

COLOUR TEXTURE REPRESENTATION BASED ON MULTIVARIATE BERNOULLI MIXTURES

Michal Haindl and Vojtěch Havlíček and Jiří Grim

Institute of Information Theory and Automation, Academy of Sciences CR
Pattern Recognition
182 08 Prague, Czech Republic

ABSTRACT

A novel generative colour texture model based on multivariate Bernoulli mixtures is proposed. A measured multispectral texture is spectrally factorised and multivariate Bernoulli mixtures are further learned from single bit planes of the orthogonal monospectral components and used to synthesise and enlarge these monospectral binary factor components. Texture synthesis is based on easy computation of arbitrary conditional distributions from the model. Finally single synthesised monospectral texture bit planes are transformed into the required synthetic multispectral texture. This model can easily serve not only for texture enlargement but also for segmentation, restoration, and retrieval or to model single factors in complex Bidirectional Texture Function (BTF) space models. The strengths and weaknesses of the presented Bernoulli mixture based approach are demonstrated on several colour texture examples.

Index Terms—Texture modeling, Bernoulli mixture, EM algorithm

1. INTRODUCTION

Generative 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 [1, 2, 3, 4, 5, 6] rely on sophisticated sampling from real texture measurements while the model-based techniques [7, 8, 9, 10, 11, 12] 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 [8]. Most published texture models are restricted only to monospectral textures for few models developed for multispectral (mostly colour) textures refer [7, 9, 10]. We introduced in our previous papers [9, 10] fast multiresolution Markov random field (MRF) based models, which are very efficient for colour or even for substantially more complex BTF [13]

texture modelling, because they do not suffer with some problems of alternative options (see [8] for details) and simultaneously they are easy to analyze as well as to synthesise. However they cannot model well regular or near-regular textures.

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

2. BERNOULLI DISTRIBUTION MIXTURE MODEL

General static multispectral (e.g. colour) texture requires a three-dimensional model (for some fixed illumination and viewing angles). The spectral factorisation alternative (using PCA decorrelation) accepted in this paper allows using simpler 2D data models with much less parameters at the cost of mostly negligible loss of spectral image information. A digitised texture image Y is assumed to be defined on a finite rectangular $N \times M \times d$ lattice I , $r = \{r_1, r_2, r_3\} \in I$ 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. Supposing now uncorrelated monospectral textures after the PCA decorrelation step of our algorithm, we assume that each pixel of the image is described by K possible grey level values, where \mathcal{K} is the set of distinguished grey levels and $\mathcal{B} = \{0, 1\}$. Single monospectral images are further factorised into separate binary bit planes (8 for $|\mathcal{K}| = 256$) of binary variables $\xi \in \mathcal{B}$ which are modelled separately. These binary plane Bernoulli mixture models can be reliably learned from much smaller training texture than the full gray scale discrete mixture models. To simplify notation we will neglect further on the spectral component in the multi-indices r, s because single submodels 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 mospectral tex-

Thanks to GAČR agency for funding.

tured image component represents a realisation 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 centred around the location r and specified by the rectangular index set $I_r \subset I$. If we denote $Y_{\{r\}}$ the corresponding subvector of $Y_{\bullet,\bullet}$, $Y_{\{r\}} = [Y_s \ \forall s \in I_r]$, $Y_{\{r\}} \in \mathcal{B}^\eta$, $\eta = \text{card}\{I_r\}$ and $P(Y_{\{r\}})$ 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. Thus, e.g., for a rectangular window of size $\eta = 20 \times 20$ pixels we have to estimate a 400-dimensional probability distribution $P(Y_{\{r\}})$. The distribution $P(Y_{\{r\}})$ is assumed to be multivariable Bernoulli mixture in the form:

$$P(Y_{\{r\}}) = \sum_{m \in \mathcal{M}} p(m) P(Y_{\{r\}} | m) \quad (1)$$

$Y_{\{r\}} \in 2^\eta$ $\mathcal{M} = \{1, 2, \dots, M\}$ where $p(m)$ are probability weights and the component distributions $P(\cdot | m)$ are multivariable Bernoulli

$$P(Y_{\{r\}}) = \sum_{m \in \mathcal{M}} p(m) \prod_{s \in I_r} p_s(Y_s | m), \quad (2)$$

$p_s(Y_s | m) = \theta_{m,s}^{Y_s} (1 - \theta_{m,s})^{1-Y_s}$, $Y_s \in \mathcal{B}$ are univariate Bernoulli distributions. The parameters of the mixture model (2) are probabilistic component weights $p(m)$ and the univariate discrete distributions of binary levels simply defined by a vector of probabilities:

$$p_s(\cdot | m) = (\theta_{m,s}, 1 - \theta_{m,s}). \quad (3)$$

The total number of mixture (2) parameters is thus $M(1 + 2\eta)$ - confined to the appropriate norming conditions. Note that the form of the univariate discrete distributions (3) is fully general without any constraint. The strong motivation for the multivariable Bernoulli model (2) is a simple switch-over to any marginal distribution by deleting superfluous terms in the products $P(Y_{\{r\}} | m)$.

3. EM ALGORITHM

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

$$\mathcal{S} = \{Y_{\{r\}} \ \forall r \in I, I_r \subset I\} \quad Y_{\{r\}} \in \mathcal{B}^\eta. \quad (4)$$

The unknown parameters of the approximating mixture can be estimated by means of the iterative EM algorithm [14], [15]. In order to estimate the unknown distributions $p_n(\cdot | m)$ and the component weights $p(m)$ we maximize the likelihood function corresponding to (4)

$$L = \frac{1}{|\mathcal{S}|} \sum_{Y_{\{r\}} \in \mathcal{S}} \log \left[\sum_{m \in \mathcal{M}} P(Y_{\{r\}} | m) p(m) \right] \quad (5)$$



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.

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

$$q^{(t)}(m | Y_{\{r\}}) = \frac{P^{(t)}(Y_{\{r\}} | m) p^{(t)}(m)}{\sum_{j \in \mathcal{M}} P^{(t)}(Y_{\{r\}} | j) p^{(t)}(j)}, \quad (6)$$

$$p^{(t+1)}(m) = \frac{1}{|\mathcal{S}|} \sum_{Y_{\{r\}} \in \mathcal{S}} q^{(t)}(m | Y_{\{r\}}), \quad (7)$$

$$p_n^{(t+1)}(\xi | m) = \frac{1}{|\mathcal{S}| p^{(t+1)}(m)} \sum_{Y_{\{r\}} \in \mathcal{S}} \delta(\xi, Y_n) q^{(t)}(m | Y_{\{r\}}), \quad \xi \in \mathcal{B}. \quad (8)$$

The mixture parameters are initialised by random numbers. The iteration process is stopped when the criterion increments are sufficiently small. The iteration scheme (6) – (8) 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 extreme 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.

4. TEXTURE SYNTHESIS

Let I_r be a fixed position of the generating window. If $Y_{\{\rho\}} \subset Y_{\{r\}}$ is a subvector of all pixels previously specified within this window and $\rho \subset I_r$ 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 $p_n |_\rho$:

$$p_n |_\rho(Y_n | Y_{\{\rho\}}) = \sum_{m=1}^M W_m(Y_{\{\rho\}}) p_n(Y_n | m), \quad (9)$$

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

$$W_m(Y_{\{\rho\}}) = \frac{p(m)P_\rho(Y_{\{\rho\}}|m)}{\sum_{j=1}^M p(j)P_\rho(Y_{\{\rho\}}|j)}, \quad (10)$$

$$P_\rho(Y_{\{\rho\}}|m) = \prod_{n \in \rho} p_n(Y_n|m).$$

The binary level y_n can be randomly generated by means of the conditional distribution $p_{n|C}(y_n|Y_{\{\rho\}})$ whereby Eqs. (9) 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 (9) 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 (1) synthesise single bit planes of the decorrelated monospectral components. The resulting synthesised colour (multi-spectral) texture is obtained from combining corresponding bit planes into three (several for multispectral) synthesised monospectral images and inverting the decorrelation process.

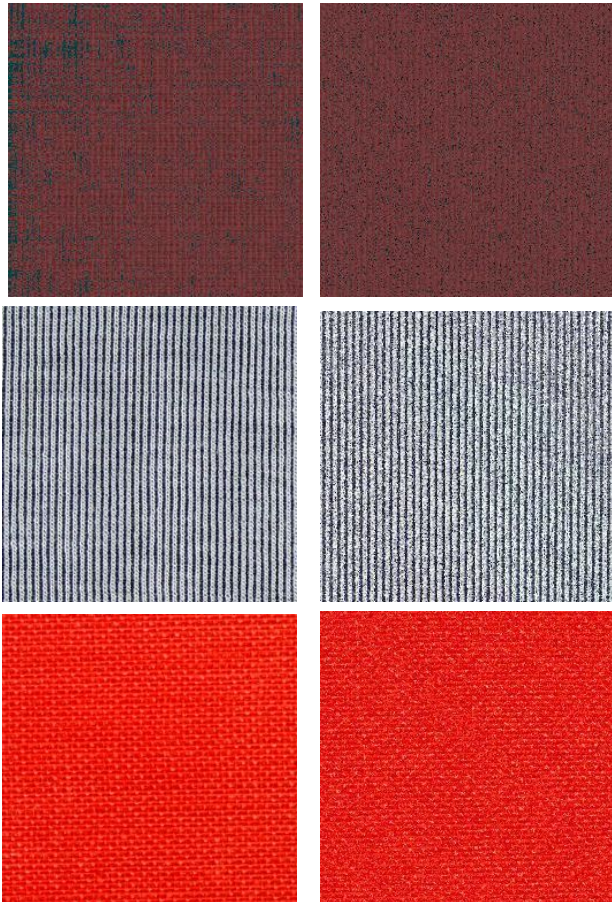


Fig. 2. Natural and synthetic (right) textile textures.

5. 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 ($N \approx 10^1 - 10^2$) 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 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 with one iteration step of the EM algorithm. The examples

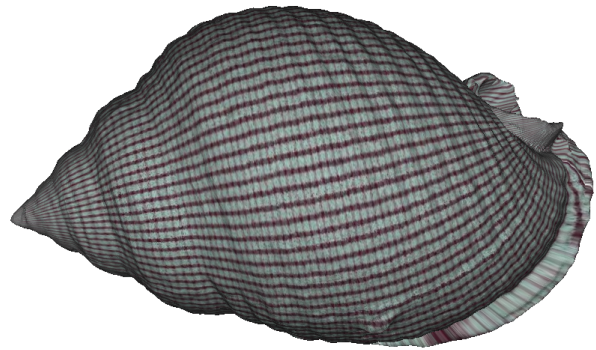


Fig. 3. Synthetic gingham texture mapped on a snail shell model.

Figs. 1,2,3 illustrate properties of our BM model on natural colour textures. The carpet texture on Fig. 1 or gingham texture on Fig. 3 represent relatively regular texture which is notoriously difficult for some alternative texture models like for example Gaussian Markov random field models (Fig. 1 - top right) but the presented model produced very good synthesis result (Fig. 1 - top middle). Similarly the jute example (Fig. 1 - bottom) or the buckram texture (Fig. 2 - top) demonstrate very good performance of the presented model.

Similarly as all other known texture models also our BM model has its strong as well as weak sides. While the presented model can realistically synthesise natural or man-made textures with strong periodicity, which are notoriously difficult for most of alternative approaches, its major weakness is lesser robustness than the Markovian models family. A BM model has strong tendency either to produce high quality synthetic texture or to completely fail with resulting noise field. Markovian models in these cases demonstrate clear effort to grasp at least some of the difficult texture features. The computationally most ef-

efficient Markovian models are much faster than the presented model, but general Markovian models which require Markov chain Monte Carlo methods for their analysis as well as synthesis are comparable.

6. CONCLUSION

The application of EM algorithm to colour texture modelling has some specific features. Generally the dimension of the sample space is relatively high and the corresponding sample size appears to be sometimes insufficient. 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 mixtures is a difficult task. Our extensive BM models simulations suggest that often the model requires a relatively large training data set and powerful computing resources to successfully reproduce any given natural texture. While the computational complexity is going to be less important in near future and on top of that this model is ideal for parallelization, the requirement for large learning data set is more difficult to overcome and can be restrictive in some texture modelling applications.

Acknowledgements

This research was supported by the grant GAČR 102/08/0593 and partially by the MŠMT grants 1M0572 DAR, 2C06019.

7. REFERENCES

- [1] J.S. De Bonet, "Multiresolution sampling procedure for analysis and synthesis of textured images," in *ACM SIGGRAPH 97*. 1997, pp. 361–368, ACM Press.
- [2] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *ACM SIGGRAPH 2001*, Eugene Fiume, Ed. 2001, pp. 341–346, ACM Press.
- [3] Alexei A. Efros and Thomas K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. Int. Conf. on Computer Vision (2)*, Corfu, Greece, 1999, pp. 1033–1038.
- [4] D.J. Heeger and J.R. Bergen, "Pyramid based texture analysis/synthesis," in *ACM SIGGRAPH 95*. 1995, pp. 229–238, ACM Press.
- [5] Y. Xu, B. Guo, and H. Shum, "Chaos mosaic: Fast and memory efficient texture synthesis," Tech. Rep. MSR-TR-2000-32, Redmont, 2000.
- [6] S. Zelinka and M. Garland, "Towards real-time texture synthesis with the jump map," in *13th European Workshop on Rendering*, 2002, p. 99104.
- [7] J. Bennett and A. Khotanzad, "Multispectral random field models for synthesis and analysis of color images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 327–332, March 1998.
- [8] M. Haindl, "Texture synthesis," *CWI Quarterly*, vol. 4, no. 4, pp. 305–331, December 1991.
- [9] M. Haindl and V. Havlíček, "Multiresolution colour texture synthesis," in *Proceedings of the 7th International Workshop on Robotics in Alpe-Adria-Danube Region*, K. Dobrovodský, Ed., Bratislava, June 1998, pp. 297–302, ASCO Art.
- [10] M. Haindl and V. Havlíček, "A multiresolution causal colour texture model," in *Advances in Pattern Recognition, Lecture Notes in Computer Science 1876*, F. J. Ferri, J. M. Inesta, A. Amin, and P. Pudil, Eds., chapter 1, pp. 114 –122. Springer-Verlag, Berlin, August 2000.
- [11] R. Paget and I. D. Longstaff, "Texture synthesis via a noncausal nonparametric multiscale markov random field," *IEEE Trans. on Image Processing*, vol. 7, no. 8, pp. 925–932, 1998.
- [12] 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. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 554–569, June 2000.
- [13] Michal Haindl and Jiří Filip, "Extreme compression and modeling of bidirectional texture function," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1859–1865, 2007.
- [14] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, B*, vol. 39, no. 1, pp. 1–38, 1977.
- [15] J. Grim, "On numerical evaluation of maximum likelihood estimates for finite mixtures of distributions," *Kybernetika*, vol. 18, pp. 173–190, 1982.