A Moving Average Bidirectional Texture Function Model

Michal Havlíček and Michal Haindl

Institute of Information Theory and Automation of the ASCR
Prague, Czech Republic
{haindl,havlimi2}@utia.cz

Abstract. The Bidirectional Texture Function (BTF) is the recent most advanced representation of visual properties of surface materials. It specifies their appearance due to varying spatial, illumination, and viewing conditions. Corresponding enormous BTF measurements require a mathematical representation allowing extreme compression but simultaneously preserving its high visual fidelity. We present a novel BTF model based on a set of underlying mono-spectral two-dimensional (2D) moving average factors. A mono-spectral moving average model assumes that a stochastic mono-spectral texture is produced by convolving an uncorrelated 2D random field with a 2D filter which completely characterizes the texture. The BTF model combines several multi-spectral band limited spatial factors, subsequently factorized into a set of mono-spectral moving average representations, and range map to produce the required BTF texture space. This enables very high BTF space compression ratio, unlimited texture enlargement, and reconstruction of missing unmeasured parts of the BTF space.

Keywords: BTF, texture analysis, texture synthesis, data compression, virtual reality, moving average random field.

1 Introduction

Realistic virtual reality scenes require objects covered with synthetic textures visually as close as possible to the corresponding real surface materials appearance they emulate under any required viewing conditions. Such textures have to model real non-Lambertian rugged surfaces whose reflectance is illumination and view angle dependent. Recent most advanced visual representation of such surfaces is the Bidirectional Texture Function (BTF) [1, 2] which is a 7-dimensional function describing surface appearance variations due to varying spatial position and illumination and viewing angles. Such a function is typically measured by thousands of images per material sample, each taken for a specific combination of the illumination and viewing condition. Visual textures can be either represented by digitized measured textures or textures synthesized from an appropriate mathematical model. Using digitized textures directly suffers among others with extreme memory requirements for storage of a large number of digitized cross

R. Wilson et al. (Eds.): CAIP 2013, Part II, LNCS 8048, pp. 338–345, 2013.

sectioned slices through different material samples of the measured BTF space [2]. Moreover this solution become even unmanageable for physically correctly modeled scenes with BTF surfaces representation because even a simple scene with only several materials requires to store tera bytes of textural data which is still far out of limits for any current and near-future hardware. Several so called intelligent sampling methods (for example [4–6] and some others) were proposed to reduce these extreme 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, they still require to store thousands images for every combination of viewing and illumination angle of the original target texture sample and additionally they often produce images with undesirable visual artifacts like visible seams (except for the method presented in [7]). Some of them are very computationally demanding and none of them is able to generate previously unseen textures (i.e., BTF space reconstruction). Contrary to the sampling approaches, the synthetic textures generated from mathematical models are more flexible and extremely compressed, because only tens of parameters have to be stored only instead of the original visual measurements. They may be evaluated directly in a procedural form and can be used to fill virtually infinite texture space without visible discontinuities. On the other hand, mathematical models can only approximate real measurements, which might result in visual quality compromise. A BTF texture representation requires seven dimensional mathematical models, but it is possible to approximate this general BTF model with a set of much simpler three or two dimensional factorial models. Such a compromise obviously leads to some information loss.

The proposed underlaying moving average model suffers from inability to represent low frequencies present in natural textures. But this problem can be negotiated by utilizing a multiple resolution decomposition such as the Gaussian Laplacian pyramid. The hierarchy of different resolutions of an input image provides a transition between pixel level features and region or global features and hence such a representation simplify modelling a large variety of possible textures. Each band limited component is modeled independently. BTF moving average model represents a novel method for efficient rough texture modelling which combines an estimated range map with synthetic smooth multi-spectral texture generated by the set of multiscale mono-spectral moving average models. The texture visual appearance during changes of viewing and illumination conditions are simulated using either the bump mapping [8] or displacement mapping [9] technique. The obvious advantage of this solution is the possibility to exploit direct support for both bump and displacement mapping techniques in the contemporary graphics hardware.

2 Moving Average BTF Model

The BTF model combines an estimated enlarged material range map (section 2.1) with synthetic multiscale multi-spectral smooth texture (sections 2.2-2.6). We seek a trade-off between an extreme compression ratio and the visual quality by using several probabilistic BTF subspace dedicated models. The intrinsic

BTF space dimensionality is estimated using the eigenanalysis approach and the segmentation is done using the K-means clustering in the perceptually uniform CIE Lab color-space (see details in [2, 10]). Each modeled BTF subspace is further spectrally decorrelated (section 2.2) and decomposed into several spatial factors (section 2.3). The mono-spectral band limited parts of single BTF subspaces are modeled using the 2D moving average models (section 2.4). Finally, the overall BTF texture visual appearance during changes of viewing and illumination conditions is simulated using either the bump [8] or displacement mapping [9] techniques.

2.1 Range Map Modelling

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 either measured or estimated by several existing approaches such as the shape from shading [11], shape from texture [12] or photometric stereo [13]. The photometric stereo enables to acquire the normal and albedo fields from at least three intensity images obtained for different illuminations but fixed camera position while a Lambertian opaque surface is assumed. The BTF model range map estimate can benefits from tens of ideally mutually registered BTF measurements (e.g., 81 for a fixed view of the University of Bonn data [3]) and uses the overdetermined photometric stereo from mutually aligned BTF images. However, the photometric stereo method is not well suited for surfaces with highly specular reflectance, highly subsurface scattering or strong occlusion, since it breaks the Lambertain reflectance assumption. Estimated range map is enlarged into any required size using the roller method [7].

2.2 Spectral Decorrelation

Measured visual surface data can be spectrally decorrelated only approximately, therefore this step leads to certain loss of information. Spectral factorization is performed by the Karhunen-Loeve transformation (K-L). The original data space $\tilde{Y}$ is transformed into new one with coordinate axes $\bar{Y}$. New basis consists of the eigenvectors of the second-order statistical moments matrix $V = E\{\tilde{Y}_r \tilde{Y}_r^T\}$ where r denotes a multiindex $r = (r_1, r_2)$, $r \in I$, with the row and column indices, $\bullet$ all possible values of the corresponding index, and I is a finite discrete 2-dimensional rectangular $M \times N$ index lattice. The projection of a random vector $\tilde{Y}_r$ onto the K-L coordinate system uses transformation matrix which consists of eigenvectors of V. The total number of those eigenvectors depends on the number of spectral bands in the original data. If we assume that components of the transformed data $\bar{Y}_{r,\bullet} = T\tilde{Y}_{r,\bullet}$ are Gaussian then they are independent and thus each mono-spectral factor can be modeled independently.

2.3 Spatial Factorization

The spatial factorisation is technique that enables separate modelling of individual band limited frequency components of input image data and thus to use random field models with small compact contextual support. Each grid resolution represents a single spatial frequency band of the texture which corresponds to one layer of Gaussian-Laplacian pyramid (G-L) [2]. The input data are decomposed into a multi-resolution grid and all resolution data factors represents the Gaussian pyramid of level k which is a sequence of k images in which each one is a low-pass down-sampled version of its predecessor. An analysed data are decomposed into multiple resolutions factors using the Laplacian pyramid and the intermediary Gaussian pyramid. Each level of Laplacian pyramid generates a single spatial frequency band of the data. Laplacian pyramid contains band-pass components and provides a good approximation to the Laplacian of the Gaussian kernel. It can be computed by differencing single Gaussian pyramid layers.

2.4 2D Moving Average Texture Model

Single mono-spectral smooth texture factors are modelled using the moving average model [14] (MA^{2D}). A stochastic mono-spectral texture can be considered to be a sample from 2D random field defined on an infinite 2D lattice. Let us denote I a finite discrete 2-dimensional rectangular index lattice for some input factor Y represented by the MA^{2D} random field model, Y_r is the intensity value of a mono-spectral pixel $r \in I$ in the image space. The model assumes that each factor is the output of an underlying system which completely characterizes it in response to a 2D uncorrelated random input. This system can be represented by the impulse response of a linear 2D filter. The intensity values of the most significant pixels together with their neighbours are collected and averaged, and the resultant 2D kernel is used as an estimate of the impulse response of the underlying system. A synthetic mono-spectral factor can be generated by convolving an uncorrelated 2D random field with this estimate. Suppose a stochastic mono-spectral texture denoted by Y is the response of an underlying linear system which completely characterizes the texture in response to a 2D uncorrelated random input e_r, then Y_r is determined by the following difference equation:

$$Y_r = \sum_{s \in I_r} b_s e_{r-s} \tag{1}$$

where b_s are constant coefficients and $I_r \subset I$. Hence Y_r can be represented $Y_r = h(r) * e_r$ where the convolution filter $h(r)$ contains all parameters b_s. In this equation, the underlying system behaves as a 2D filter, where we restrict the system impulse response to have significant values only within a finite region. The geometry of I_r determines the causality or non-causality of the model. The selection of an appropriate model support region is important to obtain good results: small ones cannot capture all details of the texture and contrariwise, inclusion of the unnecessary neighbours adds to the computational burden and can potentially degrade the performance of the model as an additional source of noise.

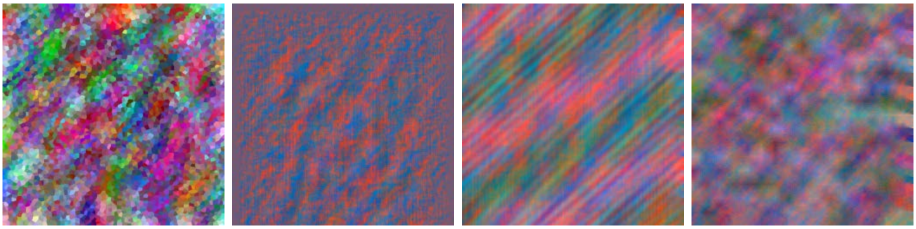

Fig. 1. Original colorful texture and it synthesis using CAR^{2D}, $GMRF^{2D}$, and MA^{2D} models (from left to right)

2.5 Parameter Estimation

To fit the model given in equation (1) to a given image Y, the parameters of $h(r)$ have to be estimated. This may be performed by using a method [14] similar to the one-dimensional Random Decrement Technique [15]. The procedure begins by arbitrarily selecting a threshold, γ usually chosen as some percentage of the standard deviation (σ) of the intensities of the input. All results in the paper use $\gamma = 0.5\sigma$. The analysis starts from the top left corner of Y and proceeds to the bottom right corner identifying the pixels at which the intensity crosses the threshold. When a threshold crossing occurs at location r, the intensity values of the support region defined by I_r around the crossing point are saved in memory (index set Γ), if among the four adjacent pixels to r, at least one in the same row and one in the same column are less than the threshold. The same procedure is followed at the next threshold crossing point and these intensity values are added to the previously saved. The summed up segments are divided by the total number of segments for the corresponding parameter estimates, i.e.,

$$\hat{b}_s = \frac{1}{cardinality\{\Gamma\}} \sum_{\forall r \in \Gamma} Y_{r+s} \quad \forall s \in I_r \ . \tag{2}$$

Additional details can be found in [14].

2.6 Model Synthesis

The underlying MA^{2D} model is able to generate synthetic images from the model parameters. Synthetic mono-spectral factor can be generated by convolving an uncorrelated 2D random field with the estimate of $\hat{h}(r)$ according (1). It has been proved [16] that the synthesized image closely approximates the first and second order statistics of the original one when e_r is the white noise. The synthetic band limited multi-spectral factors are created by the inverse K-L transformation $\tilde{Y}_{r,\bullet} = T^{-1}\bar{Y}_{r,\bullet}$ from the corresponding monospectral factors. Fine-resolution synthetic multi-spectral smooth texture is then obtain by the G-L pyramid collapse which is inverse procedure to that described in section 2.3.

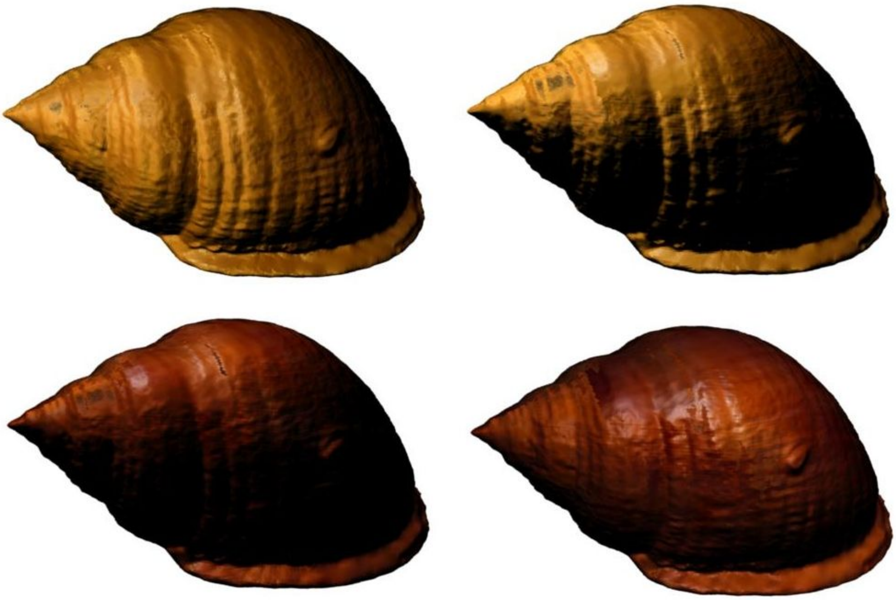

Fig. 2. Two BTF wood synthetic materials mapped on the 3D shell model under two different illumination angles

3 Results

We have tested the model on BTF colour textures from the University of Bonn BTF measurements [3] consist of several materials such as wood (Fig.2) or leather. Each BTF material sample comprised in the University of Bonn database is measured in 81 illumination and 81 viewing angles and has resolution 800×800 pixels. The resulting texture quality is approaching existing alternative BTF models based on 2D random fields: Causal Auto-Regressive model (CAR^{2D}) [17] and Gaussian Markov random field model ($GMRF^{2D}$) [18].

Table 1. Processing time for single models

model	analysis $[s]$	synthesis $[s]$
$GMRF^{2D}$	5.63	21.68
CAR^{2D}	8.49	3.62
MA^{2D}	2.32	3.66

BTF moving average model represents a simple alternative to these BTF models. Multi-spectral (both BTF or non-BTF) models based on spectral factorization (2D random field models) have problems to correctly represent spectrum of motley textures (Fig. 1-left). The MA^{2D} models spectrally outperforms (Fig. 1) both these alternative models due to its weak spatial correlations. The main advantage of the moving average model is its stability, which is a problem which has to be occasionally treated for CAR type models. The GMRF models require approximate parameters estimation and demanding texture synthesis. Another advantage of the model is its numerical efficiency, Tab.1 compares analysis and synthesis times for a 128×128 texture with 4 pyramid levels on the 2GHz Pentium 4 processor.

4 Conclusion

The presented BTF moving average model offers the possibility to describe and enlarge BTF textures and represents the simple alternative to existing 2D BTF models. The preliminary test results of the model on available BTF data are promising although they are only approximation of the original measurements. Even not so successful results can be used for the preattentive BTF textures applications. The presented BTF moving average model enables fast seamless enlargement of BTF texture to arbitrary size and very high BTF texture compression ratio which cannot be achieved by any alternative sampling based BTF texture enlargement method. This is advantageous for transmission, storing or modelling visual surface texture data. Model has low computation complexity, does not need any time consuming numerical optimisation like the usually employed Markov chain Monte Carlo method or some of their deterministic approximation, or Fourier transformation. On the other hand, the necessary spectral and spatial factorizations increase overall time and computing demands. This model may be also used to reconstruct BTF space (i.e., missing parts of the BTF measurement space) or even non existing (i.e., previously not measured) BTF textures. Due to its simplicity, the model is also potentially capable of direct implementation inside the graphical card processing unit or a multithreaded implementation.

Acknowledgements. This research was supported by the grant GAČR 102/08/0593 and partially by the project GAČR 103/11/0335.

References

1. Dana, K.J., Nayar, S.K., van Ginneken, B., Koenderink, J.J.: Reflectance and texture of real-world surfaces. In: CVPR, pp. 151–157. IEEE Computer Society (1997)
2. Haindl, M., Filip, J.: Visual Texture. Advances in Computer Vision and Pattern Recognition. Springer, London (January 2013)

3. Müller, G., Meseth, J., Sattler, M., Sarlette, R., Klein, R.: Acquisition, synthesis and rendering of bidirectional texture functions. In: Eurographics 2004, STAR - State of The Art Report, Eurographics Association, Eurographics Association, pp. 69–94 (2004)
4. Kawasaki, H., Seo, K.D., Ohsawa, Y., Furukawa, R.: Patch-based btf synthesis for real-time rendering. In: IEEE International Conference on Image Processing, ICIP, September 11-14, vol. 1, pp. 393–396. IEEE (2005)
5. Lefebvre, S., Hoppe, H.: Appearance-space texture synthesis. ACM Trans. Graph 25(3), 541–548 (2006); BTF sampling
6. Leung, C.S., Pang, W.M., Fu, C.W., Wong, T.T., Heng, P.A.: Tileable btf. IEEE Transactions on Visualization and Computer Graphics 13(5), 953–965 (2007)
7. Haindl, M., Hatka, M.: BTF Roller. In: Chantler, M., Drbohlav, O. (eds.) Proceedings of the 4th International Workshop on Texture Analysis, Texture 2005, pp. 89–94. IEEE, Los Alamitos (2005)
8. Blinn, J.: Simulation of wrinkled surfaces. SIGGRAPH 1978 12(3), 286–292 (1978)
9. Wang, L., Wang, X., Tong, X., Lin, S., Hu, S., Guo, B., Shum, H.: View-dependent displacement mapping. ACM Transactions on Graphics 22(3), 334–339 (2003)
10. Haindl, M., Filip, J.: Extreme compression and modeling of bidirectional texture function. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(10), 1859–1865 (2007)
11. Frankot, R.T., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence 10(7), 439–451 (1988)
12. Favaro, P., Soatto, S.: 3-D shape estimation and image restoration: exploiting defocus and motion blur. Springer-Verlag New York Inc. (2007)
13. Woodham, R.: Photometric method for determining surface orientation from multiple images. Optical Engineering 19(1), 139–144 (1980)
14. Li, X., Cadzow, J., Wilkes, D., Peters, R., Bodruzzaman II, M.: An efficient two dimensional moving average model for texture analysis and synthesis. In: Proceedings IEEE Southeastcon 1992, vol. 1, pp. 392–395. IEEE (1992)
15. Cole Jr., H.A.: On-line failure detection and damping measurement of aerospace structures by random decrement signatures. Technical Report TMX-62.041, NASA (May 1973)
16. Li, X.: An efficient two-dimensional FIR model for texture synthesis. PhD thesis, Vanderbilt University (1990)
17. Haindl, M., Filip, J.: A fast probabilistic bidirectional texture function model. In: Campilho, A.C., Kamel, M.S. (eds.) ICIAR 2004. LNCS, vol. 3212, pp. 298–305. Springer, Heidelberg (2004)
18. Haindl, M., Filip, J.: Fast BTF texture modelling. In: Chantler, M. (ed.) Texture 2003. Proceedings, pp. 47–52. IEEE Press, Edinburgh (2003)