

COLOR TEXTURE SYNTHESIS WITH 2-D MOVING AVERAGE MODEL

Glen Andrews and Kie B. Eom

Department of Electrical Engineering and Computer Science
The George Washington University
Washington, DC 20052

ABSTRACT

An algorithm for synthesizing color textures from a small set of parameters is presented in this paper. The synthesis algorithm is based on the 2-D moving average model, and realistic textures resembling many real textures can be synthesized using this algorithm. A maximum likelihood estimation algorithm to estimate parameters from a sample texture is also presented. Using the estimated parameters, a texture larger than the original image can be synthesized from a small texture sample. In the experiment, various textures suitable for multimedia applications are synthesized from parameters estimated from real textures.

1. INTRODUCTION

Color textures play important roles in multimedia applications. Textures provide rich array of background, diverse surface of objects, visual discrimination for different regions, etc. We suggest an approach for synthesizing color textures from a small number of parameters. This approach can synthesize both stochastic textures and structural textures. The stochastic textures, such as cloud, grass, sand, etc, contain no structural patterns, and the structural textures, such as brick wall, herringbone weave, etc, contain strong structural patterns. The estimation of model parameters are done by the maximum-likelihood (ML) method. Using this modeling approach, parameters are estimated from the real textures, and they can be stored efficiently. Diverse images resembling real textures can be efficiently synthesized from the estimated parameter values. The approach of synthesizing color textures from a sample texture is illustrated by the block diagram in Figure 1.

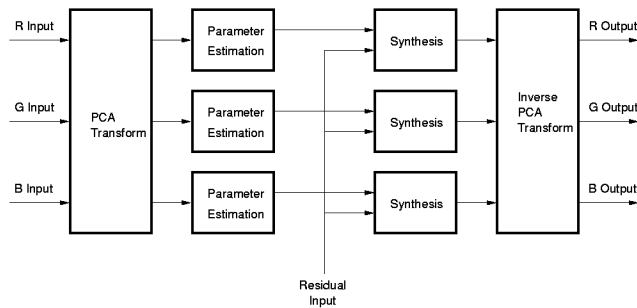


Figure 1: Color texture analysis and synthesis algorithm.

The synthesis approach is based on the two-dimensional (2-D) moving average (MA) model. Under this model, a monochrome image is assumed to be generated by a circular convolution of a point spread function with input excitation process. The input excitation process is assumed as a white Gaussian process for stochastic textures, and a mixture of structured process and white Gaussian process for structured textures. The point spread function is modeled by a linear geometric transform of an isotropic function. The ML estimation of parameters is done efficiently in the frequency domain. The synthesis of each color component is also done in the frequency domain. The synthesis does not require an iterative algorithm, and the MA synthesis algorithm is computationally attractive.

The MA model-based approach is tested with real textures. The real textures are selected from Brodatz [2] texture album and Vision Texture [7] database. In the experiment with monochrome textures, images are synthesized from the white Gaussian process using parameters estimated from textures selected from Brodatz [2] texture album. The synthesized stochastic textures are similar to original textures. In the experiment with color textures, both stochastic textures and structured textures are selected from Vision Texture [7] database. The synthesis of color textures is done with parameters estimated from a small texture. Using the estimated parameter, larger images resembling original textures are synthesized. The color and patterns of synthesized images are similar to the original images.

2. 2-D MOVING AVERAGE MODEL

The 2-D MA model is represented in the spatial domain as follows. Let $\{y(s) \mid s \in \Omega_s\}$, where $\Omega_s = \{s = (s_1, s_2) \in \mathbf{Z} \times \mathbf{Z} \mid 0 \leq s_1, s_2 \leq N-1\}$ be a monochrome image of size $N \times N$ following a 2-D MA model. Then $y(s)$ is represented by the following convolution equation:

$$y(s) = \sum_{r \in K_g} g(r)w(s \ominus r) = g(s) * w(s), \quad (1)$$

where $s, r \in \mathbf{Z} \times \mathbf{Z}$, K_g is a finite index set where $g(s)$ is defined, $\ominus$ is a *modulo* $-(N, N)$ subtraction, $*$ is a 2-D circular convolution on $N \times N$ lattice, and $w(s)$ is a 2-D stationary random process.

With the circular closure (torus) assumption at the boundary, (1) is represented in the frequency domain by applying

DFT to both sides of (1).

$$Y(u) = NG(u)W(u), \quad (2)$$

where $Y(u)$, $G(u)$ and $W(u)$ are DFT's of $y(s)$, $g(s)$ and $w(s)$, respectively.

Without any restriction on the transfer function $G(u)$, the MA model depends on large number of parameters, and the accuracy of estimators will not be satisfactory since there are not enough samples for the estimation. Therefore, the transfer function $G(u)$ needs be parameterized with small set of parameters. For modeling textures having directional and elongated patterns, the transfer function $G(u)$ should also be directional and elongated. For representing such transfer functions with small set of parameters, the transfer function $G(u)$ is assumed to be obtained by a linear geometric transform of an isotropic function. Then $G(u)$ is related to a one-dimensional function $H(\cdot)$ by

$$NG(u) = H(\|Au\|), \text{ where } A = \begin{bmatrix} \alpha \cos \theta & -\sin \theta \\ \alpha \sin \theta & \cos \theta \end{bmatrix} \quad (3)$$

and $\|\cdot\|$ is the Euclidian norm. The geometric transform parameters α and θ are elongation and orientation parameters, respectively. For isotropic textures, the elongation parameter α is unity. Thus (2) can be rewritten as

$$Y(u) = H(\|Au\|)W(u). \quad (4)$$

2.1. Stochastic Texture Modeling

Many natural textures such as sand, pressed cork, grass lawn, etc. do not have strong structural trend, and the orientation or alignment of patterns are random. For modeling such stochastic textures using 2-D MA models, the input process $w(s)$ is assumed as a white Gaussian process with zero mean and unit variance. Then by the Theorem 4.4.1 of Brillinger [1], DFT of the input process $w(s)$, $W(u)$ in (4), is a white complex Gaussian process with zero mean and covariance $\frac{1}{2}\mathbf{I}$, where $\mathbf{I}$ is a 2×2 identity matrix. A complex Gaussian process is defined as a vector process with real and imaginary Gaussian components, and detailed discussions on the complex Gaussian process and the DFT of random processes can be found in [1]. Therefore, the DFT $Y(u)$ of the 2-D MA process $y(s)$ is also a white complex Gaussian random process with zero mean and covariance $\frac{|H(\|Au\|)|^2}{2}\mathbf{I}$, where A is the geometric transform matrix defined in (3).

The model parameters are estimated by a maximum likelihood (ML) approach, and the detailed discussion on ML estimators of the 2-D MA model can be found in [3]. The theoretical properties of ML estimators in 2-D MA model can also be found in [3]. The ML estimators of the 2-D MA model and their properties are summarized in the following. Define $\eta_i = |H(\rho_i)|^2$ and $\mathbf{H} = \{\eta_i \mid i = 1, \dots, n\}$, where $\{\rho_i\} = \{\|Au\| \mid u \in \Omega_u\}$ and n is the cardinality of $\{\rho_i\}$. Let $\mathbf{Y} = \{Y(u), u \in \Omega_u\}$. By defining $\Omega_i = \{u \in \Omega_u \mid \|Au\| = \rho_i\}$ and by defining N_i as the number of elements in Ω_i , the likelihood function of $\mathbf{H}$, θ and α is written

as

$$p(\mathbf{Y}|\mathbf{H}, \theta, \alpha) = \prod_{i=1}^n \prod_{u \in \Omega_i} \frac{1}{\pi \eta_i} \exp\left(-\frac{|Y(u)|^2}{\eta_i}\right). \quad (5)$$

The estimation is done in two stages. In the first stage, the parameters η_i are estimated assuming that the geometric parameters α and θ are known. Then the geometric parameters α and θ are estimated by maximizing the likelihood function in 5. The ML estimator of η_i when the orientation and elongation parameters are known is obtained as

$$\hat{\eta}_i = \frac{1}{N_i} \sum_{u \in \Omega_i} |Y(u)|^2, \quad (6)$$

where Ω_i and N_i are defined in (5). It can be easily shown that the ML estimators $\{\hat{\eta}_i\}$ are unbiasedness and consistent. The accuracy of an estimator can be measured by comparing its mean square error with the Cramer-Rao lower bound. The Cramer-Rao lower bound is derived in [3] and we have the following relation.

$$MSE[\hat{\eta}_i] \geq \eta_i^2 / N_i. \quad (7)$$

Since the ML estimator is efficient, the mean square error of the ML estimator given in (6) asymptotically approaches to the Cramer-Rao lower bound given in (7).

The elongation and the orientation of a texture can be represented by a set of discrete values. For example, eight different orientations are sufficient for many applications, and the orientation parameter θ can be discretized to the finite set $\Theta = \{\frac{k\pi}{8}, k = 0, \dots, 7\}$. Similarly, eight different elongated values are sufficient for many applications, and the elongation parameter α can be discretized to the finite set $\mathcal{A} = \{1, 2, \dots, 8\}$. We need to consider only for $\alpha \geq 1$ since values lower than 1 can be achieved by rotation. Thus, the parameters α and θ can be estimated by maximizing the log-likelihood function $J(\alpha, \theta)$ over the set $\Theta \times \mathcal{A}$. The log-likelihood function of θ and α , $J(\theta, \alpha)$, is derived from (5). The detailed derivation can be found in [3].

$$J(\theta, \alpha) = \sum_{i=1}^n \left[(N_i - 1) \log(\hat{\eta}_i) + \frac{1}{2} \log(N_i) \right] \quad (8)$$

From the above results, we get the following estimation algorithm for transfer function parameters $\{\eta_i\}$, orientation parameter θ and the elongation parameter α .

2.2. Structured Texture Modeling

Structured textures are defined as textures having strong deterministic trend. For example, brick wall, herringbone weave, raffia mat, etc. show structured patterns. Modeling structured textures with a stochastic MA model will not be satisfactory since the structured information is not represented by the stochastic model. For synthesizing structured textures, the input process in the MA model should include deterministic trend representing structured patterns. In addition to the estimation of transfer function parameters, η_i , and geometric parameters α and θ , the information on the structured input process $r(s)$ needs be extracted from the

sample texture where parameters are estimated. The residual in the frequency domain $W(u)$ is computed using (4) with estimated parameters.

$$W(u) = Y(u)/\hat{H}(\|\hat{A}u\|), \quad (9)$$

where $\hat{A}$ and $\hat{H}$ are obtained from the ML estimators.

The structured input process thresholded by the 3σ rule, and pixels with $|W(u)| < 3\sigma$ is replaced by zero. In our experiment, less than 20 percent of residuals are nonzero after thresholding in most of textures. Further, non-zero pixels can be quantized with small number of bits without degrading the quality of synthesized textures. The quantization into 2 bits/pixel is sufficient to synthesize structured textures similar to real textures.

3. COLOR TEXTURE SYNTHESIS

By summarizing the discussions in Section 2, we have the following synthesis algorithm for structured textures. For synthesizing stochastic textures by using the algorithm given below, the estimation of structured trend is unnecessary and a stochastic texture is synthesized by setting the structured trend term by zero.

Texture Synthesis Algorithm

1. Decorrelate the original color image, and obtain three decorrelated components by principal component analysis (PCA).
2. Apply the ML estimation algorithm to each decorrelated component, and estimate $\mathbf{H}$, α and θ .
3. Using the estimated parameters, estimate the structured trend and threshold them by the 3σ rule. The non-zero elements of estimated trend are quantized into 2 bits/pixel.
4. Scale the estimated MA parameters η_i and the estimated deterministic by the ratio between the sizes of the original image and the image to be synthesized.
5. Reconstruct the input residual process by (9) and each decorrelated component is synthesized by (4).
6. Repeat steps 2-5 for three components.
7. Apply the inverse PCA transform to the synthesized image, and a color texture is obtained.

4. EXPERIMENTAL RESULTS

Both stochastic and structured textures are synthesized by the 2-D MA modeling approach. Stochastic textures are defined as those without any deterministic trend, and are synthesized by a 2-D MA model with white Gaussian input process $w(s)$. Structured textures are defined as those having a strong deterministic trend, and are synthesized by a 2-D MA model with a composite input process which is a sum of structured residual process and white Gaussian process.

Monochrome textures are special cases of color textures, and the validity of texture synthesis algorithm can be demonstrated by synthesizing realistic monochrome textures. Figure 2 shows original and synthesized textures. The monochrome

textures shown in the first column of Figure 2 are selected from Brodatz texture album [2], and they are from the top: pressed cork (D04), raffia woven with cotton threads (D51), herringbone weave (D17), and cotton canvas (D77) textures. MA parameters η_i and geometric parameters α and θ are estimated by the ML estimation algorithm given in Section 2. Using the estimated parameters, a texture resembling the original is synthesized by the texture synthesis algorithm given Section 3. The synthesized textures are shown in the second column of Figure 2, and they are similar to the original images in the first column of Figure 2. Note that the textures in the last two rows are synthesized with the estimated structured input process.

For the synthesis of color textures, textures are selected from the Visual Texture [7] database. The original and synthesized textures are shown in Figure 3. The original textures are of size 128×128 , and include both stochastic and structured textures. They are from the top, Fabric15, Food6, Bark10, and Brick7 textures. Top two rows are stochastic textures, and bottom two rows are structured textures. The color, size, and orientations of texture patterns in the original images are all different. To demonstrate the synthesis of larger textures resembling original textures, 256×256 color textures are synthesized from the parameters estimated from the original textures of size 128×128 . The PCA transform is applied to each original texture, and three uncorrelated components are obtained. As explained in the synthesis algorithm in Section 3, the model parameters are estimated and scaled by the factor of 2 for each decorrelated component. Using the estimated and scaled parameters, three monochrome textures corresponding to three decorrelated components are synthesized. The synthesized components in the decorrelated space is transformed back to the RGB color space by applying the inverse PCA transform.

5. REFERENCES

- [1] D.R. Brillinger, Time series, Data Analysis and Theory, Expanded Edition, Holden Day, 1981.
- [2] P. Brodatz, Textures: A Photographic Album for Artist and Designers, Dover Publication, New York, 1966.
- [3] K. B. Eom, "2-D moving average models for texture synthesis and analysis," To appear in IEEE Trans. on Image Processing, December 1998.
- [4] J.M. Francos, A.Z. Meiri, and B. Porat, "A unified texture model based on a 2-D Wold-like decomposition," IEEE Trans. on Signal Processing, vol. 41, no. 8, pp. 2665-2677, 1993.
- [5] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis" Computer Graphics, pp. 229-238, 1995.
- [6] J.J. van Wijk, "Spot noise: Texture synthesis for data visualization," Computer Graphics, vol. 25, no. 4, pp. 309-318, 1991.
- [7] MIT Media Laboratory, Vision Texture, <http://www-white.media.mit.edu/vismod/imagery/VisionTexture/>.

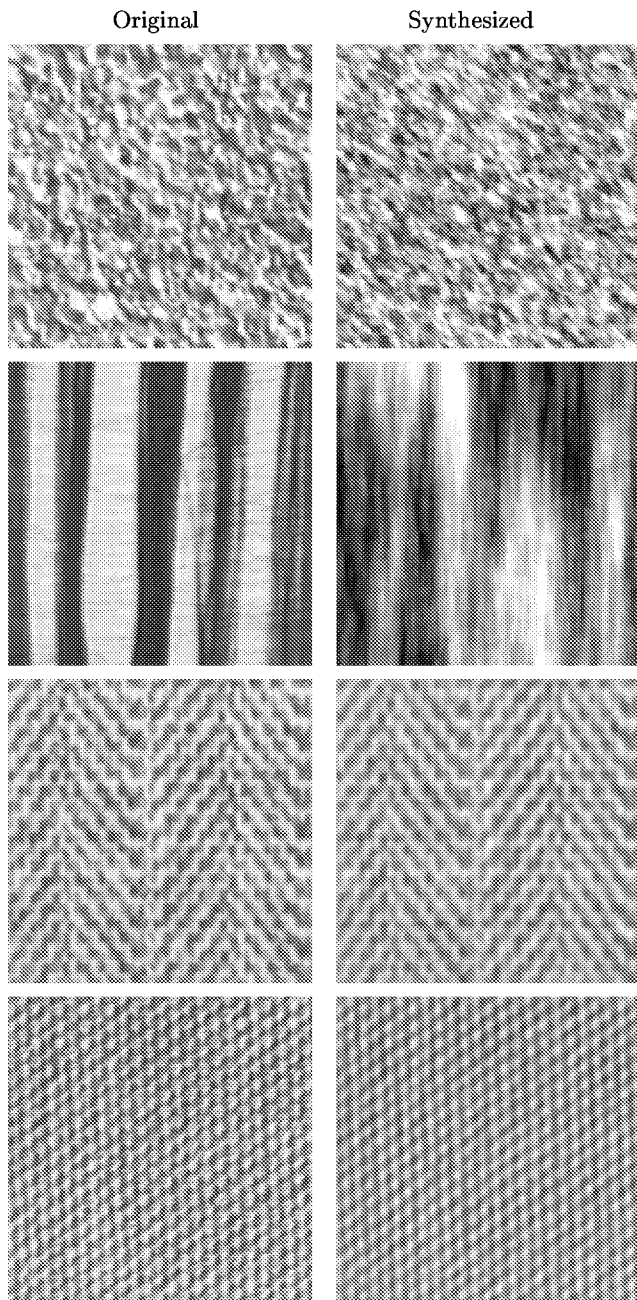


Figure 2: The original (left) and synthesized (right) monochrome textures. From the top, Pressed cork (D04), Raffia woven with cotton threads (D51), Herringbone weave (D17), and Cotton canvas (D77) textures.

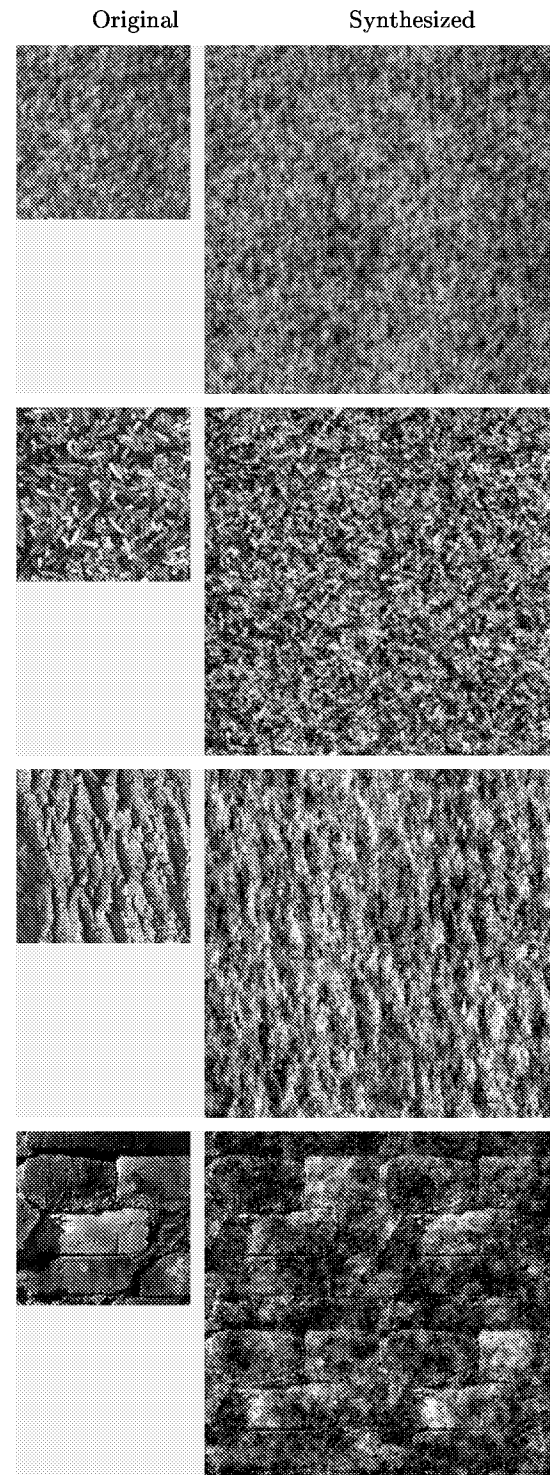


Figure 3: The original (left) and synthesized (right) color textures. They are from the top, Fabric15, Food6, Bark10, and Brick7 textures.