JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 19, 371-380 (2003)

Short Paper

Texture Synthesis: A Review and Experiments*

CHAUR-CHIN CHEN AND CHIEN-CHANG CHEN
Department of Computer Science
National Tsing Hua University
Hsinchu, 300 Taiwan
E-mail: cchen@cs.nthu.edu.tw

Textures have been used in various applications, such as carpet quality control, region recognition in satellite images, and body painting control in the vehicle industry. This paper studies texture synthesis. We survey the algorithms for synthesizing textures that have proposed over the past two decades. A variety of algorithms associated with synthesized textures are given. The goal is to further understand textures, to provide the methodology needed to generate various textures in other studies, and to instigate pursuing better models for synthesizing more interesting textures.

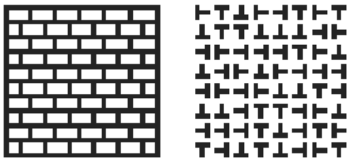
Keywords: autoregressive model, fractal, IFS, Markov random field, SVD

1. INTRODUCTION

Texture is one of the important characteristics of a digital image. Although textures have been widely studied over the past two decades, a precise definition of texture still does not exist [8]. We define a texture as an image containing no explicit objects. There are two categories of textures: structural textures and statistical textures. A structural texture consists of primitives with their relocations based on some replacement rules, such as scaling, rotation, translation, and reflection. Structural textures are frequently used to investigate the relationships among similarity, homogeneity and symmetry in the responses of the human visual system. Two examples of structural textures are given in Fig. 1. Statistical textures are formed based on probabilistic models, such as fractal models [2], time series models [6], and random field models [3].

This paper will focus on statistical textures. Although a variety of algorithms based on generative mathematical models for texture synthesis have been published, many of these works did not report parameters. This paper reviews and reports a variety of experiments on texture synthesis based on mathematical models:

Received March 15, 2001; revised July 11 & September 5, 2001; accepted October 3, 2001.
Communicated by Kuo-Chin Fan.
* This work was partially supported by NSC 89-2213-E-007-131.



(a) (b)

Fig. 1. Structural textures.

(1) the autoregressive (AR) model [6],
(2) the fractal model [2, 9],
(3) the Markov random field (Mrf) model [3, 7],
(4) the matrix decomposition model [14].

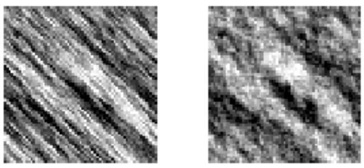
There does not exist a versatile model which can produce any desired texture, and textures generated based on these four models are visually different. The details of these models will be discussed in the following sections.

2. TEXTURES GENERATED BY THE AR MODEL

Let a gray level image of size $N \times N$ be represented by $X(i, j)$, where $0 \le i, j \le N - 1$. Suppose that the first column and the first row of the image X are known, the remainder of the image can be generated based on the 2D autoregressive model [6] by means of the following formula according to lexicographic order:

$$X(i, j) = aX(i, j-1) + bX(i-1, j-1) + cX(i-1, j) + s\varepsilon_{ij},\ \varepsilon_{ij} \sim N(0, 1) \text{ are i.i.d.} \tag{1}$$

Let the first row and the first column of the image X be randomly given, which follow the i.i.d. Gaussian distribution with mean 128 and variance 900. Two synthesized textures with parameters (a) $(a, b, c, s) = (0.1, 0.8, 0.1, 30)$ and (b) $(0.45, 0.1, 0.45, 30)$ are shown in Fig. 2.



(a) (b)

Fig. 2. Textures synthesized using the AR model.

The AR model can generate textures that look like cloud, grass, and/or rain patterns. The parameters a, b, and c affect the directionality of the synthesized texture, and the parameter s affects the spread of the gray values of the texture image. The ranges $a, b, c \in (-1, 1)$ and $s \le 40$ are suggested.

3. TEXTURES GENERATED BY FRACTAL MODELS

Fractal models used to generate such textures as *ferns*, *Sierpinski triangles*, and *snowflakes* have recently received attention in the image compression field [2]. Synthesis is based on the iterated function system (IFS) codes, which are nothing but a set of affine transformations. We list the algorithm and specify some parameters to generate two fractal textures. Let $A \in R^{2\times 2}$ and $\mathbf{t} \in R^2$. An affine transform on $\mathbf{x} \in R^2$ is defined as $A\mathbf{x} + \mathbf{t}$. To describe the algorithm, we denote $A_i = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}$ with $p_i = | a_i d_i - b_i c_i | \neq 0$, and let $\mathbf{t}_i$ denote $\begin{bmatrix} e_i \\ f_i \end{bmatrix}$. Then an algorithm based on IFS codes with K affine transforms is listed below. Experiments conducted using two sets of affine transformations to generate textures are given. The parameters of this fractal model are given in Tables 1 and 2, respectively. Two such synthesized textures are shown in Fig. 3.

Table 1. IFS codes for a fern.

i	a_i	b_i	c_i	d_i	e_i	f_i
1	0	0	0	0.16	0	0
2	0.85	0.04	–0.04	0.85	0	1.60
3	0.20	–0.26	0.23	0.22	0	1.60
4	– 0.15	0.28	0.26	0.24	0	0.44

Table 2. IFS codes for Sierpinski triangles.

i	a_i	b_i	c_i	d_i	e_i	f_i
1	0.5	0	0	0.5	0	0
2	0.5	0	0	0.5	1.0	0
3	0.5	0	0	0.5	0.5	0.5

(a) (b)

Fig. 3. Textures synthesized using IFS codes.

Fractal Generating Algorithm

(1) Set $m = 0$ and randomly pick up an initial point $\mathbf{x}^{(0)} \in R^2$.
(2) Select $\{A_j, \mathbf{t}_j\}$ according to the probability distribution of $\{r_j, 1 \le j \le K\}$, where $r_j = p_j / \sum_{i=1}^{K} p_i$ for $1 \le j \le K$.
(3) $\mathbf{x}^{(m+1)} \leftarrow A_j \mathbf{x}^{(m)} + \mathbf{t}_j$.
(4) $m \leftarrow m + 1$.

(5) Repeat steps 2, 3, 4 until "*convergence*," for example, $m = 1000$, is achieved.
(6) Plot $\{\mathbf{x}^{(i)}\}$ for $i = L$ to 1000, say $L = 100$.

Convergence of this algorithm was studied by Barnsley [2] and Chu and Chen [9]. An IFS code consisting of two to five contractive affine transforms has been suggested [2, 9] which generates self-similar images.

4. TEXTURE GENERATED BY MRF MODELS

Using Markov random fields to synthesize textures is a challenging task. We will review Mrf and give algorithms for synthesizing textures [7, 15].

4.1 Background of the Markov Random Field

Let x, an $M \times N$ texture pattern, be represented as a matrix whose elements take values from the set $A = \{0, 1, ..., G - 1\}$. Let $\Omega = \{x \mid x_t = x(i, j) \in A\}$ be the set of all possible texture patterns, and let $S = \{1, ..., MN\}$ be the sites of a matrix ordered by a raster scan. A Gibbs random field (Grf) is a joint probability mass function defined on Ω which satisfies

$$P(x) = e^{-U(x)} / Z, \tag{2}$$

where $U(x)$ is the energy function and $Z = \sum_{y \in \Omega} e^{-U(y)}$ is the partition function.

A Markov random field is a Gibbs random field whose probability mass function satisfies the following conditions.

(a) *Positivity*: $P(X = x) > 0$ for all $x \in \Omega$.
(b) *Markov property*: For all $t \in S$, $P(X_t = x_t \mid X_r = x_r, r \neq t) = P(X_t = x_t \mid X_r = x_r, r \in R_t)$, where R_t is the ordered set of neighbors of site t.
(c) *Homogeneity*: $P(x_t \mid R_t)$ does not depend on a particular site t.

Fig. 4 defines the relative sites and orders of neighbors of site t. A Grf and an Mrf are equivalent [7] with respect to a specified neighborhood system.

t:-3	t:-2	t:+4
t:-1	t	t:+1
t:-4	t:+2	t:+3

(a)

2	1	2
1	t	1
2	1	2

(b)

Fig. 4. The relative neighboring sites and orders of site t.

A Gibbs random field is completely characterized by its energy function. In this paper, two commonly used Mrf models whose energy functions have the following form are introduced:

$$U(x)=\sum_{t=1}^{MN}F(x_t)+\sum_{t=1}^{MN}\sum_{r=1}^{c}H(x_t,x_{t:+r}), \tag{3}$$

where $H(a, b) = H(b, a)$ and c depends on the size of the neighborhood. For example, $c = 2, 4$ for 1st-order and 2nd-order neighborhoods, respectively. Two Mrf models are defined below [3, 7, 15].

4.1.1 Generalized Ising model (GIM)

Let $A = \{0, 1, ..., G - 1\}$; the F and H functions of (2) in the generalized Ising model are defined as $F(x_t) = \alpha_{x_t}$ and $H(x_t, x_{t:+r}) = \theta_r I(x_t, x_{t:+r})$, where $I(a, b) = -1$ if $a = b$ and $I(a, b) = 1$, otherwise.

Simple derivation gives the conditional density:

$$P(x_t|R_t)=\exp[-\alpha_{x_t}-\sum_{r=-c}^{c}\theta_r I(x_t,x_{t:+r})]/\sum_{s\in A}\exp[-\alpha_s-\sum_{r=-c}^{c}\theta_r I(s,x_{t:+r})]. \tag{4}$$

An algorithm for simulating the generalized Ising model (GIM) is given below. The synthesized textures obtained based on GIM with the parameters $M = N = 128$ and $\theta = (1, 1, 1, -1)$ [7] are shown in Fig. 5(a).

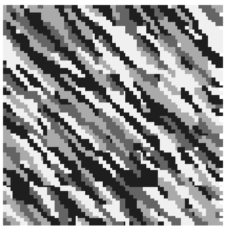


Fig. 5. (a) Textures synthesized using Mrf models.

Algorithm GIM

(1) For s = 1 to MN, randomly assign a $g \in A$ for x_s to give an initial image x.
(2) For s = 1 to MN Do
    (a) Let $y_t = x_t$ for all $t \neq s$. Choose $g \in A$ at random and let $y_s = g$.
    (b) Let $r = min\{1, P(y)/P(x)\}$, where P is as defined in eq. (1).
    (c) $x \leftarrow y$ with probability r.
(3) Repeat step (2) until "*convergence*" is achieved, for example, 50 iterations.

Each of the four parameters of the 2nd-order GIM model is restricted to be between –2 and 2 to avoid the phase-transition phenomenon [7]. In practice, this model assumes that a texture will consist of a small number of gray levels, for example, 8 or less. Each parameter determines a directionality; the larger the negative value of the parameter, the stronger the direction.

4.1.2 Gaussian Markov random field (Gmrf)

A Gmrf was first proposed by Besag as a model for analyzing crop yields in plant ecology [3]. This model has also been used to model natural textures [4, 7]. Let $A = R$; the corresponding F and H functions of Gmrf in eq. (2) are defined as

$$F(x_t) = (x_t - \mu_t)^2/2\sigma^2, \quad H(x_t, x_{t:+r}) = -\theta_r(x_t - \mu_t)(x_{t:+r} - \mu_{t:+r})/\sigma^2. \tag{5}$$

Simple derivation gives the conditional density:

$$P(x_t|R_t) = \frac{1}{2\pi\sigma^2}\exp\left[(x_t - \mu_t - \sum_{-c}^{c}\theta_r(x_{t:+r} - \mu_{t:+r}))^2/2\sigma^2\right]. \tag{6}$$

The distribution of X under the Gmrf model is a multivariate normal distribution [1] with the block circulant covariance matrix B^{-1} given below:

$$f(x) = \frac{|B|^{\frac{1}{2}}}{(2\pi\sigma^2)^{MN/2}}\exp\left[-(x-\mu)^t B(x-\mu)/2\sigma^2\right]. \tag{7}$$

The matrix B is an $MN \times MN$ block circulant matrix with M^2 blocks of N^2 circulant matrices $B'_{ij}s$, defined as

$$B = \begin{bmatrix} B_{11} & B_{12} \ldots B_{1M} \\ B_{1M} & B_{11} \ldots B_{1,M-1} \\ \vdots & \vdots \; \vdots \; \vdots \\ B_{12} & B_{13} \cdots B_{11} \end{bmatrix}. \tag{8}$$

For the 2nd-order neighborhood,

B_{11} = circulant $(1, -\theta_1, 0, \ldots, 0, -\theta_1)$,
B_{12} = circulant $(-\theta_2, -\theta_3, 0, \ldots, 0, -\theta_4)$,
B_{1M} = circulant $(-\theta_2, -\theta_4, 0, \ldots, 0, -\theta_3)$,
$B_{1j} = O$ for $2 < j < M$.

The probability density (7) is valid only if B is positive definite, and it is identifiable if no different parameter sets lead to the same eigenvalues of matrix B. Sampling a

Gmrf is nothing but sampling a multivariate normal distribution [1]. However, in image analysis, the B matrix is of order $MN \times MN$, that is, 16384×16384, when $M = N = 128$, so the traditional method for simulating the multivariate normal distribution based on Cholesky decomposition is infeasible. An algorithm using the properties of block circulant matrices is given below [15]. Let y be an $M \times N$ array, and assign the first row of matrix B of order $MN \times MN$ to an $M \times N$ matrix A by means of $A(i, j) = B(1, j + N \times (i - 1))$. An algorithm for simulating Gmrf by adopting fast Fourier transform (FFT) is listed below. The details can be found in [7, 15]. A synthesized texture based on a Gmrf model with the parameters $M = N = 128$, $\mu = 128$, $\sigma = 64$, and $\theta = (0.07, -0.32, 0.07, 0.12)$ is shown in Fig. 5(b).

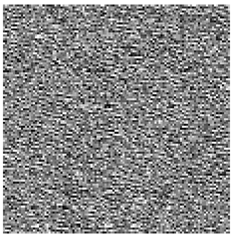


Fig. 5. (b) Textures synthesized using Mrf models.

Algorithm Gmrf

(1) Generate an $M \times N$ array Y with each element $Y(i, j) \sim N(0, \sigma^2)$ being independent.
(2) $Y \leftarrow$ apply 2D FFT on Y.
(3) $A \leftarrow$ apply 2D inverse FFT on A (formed from the first row of matrix B).
(4) $Y(u, v) \leftarrow Y(u,v) / \sqrt{A(u,v)}$, $0 \le u < M$, $0 \le v < N$.
(1) $Y \leftarrow$ apply 2D inverse FFT on Y.
(2) $Y + \mu$ is a realization.

This model is nothing but a multivariate normal distribution with the covariance matrix $\sigma^2 B^{-1}$ being a large block-circulant matrix. The parameter θ affects the directionality, and the parameter σ describes the spread of the gray values. It should be mentioned that the selected parameter θ must make the matrix B positive definite. It seems that this model tends to generate more textures as its order increases. However, the restriction of positive definiteness of matrix B results in the problem of parameter selection.

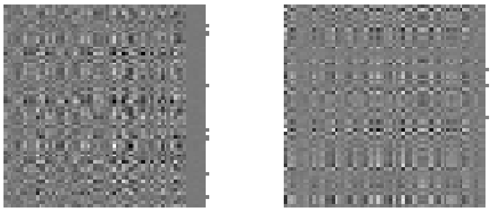
5. TEXTURES GENERATED BY THE MATRIX-DECOMPOSITION MODEL

Matrix-decomposition used to synthesize textures is motivated by image transforms. An M by N gray level image X can be regarded as an M by N matrix. If the matrix can be decomposed as the product of simple matrices, the analysis of texture analysis becomes simpler. We will review an algorithm based on singular value decomposition (SVD) for synthesizing textures [14].

SVD-Based Algorithm

(1) Generate unit vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K \in R^M$.
(2) Generate unit vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K \in R^N$.
(3) Apply the Gram-Schmidt orthogonalization process to $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K \in R^M\}$ and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K \in R^N\}$ to get orthonormal sets, and call them $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K \in R^M\}$ and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K \in R^N\}$, respectively.
(4) Generate $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_K > 0$ according to an exponential distribution with a specified mean value μ, for example, $\mu = 16\sqrt{MN}$.
(5) Get an image $X = \sum_{i=1}^{K} \sigma_i \mathbf{u}_i \mathbf{v}_i^t$.

Two synthesized textures with $M = N = 256$, $K = 16$, and $\mu = 16384$ are shown in Fig. 6. The parameters $\{\sigma_i\}$ and μ are selected based on the fact that the trace, the sum of the diagonal elements in a matrix, is equal to the sum of the eigenvalues which are closely related to singular values. The synthesized patterns look like the surface of an IC chip.



(a) (b)

Fig. 6. Textures synthesized using the SVD method.

6. DISCUSSION

We have reviewed four statistical texture synthesis models and given examples. Structure textures are formed by selecting a texton and then placing this texton in a position based on scaling, translation, and/or rotation. The models introduced in this paper are essentially statistical models, which are not appropriate for synthesizing structure textures although the GIM random field model, with parameters carefully specified, may generate those structure textures used in psychological tests. Using the other models to synthesize structure textures is inappropriate.

The AR model can produce natural textures like clouds and is easy to implement. However, inappropriate selection of the initial gray values of the pixels in the top row and the leftmost column, and of the parameter s may result in an image in which most of the pixels have very large gray values (much greater than 255). The restrictions $s \leq 40$ and a, b, $c \in (-1, 1)$ are suggested. The inverse problem of finding the parameters given a texture can be solved using a least squares method. The AR model parameters can be used to discriminate textures even when the model does not fit the image data.

The fractal model based on an IFS code generates a self-similar texture. In practice, an IFS code composed of two to five contractive affine transforms is suggested [2, 9]. The parameters must be selected such that each affine transform is contractive. The inverse problem of finding the number of affine transforms associated with their corresponding parameters based on a given image has been well studied and used to compress images [2].

The Markov random field model can be used to synthesize textures and discriminate textures. Given model parameters, we can generate textures like clouds, maze, and grass. On the other hand, by fitting a GIM or Gmrf model to a given texture image, we can get the model parameters as the texture features for discrimination although the model may not fit the image data [7].

The matrix decomposition method exploits the fact that an image is essentially a matrix. SVD treats an image, a matrix, as a product of three matrices, among which the matrix consisting of singular values characterizes the original matrix (image). Selecting parameters $\{\sigma_i\}$ as singular values can produce a texture that looks like the surface of an IC chip. This method can also be applied to perform texture image compression and discrimination for a limited number of textures.

7. CONCLUSION AND FUTURE STUDIES

We have reviewed several mathematical models for texture synthesis. Algorithms for texture synthesis based on models associated with generated textures have also been given. Each model seems to synthesize visually different textures. The matrix decomposition method based on SVD has recently been investigated, [14] but more experimental works are needed. A future direction in SVD decomposition research may be to search for orthogonal matrices established based on sinusoidal functions, such as FFT or DCT, or for orthogonal matrices derived from a wavelet transform [5, 12]. Another matrix decomposition approach to texture analysis exploits nearly orthogonal matrix transform, such as Gabor transform, which is believed to be similar to the response of the human visual system [10, 11, 13] and has recently been investigated with regard to texture discrimination. Texture synthesis by means of Gabor transform and wavelet transform, using the concept behind the matrix decomposition approach also merits further study.

REFERENCES

1. T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, John Wiley, 1984.
2. M. F. Anderson and L. P. Hurd, *Fractal Image Compression*, Addison Wesley, 1993.
3. J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of Royal Statistical Society, Serial B*, Vol. 36, 1974, pp. 192-236.

4. C. C. Chen and C. C. Chen, "Filtering methods for texture discrimination," *Pattern Recognition Letters*, Vol. 20, 1999, pp. 783-790.
5. G. C. H. Chuang and C. C. J. Kuo, "Wavelet descriptor of planar curves: Theory and applications," *IEEE Transactions on Image Processing*, Vol. 5, 1996, pp. 56-70.
6. E. J. Delp, R. L. Kashyap, and O. R. Mitchell, "Image data compression using autoregressive times series models," *Pattern Recognition*, Vol. 11, 1979, pp. 313-323.
7. R. C. Dubes and A. K. Jain, "Random field models in image analysis," *Journal of Applied Statistics*, Vol. 16, 1989, pp. 131-164.
8. R. M. Haralick and L. G. Shapiro, *Computer and Robert Vision*, Volume I, Addison Wesley, 1992.
9. H. T. Chu and C. C. Chen, "A fast algorithm for generating fractals," in *Proceedings of IEEE International Conference on Pattern Recognition*, Vol. 3, 2000, pp. 306-309.
10. A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, Vol. 24, 1991, pp. 1167-1186.
11. T. R. Reed and H. Wechsler, "Segmentation of textured images and gestalt organization using spatial/spatial-frequency representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, 1990, pp. 1-12.
12. G. Strang, "Short wavelets and matrix dilation equations," *IEEE Transactions on Signal Processing*, Vol. 43, 1995, pp. 108-115.
13. H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 8, 1978, pp. 460-470.
14. C. C. Chen and J. H. Luo, "Textural discrimination using singular value decomposition," in *Proceedings of Internatioal Conference on Applied Modeling and Simulation*, 1993, pp. 75-77.
15. C. C. Chen and C. C. Chen, "Texture synthesis: algorithms and experiments," in *Proceedings of the 9th Chinese Image Processing and Pattern Recognition Conference*, 1996, pp. 31-37.

Chaur-Chin Chen (陳朝欽) received the B.S. degree in Mathematics from National Taiwan University, Taipei, in 1977, the M.S. degrees in both Mathematics and Computer Science, and the Ph.D. degree in Computer Science, all from Michigan State University, East Lansing, Michigan, in 1982, 1984, and 1988, respectively. He worked as a research associate at MSU from January to June in 1989 and has been an Associate Professor in Department of Computer Science at National Tsing Hua University since August, 1989. He was a visiting scholar at MSU from September to December in 1997. His current research interests are in the areas of texture analysis, biometrics, and information concealment.

Chien-Chang Chen (陳建彰) received the B.S. degree from Department of Computer and Information Science at Tung Hai University, Taichung, in 1991, and the Ph.D. degree in Computer Science from National Tsing Hua University in 1999. He has been an Assistant Professor in Department of Information Management at Hsuan Tzarn College.