

PROBABILISTIC MIXTURE-BASED IMAGE MODELLING

MICHAL HAINDL, VOJTĚCH HAVLÍČEK AND JIŘÍ GRIM

During the last decade we have introduced probabilistic mixture models into image modelling area, which present highly atypical and extremely demanding applications for these models. This difficulty arises from the necessity to model tens thousands correlated data simultaneously and to reliably learn such unusually complex mixture models. Presented paper surveys these novel generative colour image models based on multivariate discrete, Gaussian or Bernoulli mixtures, respectively and demonstrates their major advantages and drawbacks on texture modelling applications. Our mixture models are restricted to represent two-dimensional visual information. Thus a measured 3D multi-spectral texture is spectrally factorized and corresponding multivariate mixture models are further learned from single orthogonal mono-spectral components and used to synthesise and enlarge these mono-spectral factor components. Texture synthesis is based on easy computation of arbitrary conditional distributions from the model. Finally single synthesised mono-spectral texture planes are transformed into the required synthetic multi-spectral texture. Such models can easily serve not only for texture enlargement but also for segmentation, restoration, and retrieval or to model single factors in unusually complex seven dimensional Bidirectional Texture Function (BTF) space models. The strengths and weaknesses of the presented discrete, Gaussian or Bernoulli mixture based approaches are demonstrated on several colour texture examples.

Keywords: discrete distribution mixtures, Bernoulli mixture, Gaussian mixture, EM algorithm, multi-spectral texture modelling, BTF texture modelling

Classification: 93E12, 62A10

1. INTRODUCTION

Generative visual texture models are useful not only for modelling physically correct virtual objects material surfaces in virtual or augmented reality environments or restoring images but also for contextual recognition applications such as segmentation, classification or image retrieval.

Texture synthesis approaches may be divided primarily into sampling and model-based methods. Sampling methods [3, 5, 6, 20, 25, 26] rely on sophisticated sampling from real texture measurements while the model-based techniques [1, 11, 14, 15, 23, 27] describe texture data using multidimensional mathematical models and their synthesis is based on the estimated model parameters only. There are several texture modelling approaches published and some survey articles are also available [7, 11].

Most published texture models are restricted only to mono-spectral textures for few models developed for multispectral (mostly colour) textures refer [1],[12]–[18]. We introduced in our previous papers [14, 15] fast multiresolution Markov random field (MRF) based models, which are very efficient for colour or even for substantially more complex BTF [16] texture modelling, because they do not suffer with some problems of alternative options (see [11] for details) and simultaneously they are easy to analyze as well as to synthesise. However they cannot model well regular or near-regular textures [17]. Nevertheless, such textures can be reliably represented using discrete mixture [9], colour Gaussian mixture [13], hybrid colour Gaussian mixture [12], subspace colour Gaussian mixture [10], or Bernoulli mixtures [18, 19] models, respectively.

General static multispectral (e.g. colour) texture requires a three-dimensional underlying mathematical models (for some fixed illumination and viewing angles) for their comprehensive description. Although full 3D models allow unrestricted spatial-spectral correlation modelling their main drawback is large amount of parameters to be estimated and in the case of some models (e.g. Markov models) also the necessity to estimate all these parameters simultaneously. If we are willing to sacrifice some spectral information, a 3D texture model can be approximated by a set of simpler 2D texture models. The spectral factorization alternative (using PCA decorrelation) accepted in this paper allows the independent spectral component modelling approach and usage of simpler 2D data models with much less parameters. This is extremely important because our mixture models have to be reliably learned from learning example images which are restrictive in view of the high problem dimensionality. Note that, generally a 3D discrete distribution for typical random vector of 1500 random variables each one having 256 possible values is difficult to learn from commonly available 512×512 training image. Unfortunately real data space can be decorrelated only approximately, hence this approach suffers with some loss of spectral image information. A full 3D mixture model will be presented elsewhere.

The present paper targets such textures using a multivariate 2D mixtures texture models with components defined as products of multivariate discrete (DM) or Gaussian distributions (GM) or univariate Bernoulli distributions (BM). The multivariate Bernoulli mixtures are used to model the local statistical texture properties separately for individual bit planes of decorrelated mono-spectral image components. In the application part we demonstrate advantages and weak points of the proposed method on several colour textured images.

2. PROBABILISTIC MIXTURE MODEL

A digitized multispectral static texture image \tilde{Y} is assumed to be defined on a finite rectangular $N_1 \times N_2 \times d$ lattice I^3 , $\tilde{r} = \{r_1, r_2, r_3\}$ and $r = \{r_1, r_2\}$ denotes a pixel multiindex with the row, columns and spectral indices, respectively. The notation \bullet has the meaning of all possible values of the corresponding index. In this notation a mono-spectral component of the original texture image can be viewed as a matrix $Y_{\bullet, \bullet, r_3} \in \mathcal{K}_{r_3}^{N_1 N_2}$.

Our 2D mixture models require spectral factorization in the wavelength spectrum

meaning. The Karhunen–Loeve expansion transforms the original centered data space \bar{Y} defined on the rectangular $N_1 \times N_2$ finite toroidal lattice I into a new data space with K-L coordinate axes \bar{Y} . Although we do not assume any data distribution at this stage, we replace the unknown covariance matrix by its sampling estimate. It allows us to significantly simplify the difficult 3D modelling task into 3 much simpler 2D learning and modelling subtasks. These new basis vectors are the eigenvectors of the $d \times d$ second-order statistical moments matrix $\Phi = E\{\bar{Y}_{r,\bullet}\bar{Y}_{r,\bullet}^T\}$. The projection of random vector $\bar{Y}_{r,\bullet}$ onto the K-L coordinate system uses the transformation matrix $T = [u_1^T, \dots, u_d^T]^T$ which has single rows u_j that are eigenvectors of the matrix Φ .

$$\bar{Y}_{r,\bullet} = T\tilde{Y}_{r,\bullet} \quad (1)$$

Components of the transformed vector $\bar{Y}_{r,\bullet}$ (1) are mutually uncorrelated and if $\bar{Y}_{r,\bullet}$ are Gaussian they are also independent hence each transformed mono-spectral factor can be modeled independently of the remaining spectral factors. Although the Gaussianity assumption generally does not hold, we assume this approximation to enable important simplifying spectral space factorization. Besides, this approximation is subsequently validated by our high quality experimental results.

Supposing now uncorrelated mono-spectral textures, we assume that each pixel of the r_3 th mono-spectral image is described by a grey level taking on K_{r_3} (often $K_{r_3} = 256$) possible values, i. e.,

$$Y_{\cdot,\cdot,r_3} \in \mathcal{K}_{r_3}, \quad \forall \tilde{r} \in I^3, \quad \mathcal{K}_{r_3} = \{1, 2, \dots, K_{r_3}\}, \quad (2)$$

where \mathcal{K}_{r_3} is the set of distinguished grey levels in the r_3 th transformed spectral band. If $K_{r_3} = 2$ we denote this set \mathcal{B} . Because the amount of information content of each transformed spectral band is proportional to the eigenvalue corresponding to the single transformation matrix row, the most descriptive and best model should be used for the $r_3 = 1$ spectral band. A reasonable approximation significantly improving the numerical efficiency of the proposed models is therefore

$$\text{card}\{\mathcal{K}_1\} = K \geq \text{card}\{\mathcal{K}_2\} \geq \text{card}\{\mathcal{K}_3\} \geq \dots \geq \text{card}\{\mathcal{K}_d\},$$

$$\frac{\text{card}\{\mathcal{K}_i\}}{\text{card}\{\mathcal{K}_{i+1}\}} \approx \begin{cases} \frac{\lambda_i}{\lambda_{i+1}} & \text{if } \frac{\lambda_i}{\lambda_{i+1}} < 5 \\ 5 & \text{otherwise} \end{cases}$$

and a re-quantized mono-spectral pixel is then

$$Y_{r_1,r_2,r_3} = \bar{Y}_{r_1,r_2,r_3} \frac{K_{r_3}}{K} \quad (3)$$

where K is the original number of quantization levels (assumed equal for all uncorrelated spectral bands $\bar{Y}_{\cdot,\cdot,r_3}$). To simplify notation we will neglect further on (Sections 3–6) the spectral component in the multiindices r, s because single sub-models describe only decorrelated mono-spectral components of the original multi-spectral texture and the lattice I will have only two dimensions.

Let us suppose that the natural homogeneous texture image represents a realization of a random vector with a probability distribution $P(Y_{\cdot,\bullet})$. The concept

of texture intuitively implies some degree of homogeneity. In other words we may assume that the local statistical properties of a texture as observed, e. g., within a small moving window should be invariant with respect to the window position. In this sense we can describe the statistical properties of interior pixels of the moving window by a joint probability distribution and that the properties of the texture can be fully characterized by statistical dependencies on a sub-field, i. e., by a marginal probability distribution of grey levels on pixels within the scope of a window centered around the location r and specified by the index set I_r .

$$I_r = \{r + s : |r_1 - s_1| \leq \alpha \wedge |r_2 - s_2| \leq \beta\} \subset I \tag{4}$$

where α, β are some chosen constants and $|\cdot|$ is the absolute value. I_r depends on a modeled visual data and can have other than this rectangular shape. If we denote $Y_{\{r\}}$ the corresponding vector containing all Y_s in some fixed order arrangement such that $s \in I_r$, $Y_{\{r\}} = [Y_s \ \forall s \in I_r]$, $Y_{\{r\}} \subset Y$, $\eta = \text{card}\{I_r\}$ and $P(Y_{\{r\}})$ is the corresponding marginal distribution of $P(Y)$ then the marginal probability distribution on the “generating” window I_r is assumed to be invariant with respect to arbitrary shifting within the original image, i. e.,

$$P(Y_{\{r\}}) = P(Y_{\{s\}}) \ , \quad \forall s, r \in I, s \neq r \ .$$

Thus, e. g., for a rectangular window of size $\eta = 20 \times 30$ pixels we have to estimate a 600-dimensional probability distribution $P(Y_{\{r\}})$. The marginal distribution $P(Y_{\{r\}})$ is assumed to contain sufficient information to synthesise the modeled texture. The distribution $P(Y_{\{r\}})$ is assumed to has the mixture probability form:

$$P(Y_{\{r\}}) = \sum_{m \in \mathcal{M}} P(Y_{\{r\}} | m) p(m) \tag{5}$$

$Y_{\{r\}} \in \mathcal{K}^\eta$ $\mathcal{M} = \{1, 2, \dots, M\}$ where $p(m)$ are probability weights and

$$\sum_{m \in \mathcal{M}} p(m) = 1 \ .$$

The component distributions $P(\cdot | m)$ are factorizable, i. e., we can write

$$P(Y_{\{r\}} | m) = \prod_{s \in I_r} p_s(Y_s | m) \quad Y_s \in Y_{\{r\}} \ . \tag{6}$$

$p_s(Y_s | m)$ are univariate (component-specific) probability distributions. It can be seen that, by Eqs. (5), (6) the variables $\{Y_s : \forall s \in I_r\}$ are conditionally independent with respect to the index variable m . From the theoretical point of view, this assumption is not restrictive. It can be easily verified that, in discrete case $Y_{\{r\}} \in \mathcal{K}^\eta$, the class of finite mixtures

$$P(Y_{\{r\}}) = \sum_{m \in \mathcal{M}} p(m) \prod_{s \in I_r} p_s(Y_s | m) \ , \tag{7}$$

is complete in the sense that any discrete probability distribution on \mathcal{K}^η can be expressed in the form (7) for sufficiently large M .

3. EM ALGORITHM

The underlying structural model of conditional independence is estimated from a data set \mathcal{S} obtained by step-wise shifting the contextual window I_r within the original texture image, i. e., for each location r one realization of $Y_{\{r\}}$.

$$\mathcal{S} = \{Y_{\{r\}} \mid \forall r \in I, I_r \subset I\} \quad Y_{\{r\}} \in \mathcal{K}^n, \tag{8}$$

where $\mathcal{K} = \begin{cases} \mathcal{K}_{r_3} & \text{for DM, GM} \\ \mathcal{B} & \text{for BM} \end{cases}$. The unknown parameters of the approximating mixture can be estimated by means of the iterative EM algorithm [4, 8]. In order to estimate the unknown distributions $p_s(\cdot \mid m)$ and the component weights $p(m)$ we maximize the likelihood function corresponding to (8)

$$L = \frac{1}{|\mathcal{S}|} \sum_{Y_{\{r\}} \in \mathcal{S}} \log \left[\sum_{m \in \mathcal{M}} P(Y_{\{r\}} \mid m) p(m) \right] \tag{9}$$

by means of the EM algorithm. The related iteration equations can be expressed as follows:

$$q^{(t)}(m \mid Y_{\{r\}}) = \frac{P^{(t)}(Y_{\{r\}} \mid m) p^{(t)}(m)}{\sum_{j \in \mathcal{M}} P^{(t)}(Y_{\{r\}} \mid j) p^{(t)}(j)}, \tag{10}$$

$$p^{(t+1)}(m) = \frac{1}{|\mathcal{S}|} \sum_{Y_{\{r\}} \in \mathcal{S}} q^{(t)}(m \mid Y_{\{r\}}), \tag{11}$$

$$P^{(t+1)}(\cdot \mid m) = \arg \max_{P(\cdot \mid m)} \left\{ \sum_{Y_{\{r\}} \in \mathcal{S}} q^{(t)}(m \mid Y_{\{r\}}) \log P(Y_{\{r\}} \mid m) \right\}. \tag{12}$$

The mixture parameters are initialized by random numbers. The iteration process is stopped when the criterion increments are sufficiently small. The iteration scheme (10)–(12) has the monotonic property: $L^{(t+1)} \geq L^{(t)}$, $t = 0, 1, 2, \dots$ which implies the convergence of the sequence $\{L^{(t)}\}_0^\infty$ to a stationary point of EM algorithm (local maximum or a saddle point of L). However, the ML estimates may be negatively influenced by the fact that the observations in \mathcal{S} are not independent, because the standard ML estimate assumes the independent observations of modelled data. The assumption which is clearly violated in our task, because the contextual windows $Y_{\{r\}}$ are partly overlapping for neighbouring locations r .

4. PROBABILISTIC DISCRETE MIXTURE MODEL

The parameters of the mixture model (7) are probabilistic component weights $p(m)$ and the univariate discrete distributions of grey levels are simply defined by a vector of probabilities:

$$p_s(\cdot \mid m) = (p_s(1 \mid m), p_s(2 \mid m), \dots, p_s(K_{r_3} \mid m)) . \tag{13}$$

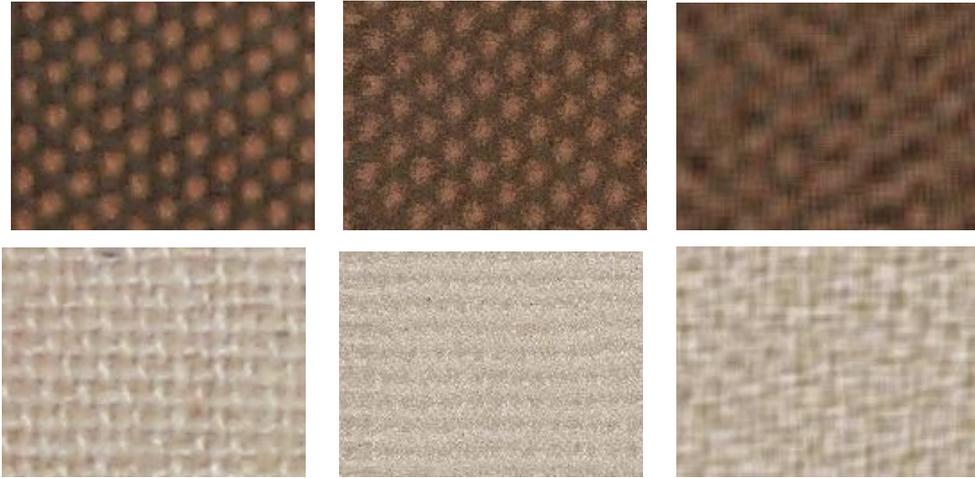


Fig. 1. Natural (left) and synthetic BM (middle) carpet (upper) and jute (bottom) textures compared with their synthetic (right) alternatives generated using Gaussian MRF models.

The M step (12) of the EM algorithms is

$$p_s^{(t+1)}(\xi | m) = \frac{1}{|\mathcal{S}| p^{(t+1)}(m)} \sum_{Y_{\{r\}} \in \mathcal{S}} \delta(\xi, Y_s) q^{(t)}(m | Y_{\{r\}}), \quad \xi \in \mathcal{K}_{r_3}. \quad (14)$$

The total number of mixture (7), (13) parameters is thus $M(1 + NK_{r_3})$ – confined to the appropriate norming conditions. Note that the form of the univariate discrete distributions (13) is fully general without any constraint. In contrast to different parametric models (e.g., normal) the K -dimensional vector $p_s(\cdot | m)$ can describe arbitrary discrete distribution. This fact is one of the main arguments for the choice of the discrete mixture model (7), (13). Another strong motivation for the conditional independence model (7) is a simple switch-over to any marginal distribution by deleting superfluous terms in the products $P(Y_{\{r\}} | m)$. On the other hand the number of parameters included became very large with negative consequences for the reliable model learning.

5. BERNOULLI DISTRIBUTION MIXTURE MODEL

Supposing now uncorrelated mono-spectral textures, we assume that each pixel of the image is described by K_{r_3} possible grey level values, where \mathcal{K}_{r_3} is the set of distinguished grey levels. Single mono-spectral images are further decomposed into separate binary bit planes (8 for $|\mathcal{K}_{r_3}| = 256$) of binary variables $\xi \in \mathcal{B}$, $\mathcal{B} = \{0, 1\}$ which are modeled separately. These binary plane Bernoulli mixture models can be reliably learned from much smaller training texture than the full gray scale discrete

mixture models. Single sub-models describe only single bit planes from decorrelated mono-spectral components of the original multi-spectral texture.

Let us suppose that a bit plane of a mono-spectral textured image component represents a realization of a random vector with a probability distribution $P(Y_{\bullet,\bullet})$ and that the properties of the texture can be fully characterised by a marginal probability distribution of binary levels on pixels within the scope of a window centered around the location r and specified by the index set $I_r \subset I$. The sub-vector $Y_{\{r\}}$ has binary components, i.e. $Y_{\{r\}} \in \mathcal{B}^\eta$ and $P(Y_{\{r\}})$ is the corresponding marginal distribution of $P(Y)$. The distribution $P(Y_{\{r\}})$ is assumed to be multivariable Bernoulli mixture in the form (5), where the component distributions $P(\cdot | m)$ (6) are multivariable Bernoulli

$$p_s(Y_s | m) = \theta_{m,s}^{Y_s} (1 - \theta_{m,s})^{1-Y_s} . \tag{15}$$

The parameters of the mixture model (7) include probabilistic component weights $p(m)$ and the univariate discrete distributions of binary levels. They are simply defined by one parameter $\theta_{m,s}$ as a vector of probabilities:

$$p_s(\cdot | m) = (\theta_{m,s}, 1 - \theta_{m,s}) . \tag{16}$$

The EM solution is again (10), (11) and

$$p_s^{(t+1)}(\xi | m) = \frac{1}{|\mathcal{S}| p^{(t+1)}(m)} \sum_{Y_{\{r\}} \in \mathcal{S}} \delta(\xi, Y_s) q^{(t)}(m | Y_{\{r\}}), \quad \xi \in \mathcal{B} . \tag{17}$$

The total number of mixture (7), (16) parameters is thus $M(1 + \eta)$ – confined to the appropriate norming conditions. Again the advantage of the multivariable Bernoulli model (16) is a simple switch-over to any marginal distribution by deleting superfluous terms in the products $P(Y_{\{r\}} | m)$.

6. GAUSSIAN MIXTURE MODEL

If we assume the joint probability distribution $P(Y_{\{r\}})$, $Y_{\{r\}} \in \mathcal{K}^\eta$ in the form of a normal mixture

$$P(Y_{\{r\}}) = \sum_{m \in \mathcal{M}} p(m) P(Y_{\{r\}} | \mu_m, \sigma_m), \quad Y_{\{r\}} \subset Y , \tag{18}$$

where $p(m)$ are probability weights and the mixture components are defined as products of univariate Gaussian densities

$$P(Y_{\{r\}} | \mu_m, \sigma_m) = \prod_{s \in I_{\{r\}}} p_s(Y_s | \mu_{ms}, \sigma_{ms}) , \tag{19}$$

$$p_s(Y_s | \mu_{ms}, \sigma_{ms}) = \frac{1}{\sqrt{2\pi}\sigma_{ms}} \exp \left\{ -\frac{(Y_s - \mu_{ms})^2}{2\sigma_{ms}^2} \right\} ,$$

i.e., the components are multivariate Gaussian densities with diagonal covariance matrices.

Obviously, assuming the Gaussian densities (19), we ignore the discrete nature of the variables Y_s (typically $Y_s \in \{0, 1, \dots, 255\}$). On the other hand we need only 2 parameters to specify the density $p_s(Y_s | \mu_{ms}, \sigma_{ms})$ in contrast to 255 parameters to be specified in case of a general discrete distribution $p_s(Y_s | m)$ as used in [9].

The maximum-likelihood estimates of the parameters $p(m), \mu_{ms}, \sigma_{ms}$ can be computed by the means of the EM algorithm [9]. Anew we use a data set \mathcal{S} obtained by pixel-wise shifting the observation window within the original texture image

$$\mathcal{S} = \{Y_{\{r\}}^{(1)}, \dots, Y_{\{r\}}^{(K)}\}, \quad Y_{\{r\}}^{(k)} \subset Y. \quad (20)$$

The corresponding log-likelihood function

$$L = \frac{1}{|\mathcal{S}|} \sum_{Y_{\{r\}} \in \mathcal{S}} \log \left[\sum_{m \in \mathcal{M}} p(m) P(Y_{\{r\}} | \mu_m, \sigma_m) \right] \quad (21)$$

is maximized by the EM algorithm ($m \in \mathcal{M}, n \in \mathcal{N}, Y_{\{r\}} \in \mathcal{S}$)

$$q^{(t)}(m | Y_{\{r\}}) = \frac{p^{(t)}(m) P^{(t)}(Y_{\{r\}} | \mu_m, \sigma_m)}{\sum_{j \in \mathcal{M}} p^{(t)}(j) P^{(t)}(Y_{\{r\}} | \mu_j, \sigma_j)}, \quad (22)$$

$$p^{(t+1)}(m) = \frac{1}{|\mathcal{S}|} \sum_{Y_{\{r\}} \in \mathcal{S}} q^{(t)}(m | Y_{\{r\}}), \quad (23)$$

$$\mu_{m,n}^{(t+1)} = \frac{1}{\sum_{Y_{\{r\}} \in \mathcal{S}} q^{(t)}(m | Y_{\{r\}})} \sum_{Y_{\{r\}} \in \mathcal{S}} Y_n q^{(t)}(m | Y_{\{r\}}), \quad (24)$$

$$(\sigma_{m,n}^{(t+1)})^2 = -(\mu_{m,n}^{(t+1)})^2 + \frac{\sum_{Y_{\{r\}} \in \mathcal{S}} Y_n^2 q^{(t)}(m | Y_{\{r\}})}{\sum_{Y_{\{r\}} \in \mathcal{S}} q^{(t)}(m | Y_{\{r\}})}. \quad (25)$$

Let us remark that in practical experiments the resulting parameters μ_m are visually well interpretable. They correspond to the typical (smoothed) variants of the mono-spectral texture pieces occurring in the observation window. In our case the dimension of the estimated distribution is not too high ($\eta \approx 10^1 - 10^2$) and the number of the training data vectors is relatively large ($|\mathcal{S}| \approx 10^4 - 10^5$). Nevertheless the window should always be kept reasonably small and the sample size as large as possible.

7. TEXTURE SYNTHESIS

The statistical description of the local texture properties naturally suggests the possibility of texture synthesis by local prediction. We assume that in a general situation, at a given position of the observation window, some part of the synthesised texture is already specified.

Let I_r be a fixed position of the generating window. If $Y_{\{\rho\}}$ is a sub-vector of all of $Y_{\{r\}}$ pixels previously specified within this window and $I_\rho \subset I_r$ the corresponding index subset, then the statistical properties of the remaining unspecified variables are fully described by the corresponding conditional distribution. In view of the

advantageous properties of our mixture model we can easily compute any univariate conditional distribution $p_{n|\rho}$:

$$p_{n|\rho}(Y_n | Y_{\{\rho\}}) = \sum_{m=1}^M W_m(Y_{\{\rho\}}) p_n(Y_n | m) , \quad (26)$$

where $W_m(Y_{\{\rho\}})$ are the a posteriori component weights corresponding to the given subvector $Y_{\{\rho\}}$:

$$\begin{aligned} W_m(Y_{\{\rho\}}) &= \frac{p(m)P_\rho(Y_{\{\rho\}} | m)}{\sum_{j=1}^M p(j)P_\rho(Y_{\{\rho\}} | j)} , \\ P_\rho(Y_{\{\rho\}} | m) &= \prod_{n \in \rho} p_n(Y_n | m) . \end{aligned} \quad (27)$$

The binary level y_n can be randomly generated by means of the conditional distribution $p_{n|\rho}(Y_n | Y_{\{\rho\}})$ whereby Eqs. (26) can be applied to all the unspecified variables $n = \eta - \text{card}\{\rho\}$ given a fixed position of the generating field. The starting pixel (e. g., left upper corner) is generated from the corresponding unconditional marginal. Simultaneously, each newly generated binary level y_n can be used to upgrade the conditional weights $W_m(Y_{\{\rho\}})$. In the next step, the generating field is shifted to a new position and the conditional distribution (26) has to be computed for a new subset of the specified pixels in ρ . In our experiments we have used a regular left-to-right and top-to-down shifting of the generating window. Specific mixture models (5) synthesise single bit planes of the decorrelated mono-spectral components. The synthesised mono-spectral textures for Bernoulli models are composed from combining corresponding bit planes into three (several for a general multi-spectral texture) synthesised mono-spectral images. Finally the resulting synthesised multi-spectral texture is obtained from the set of synthesised mono-spectral images by inverting the decorrelation process (the inverse K-L transformation):

$$\tilde{Y}_{r,\bullet} = T^{-1} \bar{Y}_{r,\bullet} . \quad (28)$$

8. STATISTICALLY OPTIMIZED SAMPLING

We assume a mixture model to represent texture frequency (i. e. to control sampling process) but not to represent any spectral information. Hence it is enough to use simpler mono-spectral normal mixture model. The texture synthesis procedure as described in Sec. 7 has been applied to different texture images with satisfactory results (cf. [13]). Nevertheless, the synthesised textures sometimes appear to be too “smooth” or “filtered” with the high frequency details missing. In order to obtain the texture with all high frequency details the component means are replaced by the most similar pieces of source colour, multi-spectral or BTF texture in the synthesis phase.

For this reason we have replaced in the synthesis phase the mean vectors μ_m of the mixture components by the most similar target texture “centroids” from the source texture subspace, i. e. by measured pieces of the original real textures.

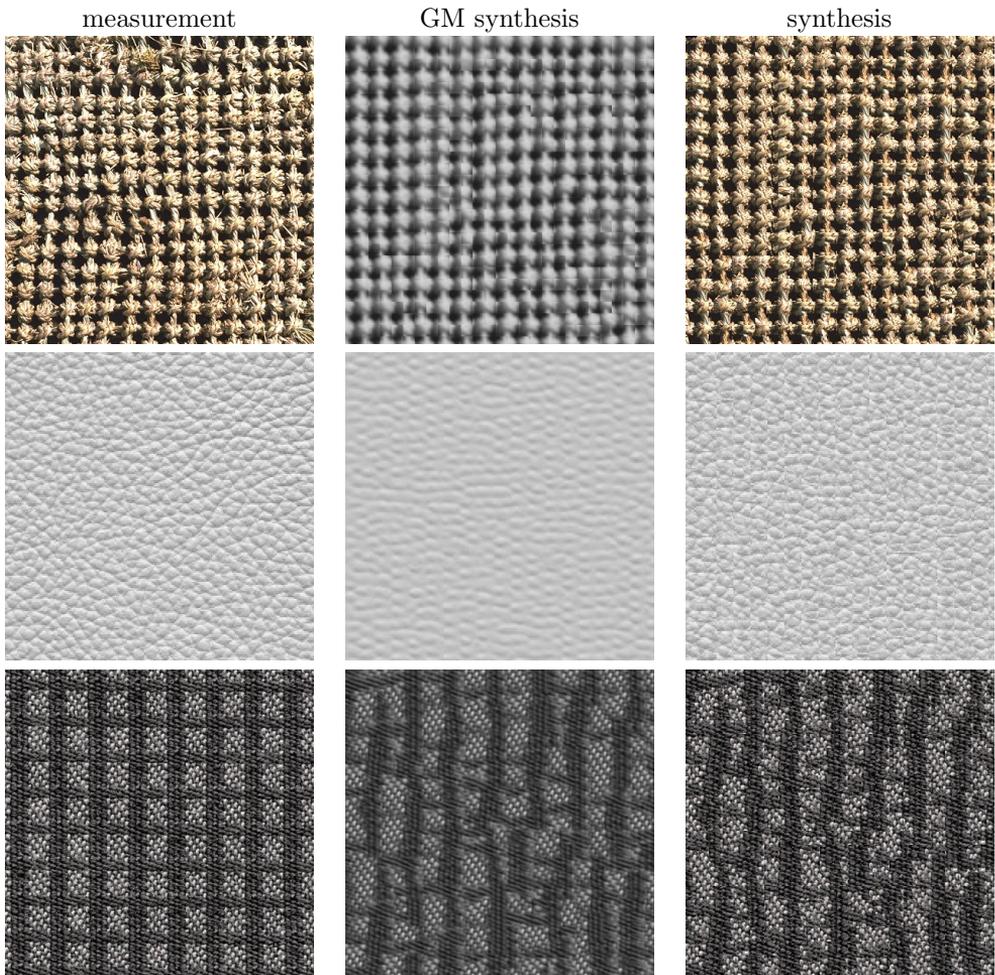


Fig. 2. Rattan, leather and carpet examples and their synthetic results.

For each vector μ_m we have found the corresponding mono-spectral centroid c_m by minimizing the Euclidean distance $\|Y_{\{r\}} - \mu_m\|$ over all possible positions of the observation window $Y_{\{r\}}$ in the mono-spectral version of the source (colour, multispectral, BTF) image:

$$c_m = (c_1, c_2, \dots, c_\eta) = \arg \min_{Y_{\{r\}}} \{\|Y_{\{r\}} - \mu_m\|\} . \quad (29)$$

Given the position of the optimal centroid c_m , we have chosen the corresponding piece of the measured texture subspace for the synthesis phase.

Let us recall that in this way the target texture synthesised by means of the estimated normal mixture is used only as a “draft layout” for the final texture subspace image set which is entirely composed of small pieces of the original texture measurements. (The last step includes also some smoothing of the boundaries between the neighbouring centroids.) The resulting synthesised texture is actually a mosaic of the original texture pieces controlled by the separately synthesised draft layout. The last version of the modelling method has been applied to different textures. In the experiments we have obtained very realistic texture images (cf. Figure 2) for the rattan and leather examples. The main reason for the carpet example failure (Figure 2 bottom) is the insufficiency of the learning texture example. The learning window I_r had to be kept too small to reliably represent all needed low frequencies in the texture.

For this reason the conditional distributions indicate only basic structure, but the majority of conditional pixel marginals are uniform.

It can be seen that in the final version the texture synthesis is much similar to standard sampling methods. By using the estimated component means μ_m we choose from the original texture image a fixed set of texture “centroids” playing the role of tiles. The synthesis based on a local mono-spectral prediction controls the tiling in a statistically optimal way by using statistically optimized set of the texture tiles.

9. BIDIRECTIONAL TEXTURE FUNCTION MODELS

Visual textures typically represent visual properties of surface materials. However physically correct reflectance model (RM) is sixteen-dimensional

$$RM(\lambda_i, x_i, y_i, z_i, t_i, \theta_i, \psi_i, \lambda_v, x_v, y_v, z_v, t_v, \theta_v, \psi_v, \theta_{i,T}, \theta_{v,T}) .$$

RM describes incident light with spectral value λ_i illuminating surface location x_i, y_i, z_i in time t_i under spherical reflectance angle θ_i, ψ_i and observed at time t_v from surface location x_v, y_v, z_v under spherical reflectance angle θ_v, ψ_v and spectrum λ_v . $\theta_{i,T}, \theta_{v,T}$ are the corresponding transmittance angles. The model height parameters z_i, z_v indicate that even radiance along light rays is not constant but depends on the height. Such a RM model is too complex and there neither exist any measurement of such data nor any mathematical representation allowing its synthesis. One of the early compromised attempts to capture real material appearance was done by Nicodemus et al. [22] and later elaborated by Dana et al. [2] in the form of *Bidirectional Texture Function* (BTF). Even if a BTF model assumes several simplifying assumptions [7, 16] its measurement, compression and synthesis is on the leading edge of current mathematical modelling and technological capabilities.

BTF is a seven-dimensional function which considers not only measurement dependency on planar material position and spectral channel but also its dependence on illumination and viewing angles:

$$\text{BTF}_{\theta_i, \phi_i, \theta_v, \phi_v}(\tilde{r}) \tag{30}$$

where θ, ϕ are elevation and azimuthal angles of illumination and view direction

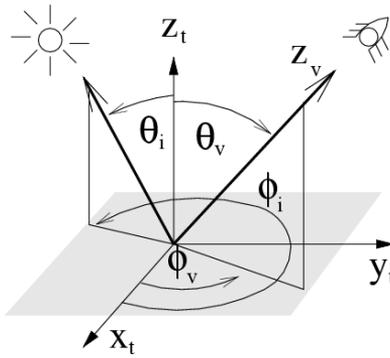


Fig. 3. Relationship between illumination and viewing angles within texture coordinate system.

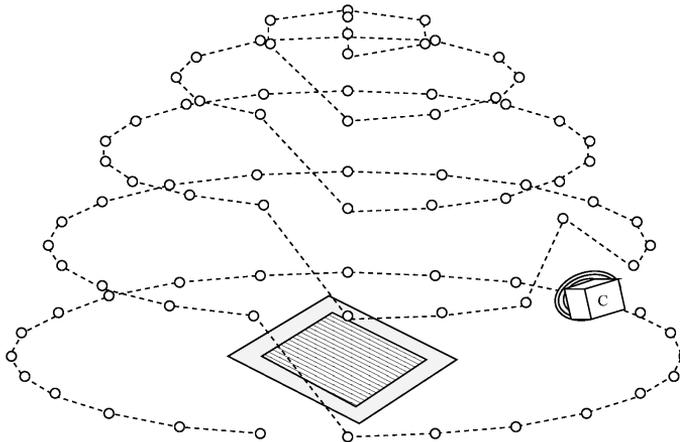


Fig. 4. Light vector trajectory above the sample. The light starts at the top.

vector (see Figure 3), the multiindex $\tilde{r} = [r_1, r_2, r_3]$ specifies planar horizontal and vertical position in a material sample image and r_3 is the spectral index.

The BTF measuring is very time consuming and such systems require high precision measuring setup hence only few such systems exist up to now [2, 21, 24]. These systems are, similarly to BRDF (spatially averaged BTF) measurement systems, based on light source, video / still camera and material sample moving using a robot arm. The main difference between individual BTF measurement systems is in the type of measurement setup (Figure 4) allowing four degree of freedom and the type of measurement sensor (CCD, video, etc.). The main requirements for BTF measurements are the accurate image rectification, i. e., aligning texture normal with view vector, mutual registration of single BTF measurements and sample visual and

illumination source constancy during the several hours long data acquisition. BTF appropriately measured from real material samples offers enough information about material properties, e. g., anisotropy, masking or self-shadowing. In contrast to a regular multispectral static 3D texture or even to BRDF, BTF is high-dimensional and involves large amounts of data. To render BTF on graphics hardware, its compact representation is needed. The best currently available BTF [24] takes up about 1.2GB of storage space per sample. Thus BTF database even for simple VR scenes requires enormous data space (TB). Some compression and modelling method of these huge data sets is inevitable. Such a method should provide compact parametric representation and preserve main visual features of the original BTF, while enabling its fast rendering taking advantage of contemporary graphics hardware.

All our proposed mixture models can be applied also to BTF modelling (Figure 5) if the BTF model separately synthesise single fixed view BTF subspace factors. A 2D mixture model can then synthesise either the corresponding image or parametric factors. Such a BTF model is on top of that very complex. It contains several hundreds of combined elemental 2D mixture models and its details are out of scope of this paper.

In this paper demonstration we use the Bonn University data [24] which have 81 measurements per fixed view BTF subspace and 81 such subspaces. The reason for this is the optimal mutual registration for all different illumination BTF measurements for a fixed view angle. All recently available BTF measurements [2, 21, 24] do not enable the errorless mutual registration of different view texture images. Single BTF sets differ in their image rectification accuracy but even the best one has several pixels error. This rectification error caused clearly visible flaw when we tried to model several view-dependent BTF subspaces using a sole Gaussian mixture model.

Single BTF measurements (Figure 5 odd rows) are at first geometrically transformed using the corresponding inverse perspective projection. The gray scale version of the image with perpendicular illumination is selected as the data source for the statistical vector sampler. This vector sampler is controlled by the primary image Gaussian mixture model.

10. EXPERIMENTAL RESULTS

The implementation of EM algorithm is simple but there are some well known computational problems, e. g., the proper choice of the number of components, the existence of local maxima of the likelihood function and the related problem of a proper choice of the initial parameter values. The above difficulties are less relevant if the sample size is sufficiently large. In our case the dimension of the estimated distribution is not too high ($\eta \approx 10^1 - 10^2$) and the number of the training data vectors relatively large ($|\mathcal{S}| \approx 10^4 - 10^5$). The number of grey levels to be distinguished is $|\mathcal{K}| = 256$ and therefore the estimated distribution becomes considerably complex. For these reasons the generating window should always be kept reasonably small and the sample size as large as possible. All BM models used the contextual window size 21×21 pixels, $M = 40$ components and about 10 iterations of the EM algorithm. The computation was rather time-consuming it took several hours in total on standard PC computer. The time needed for texture synthesis is comparable



Fig. 5. Cushion fabric and foil BTFs examples (odd rows) for three different illumination angles and their synthetic (even rows) results.

with one iteration step of the EM algorithm. The examples Figures 1, 7, 6 illustrate

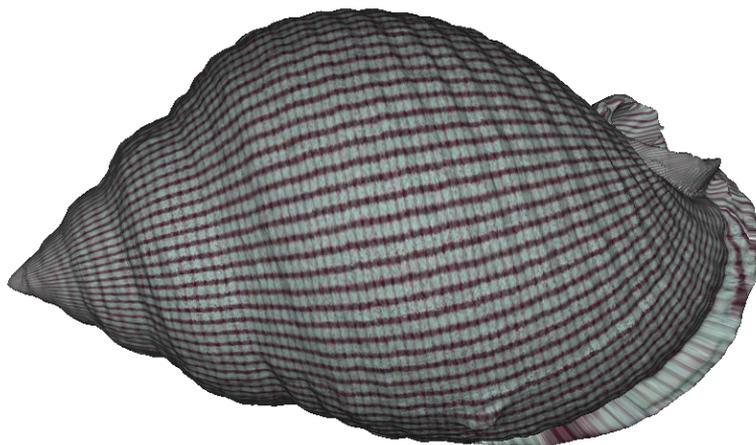


Fig. 6. Synthetic (BM) gingham texture mapped on a snail shell model.

properties of our BM model on natural colour textures. The carpet texture on Figure 1 or gingham texture on Figure 6 represent relatively regular texture which is notoriously difficult for some alternative texture models like for example Gaussian Markov random field models (Figure 1 – top right) but the presented model produced very good synthesis result (Figure 1 – top middle). Similarly the jute example (Figure 1 – bottom) or the buckram texture (Figure 7 – top) demonstrate very good performance of the presented model.

Similarly as all other known texture models also our mixture models have their strong as well as weak sides. While they can realistically synthesise natural or man-made textures with strong periodicity, which are notoriously difficult for most of other alternative approaches, their major weakness is lesser robustness than the Markovian models family. The computationally most efficient Markovian models are much faster than the presented BM or DM models, but general Markovian models which require Markov chain Monte Carlo methods for their analysis as well as synthesis are comparable.

Both DM and BM models have strong tendency either to produce high quality synthetic texture or to completely fail with resulting noise field, but the BM model is slightly more robust. Markovian models in these cases demonstrate clear effort to grasp at least some of the difficult texture features. A GM model is easier to learn and does not produce noisy failures but it is less general than DM.

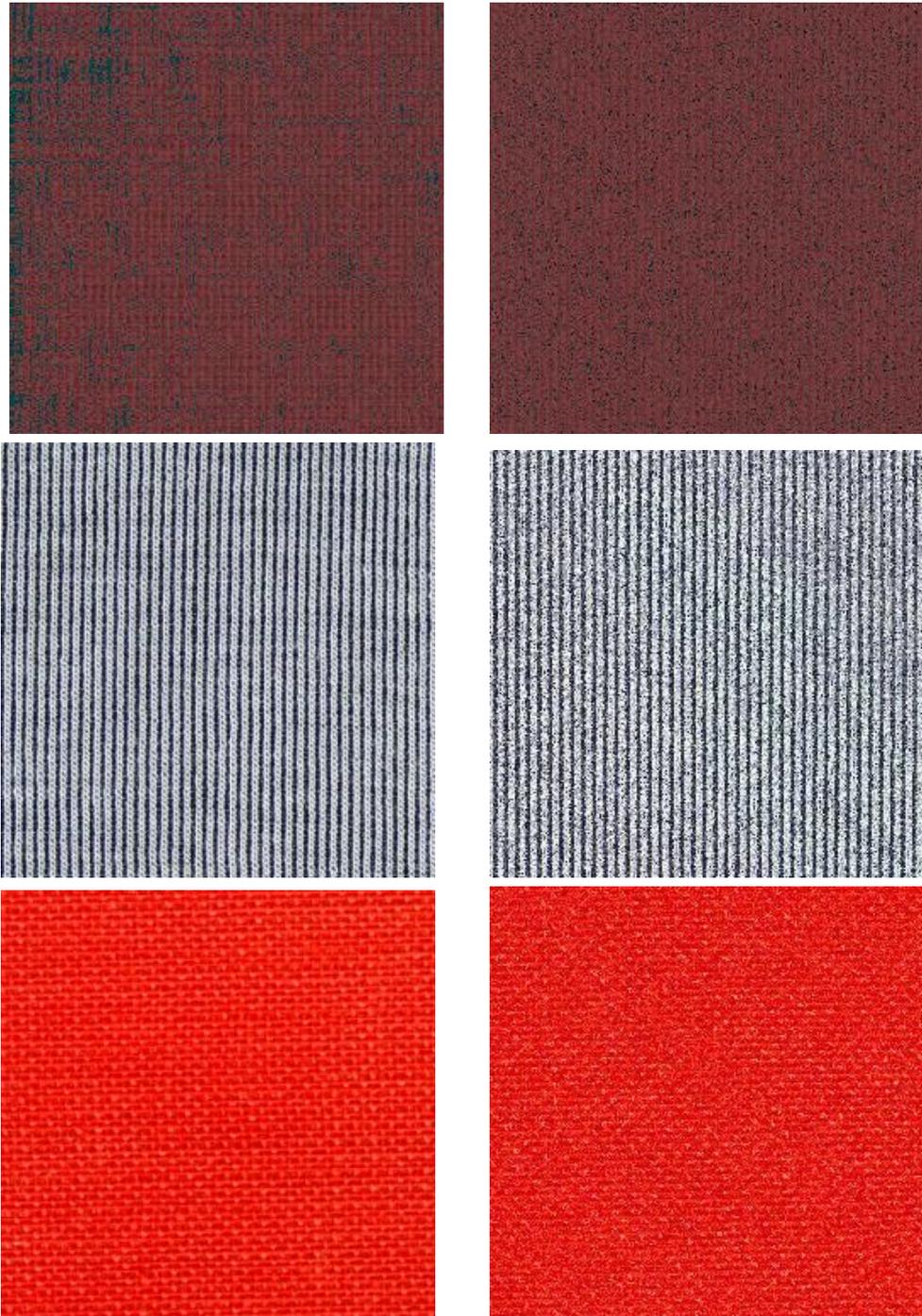


Fig. 7. Natural and synthetic BM (right) textile textures.

The dimension of the sample space is relatively high ($\eta = 200 - 400$) and the corresponding sample size appears to be often insufficient, mainly for DM models. Moreover, the data vectors obtained by shifting the window are overlapping and therefore not independent as it is assumed in the likelihood criterion.

11. CONCLUSION

The application of the EM algorithm to colour, multi-spectral or BTF visual texture modelling has some specific features. Regular or near-regular textures, such as the presented gingham and fabric textures, are notoriously difficult for Markovian texture models. Currently there is no mathematical alternative for their reliable and accurate representation than our probabilistic mixture models. However, the dimension of the sample space is generally relatively high and the corresponding sample size appears to be usually insufficient for adequate model parameters learning. Moreover, the data vectors obtained by shifting the window are not independent as it is assumed in the likelihood criterion. For these and other reasons the estimation of the texture model in the form of set of multivariate Bernoulli or general discrete mixtures is a difficult task. A Gaussian mixture model is easier to learn and less prone to produce noisy failures but also less general than general discrete mixture model. Our extensive discrete or Bernoulli mixtures models simulations suggest that often these models require a relatively large training data set and powerful computing resources to successfully reproduce any given natural texture. A hybrid mixture model using the statistically optimized sampling seems to be a reasonable compromise in situations of limited learning data available. While the computational complexity is going to be less important in near future and on top of that these models are ideal for parallelization and hence for multicore processors or GPU, the requirement for large learning data set is more difficult to overcome and can be restrictive in some texture modelling or recognition applications.

ACKNOWLEDGEMENTS

This research was supported by the grant GA ČR 102/08/0593 and partially by the Ministry of Education, Youth and Sports of the Czech Republic grants 1M0572 DAR, 2C06019, GA ČR 103/11/0335, and CESNET 387/2010.

(Received September 21, 2010)

REFERENCES

- [1] J. Bennett and A. Khotanzad: Multispectral random field models for synthesis and analysis of color images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 20 (1998), 3, 327–332.
- [2] K. J. Dana, S. K. Nayar, B. van Ginneken, and J. J. Koenderink: Reflectance and texture of real-world surfaces. In: *CVPR, IEEE Computer Society, 1997*, pp. 151–157.
- [3] J. S. De Bonet: Multiresolution sampling procedure for analysis and synthesis of textured images. In: *ACM SIGGRAPH 97, ACM Press, 1997*, pp. 361–368.

- [4] A. P. Dempster, N. M. Laird, and D. B. Rubin: Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Statist. Soc. B* 39 (1977), 1, 1–38.
- [5] A. A. Efros and W. T. Freeman: Image quilting for texture synthesis and transfer. In: *ACM SIGGRAPH 2001* (E. Fiume, ed.), ACM Press, 2001, pp. 341–346.
- [6] A. A. Efros and T. K. Leung: Texture synthesis by non-parametric sampling. In: *Proc. Internat. Conf. on Computer Vision (2)*, Corfu 1999, pp. 1033–1038.
- [7] J. Filip and M. Haindl: Bidirectional texture function modeling: A state of the art survey. *IEEE Trans. Pattern Analysis and Machine Intelligence* 31(2009), 11, 1921–1940.
- [8] J. Grim: On numerical evaluation of maximum likelihood estimates for finite mixtures of distributions. *Kybernetika* 18 (1982), 173–190.
- [9] J. Grim and M. Haindl: Texture modelling by discrete distribution mixtures. *Comput. Statist. Data Anal.* 43 (2003), 3–4, 603–615.
- [10] J. Grim, M. Haindl, P. Somol, and P. Pudil: A Subspace approach to texture modelling by using Gaussian mixtures. In: *Proc. 18th Internat. Conference on Pattern Recognition. ICPR 2006* (Y. Y. Tang, S. P. Wang, D. S. Yeung, H. Yan, and G. Lorette, eds.), Vol. II, IEEE Computer Society, Los Alamitos 2006, pp. 235–238.
- [11] M. Haindl: Texture synthesis. *Quarterly* 4 (1991), 4, 305–331.
- [12] M. Haindl, J. Grim, P. Pudil, and M. Kudo: A hybrid btf model based on gaussian mixtures. In: *Texture 2005. The 4th Internat. Workshop on Texture Analysis and Synthesis in Conjunction with ICCV2005* (M. Chantler and O. Drbohlav, eds.), Beijing 2005, Heriot-Watt University & IEEE, pp. 95–100.
- [13] M. Haindl, J. Grim, P. Somol, P. Pudil, and M. Kudo: A Gaussian mixture-based colour texture model. In: *Proc. 17th IAPR Internat. Conference on Pattern Recognition* (J. Kittler, M. Petrou, and M. Nixon, eds.), Vol. III, Los Alamitos 2004, IEEE, pp. 177–180.
- [14] M. Haindl and V. Havlíček: Multiresolution colour texture synthesis. In: *Proc. 7th Internat. Workshop on Robotics in Alpe-Adria-Danube Region* (K. Dobrovodský, ed.), Bratislava 1998. ASCO Art, pp. 297–302.
- [15] M. Haindl and V. Havlíček: A multiresolution causal colour texture model. In: *Advances in Pattern Recognition* (F. J. Ferri, J. M. Inesta, A. Amin, and P. Pudil, eds.), *Lecture Notes in Computer Science* 1876, Chapter 1, Springer-Verlag, Berlin 2000, pp. pages 114–122.
- [16] Michal Haindl and Jiří Filip: Extreme compression and modeling of bidirectional texture function. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29(2007), 10, 1859–1865.
- [17] Michal Haindl and Martin Hatka: Near-regular texture synthesis. In: *Computer Analysis of Images and Patterns* (X. Jiang and N. Petkov, eds.), *Lecture Notes in Computer Sci.* 5720, Springer 2009, pp. 1138–1145.
- [18] Michal Haindl, Vojtěch Havlíček, and Jiří Grim: Colour texture representation based on multivariate bernoulli mixtures. In: *10th Internat. Conference on Information Sciences, Signal Processing and their Applications* (B. Boashash, R. Hamila, S. Hus-sain Shaikh Salleh, and S. Abd Rahman Abu Bakar, eds.), Kuala Lumpur 2010. IEEE, pp. 578–581.

- [19] Michal Haindl, Vojtěch Havlíček, and Jiří Grim: Probabilistic discrete mixtures colour texture models. *Lecture Notes in Computer Sci.* 5197 (2008), 675–682.
- [20] D. J. Heeger and J. R. Bergen: Pyramid based texture analysis/synthesis. In: *ACM SIGGRAPH 95*, ACM Press 1995, pp. 229–238.
- [21] M. L. Koudelka, S. Magda, P. N. Belhumeur, and D. J. Kriegman: Acquisition, compression, and synthesis of bidirectional texture functions. In: *Texture 2003: Third Internat. Workshop on Texture Analysis and Synthesis, Nice 2003*, pp. 59–64.
- [22] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsburg, and T. Limperis: Geometrical Considerations and Nomenclature for Reflectance. NBS Monograph No. 160, National Bureau of Standards, U.S. Department of Commerce, Washington, D. C. 1977.
- [23] R. Paget and I. D. Longstaff: Texture synthesis via a noncausal nonparametric multi-scale markov random field. *IEEE Trans. Image Processing* 7 (1998), 8, 925–932.
- [24] M. Sattler, R. Sarlette, and R. Klein: Efficient and realistic visualization of cloth. In: *Eurographics Symposium on Rendering 2003*.
- [25] Y. Xu, B. Guo, and H. Shum: Chaos Mosaic: Fast and Memory Efficient Texture Synthesis. Technical Report MSR-TR-2000-32, Redmont 2000.
- [26] S. Zelinka and M. Garland: Towards real-time texture synthesis with the jump map. In: *13th European Workshop on Rendering 2002*, pp. 99–104.
- [27] S. C. Zhu, X. W. Liu, and Y. N. Wu: Exploring texture ensembles by efficient markov chain monte carlo – toward a “trichromacy” theory of texture. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22 (2000), 6, 554–569.

Michal Haindl, Institute of Information Theory and Automation – Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. Czech Republic.

e-mail: Haindl@utia.cas.cz

Vojtěch Havlíček, Institute of Information Theory and Automation – Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. Czech Republic.

e-mail: Havlicek@utia.cas.cz

Jiří Grim, Institute of Information Theory and Automation – Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. Czech Republic.

e-mail: Grim@utia.cas.cz