Rep. Fac. Sci. Engrg.
Saga Univ
35-2 (2006), 1~10

Reports of the Faculty of Science and Engineering,
Saga University, Vol. 35, No.2, 2006

Texture Analysis Using Gaussian Markov Random Fields

By
Tingting Cao* Hisao Tokushima* Yoshio Noguchi*

Abstract : This paper shows experimental results of texture analysis based on the Gaussian Markov random fields (GMRF) model. The GMRF parameters (mean, variance, and autocorrelation function) were estimated based on the maximum likelihood method using a simulated annealing (SA) technique. The SA technique maximizes the likelihood function of the Fourier transform of the real image. The estimated parameters were used for synthesizing artificial GMRF images which look quite similar to the originals. The developed C-language programs for texture analysis and synthesis using the GMRF model can be applied to any textual images. This paper reports essense of the master thesis of Tingting Cao.

Key words : texture, texture synthesis, Gauss-Markov random field, simulated annealing

1 Introduction

Texture is observed in the structural patterns of surfaces of objects such as wood, grain, sand, and cloth. The term texture generally refers to repetition of basic texture elements called *texels*. Natural textures are generally random, whereas artificial texture are often deterministic or periodic. Texture may be coarse, fine, smooth, granulated, regular, or linear. In an image analysis, texture is classified into two main categories, statistical and structural. For statistical analysis an image is specified by average properties. For structural analysis, shape and character of resolution cells are considered, and then the positions of these resolution cells are determined. Texture analysis is an important and useful area of machine vision. One typical application is recognition of image regions using the different texture properties in them. This is called texture classification [6].

An image is considered to be a sample function of an array of random variables called *a random field*. A two-dimensional random field is called *Markov* if at every pixel location we can find a outside partition(*future*), a bundary partition(*present*) and inside partition(*past*) of two-dimensional lattice set { (m,n) }. Every Gaussian noncausal random field is a Markov random field(MRF) [2, 7, 8].

In 1956, P. Lévy proposed a Markovian random field. In 1972, J.W. Woods defined a Markov random field on a discrete space and led a two-dimensional difference equation on the space [10]. In 1974, B.H.McCormick and S.N. Jayaramamurthy done Texture synthesis based on a time series model. In 1983, G.R.Cross and A.K.Jain showed texture analysis based on the Markov random field models [4]. In 1984, S. Geman, D. Geman used stochastic relaxation and annealing techniques for computing the maximum *a posteriori* estimate of an image [5]. In 1985, R. Chellappa, S. Chatterjee done texture classification based on Gaussian Markov random fields(GMRF) model [1]. In 1991, F. S. Cohen et al. extended the the GMRF model thory and classified rotated and scaled textures [3].

Received Nov1,2006
* Department of Electrical and Electronic Engineering

2 Textures

There is not a comprehensive and precise definition of texture of images. We recognize texture when we see it but it is not easy to define. There are many different definitions of textures. For example, two different definitions are given below.

" A region in an image has a constant texture if a set of local statistics or other local properties of the picture are constant, slowly varying, or approximately periodic."

"The notion of texture appears to depend upon three ingredients: (i) some local 'order' is repeated over a region which is larger in comparison to the order's size, (ii) the order consists in the nonrandom arrangement of elementary parts, (iii) the parts are roughly uniform entities having approximately the same dimensions everywhere in the textured region."

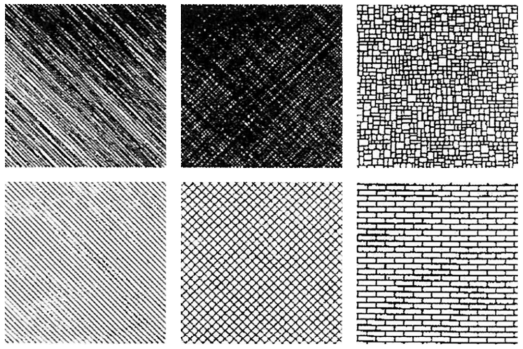


Fig.1 Artifical texture

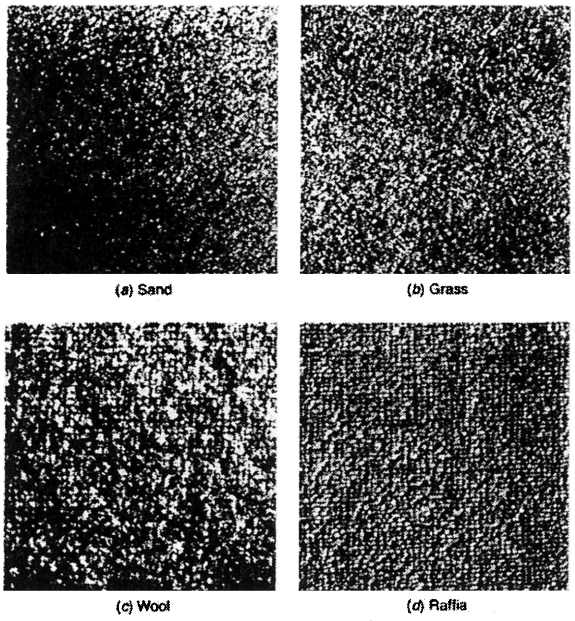


Fig.2 Natural texture

In general "texture" means repeated mode of intensity change in some region of an image. Formation mechanism of texture is changed on some area of an image.

3 Gaussian Markov Random Fields

In stochastic representation, an image can be considered to be a sample function of an array of random variables called a random field. In the discrete Gaussian Markov random fields models, the gray value of any pixel can be modeled and given by:

$$f(i,j) = \sum_{(k,j)} f(i-k, j-l)h(k,l) + n(i,j), \quad (1)$$

where $\sum$ is defined at a neighborhood set of piexls $\{ (i,j) \}$. $f(i-k,j-l)h(k,l)$ is a linear combination of the gray value on the neighborhood, $h(k,l)$ is a weight coefficient at (k,l) , and $n(i,j)$ is a linear noise.

Random field:

Z is a lattice set on a two-dimensional real plane, that is, $Z = \{(i,j)| -(N/2-1) \leq i,j \leq N/2\}$, where i,j are integers, N is an even integer. Let R_S be a random space, $\lambda = \{1,2,\ldots,l\}$ be a parameter set, and $S = \{0,1,2,\ldots,M\}$ be state space (a set of gray-levels) . Let $X_{ij} : R_S \rightarrow S$; for $i,j \in \lambda$ be a random variable, then $X = \{X_{ij} : i,j \in \lambda\}$ is called a random field on Z .

Neighborhood set:

For a given parametr set λ, if $D = \{D_{i,j} \subset \lambda \times \lambda : i,j \in \lambda\}$ satisfy:

(1) $(i,j) \notin D_{i,j}$,

(2) if $(i,j) \in D_{i_2,j_2}$ then $(i_2,j_2) \in D_{i,j}$,

then D is called a neighborhood set on $\lambda \times \lambda$. $D_{i,j}$ is called the neighbors of (i,j). The c-neighborhood $D_{i,j}^{(c)}$ is defined by $D_{i,j}^{(c)} = \{(k,l)|(k,l) \in Z, 0 \leq (k-i)^2 + (l-j)^2 \leq c\}$.

For $c = 1$,

$D_{i,j}^{(1)} = \{(i-1,j)(i+1,j),(i,j-1)(i,j+1)\}$.

For $c = 2$,

$D_{i,j}^{(2)} = \{(i-1,j-1),(i,j-1),(i+1,j-1),$
$(i-1,j),(i+1,j),(i-1,j+1),$
$(i,j+1),(i+1,j+1)\}$.

Connected set:

Let D be a neighborhood set, if $C = \{C_a : a \in \lambda\}$ satisfy:

(1) $C_a \subset \lambda \times \lambda$

(2) for all the (i_1,j_1), (i_2,j_2) $\in C_a$,
there is $(i,j) \in D_{i_2,j_2}$,

then C is called a connected set about the D.

GMRF about neighborhood set:

Let $X = \{X_{i,j} : i,j \in \lambda\}$ be a random field, and D be a neighborhood set. Let $X_{i,j}$ be a sample from a Gaussian distribution. For all the $(i,j) \in \lambda \times \lambda$, the conditional probability satisfies the equation below:

$$p\{X_{ij} = x_{ij} | X_{kl} = x_{kl},\ (k,l) \neq (i,j)\}$$
$$= p\{X_{ij} = x_{ij} | X_{kl} = x_{kl},\ (k,l) \in D_{i,j}\}.$$

X is called GMRF about the neighborhood set D .

4 GMRF Models of Textures

4.1 Power Spectral Density of Image

Let $g_0(m,n)$ be the intensity of an image at pixel (m,n) . The GMRF is stationary non-causal two-

Table 1 Neighborhood D_p of the pixel labeled " × " for different orders p of the GMRF model. The numbers indicate the order of the model relative to " × " .

9	8	7	6	7	8	9
8	5	4	3	4	5	8
7	4	2	1	2	4	7
6	3	1	×	1	3	6
7	4	2	1	2	4	7
8	5	4	3	4	5	8
9	8	7	6	7	8	9

dimensional autoregressive process described by the following equation:

$$g_0(m,n) = \mu + \sum_{k,l \in D_p} \beta(k,l)[g_0(m-k,n-l)-\mu] + \nu(m,n), \quad (2)$$

where $\{\beta(k,l)\}$ are parameters, μ is the mean of $g_0(m,n)$, and D_p is a neighborhood set. The pixels which enter D_p can be arbitrarily defined, however it is natural to consider only neighbors which are spatially close to the pixel at (m,n). Neighborhood sets D_p for $p \leq 9$, where p is the order of the model, are shown in **Table 1** .

$$R_\nu(k,l) = \begin{cases} \sigma^2, & if \ (k,l) = (0,0) \\ -\sigma^2\beta(k,l), & if \ (k,l) \in D_p \\ 0, & otherwise. \end{cases} \quad (3)$$

The symmetry property $\beta(k,l) = \beta(-k,-l)$ follows from the symmetry of the autocorrelation function associated with $g_0(m,n)$.

$$R[g_0(k,l)] = R[g_0(-k,-l)]. \quad (4)$$

The GMRF is parametrized by a relatively small parameter set $\gamma = (\mu, \sigma^2, \beta)$. $\beta = (\beta_{1,0}, \beta_{0,1}, \beta_{1,1}, \beta_{1,-1}, ...)$. The power spectral density (PSD) associated with $g(m,n) = [g_0(m,n) - \mu]$ is given by

$$S_g(\Omega_1,\Omega_2) = \sigma^2 / \left\{ 1 - \sum_{k,l \in D_p} \beta(k,l) \exp[-i(k\Omega_1 + l\Omega_2)] \right\}, \quad (5)$$

where $i = \sqrt{-1}$.

Let $\mathbf{G} = \{G(m,n) \mid -(N/2-1) \leq m,n \leq N/2\}$ be the two-dimensional discrete Fourier Transform (DFT) of $\{\mathbf{g}_0 - \mathbf{U}\}$, where $\mathbf{g}_0$ an image of $N \times N$, and $\mathbf{U}$ is the diagonal matrix of the mean μ . The discrete Fourier transform $G(m,n)$ is defined as

$$G(m,n) = \sum_{k=-N/2}^{N/2-1} \sum_{l=-N/2}^{N/2-1} [g_0(k,l) - \mu] \times \exp\{\{-\sqrt{-1}(2\pi/N)[mk+nl]\}\}. \quad (6)$$

4.2 GMRF Parameter Estimation

To estimate the parameter set $\gamma = (\mu, \sigma^2, \beta)$ of the GMRF, the maximum likelihood estimates (MLE) were calculated. The method is very powerful for estimating parameters of probability functions. The likelihood function of the $N \times N$ image $\mathbf{g}$ is given by

$$p(\mathbf{g} \mid \gamma) = \prod_{-(N/2-1) \leq m,n \leq N/2} (1/2\pi N^2 S_g(m,n))^{1/2} \times \exp\left\{ - \sum_{-(N/2-1) \leq m,n \leq N/2} |G(m,n)|^2 / 2N^2 S_g(m,n) \right\}. \quad (7)$$

The discrete power spectral density $S_g(m,n)$ is given by evaluating the continuous power spectrai density $S_g(\Omega_1, \Omega_2)$ shown in Eq.5 at $\Omega_1 = 2\pi m/N$, and $\Omega_2 = 2\pi n/N$.

4.3 Simulated Annealing Technique

Simulated Annealing (SA) is a stochastic search method in which a randomly selected purturbation to the current configuration is acceped or rejected probabilistically. The optimal answer could be found by simulating annealing process.

Let $S = \{S_1, S_2, ...S_n\}$ be a set of all possible state. $C : S \rightarrow R$ is a nonnegative function, $C(S_i) \geq 0$, where R is a real field. It shows the condition to take S_i as an answer, then the combinatorial optimization problems could be expressed:

For $S^* \in S$, let $C(S^*) = min\{C(S_i)\} \quad \forall S_i \in S$.

The essential idea of SA method is: S_i is regarded as a microcosmic state of some material system, $C(S_i)$ is regarded as the intrinsic energy of the material system in S_i state. In simulated annealing, a parameter of temperature T is used. Let T falls slowly from a enough high temperature. For every T, heat equilibrium state of the system at T is simulated on the computer using the Metropolis sampling [9]. The current state S is stirred and a new state S' would be produced, then the increment is computed: $\Delta C' = C(S') - C(S)$. S' is accepted as a new state by probability $\exp(\Delta C/kT)$. The process is repeated for enough times. The probability of the state S_i is defined by

$$p(S_i) = Z(T)^{-1} \exp(-C(S_i)/kT), \quad (8)$$

where $Z(T) = \sum_i \exp\{-C(S_i)/kT\}$ (the partition function) and k is the Boltzmann constant.
If T falls very slowly, and $T \rightarrow 0$, the current state would has the minimum $C(S_i)$.

This idea could be expressed as algorithms below, including the Metropolis sampling and the Realize annealing process.

(1) Realize annealing process algorithm

1. Select an initial state S_0 arbitrarily, $S(0) = S_0$. Set an initial temperature T_0 and let i be zero; $i = 0$.
2. Let $T = T_i$, the Metropolis sampling method is applied by T and S_i, return the current answer that is received at last as the current answer $S_i = S$.
3. Fall temperature T by a certain way, that is , $T = T_{i+1}$; $T_{i+1} < T_i$.
4. Check if the annealing process has finished. If the process had finished, go to 5: otherwise, go to 2.
5. Output the current answer as the optimal answer.

(2) Metropolis sampling

1. Let $S(0) = S$ be the current answer when $k = 0$. The next steps will be done in T condiction.
2. On the state S of the current answer $S(k)$, a new state S' is a candidate answer of the next current answer by computing $\Delta C' = C(S') - C(S(k))$. If $\Delta C' < 0$, then S' is accepted as the next current answer with probability $\exp(-\Delta C'/kT)$. If S' was accepted, then let $S(k+1) = S'$, otherwise, let $S(k+1) = S(k)$.
3. Increase k ; $k = k+1$. Check if the method has finished or not by some convergent criterion. If yes, go to 4, otherwise, go to 2.
4. Return the current answer $S(k)$ to the annealing algorithm.

4.4 Texture Synthesis

Let
$R_0 = \{(m,n) : -(N/2-1) \leq m,n \leq N/2\}$,$R_{1R} = \{(m,n) : (0,0),(0,N/2),(N/2,0),(N/2,N/2)\}$,$R_1 = \{(m,n) : 0 \leq m,n \leq N/2\} \bigcup \{(m,n) : -(N/2-1) \leq m \leq -1, 1 \leq n \leq N/2-1\}$, $R_2 = R_1 - R_{1R}$.
The discrete Fourier transform of $g(m,n)$ is given below.

1) $(m,n) \in R_2$, $G(m,n)$ is generated by randomly selecting $G(m,n)$ from a complex circulary symmetric Gaussian random number generator with zero mean and variance $N^2 S_g(m,n)$, where $S_g(m,n)$ is given in Eq.5 at $\Omega_1 = 2\pi m/N$, and $\Omega_2 = 2\pi n/N$.

2) $(m,n) \in R_{1R}$, $G(m,n)$ is generated by randomly selecting $G(m,n)$ from a real Gaussian random number generator with zero mean and variance $N^2 S_g(m,n)$.

3) $(m,n) \in R_0 - R_1$, $G(m,n)$ is determind completory from $G(m,n)$ for $(m,n) \in R_1$ because of the symmetry property of the DFT.

The inverse discrete Fourier transform (IDFT) of **G** is taken first, after that the μ is added at each pixel point. Then the required texture image based on the MRF is given.

5 Experimental Results

5.1 Objectives

This is the first simple test and shows whether the parameters have been estimated correctly. The other test is to synthesize textures using the estimated parameters and to compare them with the real textures. The models can be considered correct if the syntesized textures using the extracted parameters look exactly like the input images. To estimate statistical parameters for real textures and to generate artificial textures using the obtained parameters , two programs were writen in C language. The first program estimates the parameters of images using the GMRF model and the second one synsesizes textured images using the parameters. The program for extracting the parameters was made to use a sample image of small size, usually 32×32 or 64×64, from a much larger image. Since larger images need a significant computational burden and they are not neccessary.

5.2 Program for GMRF Parameter Estimation

The program for estimating the GMRF parameter set $\gamma = (\mu, \sigma^2, \beta)$ from an image was developed and applied to real photographs of textures. The program uses the maximum likelihood estimate (MLE) method for obtaining γ [3] . The program started by selecting a sample image from the larger black & white input image with size 512×512 pixels. The sample image was selected by taking a piece of size $N \times N$ (N is a power of two, typically 64 or 128) so that its top left corner was at the point *(CUTX,CUTY)* of the input image. This image was placed in the array $g0[N][N]$. Then the average intensity (μ) of the sample image was calculated and subtracted from each pixel, resulting in a zero-mean image, placed in the array $g[N][N]$. At that point the image variance (σ^2) was calculated as well.

The next step of the program is to calculate the forward DFT of the sample image. This is done using the routine for n-dimensional FFT *fourn* described in *Numerical Recipes in C : Cambridge University Press* . This routine converts an n-dimensional complex array, represented as an 1-D array, into its Fourier transform. The array $g[N][N]$ is placed into the 1-D array $cplg[2N^2]$.

After the end of the FFT routine, output of the *fourn* is placed again in the same array $cplg[2N^2]$. This array starts with zero frequency of the spectrum, continues to the highest positive frequency (at index N), then all the negative frequencies follow, starting with the highest to the lowest one at index $2N$. This is repeated N times, corresponding to all the columns of the original $N \times N$ array.

The resulting two arrays $Re[N][N]$ and $Im[N][N]$ have the same placement of the Fourier spectrum $G(m,n)$ as that required in [3]:

$$G(m,n) = \sum_{k=-N/2}^{N/2-1} \sum_{l=-N/2}^{N/2-1} g(k,l) \exp\{-i(2\pi/N)[mk+nl]\} . \quad (9)$$

In (9), $i = \sqrt{-1}$ and $G(m,n) = Re[m][n] + iIm[m][n]$. Further, the program calculates the likelihood function $p(\mathbf{g}|\gamma)$, given as

$$p(\mathbf{g} \mid \gamma) = \prod_{m,n=-(N/2-1)}^{N/2} \left(\frac{1}{2\pi N^2 S_g(m,n)}\right)^{1/2} \times \exp\left(-\sum_{m,n=-(N/2-1)}^{N/2} |G(m,n)|^2/2N^2 S_g(m,n)\right), \quad (10)$$

where

$$|G(m,n)|^2 = (Re[G(m,n)])^2 + (Im[G(m,n)])^2.$$

The discrete power spectral density $S_g(m,n)$ is given by

$$S_g(m,n) = \sigma^2 / \left\{1 - 2 \sum_{k,l \in D_p'} \beta(k,l) \cos[(2\pi/N)(mk+nl)]\right\}, \quad (11)$$

where D_p' is the non-symmetric half-plane of the neighborhood set D_p shown in **Table 2** . The set D_p can be used instead of D_p' because of the symmetry of the parameters β : $\beta(k,l) = \beta(-k,-l)$. This reduces the number of independent parameters. The number of parameters β as a function of the order of the model can be calculated and is given in **Table 3** .From symmetry consideration there are 8 possible ways in which the set D_p' can be selected from D_p. The set D_p' used here consists from the lower left part of the indices of D_p, as shown in **Table 4** .

Table 2 Neighborhood D_p of the pixel labeled " $\times$ " for different orders of the GMRF model. The index shows the number of elements in each order. Elements with the same index have equal values.

9_1	8_2	7_2	6_2	7_3	8_3	9_2
8_1	5_1	4_2	3_2	4_3	5_2	8_4
7_1	4_1	2_1	1_2	2_2	4_4	7_4
6_1	3_1	1_1	$\times$	1_1	3_1	6_1
7_4	4_4	2_2	1_2	2_1	4_1	7_1
8_4	5_2	4_3	3_2	4_2	5_1	8_1
9_2	8_3	7_3	6_2	7_2	8_2	9_1

Table 3 Number of parameters of the GMRF model as a function of its order. The additional 2 parameters except $\beta(k,l)$ are σ^2 and μ.

Order	Number of parameters
1	2 + 2
2	4 + 2
3	6 + 2
4	10 + 2
5	12 + 2
6	14 + 2
7	18 + 2
8	22 + 2
9	24 + 2

The MLE of the GMRF parameter set γ is obtained by maximizing (10) with respect to γ. In case of 9^{th} order GMRF 25 parameters have to be obtained because $\mu = 0$, since we use images with zero mean. The maximization of $p(\mathbf{g} \mid \gamma)$ can be reduced to the

Table 4 One of the possible neighborhoods D_p' of the pixel labeled " $\times$ " for different orders of the GMRF model.

9_1	8_2	7_2	
8_1	5_1	4_2	
7_1	4_1	2_1	
6_1	3_1	1_1	$\times$
7_4	4_4	2_2	1_2
8_4	5_2	4_3	3_2
9_2	8_3	7_3	6_2

simpler task of maximizing its logarithm, given as

$$\ln[p(\mathbf{g} \mid \gamma)] = \frac{N^2}{2} \ln \frac{1}{2\pi N^2} + \frac{1}{2} \sum_{m,n=-(N/2-1)}^{N/2} \ln\left(\frac{1}{S_g(m,n)}\right) - \frac{1}{2N^2} \sum_{m,n=-(N/2-1)}^{N/2} \frac{|G(m,n)|^2}{S_g(m,n)}. \quad (12)$$

The first term in (12) is a constant and can be omitted. The value of $S_g(m,n)$ must be positive, because it enters the denominator and is the argument of the logarithm in (12). However, the value of $S_g(m,n)$ can change its sign as the $\beta(k,l)$ vary, therefore only positive $S_g(m,n)$ should be allowed to enter (12). If $S_g(m,n)$ is not positive for certain indices (m,n) the subroutine for calculating (12) returns a value much smaller than the typical $\ln[p(\mathbf{g} \mid \gamma)]$. In this way non-positive $S_g(m,n)$ are excluded from the calculation.

The maximization procedure applied in the program uses the Simulated Annealing (SA) technique as a main algorithm. This is required because the logarithmic function of Eq.(12) has many local maxima and its maximization by other methods are difficult. The SA is much more robust and can handle to a large extent problems with many local maxima.

To make the conversion faster, a simple **Golden Section Search procedure** for maximization to only one variable is used in the main SA loop. The Search is a very reliable method for finding the global maximum of a function which can have many local maxima. The range in which ***golden1*** varies $\beta(k,l)$ in search for the maximum has been limited to $[-1,1]$. The algorithm is as follows:

1. Read initial approximations for the σ^2 and β from a file. If the file does not exist, the program put $\beta = 0$.

2. Maximize $\ln[p(\mathbf{g} \mid \gamma)]$ with respect to σ^2, using the initial β. The subroutine for that is *golden2*.

3. Randomly choose two elements (k,l) and (m,n) from the neighborhood D_p', where $(k,l) \neq (m,n)$. Store their values in the temporary variables a and b.

4. Add to $\beta(m,n)$ a random number in the range $(-scale, scale)$, where $scale < 1$.

5. Maximize $\ln[p(\mathbf{g} \mid \gamma)]$ with respect to $\beta(k,l)$, using the procedure *golden1*. This gives a new value for $\beta(k,l)$.

6. Compare the calculated $\ln[p(\mathbf{g} \mid \gamma)]$ with that from the previous iteration.
   If a new maximum has been achieved, store $\beta(m,n)$ and $\beta(k,l)$ as new estimations.
   If the new $\ln[p(\mathbf{g} \mid \gamma)]$ is not larger than the old maximum, it still can be accepted because of the method of SA. We use the **Metropolis procedure**, which compares a random number in the range $[0,1)$ with the probability $\exp(-de/T)$, where

$$de = \ln[p(\mathbf{g} \mid \gamma)]_{old} - \ln[p(\mathbf{g} \mid \gamma)]_{new} \quad (13)$$

   and T is a parameter, equivalent to temperature in real-world physical systems. Let $rand[0,1)$ be a generated random number. If

$$rand[0,1) < \exp(-de/T),$$

   then the $\ln[p(\mathbf{g} \mid \gamma)]_{new}$ is accepted as new estimation. The probability of accepting higher de decreases exponentially and can be controlled by the parameter T.

   As the number of iterations progresses, T is decreased so that accepting big values of de becomes less probable. The Metropolis procedure is implemented in the subroutine ***metrop(de,t)*** described in the *Numerical Recipes in C*.

7. If the new value of $\ln[p(\mathbf{g} \mid \gamma)]$ has been accepted, maximize again $\ln[p(\mathbf{g} \mid \gamma)]$ with respect to σ^2 using the new values of $\beta(m,n)$ and $\beta(k,l)$. If not, restore $\beta(m,n)$ and $\beta(k,l)$ to their original values a and b.

8. Check the number of times each element β from the neighborhood D_p' has changed so far. If the number of changes exceeds certain limit, exit the loop and decrease *scale* and T according to the annealing schedule. Smaller *scale* makes the deviations of $\beta(m,n)$ smaller until the desired precision is achieved.

9. Repeat steps 3 to 8 until the procedure cannot change $\ln[p(\mathbf{g} \mid \gamma)]$ any more.

10. Write the obtained parameters σ^2 and β to file. The SA reqires some tuning of the annealing schedule and the rate with which the parameter *scale* decreases with the number of iterations. This is done experimentally.

The program can handle GMRF models of any order, provided that the user writes a *mask* file giving the correct neighborhood D_p'. Those can be easily obtained from Table 2 and Table 4 . For example, the contents of the *mask* files for GMRF models with

Table 5 Mask files for GMRF models with orders from 9^{th} to 4^{th}.

9^{th} order

1	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0
1	1	1	1	0	0	0
1	1	1	1	0	0	0

8^{th} order

0	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0
1	1	1	1	0	0	0
0	1	1	1	0	0	0

7^{th} order

0	0	1	0	0	0	0
0	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	1	1	0	0	0

6^{th} order

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	1	0	0	0	0
1	1	1	0	0	0	0
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	1	0	0	0

5^{th} order

0	0	0	0	0	0	0
0	1	1	0	0	0	0
0	1	1	0	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
0	1	1	1	0	0	0
0	0	0	0	0	0	0

4^{th} order

0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	1	0	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

order from 9^{th} to 4^{th} are given in **Table 5** . The program files the test image and its Fourier spectrum $Re[N][N]$ and $Im[N][N]$ in ASCII format.

The program ***gmrf_extract.c*** runs in Linux environment and it takes significant amount of computer time to execute. The FFT of the test image is very fast, however the maximization of the likelihood function is the most time-consuming task. Because the likelihood function has many local maxima. We are quite confident that the parameters found correspond to the global maximum.

5.3 Test Images and Estimated Parameters

Two test images were used for obtaining their GMRF parameters. Both are close-up photographs of handkerchiefs, shown on **Fig.3** and **Fig. 4** . The images have 512×512 pixels with 256 gray scale levels. Several sample images with 64×64 pixels were taken from each of these images and the algorithm for extraction of GMRF parameters was applied on them. GMRF model of 9-th order was used because it can capture the details of the image than lower order models.

Both images have distinct deterministic features with large spatial periods. The image on Fig. 3 has white horizontal and black vertical lines and we can notice wide vertical stripes in the image on Fig. 4. To capture these features, we will require GMRF model of very large order. For the image on Fig. 3 it is possible to choose such a test image with size 64×64 pixels (**test image 1** shown in **Fig.5(a)**) , so that it falls between the lines. The test image can be described quite well by GMRF model. For the **test image 2** shown in **Fig.6(a)** , it is impossible to select a 64×64 pixel frame which does not touch the vertical stripes.

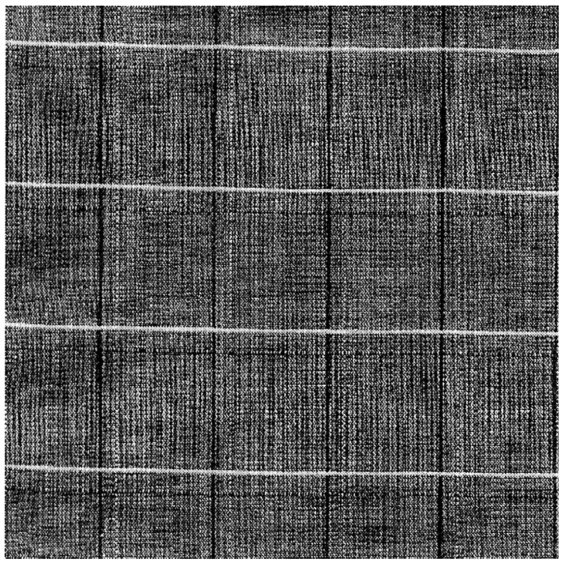


Fig.3 Input image 1 with size 512×512 pixels.

Therefore the GMRF model parameters were not be as good as that of the test image 1.

Higher values of $\ln[p(\mathbf{g}|\gamma)]$ were achived in the maximization of parameters for the picture shown on Fig. 3 than for the picture on Fig. 4. The first picture (the test image 1) is well described by the GMRF model than the second one.

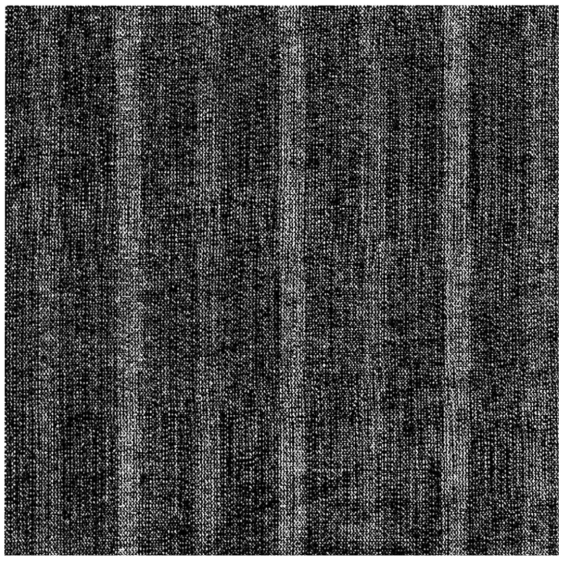


Fig.4 Input image 2.

5.4 Generation of GMRF Images

To verify the GMRF extraction algorithm, the program ***gmrf_make.c*** was created. It can generate GMRF images from the set of parameters $\gamma = (\mu, \sigma^2, \beta)$, extracted by *gmrf_extract.c* using the method

Table 6 GMRF parameters β (9-th order), obtained from a sample from image 1 with size 64×64 pixels. Element $\beta(0,0)$ is in the upper left corner; element $\beta(6,0)$ is in the lower left corner; element $\beta(6,3)$ is in the middle of the bottom row of the table. This is the output of the program *gmrf_extract.c*. The remaining 2 parameters are $\mu = 89$ and $\sigma^2 = 1439.8$; the error is below 10^{-3}.

-0.0408	-0.0597	-0.0025	
-0.0519	-0.0347	0.1044	
0.0004	-0.0598	-0.1002	
0.1691	0.2028	-0.0973	×
-0.0390	-0.0599	-0.0786	0.2331
-0.1200	-0.0660	0.1480	0.0013
-0.0049	-0.0624	-0.0739	0.1072

Table 7 GMRF parameters β (9-th order), obtained from a sample from image 2 with size 64×64 pixels. The remaining 2 parameters are $\mu = 59$ and $\sigma^2 = 1474.58$.

-0.0347	-0.0022	-0.0476	
0.0355	0.0713	0.0736	
-0.0701	-0.0195	-0.0997	
0.2895	-0.0086	0.0473	×
-0.0432	-0.0377	-0.1004	0.3159
0.0018	0.0472	0.0508	-0.1380
-0.0151	0.0180	-0.0134	0.1230

described in [3]. The generation of GMRF images is very fast. It consists of two steps: 1) generation of the FFT spectrum $G(m,n)$ of the GMRF image; and 2) taking inverse Fourier transform on the generated $G(m,n)$.

The detailed description of the algorithm on the program *gmrf_make.c* is as follows:

1. The GMRF parameter set is read form a file.

2. Two arrays, $Re[N][N]$ and $Im[N][N]$ are set initially to zero. They hold the real and the imaginary parts of $G(m,n)$.

3. $S_g(m,n)$ is calculated according to (11) for each $-(N/2-1) \le m \le N/2$, $-(N/2-1) \le n \le N/2$ and stored in an array.

4. For indices $0 \le m \le N/2$, $0 \le n \le N/2$, and $-(N/2-1) \le m \le -1$, $1 \le n \le (N/2-1)$ the arrays $Re[N][N]$ and $Im[N][N]$ are filled with random numbers with Gaussian distribution, zero mean and variance

$$\sigma^2 = N^2 S_g(m,n). \tag{14}$$

Random numbers with Gaussian propability function

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right), \tag{15}$$

corresponding to zero mean and unit variance can be generated using the method described in *Numerical Recipes in C* :

$$gasdev = \sqrt{-2\log x_1}\ \cos(2\pi x_2), \tag{16}$$

where x_1 and x_2 are two independent random numbers in the range (0,1]. From (16) it is easy to generate random numbers with arbitrary variance and mean values.

5. The elements of $Re[N][N]$ with indices $(0,0)$, $(0,N/2)$, $(N/2,0)$ and $(N/2,N/2)$ are filled with numbers generated from the same Gaussian distributed random generator (See Eq. 14). The elements of $Im[N][N]$ with the same indices are set to zero.

6. The rest of the array $G(m,n)$ is reconstructed using the symmetry property of the FFT spectrum of a real image:

$$G(-m,-n) = G^*(m,n), \tag{17}$$

where $G^*(m,n)$ is the complex conjugate of $G(m,n)$.

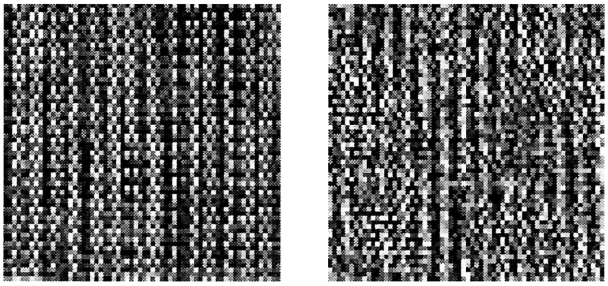
7. Inverse FFT to the $G(m,n)$ is taken, which produces real image of the GMRF model.

The synthesized images were output in ASCII format. The biggest computational burden is the IFFT calculation, however it is done only once and the algorithm is very fast.

Fig. 5 a , **Fig. 5 b** , **Fig. 6 a** , and **Fig. 6 b** show close-up views of both the sample images of size 64×64 pixels and the synthesized images with the same size. We can notice that the generated images look quite similar to the source textures. The synthesized images have the distinctive micro texture of the sample images. This shows that both algorithms, for parameter extraction and image generation, are correct and give good results. However, it can be noticed that the sample image 1 is significantly better described by GMRF model than the sample image 2. Because the likelihood function of the image 1 had higher maximum than that of the image 2.

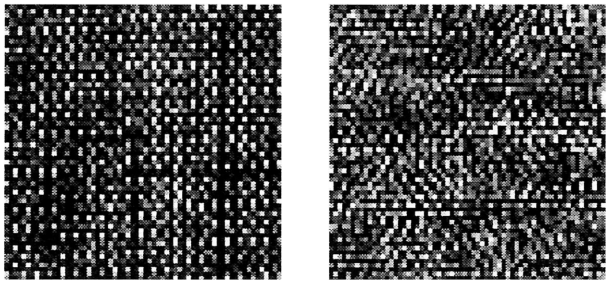
Fig.5 shows that pixels with approximately equal gray levels are separated by 2 pixels in the horizontal direction. The same patter can be observed in the matrices of GMRF parameters β in **Tables 6** and **Table 7** .

On **Fig. 7** we can see how the GMRF model has captured properly the microtexture of the image, but



a) b)

Fig.5 Close-up view of sample image with size 64×64 pixels, taken from the input image 1 (a) and synthesized GMRF image (b). The sample image was taken form the left part of the input image with cut points $(x, y) = (100, 0)$, which is away from the vertical and the horizontal stripes. The images are magnified 2 times.



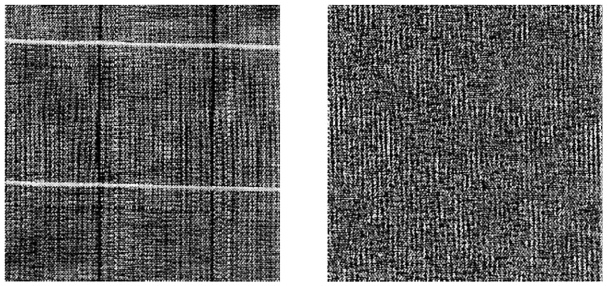
a) b)

Fig.6 Close-up view of sample image with size 64×64 pixels, taken from the input image 2 (a) and synthesized GMRF image (b). The sample image was taken form the left part of the input image with cut points $(x, y) = (100, 0)$. The images are magnified 2 times.

not the macrotexture consisting of vertical and horizontal strips. This is natural since the sample image does not include the macrotexture.

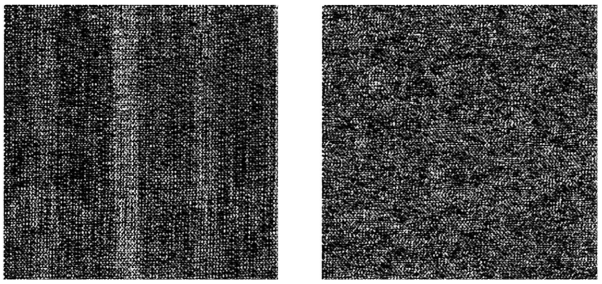
The GMRF model of the image 2 (**Fig. 8(a)**) is less successful in describing the real image because the image of size 64×64 pixels always includes part of the wide vertical strips, and therefore it is not uniformly textured. The GMRF model of such nonuniform texture is not very close to the original, which is expected. Unfortunately, sample images with sizes 32×32 pixels also include part of the strips and their GMRF model is only slightly better.

Table 8 shows the GMRF parameters of the image 2 using sample with size of 32×32. It can be noticed that they are quite similar to those extracted from 64×64 sample image (Table 7). The synthesized GMRF image, generated from the parameters from Table 8, resembles its real image only slightly better than the synthesized one using a 64×64 sample image



a) b)

Fig.7 Sample image with size 256×256 pixels, taken from input image 1 (a) and synthesized GMRF image (b). The sample image was taken form the left part of the input image with cut points $(x, y) = (0, 0)$.



a) b)

Fig.8 Sample image with size 256×256 pixels, taken from input image 2 (a) and synthesized GMRF image (b). The sample image was taken form the left part of the input image with cut points $(x, y) = (0, 0)$.

on Fig. 8(a).

6 Conclusion

The programs for analyzing and synthesizing real textured images based on GMRF model had been developed. The algorithm for extracting statistical parameters adopted the Simulated Annealing technique for maximization of the likelihood function of

Table 8 GMRF parameters β (9-th order), obtained from a sample from image 2 with size 32×32 pixels. The remaining 2 parameters are $\mu = 54.0$ and $\sigma^2 = 1762.7$

-0.0180	-0.0368	-0.0424	
0.0289	0.0832	0.0203	
-0.0961	-0.0357	-0.0952	
0.2936	-0.0265	0.0210	×
-0.0584	-0.0439	-0.0916	0.2874
0.0084	0.0313	-0.0101	-0.1561
-0.0387	-0.0027	-0.0232	0.0955

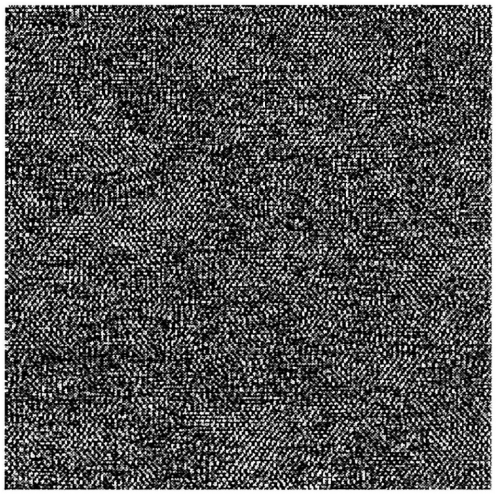


Fig.9 Synthesized GMRF image from sample of image 2 with size 32×32 pixels.

the Fourier transform of an image with respect to the GMRF parameters. The maximization used a combination of two methods (the Golden Section Search and the Metropolis procedure). The Golden Section Search was used for maximizing only one variable, here only one of the GMRF parameter set, coupled within the main SA loop which searched for the maximum in multidimensions. These two methods combine the best features of both and result in fast and reliable routine for maximization of a function of many variables with many local maxima.

The estimated parameters were used for synthesizing artificial textured images of the GMRF model. The synthesized images look similar to the original ones. However, microstructures of the texture related with more than third order statistics had been ignored. Because, extracted parameters were principally belonged to second order statistics. Some of the syntesized images are presented together with the input images used for parameter estimation. The important features of GMRF models were revealed. The features capture the microtexture of images and handle the macrotextures. The results about the image synthesis are discussed in detail. The developed programs can be used to textured images in wide range of applications.

Acknowledgements

Many people including fellow students of Tingting Cao have contributed to the work presented as the Master thesis of Tingting Cao. I would like to thank them for their help and support. I would like to thank Assoc. Prof. Hiroshi Douzono and Shigeom Hara for encouragement and help with the software and the Japanese language on numerous occasions. Tingting Cao is grateful for the generous Seikou scholarship, which made her stay in Japan possible.

References

[1] R. Chellappa and S. Chatterjee, " Classification of texture using gaussian Markov random fields", IEEE Trans. Acoustics, Speech and Signal Processing, Vol. 33, No. 4, pp. 959-963, 1985.

[2] C.H.Chen, L.F.Pau, and P.S.P. Wang(eds), Handbook of Pattern Recognition & Computer Vision, 2nd edition, World Scientific, Singapore, 1999.

[3] F.S.Cohen, Z. Fan, and M.A. Patel, " Classification of rotated and scaled textured images using gaussian Markov random field models", IEEE Trans. Pattern Anal. Mach. Intell., Vol. PAMI-13, pp. 192-202, 1991.

[4] R. Cross and A. Jain, " Markov random field texture models ", IEEE Trans. Pattern Anal. Mach. Intell., Vol. PAMI-5, pp. 25-39. 1983.

[5] S. Geman and D. Geman, " Stochastic relaxation, Gibbs distributions, and the Bayesian restroration of images ", IEEE Trans. Pattern Anal. Mach. Intell., Vol. PAMI-6, pp. 721-741, 1984.

[6] R.M. Haralick, K.Shanmugan, and Its'hak Dinstein, " Texture features for image classification", IEEE Trans. Syst., Man, Cybern., Vol. SMC-3, No.6, pp. 610-621, 1973.

[7] A. K. Jain : Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[8] S. Z. Li : Markov Random Field Modeling in Image Analysis, Springer-Verlag, Tokyo, 2001.

[9] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller, " Equation of state calculations by fast computing machines", J. of Chemical Physics, Vol. 21, No.6, pp. 1087-1092, 1953.

[10] J. Woods, " Two-dimensional discrete Markov random fields ", IEEE Trans. Inform. Theory, Vol. IT-18, pp. 232-240, 1972.