MSAR BTF Model*

Michal Havlíček †

2nd year of PGS, email: havlimi2@utia.cas.cz
Department of Mathematics
Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague

advisor: Michal Haindl, Institute of Information Theory and Automation, AS CR

Abstract. The Bidirectional Texture Function (BTF) is the recent most advanced representation of material surface visual properties. BTF specifies the changes of visual appearance due to varying illumination and viewing conditions. Such a function might be represented by thousands of images of surface taken in given illumination and viewing conditions per sample of the material. Resulting BTF size, hundreds of gigabytes, excludes its direct rendering in graphical applications, accordingly some compression of these data is obviously necessary. This paper presents a novel probabilistic model based algorithm for realistic multispectral BTF texture modelling. This complex but efficient method combines several multispectral band limited spatial factors and corresponding range map to produce the required BTF texture. Proposed scheme enables very high BTF texture compression ratio and in addition may be used to reconstruct BTF space i.e. non-measured parts of the BTF space.

Keywords: BTF, texture analysis, texture synthesis, data compression, virtual reality

Abstrakt. Obousměrná texturovací funkce je nejpokročilejší v současné době používaná reprezentace vizuálních vlastností povrchu materiálu. Popisuje změny vzhledu v důsledku měnících se úhlů osvětlení a pohledu. Tato funkce může být představována tisíci obrazy povrchu vzorku materiálu, které jsou pořízeny za různých světelných podmínek a pod různým úhlem. Výsledná velikost BTF, stovky gigabajtů, znemožňuje přímé využití v grafických aplikacích a je tedy třeba tyto data nějakým způsobem komprimovat. Tento článek představuje nový pravděpodobnostní model umožňující realistické modelování multispektrálních BTF textur. Tato složitá ale účinná metoda kombinuje několik multispektrálních frekvenčních faktorů a odpovídající hloubkovou mapu výsledkem čehož je požadovaná BTF textura. Navržený postup umožňuje velmi vysokou úroveň komprese BTF textur a navíc může být využit k rekonstrukci BTF prostoru, tj. těch částí BTF, které nebyly původně naměřeny.

Klíčová slova: BTF, analýza textur, syntéza textur, komprese dat, virtuální realita

1 Introduction

Photo realism in virtual reality scenes cannot be realized without nature like colour textures covering visualised scene objects. These textures can be either smooth or rough. The rough ones have rugged surface and do not obey Lambertian law, their reflectance

*This research was supported by the grant GAČR 102/08/0593 and partially by the projects MŠMT 1M0572 DAR, GAČR 103/11/0335, CESNET 387/2010.

†Institute of Information Theory and Automation, AS CR

is illumination and view angle dependent. Such textures might be represented by Bidirectional Texture Function (BTF) [3] which is 7-dimensional function describing texture appearance variations due to varying illumination and viewing angles. Generally, textures can be either digitised natural or artifical materials or images 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 (apposite example can be found in [15]). This solution become unmanageable for rough textures which require to store thousands of different illumination and view angle samples for every texture so that even simple scene featuring only several different textures requires to store tera bytes of texture data which is still far out of limits for any current hardware. Several so called intelligent sampling methods (for example [4], [5] and many others) were proposed to reduce these extensive memory requirements. All these methods are based on some sort of original small texture sampling and the best of them produce very realistic textures. However they require to store thousands images for every combination of viewing and illumination angle of the original target texture sample and in addition often produce visible seams (except for method presented in [10]). Some of them are computationally demanding and in addition they are not able to generate textures unseen by these algorithms.

While synthetic textures are more flexible and extremely compressed, because only several parameters have to be stored in contrast with gigabytes of original data [15]. They may be evaluated directly in 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 results in visual quality compromise of some methods. Several multispectral modelling approaches were published for example [11], [1], [12], [13]. Modelling multispectral images requires in theory three dimensional models but it is possible to approximate such model with a set of simpler two dimensional ones. Evidently this leads to certain loss of information (for example 3D Causal Autoregressive (CAR) model [7] versus 2D CAR model [8]).

Among such possible models the random fields are appropriate for texture modelling not only because they do not suffer with some problems of alternative options (see [6], [12] for details) but they provide relatively easy to implement and computational undemanding texture synthesis and sufficient flexibility to reproduce a large set of both natural and artifical textures. While the random field based models quite successfully represent high frequencies appeared in natural textures low frequencies are sometimes difficult for them. This slight drawback may be overcome by usage of a multiscale random field model. In this case 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 resolution component is both analysed and synthesised independently. Multiple resolution decomposition may be performed by means of Gaussian Laplacian pyramids, wavelet pyramids or subband pyramids. Because of its relative simplicity we decided to utilize Gaussian Laplacian pyramid decomposition for this task.

2 Smooth Texture Model

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 acquired by direct measurement of the observed surface using corresponding range cameras, however this method requires special hardware and measurement methodology [9]. Hence alternative approaches for range map estimation from surface images are more suitable. One of the most accurate approaches is the Photometric Stereo [9] which estimates surface range map from at least three images obtained for different position of illumination source and fixed camera position. This approach was used for range map estimation from textures for experiments described below. Naturally it is enough to estimate range map once per material and reuse it whenever it is needed.

We propose a novel algorithm for efficient rough texture modelling which combines an estimated range map with synthetic multiscale smooth textures generated using Multispectral Simultaneous Autoregressive Model (MSAR) [1]. The material visual appearance during changes of viewing and illumination conditions can be simulated using the bump mapping [2] or displacement mapping technique [16]. The obvious advantage of this solution is the possibility to use hardware support of bump mapping and displacement mapping in up to date visualisation hardware. The overall appearance is guided by the corresponding underlying probabilistic model.

2.1 Spatial Factorization

An analysed texture is decomposed into multiple resolution factors using Laplacian pyramid and the intermediary Gaussian pyramid $Y_\bullet''^{(k)}$ which is a sequence of images and each its element is a low pass down sampled version of its predecessor. The Gaussian pyramid for a reduction factor n is [8]:

$$Y_r''^{(k)} = \downarrow_r^n (Y_{\bullet,i}''^{(k-1)} \otimes w), \quad k = 1, 2, \ldots \quad ,$$

where $\downarrow_r^n$ denotes down sampling with reduction factor n and $\otimes$ is the convolution operation. The Laplacian pyramid $Y_r'^{(k)}$ contains band pass components and provides a good approximation to the Laplacian of the Gaussian kernel. It can be constructed by simple differencing single Gaussian pyramid layers:

$$Y_r'^{(k)} = Y_r''^{(k)} - \uparrow_r^n (Y_\bullet''^{(k+1)}), \quad k = 0, 1, \ldots \quad .$$

As previously noticed each resolution data are independently modelled by their dedicated MSAR model so that model parameters are estimated for each component independently in analysis step.

2.2 Multispectral Simultaneous Autoregressive Model

In the multispectral case random field models are defined as intensity levels on multiple two dimensional lattice planes (e.g. in case of widely used standard RGB colour model three such planes are considered). The value at each lattice location is taken to be a linear combination of neighbouring ones and an additive noise component. For mathematical simplicity, all lattices are double toroidal (the same way as Gaussian Markov Random Field model [9] for example). Let a location within an $M \times M$ two dimensional lattice be denoted by $s = (s_1, s_2)$, with $s_1, s_2 \in J$ and the set J is defined as $J = \{0, 1, \ldots, M-1\}$. The set of all lattice locations is then defined as $\Omega = \{s = (s_1, s_2) : \ s_1, s_2 \in J\}$. The value of an image observation at location s is denoted by $y(s)$. These random vectors are expected to have zero mean. Neighbour sets relating the dependence of image plane i on image plane j are defined as $N_{ij} = \{r = (k, l) : \ k, l \in \pm J\}$ with the associated neighbour coefficients $q_{ij} = \{q_{ij}(r) : r \in N_{ij}\}$. The set $\pm J = \{-(M-1), \ldots, -1, 0, 1, \ldots, M-1\}$. We also use shortened notation: $q = \{q_{ij}; \quad i, j \in \{1, \ldots, P\}\}$ and $r = \{r_i; \quad i \in \{1, \ldots, P\}\}$). Neighbour sets may be classified as symmetric or nonsymmetric. In particular, in the case of multispectral models, a symmetric neighbour set is defined as one for which $r \in N_{ij} \iff -r \in N_{ji}$. Our model is defined on symmetric neighbour set. Scale parameter ρ associated with the corresponding noise component of the model is defined for each particular lattice.

The Multispectral Simultaneous Autoregressive model (MSAR) [1] relates each lattice position $y_i(s)$ to its neighbouring pixels, both within and between image planes, according to the following equations:

$$y_i(s) = \sum_{j=1}^{P} \sum_{r \in N_{ij}} \theta_{ij}(r) y_j(s \oplus r) + \sqrt{\rho_i} w_i(s), \quad i = 1, \ldots, P \ , \tag{1}$$

where P equals number of image planes, ρ_i is noise variance of image plane i, $w_i(s)$ are i.i.d. random variables with zero mean and unit variance and $\oplus$ denotes modulo M addition in each index. Virtually the MSAR model characterizes the spatial interactions between neighbouring pixels through the parameter vectors $\theta = (\theta_{ij}; \quad i = 1, \ldots, P; j = 1, \ldots, P)^T$ and $\rho = (\rho_i; \quad i = 1, \ldots, P)^T$. Rewriting (1) in matrix form for the RGB colour model, i.e. $i \in \{r, g, b\}$, the MSAR model equations are then $B(\theta) y = w$ where

$$B(\theta) = \begin{pmatrix} B(\theta_{rr}) & B(\theta_{rg}) & B(\theta_{rb}) \\ B(\theta_{gr}) & B(\theta_{gg}) & B(\theta_{gb}) \\ B(\theta_{br}) & B(\theta_{bg}) & B(\theta_{bb}) \end{pmatrix},$$

$$y = (y_r(s), \ y_g(s), \ y_b(s))^T, \quad w = (\sqrt{\rho_r} w_r(s), \ \sqrt{\rho_g} w_g(s), \ \sqrt{\rho_b} w_b(s))^T$$

and both $y_i(s)$ and $w_i(s)$ are M^2-vectors of lexicographic ordered arrays $\{y_i(s)\}$ and $\{w_i(s)\}$, respectively. The transformation matrix $B(\theta)$ is composed of $M^2 \times M^2$ block circulant submatrices

$$B(\theta_{ij}) = \begin{pmatrix} B(\theta_{ij})_1 & B(\theta_{ij})_2 & \ldots & B(\theta_{ij})_M \\ B(\theta_{ij})_M & B(\theta_{ij})_1 & \ldots & B(\theta_{ij})_{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ B(\theta_{ij})_2 & B(\theta_{ij})_3 & \ldots & B(\theta_{ij})_1 \end{pmatrix}$$

where each element $B(\theta_{ij})_p \, p \in \{1, \ldots, M\}$ is an $M \times M$ circulant matrix whose (m,n)-th element is given by:

$$b(\theta_{ij})_p(m,n) = \begin{cases} 1, & i=j,\, m=n, \\ -\theta_{ij}(k,l), & k=p-1,\, l=((n-m) \mod M),\, (k,l) \in N_{ij}, \\ 0, & otherwise. \end{cases}$$

Writing the image observations as $y = B(q)^{-1}w$, the image covariance matrix is obtained as $\Sigma_y = \varepsilon\{yy^T\} = \varepsilon\ \{\ B(q)^{-1}\ ww^T\ [B(q)^{-1}]^T\ \} = B(q)^{-1}\Sigma_w[B(q)^{-1}]^T$ where

$$\Sigma_w = \varepsilon\{ww^T\} = \begin{pmatrix} \rho_r I & 0 & 0 \\ 0 & \rho_g I & 0 \\ 0 & 0 & \rho_b I \end{pmatrix}.$$

2.3 Parameter Estimation

It is neccesary to notice that the selection of an appropriate MSAR model support is important to obtain good results in modelling of a given random field. If the contextual neighbourhood is too small it can not capture all details of the random field. Contrariwise, inclusion of the unnecessary neighbours add to the computational burden and can potentially degrade the performance of the model as an additional source of noise.

A least squares (LS) estimate of the MSAR model parameters can be obtained by equating the observed pixel values of an image to the expected value of the model equations. As we prefer RGB colour model our task leads to three independent systems of M^2 equations:

$$y_i(s) = q_i(s)^T\theta_i, \quad s \in \Omega, \quad i \in \{r,g,b\},$$

with vectors θ_i and $q_i(s)$ formed as follows $\theta_i = (\theta_{ir}, \theta_{ig}, \theta_{ib})^T$ and
$q_i(s) = (\ \{y_r(s \oplus t) : t \in N_{ir}\}, \quad \{y_g(s \oplus t) : t \in N_{ig}\}, \quad \{y_b(s \oplus t) : t \in N_{ib}\}\ )^T$.
The LS solution $\hat{\theta}_i$ and $\hat{\rho}_i$ can be found then as

$$\hat{\theta}_i = \left(\sum_{s \in \Omega} q_i(s) q_i(s)^T \right)^{-1} \left(\sum_{s \in \Omega} q_i(s) y_i(s) \right),$$

$$\hat{\rho}_i = \frac{1}{M^2} \sum_{s \in \Omega} (y_i(s) - \hat{\theta}_i^{\,T} q_i(s))^2 \quad .$$

2.4 Texture Synthesis

The goal of texture synthesis in case of probabilistic model is to generate image of arbitrary size directly from the model parameters so that resulting texture has the same statistical properties as original measured and analysed one. Several possibilities exist for a finite lattice MSAR synthesis. The most effective method uses the discrete fast Fourier transformation (DFT). The MSAR model equations (1) may be expressed in terms of the DFT of each image plane as

$$Y_i(t) = \sum_{j=1}^{P} \sum_{r \in N_{ij}} \theta_{ij}(r) Y_j(t) e^{\sqrt{-1}\omega_{rt}} + \sqrt{\rho_i} W_i(t), \quad i = 1, \ldots, P \tag{2}$$

where $Y_i(t)$ and $W_i(t)$ are the two-dimensional DFT coefficients of the image observation $\{y_i(s)\}$ and noise sequence $\{w_i(s)\}$, respectively, at discrete frequency index $t = (m, n)$ and $\omega_{rt} = \frac{2\pi(mk+nl)}{M}$ for $r = (k, l)$. For the RGB colour model equations (2) can be written in matrix form as

$$Y(t) \;=\; \Lambda(t)^{-1} \, \Sigma^{\frac{1}{2}} \, W(t), \quad t \in \Omega$$

where the vectors $Y(t)$ and $W(t)$ are formed this way:

$$Y(t) = (Y_r(t),\ Y_g(t),\ Y_b(t))^T, \quad W(t) = (W_r(t),\ W_g(t),\ W_b(t))^T,$$

and the matrices Σ and $\Lambda(t)$ are defined as:

$$\Sigma = \begin{pmatrix} \rho_r & 0 & 0 \\ 0 & \rho_g & 0 \\ 0 & 0 & \rho_b \end{pmatrix},$$

$$\Lambda(t) = \begin{pmatrix} \lambda_{rr}(t) & \lambda_{rg}(t) & \lambda_{rb}(t) \\ \lambda_{gr}(t) & \lambda_{gg}(t) & \lambda_{gb}(t) \\ \lambda_{br}(t) & \lambda_{bg}(t) & \lambda_{bb}(t) \end{pmatrix},$$

$$\lambda_{ij}(t) = \begin{cases} 1 - \sum_{r \in N_{ij}} \theta_{ij}(r)\, e^{\sqrt{-1}\omega_{rt}} & i = j\,, \\ -\sum_{r \in N_{ij}} \theta_{ij}(r)\, e^{\sqrt{-1}\omega_{rt}} & i \neq j\,. \end{cases}$$

Apparently, the MSAR model will be stable and valid if $\Lambda(t)$ is nonsingular matrix $\forall t \in \Omega$. Given the model parameters, a $M \times M$ MSAR image can be synthesized according to the following algorithm:
1) Generate the i.i.d. noise arrays $\{w_i(s)\}$ for each image plane using a pseudo random number generator.
2) Calculate the two-dimensional DFT of each noise array i.e. produce the transformed noise arrays $\{W_i(t)\}$).
3) For each discrete frequency index t compute $Y(t) = \Lambda(t)^{-1}\, \Sigma^{\frac{1}{2}}\, W(t)$.
4) Perform the two-dimensional inverse DFT of each frequency plane $\{Y_i(t)\}$, producing the synthesized image planes $\{y_i(s)\}$.
The resulting image planes will have zero mean thus it is necessary to add desired mean to each spectral plane after step 4. Fine resolution texture is obtained from the pyramid collapse procedure that is inversion process to process described in section 2.1.

3 Results

We have tested the algorithm on colour BTF textures from the University of Bonn BTF measurements [15], namely on following materials: artifical leather, foil, glazed tails, plastic floor and two different samples of wood. Each BTF material sample comprised in mentioned database is measured in 81 illumination and 81 viewing angles and has resolution 800 × 800 pixels, so that 6561 images had to be analysed for each material.

The open source project Blender[1] with special plugin for BTF support [14] was used to render the results i.e. the scene in virtual reality featuring three-dimensional object covered with synthesised BTF texture. Figure 1 demonstrates the result for one picked material, foil in this case, i.e. synthesised BTF texture combined with its range map in a displacement mapping filter of the rendering software mapped on bumpy board. Scene was rendered in several different illumination conditions with fixed view angle to demonstrate visual quality of synthesised BTF.

3.1 Implementation Details

The source code was written in C++ and compiled in several different environments (namely with g++ versions 3.4.4, 4.1.2, 4.3.2, 4.3.4 and 4.5.0) and tested on many different systems including standard windows based operating system with cygwin environment as well as linux based systems to prove stability and portability of the program. This implementation uses many parts of library developed at Pattern Recognition Department, Institute of Information Theory and Automation[2], such as image reading and writting routines, memory managment and XML format support.

4 Summary and Conclusion

Our testing results of the 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 main benefit of this method is more realistic representation of texture colourfulness which is naturally apparent in case of very distinctively coloured textures. The multi scale approach is more robust and allows sometimes better results than the singlescale one due to capabilities of the model described above.

The proposed method allows huge compression ratio unattainable by alternative intelligent sampling approaches for transmission or storing texture data while it has still moderate computation complexity. It is neccesary to mention that the complexity of analysis is not as important as the complexity of synthesis because the parameter estimation can be performed offline unlike the synthesis which should be as fast as possible. The method does not need any time consuming numerical optimisation like for example the usually employed Monte Carlo methods. The replacement of the bump mapping technique with the displacement mapping further significantly improve the visual quality of the results. The presented method is based on the mathematical model in contrast to intelligent sampling type of methods, and as such it can only approximate realism of the original measurement. However it offers easy simulation of even non existing i.e. previously not measured BTF textures and fast seamless synthesis of texture of arbitrary size.

[1] http://www.blender.org
[2] http://www.utia.cas.cz

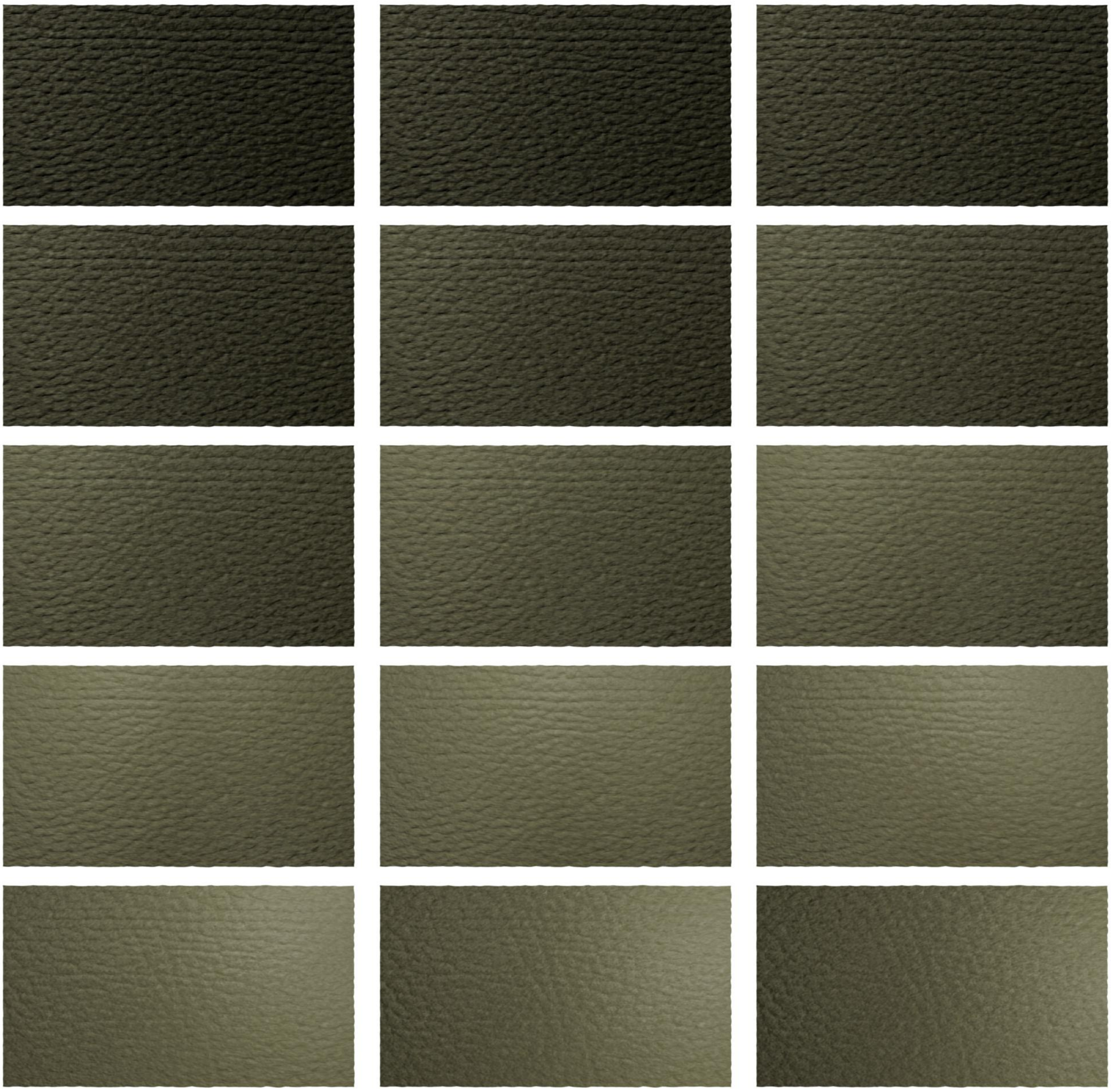

Figure 1: Resulting BTF texture of foil, synthesised texture combined with its range map mapped on bumpy board rendered with 15 different angles of illumination and fixed view angle.

5 Future Work

This developed model might be further tested on different BTF measurements and compared with other random field based models such as already mentioned CAR or Gauss-Markov random field model [9]. Though the quality of the model was proven it would be interesting to find its limitation and study the influence of the size of the neighbourhood to overall performance for example. Naturally more interesting is possible extension of current implementation by means of parallel programming with use of OpenMP[3] library wich is straightforward and would notably increase the model performance. It is also possible rewrite the source code so that program would perform all computations on GPU.

References

[1] J. Bennett, A. Khotanzad. *Multispectral Random Field Models for Synthesis and Analysis of Color Images*. IEEE Transactions on Pattern Analysis and Machine Inteligence **20**(3) (1998), 327–332.

[2] J. Blinn. *Simulation of Wrinkled Surfaces*. ACM SIGGRAPH Computer Graphics **12**(3) (1978), 286–292.

[3] K. Dana, S. Nayar, B van Ginneken, J. Koenderink. *Reflectance and Texture of Real-World Surfaces*. Proceedings of IEEE Conference Computer Vision and Pattern Recognition (1997), 151–157.

[4] J. De Bonet *Multiresolution sampling procedure for analysis and synthesis of textured images*. Proceedings of SIGGRAPH 97, ACM (1997), 361–368.

[5] W. Efros, A.A. Freeman. *Image quilting for texture synthesis and transfer*. SIGGRAPH 2001, Computer Graphics Proceedings, E. Fiume, Ed. ACM Press / ACM SIGGRAPH (2001), 341–346.

[6] M. Haindl. *Texture synthesis*. CWI Quarterly **4**(4) (1991), 305–331.

[7] M. Haindl, J. Filip, M. Arnold. *BTF Image Space Utmost Compression and Modelling Method*. Proceedings of 17th ICPR **3**, IEEE Computer Society Press (2004), 194–198.

[8] M. Haindl, J. Filip. *A Fast Probabilistic Bidirectional Texture Function Model*. Proceedings of ICIAR (lecture notes in computer science 3212) **2**, Springer-Verlag, Berlin Heidenberg (2004), 298–305.

[9] M. Haindl, J. Filip. *Fast BTF Texture Modeling*. Proceedings of the 3rd International Workshop on Texture Analysis and Synthesis (2003), 47–52.

[10] M. Haindl, M. Hatka. *BTF Roller*. Texture 2005: Proceedings of the 4th International Workshop on Texture Analysis and Synthesis (2005), 89–94.

[3] http://openmp.org

[11] M. Haindl, V. Havlíček. *Multiresolution colour texture synthesis.* Proceedings of the 7th International Workshop on Robotics in Alpe-Adria-Danube Region, K. Dobrovodský, Ed. Bratislava: ASCO Art (1998), 297–302, Berlin: Springer-Verlag (2000), 114–122.

[12] M. Haindl, V. Havlíček. *A multiresolution causal colour texture model.* Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition, Springer-Verlag (2000), 114–122.

[13] M. Haindl, V. Havlíček. *A multiscale colour texture model.* Proceedings of the 16th International Conference on Pattern Recognition (2002), 255–258.

[14] M. Hatka *Vizualizace BTF textur v Blenderu.* Doktorandské dny 2009, sborník workshopu doktorandů FJFI oboru Matematické inženýrství, České vysoké učení technické v Praze (2009), 37–46.

[15] G. Müller, J. Meseth, M. Sattler, R. Sarlette, R. Klein. *Acquisition, Compression, and Synthesis of Bidirectional Texture Functions.* State of the art report, Eurographics (2004), 69–94.

[16] X. Wang, X. Tong, S. Lin, S. Hu, B. Guo, H.-Y. Shum. *View-dependent displacement mapping.* ACM SIGGRAPH 2002 **22**(3), ACM Press (2003), 334–339.