2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)

BTF Compound Texture Model with Fast Iterative Non-Parametric Control Field Synthesis

Michal Haindl and Vojtěch Havlíček
Institute of Information Theory and Automation of the CAS
Prague, Czech Republic 182 08
Email: {haindl,havlicek}@utia.cas.cz

***Abstract*—We propose a substantial speed up a modification to our recently published novel multidimensional statistical model for realistic modeling, enlargement, editing, and compression of the recent state-of-the-art Bidirectional Texture Function (BTF) textural representation. The multispectral compound Markov random field model (CMRF) efficiently fuses a non-parametric random field model with several parametric Markovian random fields models. The principal application of our model is physically correct and realistic synthetic imitation of material texture, its enlargement, and huge compression. So that ideally, both natural and synthetic texture of a given measured natural or artificial texture will be visually indiscernible for any observation or illumination directions. The presented model can be easily applied also for BTF material texture editing to model non-measured or unmeasurable but still realistic material textures. The CMRF model consists of several parametric sub-models each having different characteristics along with an underlying switching structure model which controls transitions between these submodels. The proposed model uses the non-parametric random field for distributing local texture models in the form of analytically solvable wide-sense BTF Markov representation for single regions among the fields of a mosaic approximated by the random field structure model. The non-parametric control field of the BTF-CMRF is iteratively generated to guarantee identical region-size histograms for all material sub-classes present in the target example texture. The present iterative algorithm significantly cuts the number of iterations to converge in comparison with our previous iterative method and even sometimes skip all iteration due to its ingenious initialization. The local texture regions (not necessarily continuous) are represented by analytical BTF models modeled by the adaptive 3D causal auto-regressive (3DCAR) random field model which can be analytically estimated as well as synthesized. The visual quality of the resulting complex synthetic textures generally surpasses the outputs of the previously published simpler non-compound BTF-MRF models and allows to reach tremendous compression ratio incomparable with any standard image compression method.**

***Keywords*-BTF texture model; compound Markov random field; BTF texture synthesis;**

I. Introduction

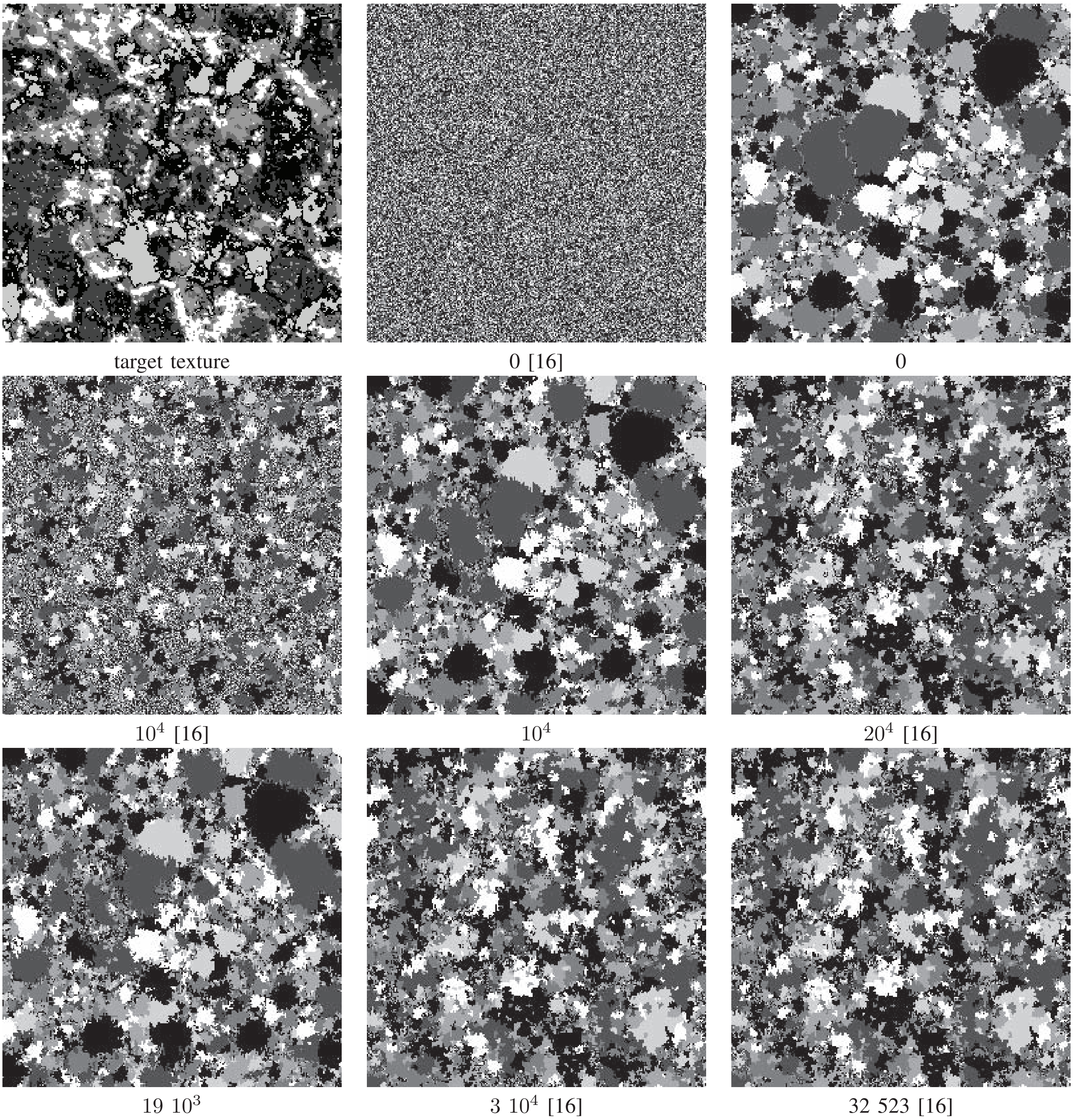
A physically correct real materials surface reflectance is a very complex, currently unfeasible to measure or to mathematically model, the function of at least 16 variables [1]. The Bidirectional Texture Function (BTF) is its state-of-the-art approximation which allows to express spectral, spatial, illumination angle, and observation angle visual dependencies of a measured material texture what significantly improves the visual realism of a modeled object at the expense of non-trivial measurements and mathematical modeling of these huge BTF data spaces. Static BTF texture modeling based on probabilistic models requires complex seven-dimensional models. It is far from being a straightforward generalization of any 3D model (required for usual static three-dimensional color textures) with just adding four additional dimensions. On the contrary, every extra model dimension multiplies difficulties encountered within all necessary modeling steps [1], i.e., optimal model selection, robust parameters estimation from always limited learning data, stability, and synthesis. A practical, reliable full 7D BTF model has not yet been developed. Thus we use two factorization levels which are the conceivable approximation for acceptable visual quality.

Compound Markov random field models (CMRF) consist of several sub-models each having different characteristics along with an underlying structure model which controls transitions between these submodels [2]. CMRF models were already applied to the image restoration [2]–[5], segmentation [6], or image or texture modeling [7]–[10]. Unfortunately, Markovian models generally require demanding numerical solutions for learning as well as for synthesis with all their well-known drawbacks. The exceptional CMRF [7], [9] models allow analytical synthesis at the cost of a slightly compromised compression rate due to the non-parametric control field data. Methods based on different Markov random fields [11]–[16] combine an estimated range-map with synthetic multiscale smooth texture using Markov models. The measured BTF data are analyzed for their intrinsic dimensionality [1] and factorized into BTF and subsequently also spatial factors. The original registered BTF illumination-/-view measurement space is segmentation into several subspace images using the K-means algorithm in the perceptually uniform CIE Lab color-space using color cumulative histograms features.

In this paper, we propose a significant speed-up modification of our previous BTF-CMRF^{NPi3AR} model [16]. This new iterative procedure allows cutting the number of iterations to converge to be about 20% of the previous

978-1-5386-9385-8/18/$31.00 ©2018 IEEE
DOI 10.1109/SITIS.2018.00025

IEEE computer society



target texture

0 [16]

0

10^4 [16]

10^4

20^4 [16]

19 10^3

3 10^4 [16]

32 523 [16]

Figure 1. Comparison of the granite control field synthesis [16] and its proposed speed up version. The proposed control field iteration converges at 19 214 cycles, while the older algorithm [16] needs 32 523 cycles.

algorithm. We propose a hierarchical BTF-CMRF^{NPfi3AR} model which combines a non-parametric Markov random field (MRF) model with local parametric MRF models [17], [18]. The parametric MRF models can be analytically solved, while the other is synthesized using a newly proposed faster iterative method for its synthesis.

II. Compound Markov Model

Let us denote a multiindex $r = (r_1, r_2)$, $r \in I$, where I is a discrete 2-dimensional rectangular lattice and r_1 is the row and r_2 the column index, respectively. $X_r \in \{1, 2, \ldots, K\}$ is a random variable with natural number value (a positive integer), Y_r is the multispectral pixel at location r and $Y_{r,j} \in \mathcal{R}$ is its j-th spectral plane component. Both random fields (X, Y) are indexed on the same $M \times N$ lattice I. Let us assume that each multispectral observed texture $\tilde{Y}$ (composed of d spectral planes, e.g., $d = 3$ for colour textures) and indexed on the $\tilde{M} \times \tilde{N}$ lattice $\tilde{I}$ (usually $\tilde{I} \subseteq I$) can be modeled by a compound Markov random field model, where the principal Markov random field (MRF) X controls switching to a regional local MRF model $Y = \bigcup_{i=1}^{K} {}^{i}Y$. Single K regional sub-models ${}^{i}Y$ are defined on their corresponding lattice subsets ${}^{i}I$, ${}^{i}I \cap {}^{j}I = \emptyset \quad \forall i \neq j$ and they are of the same MRF type. These models differ only in their contextual support sets ${}^{i}I_r$ and corresponding parameters sets ${}^{i}\theta$. The BTF-CMRF^{NPfi3AR} model has the posterior probability

$$P(X, Y \,|\, \tilde{Y}) = P(Y \,|\, X, \tilde{Y}) P(X \,|\, \tilde{Y}) \qquad (1)$$

and the corresponding optimal MAP solution is:

$$(\hat{X}, \hat{Y}) = \arg \max_{X \in \Omega_X, Y \in \Omega_Y} P(Y \,|\, X, \tilde{Y}) \, P(X \,|\, \tilde{Y}) \ ,$$

where Ω_X, Ω_Y are the corresponding configuration spaces for both random fields (X, Y). To avoid an iterative MCMC MAP solution, which cannot be managed in the huge BTF data space, we proposed the following two step approximation [7]:

$$\begin{aligned} (\breve{X}) &= \arg \max_{X \in \Omega_X} P(X \,|\, \tilde{Y}) \ , & (2) \\ (\breve{Y}) &= \arg \max_{Y \in \Omega_Y} P(Y \,|\, \breve{X}, \tilde{Y}) \ . & (3) \end{aligned}$$

This approximation substantialy simplifies the BTF-CMRF^{NPfi3AR} estimation because it allows us to take advantage of an analytical estimation of all regional MRF models ${}^{i}Y$ in (3).

A. Non-Parametric Control Field

The control random field X (Fig.1 - left upper row) is assumed to be independent on illumination and observation angles, i.e., it is identical for all possible combinations $\phi_i, \phi_v, \theta_i, \theta_v$ azimuthal and elevation illumination / viewing angles, respectively. This assumption does not compromise the resulting BTF space quality, because it influences only a material texture macro-structure which is independent on these angles.

The control random field $\breve{X}$ is estimated using simple K-means clustering of $\tilde{Y}$ in the RGB colour space into predefined number of K classes, where cluster indices ω_i are $\breve{X}_r \quad \forall r \in I$ estimates. The number of classes K can be estimated using the Kullback-Leibler divergence and considering a sufficient amount of data necessary to reliably estimate all local Markovian models. The clustering resulting thematic map is used to compute region size histograms $\tilde{h}_i$ for all $i = 1, \dots, K$ classes. Let us order classes according the decreasing number of pixels $\tilde{n}_i$ belonging to each class, i.e., $\tilde{n}_1 \geq \tilde{n}_2 \geq \dots \geq \tilde{n}_K$. Histograms $\tilde{h}_i$ are the only parameters required to store for the control field.

1) Iterative Control Field Synthesis: The iterative algorithm (Fig.1) is based on a data structure which describes for each pixel a membership in the region. For each region the class membership, a size of the region and the requested number of regions of it's size, all border pixels from both sides of the border, possibility to decrease or increase of the region, and for all classes the histogram and regions, which can be increased or decreased. After any change in a pixel class assignment, this structure has to be updated.

0. The synthesized $M \times N$ required control field is initialized to the value ω_0 it means, that pixel was not assigned to any class ω_i for $i = 1, \dots, K$. All histograms cells are rescaled using the scaling factor $\frac{MN}{\tilde{M}\tilde{N}}$, i.e., $X_r^{(0)} = \omega_1 \ \ \forall r \in I$ and $\tilde{h}_i \rightarrow h_i$ for $i = 1, \dots, K$. All regions from all classes $i = 1, \dots, K$ are sorted by region size. Starting from the biggest region A_1 till the smallest region A_M, where M is number of all regions, a lattice multiindex r is randomly generated. First pixel X_r of the region A_j where $j = 1, \dots, M$ and class ω_i is randomly selected and is changed to new value $X_r = \omega_i$ only if its previous value was $X_r = \omega_0$ All neighbours X_s of the pixel X_r which fulfil conditions $X_s = \omega_0$ and pixel X_s has no neighbour from the class ω_i are added to the queue Q. Till the size of region A_j is higher than number of actually added pixels, next pixel X_r is randomly selected from the queue Q the values is changed to $X_r = \omega_i$ and its neigbours are added to the queue Q if they meets mentioned conditions. If the queue Q is empty and size of the region A_j is higher than the number of actually assigned pixels, the rest of the pixels is randomly assigned to the class ω_i after the initialization of the last region A_M. After this initialization step, all classes have their correct required number of pixels but not yet their correct region size histograms.
1. Pixels r and s are randomly selected with the following properties: The pixel r from the class ω_i is on the border between region $\downarrow \omega_i^A$ (a region A which can be decreased) and a region $\uparrow \omega_j^B$ (a region B which can be increased). The pixel s from the class ω_j is on the border between region $\downarrow \omega_j^C$ (a region C which can be decreased) and a region $\uparrow \omega_i^D$ (a region D which can be increased). These regions have to be distinct, i.e., $A \cap D = \emptyset$ and $B \cap C = \emptyset$. If such pixels r, s exist go to the step 5. If not repeat this step once more.
2. Gradually check all class couples starting from $\omega_1, \omega_2, \dots, \omega_K$ to find pixels r, s which meet conditions in step 1. All regions corresponding to

the chosen classes ω_i and ω_j are selected randomly. If such pixels r, s exist, go to step 5.

3. Randomly select a region from class ω_i which has two neighbouring regions of class ω_j such as one can be decreased and another increased. If there exist two border pixels r, s in the region ω_j, where r is a border pixel with a region to be increased and s with a region to be decreased, go to the step 5.
4. Gradually check all classes with incorrect histogram, starting from $\omega_1, \omega_2, \ldots, \omega_K$, for every class ω_i gradually check all its regions $\uparrow \omega_i^A$ which can be increased, for each region $\uparrow \omega_i^A$ check every region neighbouring border pixel r from class ω_j and region $\downarrow \omega_j^B$ (a region B which can be decreased) and find pixel s with the following properties: pixel s is from the class ω_i and region $\downarrow \omega_i^C$ (a region C which can be decreased), pixel s is on the boarder of the region $\uparrow \omega_j^D$ from class ω_j (a region which can be increased). These regions have to be distinct, i.e., $A \cap C = \emptyset$ and $B \cap D = \emptyset$. If such pixels do not exist go to step 7.
5. $X_r = \omega_j, X_s = \omega_i$ update the data structure.
6. If the number of iterations is less than a selected limit go to 1.
7. Store the resulting control field and stop.

The steps 1.,2. allow simultaneous improvement of four regions while step 3. improves two regions only. The algorithm converges to the correct class histograms $h_i \quad i = 1, \ldots, K$.

Table I
THE NUMBER OF ITERATIONS FOR THE CONTROL FIELD SYNTHESIS FOR THE CONVERGENCE.

texture	256×256 [16]	256×256	512×512 [16]	512×512
moss	178 167	3 877	392 954	470 022
bark16	40 193	7 957	141 683	40 985
bark4	45 796	12 666	128 108	41 924
begonia	198 430	9 961	216 328	74 064
blossoms1	9 043 864	0	400 000	0
floor tile	4 792 656	870 000	14 215 445	500 000
floor plastic	9 906 941	1 650 000	12 787 487	500 000
clay06	12 227	0	45 209	0
clay14	28 073	8 609	99 465	11 885
fabric16	18 603	9 080	77 959	19 542
granite2	32 523	19 214	260 779	171 611
grass38	23 502	18 542	93 131	31 830
lichen9	63 931	9 655	140 146	29 137
marble4	75 374	12 581	480 000	55 519
meadow3	67 041	51 727	500 000	500 000
stone90	18 369	4 313	66 891	7 627
rusty plate	55 065	7 175	92 490	48 224
stone28	15 956	0	255 618	200 410
stone29	13 902	4 780	56 803	24 249
median	45 796	9 080	141 683	41 924

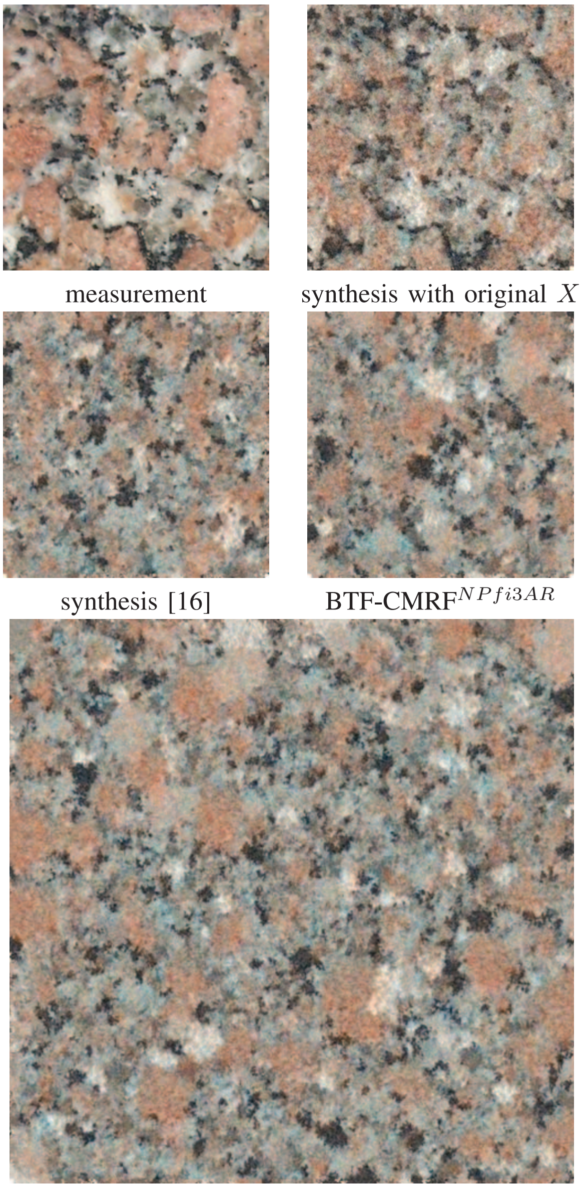


Figure 2. The granite texture synthesis, target texture (left upper row), its synthesis and enlargement right, middle, and bottom, respectively.

B. Local BTF Markov Models

Local i-th texture region (not necessarily continuous) is represented by the adaptive 3D causal auto-regressive random (3DCAR) field model [17], [18]. This model can be analytically estimated as well as easily synthesised. The model can be defined in the following matrix equation (i-th model index is further omitted to simplify notation):

$$Y_r = \gamma Z_r + \epsilon_r \ , \qquad (4)$$

where $Z_r = [Y_{r-s}^T : \forall s \in I_r]^T$ is the $\eta d \times 1$ data vector with multiindices r, s, t, $\gamma = [A_1, \ldots, A_\eta]$ is the $d \times d \, \eta$ unknown parameter matrix with parametric submatrices A_s. The model functional contextual neighbour index shift set is denoted I_r and $\eta = cardinality(I_r)$. All 3DCAR model statistics can be efficiently estimated analytically [17]. Given the known 3DCAR process history $Y^{(t-1)} = \{Y_{t-1}, Y_{t-2}, \ldots, Y_1, Z_t, Z_{t-1}, \ldots, Z_1\}$ the parameter estimation $\hat{\gamma}$ can be accomplished using fast, numerically robust and recursive statistics [17]:

$$\begin{aligned}
\hat{\gamma}_{t-1}^T &= V_{zz(t-1)}^{-1} V_{zy(t-1)} \ , \\
V_{t-1} &= \tilde{V}_{t-1} + V_0 \ , \\
\tilde{V}_{t-1} &= \begin{pmatrix} \sum_{u=1}^{t-1} Y_u Y_u^T & \sum_{u=1}^{t-1} Y_u Z_u^T \\ \sum_{u=1}^{t-1} Z_u Y_u^T & \sum_{u=1}^{t-1} Z_u Z_u^T \end{pmatrix} \\
&= \begin{pmatrix} \tilde{V}_{yy(t-1)} & \tilde{V}_{zy(t-1)}^T \\ \tilde{V}_{zy(t-1)} & \tilde{V}_{zz(t-1)} \end{pmatrix} \ ,
\end{aligned}$$

where V_0 is a positive definite matrix (see [17]). Although, an optimal causal functional contextual neighbourhood I_r can be solved analytically by a straightforward generalisation of the Bayesian estimate in [17], we use the faster approximation which does not need to evaluate statistics for all possible I_r configurations. This approximation is based on spatial correlations. Starting from the causal part of a hierarchical non-causal neighbourhood, neighbours locations corresponding to spatial correlations larger than a specified threshold (> 0.6) are selected. The i-th model pixel-wise synthesis is the simple, direct application of (4) for all 3DCAR models. 3DCAR models provide better spectral modelling quality than the alternative spectrally decorrelated 2D models for motley textures at the cost of a small increase in the number of parameters to be stored.

Parameters of the selected local subspace BTF Markov models are estimated and stored in a small parametric database. These spectral models are finally fused with the estimated range map. The BTF range map estimate could benefit from tens of ideally mutually registered BTF measurements, thus it is advantageous to use the over-determined photometric stereo from among the possible estimation alternatives of the range map. The required synthetic factors are generated on request, the factorization process of the synthetic BTF subspace is inverted, and then this inversion is used in a virtual scene mapping. Finally, the overall BTF texture's visual appearance during changes of viewing and illumination conditions is simulated using the displacement mapping technique [1].

III. Results

The ideal synthetic texture should be visually indiscernible for any observation or illumination directions from the given measured natural texture, but pixelwise identical. Such visual similarity between a synthetic and measured BTF space would ideally be measured by some mathematical criterion. Unfortunately, an automatic texture quality evaluation is important but still unsolved difficult problem, and qualitative evaluation is, for now, possible only using impractical and expensive visual psycho-physics. We have recently tested [19] on our texture fidelity benchmark (**http://tfa.utia.cas.cz**) several published state-of-the-art image quality measures and also a dedicated texture measure (STSIM) [20] in several variants or our textural qualitative criterion based on the generative Markovian texture model statistics ζ [21] which slightly outperforms the best alternative - the STSIM fidelity criterion.

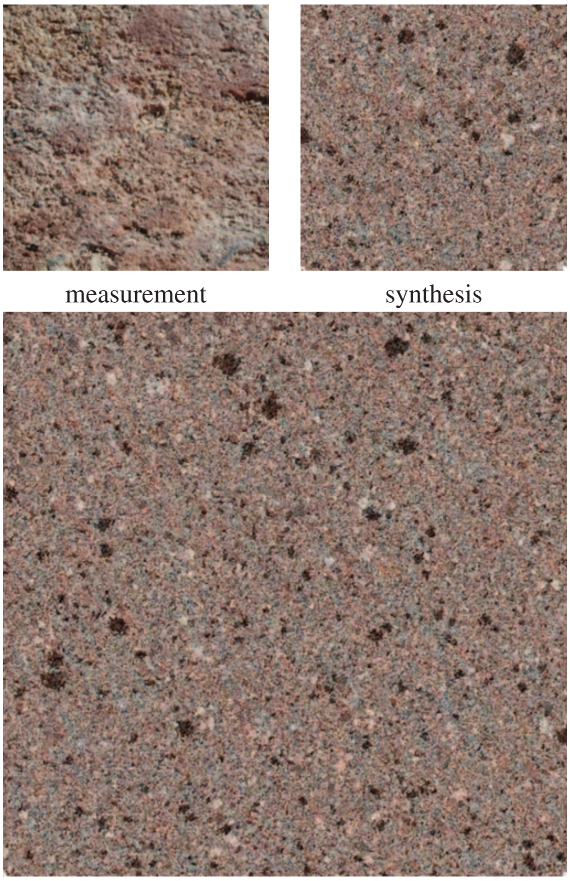

measurement synthesis

Figure 3. The clay texture synthesis, target texture (left upper row), its synthesis and enlargement right and bottom, respectively.

These results demonstrate that neither the standard image quality criteria (MSE [22], VSNR [23], VIF [24], SSIM [25], CW-SSIM [26]) nor the STSIM texture criterion can be reliably used for texture quality validation (see for details [19]). It is easy to manifest failure counterexamples for each of these published quality criteria. Thus our results can be checked only visually.

Fig.1 visualizes a comparison between our previous [16] iterative control field synthesis and the presented one on the granite control field. It possible to note an important initialization improvement and the overall convergence speed up to 59% of previously required cycles. This speed up depends on the texture. Tab.I shows the required number of cycles for the

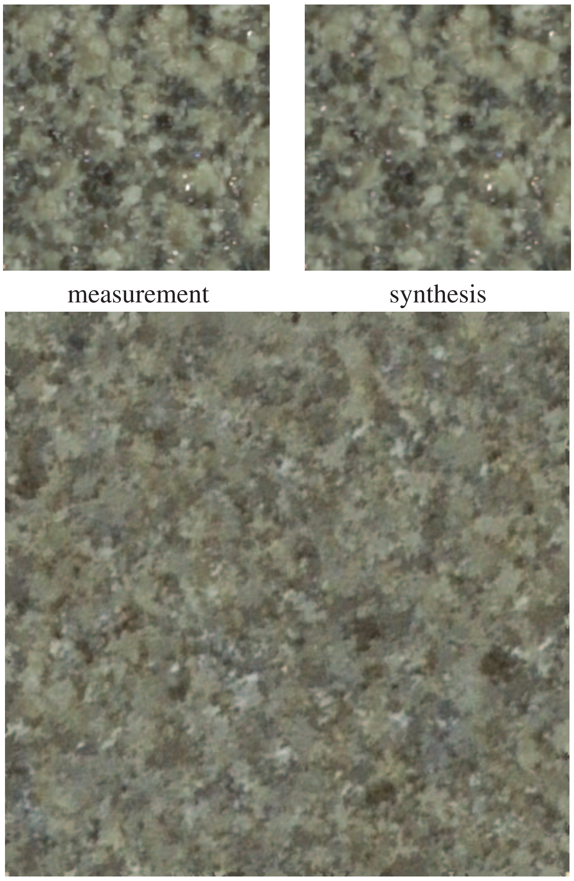


Figure 4. The floor tile texture synthesis, target texture (left upper row), its synthesis and enlargement right and bottom, respectively.

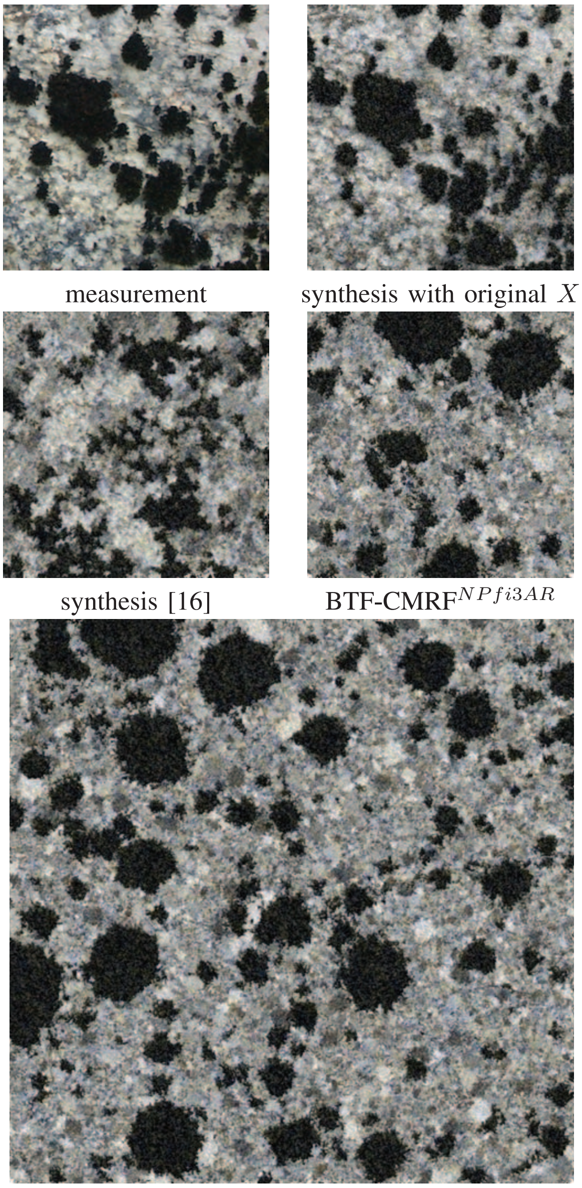


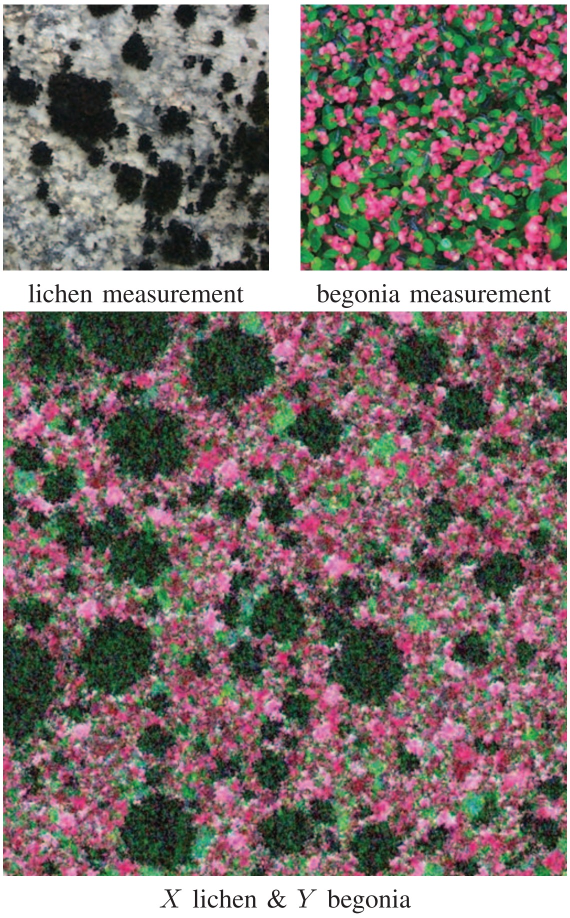
Figure 5. The lichen texture synthesis, target texture (left upper row), its synthesis and enlargement right, middle, and bottom, respectively.

convergence for nineteen materials, two texture sizes, and the previous and presented iterative methods. Even for three textures (blossoms1, clay06 Fig.3, stone28), the initialization is sufficient, and there is no need for any iterations. The median speed up between both methods is one-fifth of previous cycles. The number of convergence cycles (Tab.I) is linearly dependent on the required control field size in average.

Figs.2,5 allow to compare synthesis results for the previous [16] and presented methods on granite and lichen textures, respectively. Although both methods produce high-quality results, they differ. The presented method tends to create large convex regions, if possible. Both figures also illustrate a negligible visual difference between the original measurement and its synthetic version if the control field was just estimated and not synthesized (upper rows right). This is another demonstration of the high model quality as well as its huge compression capability.

Figs.2-6 demonstrate various natural (granite, clay, floor tile, lichen, grass, and begonia) texture synthesis results using the proposed model. Due to space constraint, we present only one texture from each corresponding BTF measurement space. For some textures (lichen, grass, begonia) we do not have BTF measurements. Thus they illustrate the modeling application of our model on standard color textures. However, even these non-BTF textures can be used as control field learning data for BTF space editing if the parametric models Y are estimated from some BTF data space.

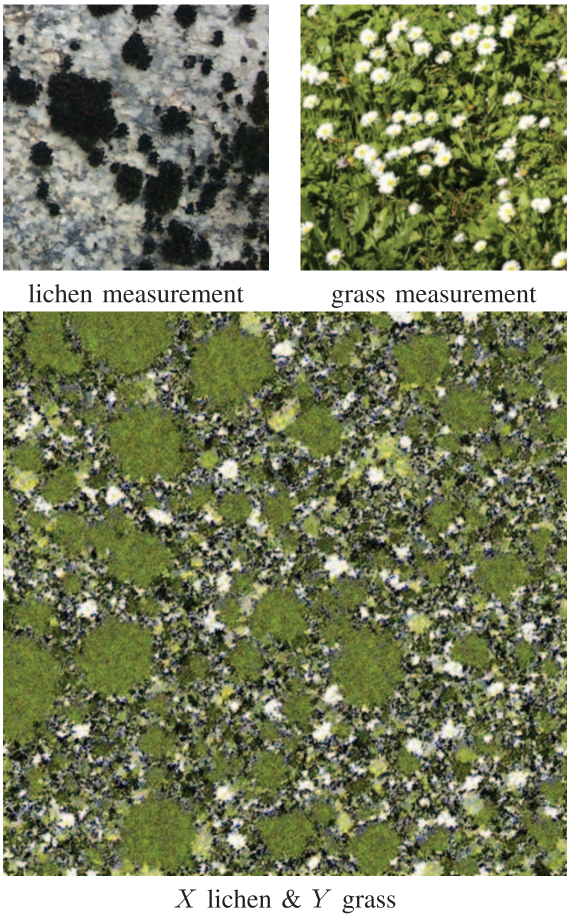
Figs.6,7 show the texture editing capability of the model. Its top row are measured natural lichen and begonia or grass textures and the bottom row contains their corresponding editing results. The bottom texture was created by combing an estimated control field from the lichen texture with local



lichen measurement begonia measurement

X lichen & Y begonia

Figure 6. The measured bark (source of the control field) and begonia textures (upper row), their synthesis (middle), and edited texture synthesis with the bark control field and local begonia models.



lichen measurement grass measurement

X lichen & Y grass

Figure 7. The measured bark (source of the control field) and grass textures (upper row), their synthesis (middle), and edited texture synthesis with the bark control field and local grass models.

Markovian models (4) estimated from the begonia or grass texture. Although the edited textures change their appearance, they still look convincingly and physically correct.

IV. Conclusion

The presented BTF-CMRF^{NPfi3AR} method exhibits very good results on the selected texture categories, i.e., textures with the random type of their macro-structure. The model combines the iteratively solved non-parametric random field for distributing local texture models with the analytically solvable wide-sense BTF Markovian submodels. The median speed up between the presented and our previous method for the non-parametric control field synthesis is one-fifth of the required cycles to converge. For some textures, the control field synthesis even does not need any iterations. The synthetic texture can be easily modified by changing the underlying mosaic generation model, which is now performed by fast iterative modification of class region histograms, to fit better with different random texture types. The proposed BTF-CMRF^{NPfi3AR} model is well suited to model various types of natural materials surfaces such as clay, lichens, stones, barks, rusty materials, or meadows. The model allows for seamless multispectral texture synthesis and enlargement with an extremely high compression rate independent of the size of the desired resulting texture. The model does not generate any repetitions contrary to the most sampling alternatives. The data needed to be stored comprised of only several dozens of parameters and few region size histograms. Using a simple modification of the method, we can use it for texture editing (by changing the local texture models for several indexes of the control field), we can use it for modeling BTF textures or even the synthesis of new, unmeasured textures by manually or automatically assigning the model's parameters. The visual quality of the resulting complex synthetic textures generally surpasses the outputs of the previously published simpler non-compound BTF-MRF models.

References

[1] M. Haindl and J. Filip, *Visual Texture*, ser. Advances in Computer Vision and Pattern Recognition. London: Springer-

Verlag London, January 2013.

[2] F.-C. Jeng and J. W. Woods, "Compound gauss-markov random fields for image estimation," *IEEE Transactions on Signal Processing*, vol. 39, no. 3, pp. 683–697, 1991.

[3] S. Geman and D. Geman, "Stochastic relaxation , gibbs distributions and bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 6, no. 11, pp. 721–741, November 1984.

[4] M. Figueiredo and J. Leitao, "Unsupervised image restoration and edge location using compound gauss - markov random fields and the mdl principle," *Image Processing, IEEE Transactions on*, vol. 6, no. 8, pp. 1089–1102, April 1997.

[5] R. Molina, J. Mateos, A. Katsaggelos, and M. Vega, "Bayesian multichannel image restoration using compound gauss-markov random fields," *IEEE Trans. Image Proc.*, vol. 12, no. 12, pp. 1642–1654, December 2003.

[6] J. Wu and A. C. S. Chung, "A segmentation model using compound markov random fields based on a boundary model," *IEEE Trans. Image Processing*, vol. 16, no. 1, pp. 241–252, jan 2007.

[7] M. Haindl and V. Havlíček, "A compound MRF texture model," in *Proceedings of the 20th International Conference on Pattern Recognition, ICPR 2010*. Los Alamitos: IEEE Computer Society CPS, August 2010, pp. 1792–1795.

[8] M. Haindl, V. Remeš, and V. Havlíček, "Potts compound markovian texture model," in *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012*. Los Alamitos: IEEE Computer Society CPS, November 2012, pp. 29 – 32.

[9] M. Haindl and V. Havlíček, "A plausible texture enlargement and editing compound markovian model," in *Computational Intelligence for Multimedia Understanding*, ser. Lecture Notes in Computer Science, E. Salerno, A. Cetin, and O. Salvetti, Eds. Springer Berlin / Heidelberg, 2012, vol. 7252, pp. 138–148, 10.1007/978-3-642-32436-9_12.

[10] M. Haindl and V. Havlíček, "Two compound random field texture models," in *2016 the 21st IberoAmerican Congress on Pattern Recognition (CIARP 2016)*, ser. Lecture Notes in Computer Science, C. Beltrán-Castañón, I. Nyström, and F. Famili, Eds., vol. 10125. Gewerbestrasse 11, Cham, CH-6330, Switzerland: Springer International Publishing AG, November 2017, pp. 44 – 51.

[11] M. Haindl and J. Filip, "Fast BTF texture modelling," in *Texture 2003. Proceedings*, M. Chantler, Ed. Edinburgh: IEEE Press, October 2003, pp. 47–52.

[12] M. Haindl, J. Filip, and M. Arnold, "BTF image space utmost compression and modelling method," in *Proceedings of the 17th IAPR International Conference on Pattern Recognition*, J. Kittler, M. Petrou, and M. Nixon, Eds., vol. III. Los Alamitos: IEEE Press, August 2004, pp. 194–197.

[13] M. Haindl and J. Filip, "A fast probabilistic bidirectional texture function model," *Lecture Notes in Computer Science*, no. 3212, pp. 298 – 305, 2004.

[14] ——, "Extreme compression and modeling of bidirectional texture function," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1859–1865, 2007.

[15] M. Haindl and M. Havlíček, "Bidirectional texture function simultaneous autoregressive model," in *Computational Intelligence for Multimedia Understanding*, ser. Lecture Notes in Computer Science, E. Salerno, A. etin, and O. Salvetti, Eds. Springer Berlin / Heidelberg, 2012, vol. 7252, pp. 149–159, 10.1007/978-3-642-32436-9_13.

[16] M. Haindl and V. Havlíček, "Btf compound texture model with non-parametric control field," in *The 24th International Conference on Pattern Recognition (ICPR 2018)*. IEEE, August 2018, pp. 1151 – 1156.

[17] M. Haindl, "Visual data recognition and modeling based on local markovian models," in *Mathematical Methods for Signal and Image Analysis and Representation*, ser. Computational Imaging and Vision, L. Florack, R. Duits, G. Jongbloed, M.-C. Lieshout, and L. Davies, Eds. Springer London, 2012, vol. 41, ch. 14, pp. 241–259, 10.1007/978-1-4471-2353-8_14.

[18] M. Haindl and V. Havlíček, "A multiscale colour texture model," in *Proceedings of the 16th International Conference on Pattern Recognition*, R. Kasturi, D. Laurendeau, and C. Suen, Eds. Los Alamitos: IEEE Computer Society, August 2002, pp. 255–258.

[19] M. Haindl and M. Kudělka, "Texture fidelity benchmark," in *Computational Intelligence for Multimedia Understanding (IWCIM), 2014 International Workshop on*. Los Alamitos: IEEE Computer Society CPS, November 2014, pp. 1 – 5.

[20] J. Zujovic, T. Pappas, and D. Neuhoff, "Structural texture similarity metrics for image analysis and retrieval," *Image Processing, IEEE Transactions on*, vol. 22, no. 7, pp. 2545–2558, 2013.

[21] M. Kudělka and M. Haindl, "Texture fidelity criterion," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, September 2016, pp. 2062 – 2066.

[22] Z. Wang and A. Bovik, "Mean squared error: Lot it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98 – 117, 2009.

[23] D. M. Chandler and S. S. Hemami, "Vsnr: A wavelet-based visual signal-to-noise ratio for natural images," *Image Processing, IEEE Transactions on*, vol. 16, no. 9, pp. 2284–2298, 2007.

[24] H. Sheikh and A. Bovik, "Image information and visual quality," *Image Processing, IEEE Transactions on*, vol. 15, no. 2, pp. 430–444, 2006.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, apr 2004.

[26] Z. Wang and E. P. Simoncelli, "Translation insensitive image similarity in complex wavelet domain," in *In Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP 05). IEEE International Conference on*, 2005, pp. 573–576.