

# A Fast Probabilistic Bidirectional Texture Function Model

M. Haindl & J. Filip  
{haindl,flipj} @utia.cas.cz

Dept. of Pattern Recognition, Institute of Information Theory and Automation,  
Academy of Sciences of the Czech Republic, Prague, Czech Republic

**Abstract.** The bidirectional texture function (BTF) describes rough texture appearance variations due to varying illumination and viewing conditions. Such a function consists of thousands of measurements (images) per sample. Resulted 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 modelling allowing such an efficient compression with possibility of direct implementation inside the graphics card. The analytical step of the algorithm starts with the BTF space segmentation and range map estimation of the BTF surface followed by the spectral and spatial factorisation of selected sub-space multispectral texture images. Single monospectral band-limited factors are independently modelled by their dedicated causal autoregressive models (CAR). During rendering the corresponding sub-space images of arbitrary size are synthesised and both multispectral and range information is combined in a bump mapping filter of the rendering hardware according to view and illumination positions. The presented model offers 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.

## 1 Introduction

A physically correct virtual models visualisation cannot be accomplished without nature-like colour textures covering virtual or augmented reality scene objects. These textures can be either smooth or rough (also referred as the bidirectional texture function - BTF [31]). The rough textures which have rugged surfaces do not obey the Lambert law and their reflectance is illumination and view angle dependent. Both types of textures occur in virtual scenes models can be either digitised natural textures or textures synthesised from an appropriate mathematical model. The former simplistic option suffers among others with extreme memory requirements for storage of a large number of digitised cross-sectioned slices through different material samples. Sampling solution become unmanageable for rough textures which require to store thousands of different illumination and view angle samples for every texture. Such a simple VR scene

requires to store tera bytes of texture data which is far out of limits for any current real time hardware. Several intelligent sampling methods [2], [4], applied either to smooth or BTF data were proposed with the aim to diminish these huge memory requirements. All these methods are based on some sort of original small texture sampling and the best of them produce very realistic synthetic textures. However these methods require to store thousands images for every combination of viewing and illumination angle of the original target texture sample, they often produce visible seams, some of them are computationally demanding and they cannot generate textures unseen by these algorithms.

Synthetic textures are far more flexible, extremely compressed (few tens of parameters have to be stored only), they may be evaluated directly in procedural form and can be designed to meet certain constraints or properties, so that they can be used to fill an infinite texture space without visible discontinuities. Several smooth texture modelling approaches were published, e.g., [?], [15] and some survey articles are available [?], [?] as well. The random field based models quite successfully represent high frequencies present in natural textures low frequencies are much more difficult for them. One possibility how to overcome this drawback is to use a multiscale random field model. Unfortunately autoregressive random fields, similarly as the majority of other Markovian types of random field models, are not invariant to multiple resolution decomposition (MRD) even for simple MRD like subsampling and the lower-resolution images generally lose their autoregressive property (or wide-sense Markovianity property) and become ARMA random fields instead. Fortunately we can avoid computationally demanding approximations of an ARMA multigrid random field by an infinite order (i.e., high order in practice) autoregressive random fields because there is no need to transfer information between single spatial factors hence it is sufficient to analyse and synthesise each resolution component independently.

The rest of the paper is organised as follows. The following section describes probabilistic BTF synthesis algorithm composed from BTF space segmentation, range-map estimation and multiscale smooth texture synthesis. We discuss the underlying CAR model parameter estimation and synthesis solutions. Results are reported in the section 3, followed by a conclusions in the last section.

## 2 BTF Model

We propose a novel algorithm Fig.1 for efficient rough texture modelling which combines an estimated range map with synthetic multiscale smooth textures generated using a set of simultaneous causal autoregressive random (CAR) field based models. The material visual appearance during changes of viewing and illumination conditions is simulated using the bump-mapping technique. This idea of perturbing surface normals using a range-map was introduced by Blinn [23]. The overall appearance is guided by the corresponding underlying sub-space model. The obvious advantage of this solution is the possibility to use hardware support of bump technique in contemporary visualisation hardware.

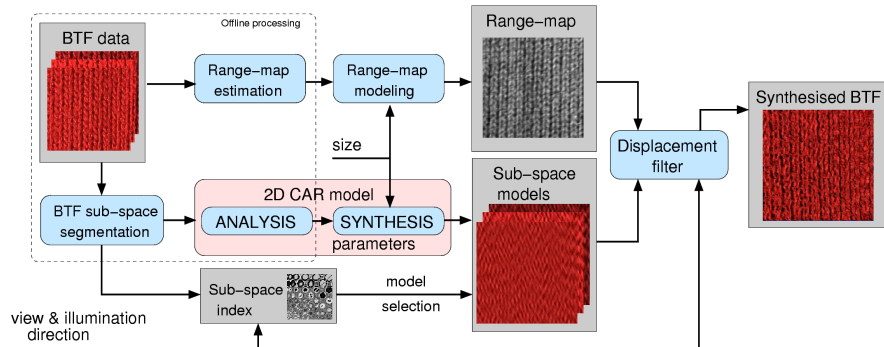


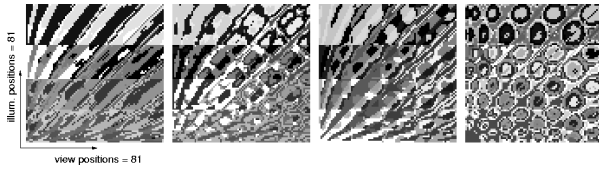
Fig. 1. The overall BTF algorithm scheme.

## 2.1 BTF Space Segmentation

A single probabilistic BTF model combined with the bump-mapping filter can potentially approximate the whole BTF measurement space. However approximation errors for significantly different illumination and viewing angles can worsen visual realism of certain virtual textured object faces. Thus we propose to compromise extreme compression ratio for the visual quality and to use several probabilistic BTF subspace dedicated models. The off-line part of our algorithm Fig.1 starts with the BTF space segmentation into several subspaces using the simple K-means algorithm, in representation of  $81 \text{ view} \times 81 \text{ illum.}$  positions, using colour histogram data features. Such a representation is numerically more efficient than the alternative texture spatial representation because the sample resolution is usually larger than the number of angle discretization steps. The eigen value analysis (PCA) lead us to conclusion that the intrinsic BTF space dimensionality for most BTF texture samples is between ten and twenty. Hence the first 10-20 eigen-values contain 95% of the whole information. BTF space segmentation for different materials are depicted in Fig.2.

## 2.2 Range Map Estimation

The overall roughness of a textured surface significantly influences a BTF texture appearance. Such a surface can be specified using its range map, which can be estimated by several existing approaches. The most accurate range map can be estimated by direct measurement 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 images are more appropriate. One of the most accurate approaches used in this paper is the *Photometric Stereo* which estimates surface range map from at least three images obtained for different position of illumination source while the camera position is fixed. In the case of full BTF measurement space such mutually registered images are always available for free. The photometric stereo enables to acquire the normal and albedo fields from no less than three intensity images while Lambertian opaque surface is assumed. For details see [?].



**Fig. 2.** Sub-space index images for four different materials: leather (12 clusters), fabric (13 clusters), wool (15 clusters) and lacquered wood (15 clusters).



**Fig. 3.** Lacquered wood synthesised BTF rendered on the sphere.

### 2.3 Smooth Texture Model

Modelling general multispectral (e.g., colour) texture images requires three dimensional models. If a 3D data space can be factorised then these data can be modelled using a set of less-dimensional 2D random field models, otherwise it is necessary to use some 3D random field model. Although full 3D models allows unrestricted spatial-spectral correlation modelling its main drawback is large amount of parameters to be estimated. The factorisation alternative is attractive because it allows using simpler 2D data models with less parameters.

**Spectral Decorrelation** A real data space can be decorrelated only approximately, hence the independent spectral component modelling approach suffers with some loss of image information, however this loss of spectral information is only visible in textures with many substantially different colours. Spectral factorisation using the Karhunen-Loeve expansion transforms the original centered data space  $\tilde{Y}$  defined on the rectangular  $M \times N$  finite lattice  $I$  into a new data space with K-L coordinate axes  $\tilde{Y}$ . This new basis vectors are the eigenvectors of the second-order statistical moments matrix (1)

$$\Phi = E\{\tilde{Y}_r \tilde{Y}_r^T\} \quad (1)$$

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

$$T = [u_1^T, u_2^T, u_3^T]^T \quad (2)$$

which has single rows  $u_j$  that are eigenvectors of the matrix  $\Phi$ . Components of the transformed vector  $\tilde{Y}_r = T\tilde{Y}_r$  are mutually uncorrelated and if we assume that they are also Gaussian then they are independent thus each transformed monospectral factor can be modelled independently of remaining spectral factors.

**Spatial Factorization** Input multispectral image is factorised into  $d$  monospectral images  $\tilde{Y}_\bullet$  for  $i = 1, \dots, d$  (the notation  $\bullet$  has the meaning of all possible values of the corresponding index). These components are further decomposed into a multi-resolution grid and each resolution data are independently modelled by their dedicated CAR models. Each one generates a single spatial frequency

band of the texture. An analysed texture is decomposed into multiple resolutions factors using 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 a reduction factor  $n$  is

$$\ddot{Y}_r^{(k)} = \downarrow_r^n (\ddot{Y}_{\bullet,i}^{(k-1)} \otimes w) \quad k = 1, 2, \dots, \quad (3)$$

where  $\ddot{Y}_{\bullet}^{(0)} = \bar{Y}_{\bullet}$ ,  $\downarrow^n$  denotes down-sampling with reduction factor  $n$  and  $\otimes$  is the convolution operation. The convolution mask based on weighting function (FIR generating kernel)  $w$  is assumed to execute separability, normalization, symmetry and equal contribution constrains. The FIR equation is then

$$\ddot{Y}_r^{(k)} = \sum_{i,j=-l}^l \hat{w}_i \hat{w}_j \ddot{Y}_{2r+(i,j)}^{(k-1)}. \quad (4)$$

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:

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

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

**CAR Factor Model** Single orthogonal monospectral components are thus decomposed into a multi-resolution grid and each resolution data 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 realisations  $Y$  of the  $M \times N$  lattice  $I$ , subject to following condition:

$$p(Y | \gamma, \sigma^{-2}) = (2\pi)^{-\frac{(MN-1)}{2}} |\sigma^{-2}|^{\frac{(MN-1)}{2}} \exp \left\{ \frac{-1}{2} tr \left\{ \sigma^{-2} \begin{pmatrix} -\alpha \\ \gamma^T \end{pmatrix}^T \tilde{V}_{MN-1} \begin{pmatrix} -\alpha \\ \gamma^T \end{pmatrix} \right\} \right\}, \quad (6)$$

where  $\alpha$  is a unit vector and the following notation is used

$$\tilde{V}_{r-1} = \begin{pmatrix} \tilde{V}_{y(r-1)} & \tilde{V}_{xy(r-1)}^T \\ \tilde{V}_{xy(r-1)} & \tilde{V}_{x(r-1)} \end{pmatrix},$$

where  $\tilde{V}_{a(r-1)} = \sum_{k=1}^{r-1} A_k A_k^T$  and  $\tilde{V}_{ab(r-1)} = \sum_{k=1}^{r-1} A_k B_k^T$ .

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

$$Y_r = \gamma X_r + e_r, \quad (7)$$

where  $\gamma = [a_1, \dots, a_\eta]$  is the parameter vector,  $I_r^c$  is a causal neighbourhood with  $\eta = \text{card}(I_r^c)$  and  $e_r$  is a white Gaussian noise with zero mean and a constant but unknown variance  $\sigma^2$  and  $X_r$  is a corresponding vector of  $Y_{r-s}$ .

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

$$\hat{\gamma}_{r-1}^T = V_{x(r-1)}^{-1} V_{xy(r-1)} \quad , \quad \hat{\sigma}_{r-1}^2 = \frac{\lambda_{(r-1)}}{\beta(r)} \quad (8)$$

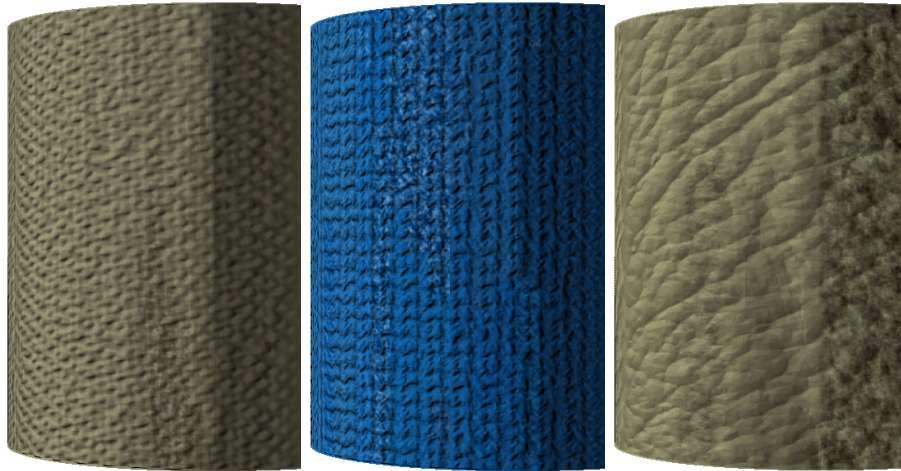
and where  $V_{z(r-1)} = \tilde{V}_{z(r-1)} + V_{z(0)}$  and matrices  $V_{z(0)}$  are from parameter prior. The estimates (8) can be also evaluated recursively if necessary.

**Model Synthesis** The CAR model synthesis is very simple and a causal CAR random field can be directly generated from the model equation (7). Single CAR models synthesize spatial frequency bands of the texture. Each monospectral fine-resolution component is obtained from the pyramid collapse procedure (inversion process to (3),(5)). Finally the resulting synthesized colour texture is obtained from the set of synthesized monospectral images using the inverse K-L transformation. If a single visualized scene simultaneously contains BTF texture view and angle combinations which are modelled by different probabilistic models (i.e., models supported by different BTF subspaces) for the same material all such required textures are easily synthesized simultaneously. Simultaneous synthesis allows to avoid difficult texture registration problems.

During rendering the four closest BTF view and illumination positions from the BTF sub-space index, computed during segmentation, are taken along with corresponding weights. The resulted image is interpolation of these synthesized images. The resulted image is finally combined with range-map according to illumination direction and sent to a polygon being processed.

### 3 Results

We have tested the algorithm on BTF colour measurements mostly from the University of Bonn such as upholstery, lacquered wood, knitwear or leather textures. Each BTF material sample comprised in Bonn database is measured in 81 illumination and viewing angles, respectively and has resolution  $800 \times 800$ . Fig.4 demonstrates synthesised results for three different materials: fabric, wool and leather rendered on cylinders consisting of  $64 \times 64$  polygon mesh. Fig. 3 illustrates the lacquered wood smooth BTF synthesis rendered on the sphere (without bump). Even if the analysed textures violates the CAR model stationarity assumption, the proposed algorithm demonstrates its ability to model such BTF textures with acceptable visual realism. We used only low order CAR models for experiments in Figs.4 with less than 7 contextual neighbours. Figures above demonstrate also the colour quality of the model. Our synthesised images manifest comparable colour quality with the much more complex 3D models [15]. The overall BTF space in our parametric extremely compressed representation



**Fig. 4.** Three distinct BTF materials - fabric, wool and leather are rendered on a cylinder using the proposed BTF model combined with the bump mapping filter.

requires 300 KB space (range-map, sub-space index, 2D CAR model parameters). BTF measurements per sample for the Bonn data are 40GB hence we are able to reach the compression ratio  $1 : 8 \times 10^6$ . Moreover the proposed algorithm is very fast, since BTF synthesis of sub-space images from 2D CAR model parameters takes up about 3 seconds in average on Athlon 1.9GHz. In our case the bump mapping was computed on CPU, but fast implementation on GPU using fragment programs can be exploited as well [25].

#### 4 Summary and Conclusions

The test results of our algorithm on available BTF data are encouraging. Some synthetic textures reproduce given measured texture images so that both natural and synthetic texture are almost visually indiscernible. The overwhelming amount of original colour tones were reproduced realistically in spite of restricted spectral modelling power of the model due to spectral decorrelation error. The proposed method allows huge compression ratio (unattainable by alternative intelligent sampling approaches) for transmission or storing texture information while it has low computation complexity. The method does not need any time-consuming numerical optimization like for example the usually employed Markov chain Monte Carlo methods. The CAR model synthesis is much faster than the MRF model synthesis even faster than the exceptionally fast Gaussian Markov random field model [19] implemented on a toroidal underlying lattice. The CAR model is better suited for real time (e.g. using a graphical card processing unit) or web applications than most other published alternatives. The presented technique allows to avoid difficult texture registration problem with simple simultaneous synthesis of all necessary sub-space textures. The method allows also a BTF data space restoration, in the extreme situation the whole

BTF space can be approximated from a single measurement. The bump filter can be implemented using fragment programs with the novel graphics cards to speed up the rendering of our algorithm results. The presented method is based on the estimated model in contrast to prevailing intelligent sampling type of methods, and as such it can only approximate realism of the original measurement. However it offers unbeatable data compression ratio (tens of parameters per BTF only), easy simulation of even non existing (previously not measured) BTF textures and fast seamless synthesis of any texture size.

## Acknowledgements

This research was supported by the EC project no. IST-2001-34744 RealReflect, grant No.A2075302 of the Grant Agency of the Academy of Sciences CR and partially by the grant MŠMT No. ME567 MIXMODE. The authors wish to thank R. Klein of the University of Bonn for providing us with the BTF measurements.

## References

1. De Bonet, J.: Multiresolution sampling procedure for analysis and synthesis of textured images. In: Proc. SIGGRAPH 97, ACM (1997) 361–368
2. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proc. Int. Conf. on Computer Vision (2). (1999) 1033–1038
3. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In Fiume, E., ed.: SIGGRAPH 2001, Computer Graphics Proceedings, ACM Press / ACM SIGGRAPH (2001) 341–346
4. Heeger, D., Bergen, J.: Pyramid based texture analysis/synthesis. In: Proc. SIGGRAPH 95, ACM (1995) 229–238
5. Xu, Y., Guo, B., Shum, H.: Chaos mosaic: Fast and memory efficient texture synthesis. Technical Report MSR-TR-2000-32, Redmont (2000)
6. Dong, J., Chantler, M.: Capture and synthesis of 3d surface texture. In: Texture 2002. Volume 1., Heriot-Watt University (2002) 41–45
7. McAllister, D., Lastra, A., Heidrich, W.: Efficient rendering of spatial bi-directional reflectance distribution functions. In: Eurographics, Blackwell Publishers (2002)
8. Meseth, J., Müller, G., Klein, R.: Preserving realism in real-time rendering. In: OpenGL Symposium, Eurographics Association, Switzerland (2003) 89–96
9. Besag, J.: Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society, Series B **B-36** (1974) 192–236
10. Chellappa, R.: Two-dimensional discrete gaussian markov random field models for image processing. In Kanal, L., Rosenfeld, A., eds.: Progress in Pattern Recognition 2, North-Holland, Elsevier (1985) 79–112
11. Kashyap, R.: Analysis and synthesis of image patterns by spatial interaction models. In Kanal, L., A.Rosenfeld, eds.: Progress in Pattern Recognition 1, North-Holland, Elsevier (1981)
12. Portilla, J., Simoncelli, E.: A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision **40** (2000) 49–71
13. Grim, J., Haindl, M.: Texture modelling by discrete distribution mixtures. Computational Statistics & Data Analysis **41** (2003) 603–615





34. Liu, X., Yu, Y., Shum, H.: Synthesizing bidirectional texture functions. ACM SIGGRAPH (2001)
35. Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B., Shum, H.Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. ACM Transactions on Graphics (TOG) **21** (2002) 665–672