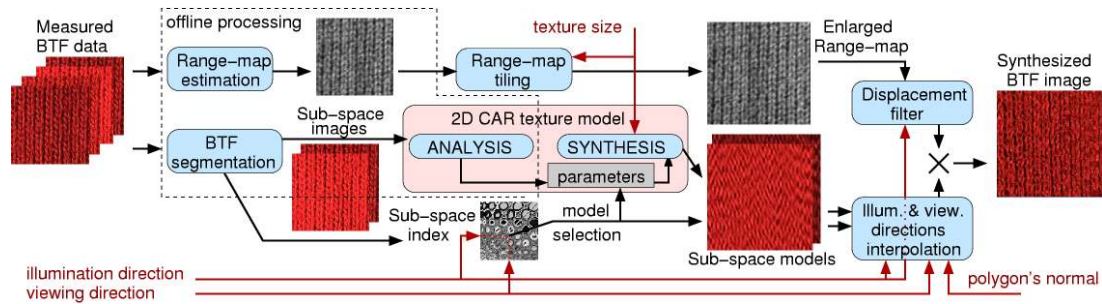


Fig. 3. The overall scheme of the proposed BTF modeling approach.

The visual appearance of rough material macro-structure during changes of viewing and illumination conditions is simulated using the bump-mapping filter technique [2], [29]. The overall material appearance (i.e. without macro-structure geometry) is guided by the corresponding underlying subspace model. An obvious advantage of this solution is the possibility of using the hardware support of the bump / displacement techniques in contemporary visualization hardware.

The BTF space model utilizes three functional blocks - BTF space segmenter, range map estimator and a probabilistic data model, respectively. The inclusion of the first two processing units permits a significant reduction of the number of BTF subspace color textures required in order to represent the whole measured BTF space.

#### A. BTF Space Segmentation

A single probabilistic BTF model [10], [12] (i.e., for a single view and illumination direction) combined with the displacement or bump-mapping filter can potentially approximate the whole BTF measurement space. However, approximation errors for significantly different illumination and viewing angles can worsen the visual realism of certain polygons of the textured virtual object. Thus we trade off between an extreme compression ratio and the visual quality by using several probabilistic BTF subspace dedicated models. The off-line part of our algorithm (see Fig. 3) starts with the BTF illumination / view  $(\theta_i, \phi_i / \theta_v, \phi_v)$  space segmentation into several subspace images using the K-means algorithm. Since the pixel-wise matching of BTF images is inconvenient, due to presence of masking effects caused by different viewing positions, the BTF space is represented by features in  $81 (\text{view}) \times 81 (\text{illumination})$  space (for University of Bonn data [25]). As the data features we use color cumulative histograms of individual BTF images

in perceptually uniform CIE Lab color-space. The histograms were used as a relatively reliable representation of the color distribution in BTF images while the material spatial distribution depends mostly on the actual view direction. An important issue is to determine the optimal number of subspaces. The PCA analysis of the whole BTF dataspace shows that the intrinsic BTF space dimensionality for the most BTF samples is between ten and thirty eigenimages (containing 95% of the entire information). Significant drop of four materials eigenvalues is illustrated in Fig. 4. In consideration of limited GPU memory, computational demands as well as satisfactory visual BTF reconstruction, we set the maximum number of subspace images to twenty. In fact, the improvement of the visual quality when adding more subspace images turned out to be indiscernible for our BTF materials. Several examples of BTF illumination / view space segmentations for different materials are depicted in Fig. 5. These tables of indices, presented in a form of images, provide us with information which cluster (i.e. subspace index) is used for a given combination of illumination direction (axis y, in the range of 1-81) and viewing direction (axis x, in the range of 1-81). The pattern of these images is determined by the K-means clustering and is strongly related to the local reflectance of material micro-geometry. The circular structures in the images are caused by sub-surface scattering effects (which are more common in partly translucent material containing a light-transferring layer, such as lacquered wood or fibers of fabrics) while the stripes refer to shadowing and masking effects (e.g. in leathers or structured fabrics). The remaining visible vertical/horizontal stripes along the whole image length are caused by an elevation angle step of the viewing / illumination direction in the measured data.

### B. Range Map Estimation

The overall roughness of a textured surface significantly influences the BTF texture appearance. Such a surface can be specified using its range map, which can be estimated by several existing approaches [23], [28]. The most accurate range map can be estimated by direct measurements of the observed surface using corresponding range cameras. However, this method requires special hardware and measurement methodology. Hence, alternative approaches for range map estimation from surface spectral images are more appropriate. One of the most accurate approaches also used in this paper is the *Photometric Stereo* approach. It estimates the surface range map from at least three images obtained for different positions of the illumination source while the camera

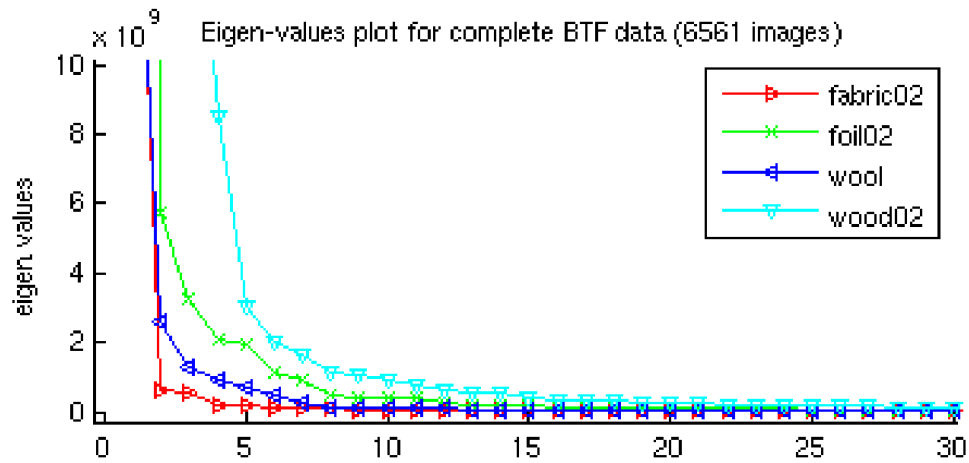


Fig. 4. The 30 largest eigenvalues for leather, fabric, wool and wood BTF measurements.

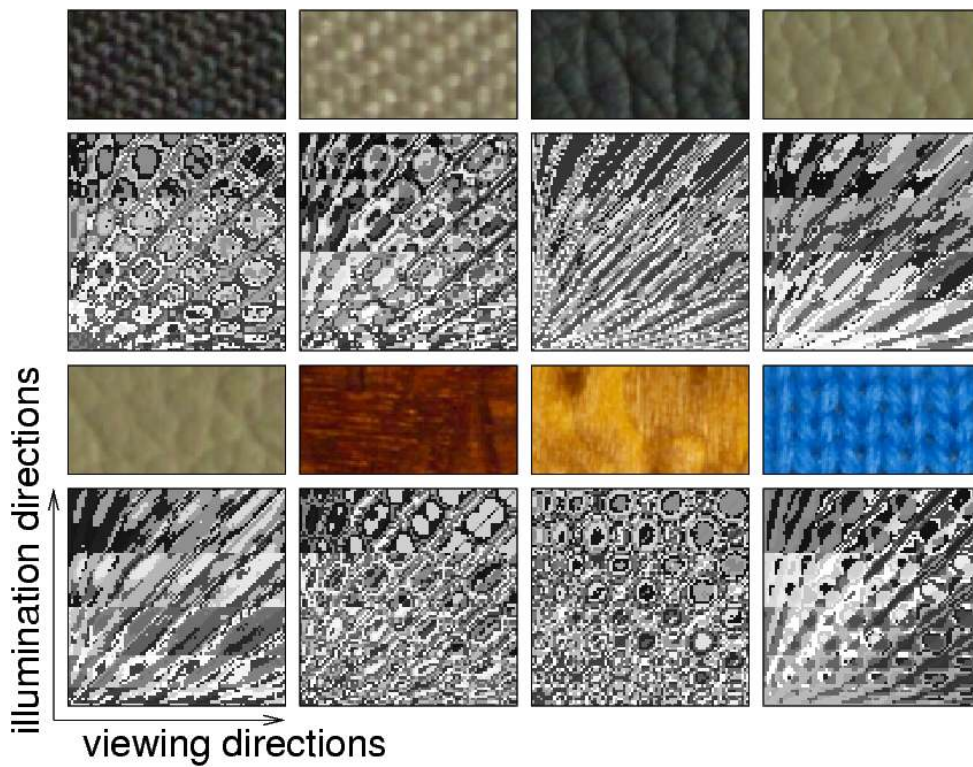


Fig. 5. Subspace index tables for eight materials in two rows: fabrics, leathers, artificial leather, lacquered woods and wool.

position is fixed. In the case of a full BTF measurement space such mutually ideally registered images are always available for free. The photometric stereo enables to acquire the normal and



albedo fields from intensity images while a Lambertian opaque surface is assumed (for details see [30]). We obtained satisfactory results for many real-world materials, e.g. plastics, leathers, fabrics, etc. while the most difficult were translucent and lacquered materials, e.g. knitted wool.

### C. Multi-Scale Colour Probabilistic Texture Model

Modeling general color subspace images (e.g., color BTF with fixed viewing and illumination angles) requires three dimensional models. If a 3D data space can be factorized, then these data can be modeled using a set of fewer-dimensional 2D random field models (3 for color), otherwise it is necessary to use a 3D random field model (see [12]). Although full 3D models allow unrestricted spatial-spectral correlation modeling, its main drawback is the large amount of parameters to be estimated and more difficult implementation in GPU. The factorization alternative is attractive, because it allows the use of simpler 2D data models with fewer parameters and consequently also easier implementation.

The multi-scale color 2D subspace probabilistic model scheme is illustrated in Fig. 6. To simplify notation we will further drop the BTF subspace indices  $(\theta, \phi)$  because the rest of this section relates to one subspace image modeling. The subspace texture model starts with a spectral PCA-based decorrelation of subspace image (result of BTF segmentation) into mono-spectral factors. Each of these factors is subsequently decomposed into sub-band components using the multi-resolution Gaussian-Laplacian pyramid. This allows to use simpler CAR models to model wide range of textures. Each such sub-band component is analyzed by a dedicated CAR factor model to obtain a compact set of model parameters.

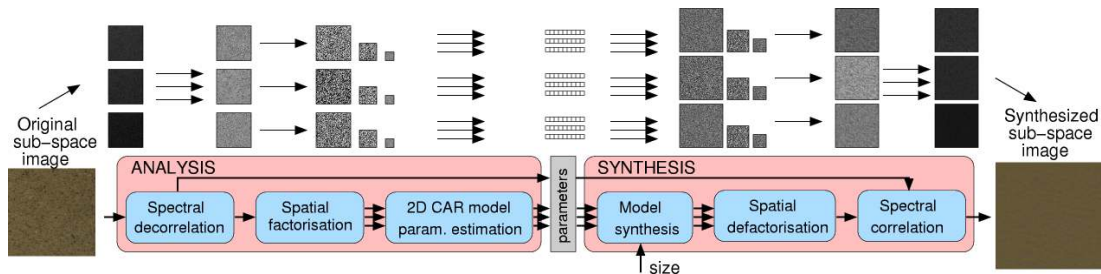


Fig. 6. The scheme of underlying subspace 2D causal auto-regressive MRF model.

*2D CAR Factor Model:* The obtained orthogonal mono-spectral sub-band components of subspace images in the form of mono-spectral images (the spectral index is further omitted) are independently modeled by their dedicated independent Gaussian noise-driven autoregressive random field model (CAR), because this model has good modeling performance [9] and allows analytical treatment [14]. 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  $M \times N$  lattice  $I$ , subject to the following condition:

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

where  $\alpha$  is a unit vector,  $\text{tr}()$  is a trace of a corresponding matrix, and the following notation is used

$$\tilde{V}_{r-1} = \sum_{k=1}^{r-1} \begin{pmatrix} Y_k Y_k^T & Y_k X_k^T \\ X_k Y_k^T & X_k X_k^T \end{pmatrix} = \begin{pmatrix} \tilde{V}_{y(r-1)} & \tilde{V}_{xy(r-1)}^T \\ \tilde{V}_{xy(r-1)} & \tilde{V}_{x(r-1)} \end{pmatrix} .$$

Here,  $r = [r_1, r_2]$  is spatial index denoting history of movements on the lattice  $I$ .

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

$$Y_r = \gamma X_r + e_r , \quad (1)$$

where  $\gamma = [a_1, \dots, a_\eta]$  is the parameter vector,  $\eta = \text{cardinality}(I_r^c)$ ,  $I_r^c$  denotes a causal (or alternatively unilateral) contextual neighborhood (CN) (i.e. all support pixels were previously visited and they are known). Furthermore,  $e_r$  denotes white Gaussian noise with zero mean and a constant but unknown variance  $\sigma^2$ , and  $X_r$  is a support vector of  $Y_{r-s}$  where  $s \in I_r^c$ .

*Parameter Estimation:* Parameter estimation of a CAR model using either the maximum likelihood, or the least square or Bayesian methods can be found analytically. The Bayesian parameter estimates of the causal AR model using the normal-gamma parameter prior are:

$$\hat{\gamma}_{r-1}^T = V_{x(r-1)}^{-1} V_{xy(r-1)} \quad \hat{\sigma}_{r-1}^2 = \frac{\lambda_{(r-1)}}{\beta(r)} \quad (2)$$

where

$$\begin{aligned} \lambda_{(r-1)} &= V_{y(r-1)} - V_{xy(r-1)}^T V_{x(r-1)}^{-1} V_{xy(r-1)} , \\ V_{(r-1)} &= \tilde{V}_{(r-1)} + V_{(0)} , \\ \beta(r) &= \beta(0) + r - 1 , \end{aligned}$$

and  $\beta(0)$  is an initialization constant and submatrices in  $V_{(0)}$  are from the parameter prior. The estimates (2) can also be evaluated recursively if necessary.

*Model Synthesis:* The CAR model synthesis is very simple. The mono-spectral sub-band components of each subspace image can be directly generated from the model equation (1). Thus a single CAR model synthesizes the individual spatial frequency bands of the subspace image. Each mono-spectral fine-resolution component is obtained from the pyramid collapse procedure (i.e. the interpolation of sub-band components - the inversion process to the creation of the Gaussian-Laplacian pyramid). Finally, the resulting synthesized color subspace image is obtained from the set of synthesized mono-spectral factors using the inverse spectral decorrelation.

During polygon rendering in a virtual scene its actual illumination and view directions are determined. Subsequently the subspace images corresponding to the three closest BTF view and illumination positions using the BTF subspace index (see Fig. 5) are determined and synthesized. Finally, they are interpolated (see Section III) to obtain BTF meso-structure corresponding to the actual polygon's illumination and view direction. A part of the interpolated image (i.e. synthesized BTF), specified by the polygon's texture coordinates is mapped onto a 3D object and subsequently used in a bump-mapping filter instead of standard albedo / color image of the material. The filter multiplies textured polygons of the 3D object with rough structure defined by the material range / normal map and the given illumination direction (for details see Section III).

### III. GPU IMPLEMENTATION

A fast implementation of the proposed BTF model requires the following three steps.

*Synthesis of the BTF Subspace Images:* As seen on the right hand side of Fig. 6, the subspace image synthesis starts with individual G-L pyramid layers synthesis using parameters of the CAR model and a Gaussian white-noise generator. Then, all synthesized pyramid layers (at most 3) of a given size are interpolated. The noise generator can be pre-computed in the form of noise texture or it can be implemented in the GPU as well. Finally, each color pixel is multiplied by the inverse spectral decorrelation  $3 \times 3$  matrix (which is stored together with the CAR parameters) to preserve the original color correlations. The output is the synthesized color subspace image. All such computations can be performed using a fragment program. This requires a causal knowledge of spatially neighboring data during the subspaces synthesis, which is completely orthogonal to the contemporary GPU hardware philosophy. This problem can



Fig. 7. Results of BTF synthesis mapped on part of a car interior. The original BTF data (left), the BTF synthesis (right). Tested materials: *lacquered woods, floor plastic, tile.*

be avoided by subsequent reading of previously synthesized pixel values from a pixel buffer. However, such an operation can be time consuming and thus, the final computation time can be similar to the standard CPU computation. On the other hand, such a problem can be circumvented in the near future with oncoming graphics hardware using faster memory chips. For example, the CPU synthesis of all the sub-space images of size  $512 \times 512$  takes about 3 seconds on average when run on a PC Athlon 1.9 GHz. Therefore, a well designed hardware implementation can considerably speed up this process. The subspace synthesis is performed only once when the corresponding material appears on the scene for the first time, and consequently, it does not require any additional computation later during online rendering. If a huge BTF space synthesis or a Monte Carlo type of scene rendering is required and simultaneously the memory is limited, the model can be used for a BTF block-wise synthesis when additional block initialization values are stored.



Fig. 8. Results of BTF synthesis mapped on part of a car interior. The original BTF data (enlarged by image tiling) (first row), smooth part of the proposed BTF synthesis (second row), relighted range-map (third row), combination of synthesized data with the range-map (fourth row). Materials from the left: two leathers *foil01*, *foil02*, three fabrics *fabric01*, *fabric02*, *knitted wool*.

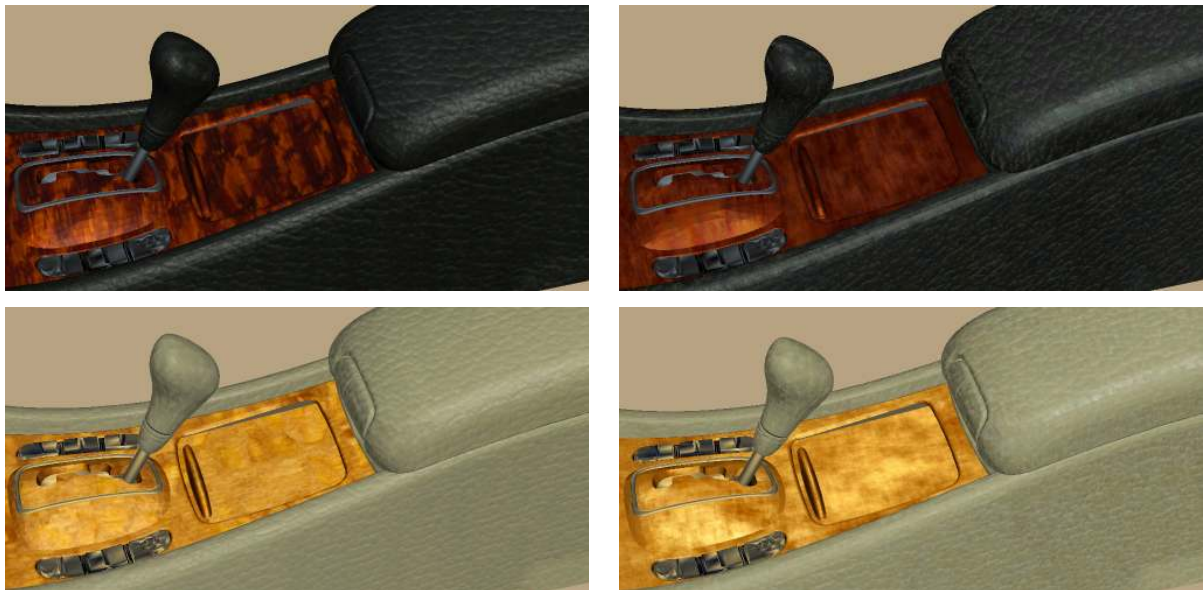


Fig. 9. BTF measurements mapped on part of a car gearbox. Original BTF data (enlarged using image tiling) (left) compared with synthesized BTF (right) for four distinct materials: *wood01*, *foil01*, *wood02*, *foil02* (3D model courtesy of DaimlerChrysler).

*BTF Interpolation:* There are seams produced that are visible on the surface of the textured objects. This is due to the relatively sparse measurements of the original BTF data space and the consequent replacement of unmeasured BTF data by the nearest available illumination and view directions. Furthermore, these inputted data are then used for either a raw data mapping or learning the BTF texture synthesis model according to a cluster index file. These artifacts were considerably suppressed when an interpolation scheme based on barycentric coordinates [4] was applied. In this approach the three closest BTF images, i.e. corresponding synthesized sub-space images, in the sense of Euclidean distance on the hemisphere, containing individual measurement points (Fig. 2), are found for both given view and illumination directions. The resulted barycentric weights for both the three closest view and illumination directions are computed. By their multiplication nine weights are obtained corresponding to nine synthesized sub-space images which are selected according to cluster index table (Fig. 5) and combined by means of the GPU fragment program.

*Bump-Map Modeling:* Rough and regular material structures with corresponding shading effects were introduced into our model using of the parallax bump mapping technique [29]. This method enables fast bump-mapping effects by means of simple vertex and fragment programs. The method approximates the correct appearance of rough surfaces by modifying the texture coordinate for each pixel with no extra polygon requirements, using only surface range and normal maps. These maps are estimated from BTF measurements, using the photometric stereo and further enlarged by image-tiling technique [26]. Thus, only a few additional range/normal-map tiles are stored together with the corresponding tile-index. The final rendering is obtained as a multiplication of relighted range-map for actual illumination direction with a smooth BTF texture obtained by the interpolation scheme mentioned in previous paragraph.

#### IV. RESULTS

We have tested the presented algorithm on BTF color measurements obtained mostly from the University of Bonn BTF database [25], such as upholstery, lacquered wood, knitwear or leather. Each BTF material sample included in the database is measured in 81 illumination and viewing angles, respectively (see Fig. 2). Final rectified BTF images have a resolution of  $800 \times 800$ . There is no robust texture similarity measure available, therefore we use only visual comparison of the observed results. Generally, we performed a CAR 2D model synthesis for different CNs and for

a different number of pyramid layers and visually chose the synthesis of subspace images which fit the BTF meso-structure best. Fig. 7 illustrates the BTF synthesis of several smooth materials, such as lacquered woods, floor plastic and tile, rendered on the car armrest, in comparison with the original BTF measurements. The visual quality of the synthetic BTFs of rough leather and fabric materials is demonstrated in Fig. 8. The synthesized images in the forth row are again compared with the enlarged original BTF measurements in the first row. The second row shows the smooth BTF interpolation while the third row illustrates the relighted range-map used to introduce rough regular macro-structure into the model. Fig. 9 shows the original BTF data and their approximation by the proposed BTF modeling approach mapped onto the same part of car gearbox and visually compared using the same view and light directions. Note that the global shadows of the gear-shift bar etc. are intentionally not included.

Even if the analyzed subspace images violate the CAR model stationary assumption, the proposed BTF modeling approach demonstrates its ability to model BTFs of different materials with an acceptable visual realism. We used only low-order CAR models for experiments with less than four contextual neighbors. Compromise results were obtained for translucent and glossy materials, where the Lambertian assumption was breached, e.g., *knitted wool* in Fig. 8. The BTF synthesis for this material does not accurately correspond to the original and the synthesized images look more or less like a rigid surface. In spite of this shortcoming, the proposed BTF model is capable of reliable approximation of many real-world materials. The CAR model is a probabilistic random field type of model and as such it has problems to model strictly spectrally regular textures (e.g. chequer-board, stripes) or some clearly identifiable texture primitives. However the model always tries to approximate such patterns and this approximation can be further improved by increasing the cardinality of the contextual neighborhood  $I_r^c$ . If the regularity is present in the estimated range map (e.g. warp) the model can reproduce it precisely. The best model performance was obtained for wood and leather. For instance, the synthesized results for leather and wood in Figures 8 and 9 are almost visually indistinguishable from their original patterns. However, since the overall meso-structure appearance is driven by the underlying smooth MRF model, the selection of suitable synthesis is very important. The overall BTF space in our parametric extremely compressed representation requires at most 300 KB (including tiled range-map and normal-map, subspace index, 2D CAR model parameters). In comparison, BTF measurements per material for the Bonn data are 1.2 GB. Hence we are able

to reach the compression ratio  $\sim 1 : 1 \times 10^5$ . We have implemented several recently published BTF modeling approaches [8], [10], [12], [13], [26] and compared their basic features. All the intelligent sampling methods [13], [26] (BTF tiling), the reflectance model [8] (LMP), the GMRF model [10], the CAR 3D model [12] and the proposed CAR 2D model fulfill the successful texture synthesis prerequisites: a seamless BTF enlargement and having strictly separated analytical and synthesis parts. Several of these methods enable a block-wise synthesis, parallelization and GPU implementation, but only the GMRF, CAR 2D and CAR 3D models offer an extreme BTF data compression and even a modeling of previously unseen (unlearned) BTF data. Although the underlying model is Markovian, the method does not need any time-consuming numerical optimization, such as the usually employed Markov chain Monte Carlo methods. The CAR model synthesis is much faster than the exceptionally fast GMRF model [10] implemented on a toroidal underlying lattice. However, the CAR 2D model in comparison with the CAR 3D model involves storage of significantly fewer parameters since the individual subspace images are synthesized separately. This enables faster synthesis of individual subspace images implemented directly in graphics hardware. The BTF data analysis takes up to several minutes for all mentioned MRF models. The time required for the synthesis of 15 BTF subspace images (of size  $512 \times 512$ ) by the proposed method is 3 s (GMRF 16 s, CAR 3D 6 s) on a PC Athlon 1.9 GHz.

## V. CONCLUSION

The test results of our BTF compression and modeling algorithm on recently available BTF data are very promising. Some synthetic BTFs reproduce the given measured BTFs in such a way that both natural and synthetic BTFs are visually almost indiscernible. However, the method is a random field based model and thus, no two realizations of the same texture are pixel-wise identical. Nevertheless, they create a visual impression of the same material. The method shows good performance on most of the real-world materials with non-regular surface structure (leathers, wood, etc.) while it has compromised results for translucent, highly non-Lambertian materials with regular surface structure (wool, some kinds of fabrics, etc.). Using any fast displacement mapping technique instead of a bump filter (which is only an approximation) can further improve visual quality of our model. The proposed method allows huge compression ratio (only tens of parameters per BTF - i.e. unattainable by any alternative BTF compression method) for



transmission or storing texture information, while it has a low computation complexity. Therefore, CAR model is significantly better suited for real time or web applications than most other published alternatives. The method also allows a BTF data space restoration. In an extreme situation the whole BTF space can be approximated from a single measurement. The bump filter is implemented using fragment and vertex programs of the current graphics cards to speed up the rendering of our algorithm. The presented method is based on an estimated model (in contrast to the prevailing intelligent sampling type of methods) and as such it can only approximate the reality of the original measurement. However, it offers easy simulation and exceptionally fast seamless synthesis of any texture size and even synthesis of non-existing (previously unmeasured) BTF samples.

#### ACKNOWLEDGMENT

This research was supported by the EC project no. FP6-507752 MUSCLE and grants A2075302, 1ET400750407, of GAAV CR and 1M0572, 2C06019 of MŠMT. The authors wish to thank R. Klein of the University of Bonn for providing us with the BTF measurements.

#### REFERENCES

- [1] J. Bennett and A. Khotzand, "Multispectral random fields for synthesis and analysis of colour images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 327–332, 1998.
- [2] J. Blinn, "Models of light reflection for computer synthesized pictures," in *Computer Graphics Proceedings, Annual Conference Series*. ACM SIGGRAPH 1977, ACM Press, 1977, pp. 192–198.
- [3] M. Cohen, J. Shade, S. Hiller, and O. Deussen, "Wang tiles for image and texture generation," in *ACM SIGGRAPH 2003*, ACM Press, vol. 22, New York, NY, USA, July 2003, pp. 287–294.
- [4] H. S. M. Coxeter, *Introduction to Geometry*. New York: Wiley, 1969.
- [5] K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, 1999.
- [6] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *ACM SIGGRAPH 2001*, ACM Press, E. Fiume, Ed., 2001, pp. 341–346.
- [7] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *ICCV (2)*, 1999, pp. 1033–1038.
- [8] J. Filip and M. Haindl, "Non-linear reflectance model for bidirectional texture function synthesis," in *Proceedings of 17th International Conference on Pattern Recognition*, vol. 1. IEEE Computer Society Press, August 2004, pp. 80–84.
- [9] M. Haindl, "Texture synthesis," *CWI Quarterly*, vol. 4, no. 4, pp. 305–331, December 1991.
- [10] M. Haindl and J. Filip, "Fast BTF texture modelling," in *Texture 2003*, October 2003, pp. 47–52.
- [11] —, "A fast probabilistic bidirectional texture function model," in *Lecture Notes in Computer Science. 3212*, vol. 2. Berlin Heidelberg: Springer-Verlag, September 2004, pp. 298–305.

- [12] M. Haindl, J. Filip, and M. Arnold, "BTF image space utmost compression and modelling method," in *Proceedings of 17th International Conference on Pattern Recognition*, vol. 3. IEEE Computer Society Press, August 2004, pp. 194–198.
- [13] M. Haindl and M. Hatka, "BTF roller," in *Texture 2005*, Edinburgh: Heriot-Watt University, October 2005, pp. 89–94.
- [14] M. Haindl and V. Havlíček, "A multiresolution causal colour texture model," in *Advances in Pattern Recognition, Lecture Notes in Computer Science 1876*. Berlin Heidelberg: Springer-Verlag, 2000.
- [15] M. Koudelka, S. Magda, P. Belhumeur, and D. Kriegman, "Acquisition, compression, and synthesis of bidirectional texture functions," in *Texture 2003*, October 2003, pp. 47–52.
- [16] E. P. Lafortune, S. C. Foo, K. E. Torrance, and D. P. Greenberg, "Non-linear approximation of reflectance functions," *Computer Graphics*, vol. 31, no. Annual Conference Series, pp. 117–126, 1997.
- [17] X. Liu, Y. Yu, and H. Y. Shum, "Synthesizing bidirectional texture functions for real-world surfaces," in *ACM SIGGRAPH 2001*, ACM Press, E. Fiume, Ed., 2001, pp. 97–106.
- [18] 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*, vol. 10, no. 3, pp. 278–289, June 2004.
- [19] T. Malzbender, D. Gelb, and H. Wolters, "Polynomial texture maps," in *ACM SIGGRAPH 2001*, ACM Press. Eurographics Association, Switzerland, 2001, pp. 519–528.
- [20] D. K. McAllister, A. Lastra, and W. Heidrich, "Efficient rendering of spatial bi-directional reflectance distribution functions," *Graphics Hardware*, pp. 77–88, 2002.
- [21] J. Meseth, G. Müller, and R. Klein, "Preserving realism in real-time rendering of bidirectional texture functions," in *OpenSG Symposium 2003*. Eurographics Association, Switzerland, April 2003, pp. 89–96.
- [22] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein, "Acquisition, synthesis and rendering of bidirectional texture functions," in *Eurographics 2004, STAR - State of The Art Report*, Eurographics Association, 2004, pp. 69–94.
- [23] A. Neubeck, A. Zalesny, and L. Gool, "3d texture reconstruction from extensive btf data," in *Texture 2005*, Edinburgh: Heriot-Watt University, October 2005.
- [24] F. Nicodemus, R. J.C., J. Hsia, I. Ginsburg, and T. Limperis, "Geometrical considerations and nomenclature for reflectance," *NBS Monograph 160, National Bureau of Standards, U.S. Department of Commerce*, pp. 1–52, October 1977.
- [25] M. Sattler, R. Sarlette, and R. Klein, "Efficient and realistic visualization of cloth," in *Eurographics Symposium on Rendering 2003*, June 2003.
- [26] P. Somol and M. Haindl, "Novel path search algorithm for image stitching and advanced texture tiling," in *Proceedings of 13-th WSCG05*, February 2005.
- [27] M. Vasilescu and D. Terzopoulos, "TensorTextures: Multilinear image-based rendering," *ACM SIGGRAPH 2004*, ACM Press, vol. 23, no. 3, pp. 336–342, August 2004.
- [28] J. Wang and K. J. Dana, "Hybrid textons: Modeling surfaces with reflectance and geometry," in *CVPR (1)*, 2004, pp. 372–378.
- [29] T. Welsch, "Parallax mapping with offset limiting: A per-pixel approximation of uneven surfaces," Infiscape Corporation, Tech. Rep., 2004.
- [30] R. Woodham, "Analysing images of curved surface," *Artificial Intelligence*, vol. 17, no. 5, pp. 117–140, 1981.
- [31] H. Yacov, T. Malzbender, D. Gelb, "Synthesis of reflectance function textures from examples," in *Texture 2003*, pp. 47–52.