Texture Synthesis*

Michal Haindl
Institut National de Recherche en Informatique et en Automatique
Domaine de Voluceau - Rocquencourt,
B.P.105 - 78153 Le Chesnay, France

To enhance realism in a graphics system it is necessary to cover generated surfaces with realistic textures. A texture in this paper is assumed to be a realization of a random field. Problems of parameter estimation and random field synthesis of a given random field model are studied. The most flexible models for realistic texture synthesis seem to be simultaneous autoregressive and Gaussian Markov random field models.

1 Introduction

In this article, we consider texture synthesis methods. The intention is to overview methods which are capable of reproducing realistic textures for enhancing realism in graphics systems. Digitized solid 3D textures are far less convenient, since they involve the 2D digitization of a large number of cross-sectional slices through some material. Synthetic textures are more flexible than digitized textures, in that synthetic textures can be designed to have certain desirable properties or meet certain constraints; for example, it can be made smoothly periodic, so that it can be used to fill an infinite texture space without visible discontinuities. While a digitized texture must be stored in a tabular form and evaluated by table lookup, a synthetic texture may be evaluated directly in procedural form. The purpose of a synthetic texture is to reproduce a given digitized texture image so that both natural and synthetic texture will be indiscernible. Therefore we limit ourselves to synthesis methods for which the analysis step is known, neglecting different ad hoc methods [48] or models with unknown parameter estimation. In the field of texture analysis, we limit our attention to a subset of analysis methods, which are useful for described texture synthesis methods. We will neglect a large area of analysis methods useful for texture segmentation like for example the co-occurrence matrices, the gray level difference method, etc. The interested reader can consult any standard image processing textbook. In this attitude we differ from the usual meaning of a texture analysis problem, where the goal is not to reproduce a given texture but to discriminate between several different textures instead. Texture model parameters can serve as a basis for texture discrimination too. While there exists

*This work was supported by ERCIM Fellowship while the author was visiting at CWI, Amsterdam.

an abundance of literature covering the texture analysis, much lesser attention has been paid to the texture synthesis problem. So far we even do not know a survey covering the state of the art in this field our selection of methods is based on referenced papers only.

There is no exact definition for texture, although some authors claim to give one. Texture is generally a visual property of a surface, representing the spatial information contained in object surfaces. Texture may represent [5] information that permits the human eye to differentiate between image regions. Another attempt to define texture is [19]: A visual texture is a sensory impression obtained through the eyes, and thus is a visual perception. Texture is considered here to be the visual perception, by which two neighboring, possibly structured, parts of the visual field may be effortlessly (spontaneously) separated by means of observation with fixed eyes. Another definition [23] states: Texture is a structure which is made of a large ensemble of elements that resemble each other 'very much', with some kind of an 'order' in their locations, so that there is no single element which attracts the viewer's eye in any special way. The human viewer gets an impression of uniformity when he looks at a 'texture'. Some authors [42] use the term texel or texon for a textured area element. We will not repeat here the number of rather philosophical definitions of texture, instead we understand a textured image to be a realization of a random field and our effort is simply to produce a copy of a given image using modelling techniques.

We use further on the following symbols: $M \times N$ is an image size, K is the number of gray levels (classes), ω_i is the true class, r, s, u are double indices ($r = \{i, j\}$), I is the image lattice (set of all indices), Y denotes a random vector (or its realization) of an image in some arrangement, Y_r is a random variable (pixel) at location r, $Y_{(r)}$ is a random vector Y with excluded Y_r, γ is a set of model parameters, I_r or $I_{i,j}$ is a neighborhood of pixel $r = \{i, j\}$, μ is a mean value and σ^2 a variance.

2 Texture synthesis methods

Several approaches to texture synthesis exist, each of them has its advantages and also its limitations. Existing methods can be categorized using different criteria. Englert and Sakas [19] divide methods into 'models focusing on the description of spatial organization and the visual appearance of a texture field' and 'models arising from texture generation methods', another possible division is into deterministic and stochastic [23] methods. Similarly Bennis and Gagalowicz [5] have also two major groups : structural (macroscopic) methods [4] consider texture as a spatial arrangement of a set of basic patterns according to some placement rules, while microscopic methods do not assume a well organized structure. These textures present a homogeneous visual aspect even without discernible spatial arrangement. In the first group we will mention random mosaic models, fractals and syntactic models. We will focus our attention on the second group and especially on the autoregressive and Markov random field models, because of our strong belief that these models are far more flexible to cover realistic textures than other approaches surveyed in this paper.

3 Random mosaic models

In the random mosaic models, the texture region is first divided into convex polygonal cells, then every cell is independently assigned a class (gray level, color) ω_i according to a fixed set of probabilities $P_1, \ldots, P_K$. Let r, s be two points a distance d apart, $P(d)$ the probability that both points are in the same cell, then [2]:

$$P(Y_r = \omega_j | Y_s = \omega_i) = P_j(1 - P(d)), \tag{1}$$

$$P(Y_r = \omega_i | Y_s = \omega_i) = P_i + (1 - P_i)P(d). \tag{2}$$

3.1 The Poisson line model

The texture area is divided into convex cells by a set of straight lines with random positions and orientations. Points (α, r) are generated [59] by a Poisson process of intensity τ/π, where $0 \leq \alpha < \pi$ and $-\infty < r < \infty$. Each line is defined as follows:

$$r = x \cos\alpha + y \sin\alpha. \tag{3}$$

Probability $P(d)$ can be found to be

$$P(d) = \exp\{-2\tau d/\pi\}, \tag{4}$$

for some other statistical characteristics see [2, 59]. The Poisson line model does not fit very well [56, 58] with natural textures.

3.2 The rotated checkerboard model

The origin of a coordinate system as well as the orientation of axes is chosen with uniform probability (orientation in the interval $[0, \pi]$). Using the resulting coordinate grid, the plane is tessellated into square cells of side length b. Probability of the same class for two distant points [56]:

$$P(d) = \begin{cases} 1 - 4d/\pi b + d^2/\pi b^2 & d \leq b \\ 1 - \dfrac{2}{\pi} - \dfrac{4}{\pi \cos(b/d)} - \dfrac{d^2}{\pi b^2} + \dfrac{4}{\pi(d^2/b^2 - 1)^{1/2}} & b < d \leq b\sqrt{2} \\ 0 & d > b\sqrt{2} \end{cases} \tag{5}$$

Because of the rectangular character of the rotated checkerboard model (the square model [2]) generated texture, we cannot expect it to fit very much with natural textures. This model can be useful for certain man-made textures however. A rotated hexagon model is analogous [57] to the checkerboard model, except that hexagons are used in lieu of squares. Another possibility is the rotated triangular model [2], where an equilateral triangular tessellation can be formed by connecting the growth centers of neighboring cells in a hexagonal tessellation.

3.3 Voronoi tessellation

The principle of this method (known also as the Dirichlet tessellation or the occupancy model) is to create a cell around each of the seed points (generated for

example by a Poisson process). Each cell contains all plane points for which the corresponding seed point is the nearest one. Voronoi tessellation shares similar disadvantages in a low natural texture modelling with the both previously mentioned methods, while the parameter estimation [59] is more difficult and even problematic on images with nonlinear or unclear borders between homogeneous areas.

3.4 Random mosaic models analysis

Random mosaic models parameter estimation is based on model fitting techniques using known dependences between the observed features (variogram, component width, perimeter, etc.) and the model parameters. If there is defined a distance between two classes (for example gray levels), we can modify the definition of the variogram ([56, 57]) of a random mosaic:

$$V(d) = E\{(Y_r - Y_{r+d})^2\}. \tag{6}$$

In a special case of constant gray level in each cell and uncorrelated gray levels in different cells the variogram can be simplified [56]

$$V(d) = 2\sigma^2(1 - P(d)), \tag{7}$$

where σ^2 is assumed to be the standard deviation of texture's classes (gray levels) and $P(d)$ is the probability that two points distance d apart are in the same cell. For the Poisson line model and the rotated checkerboard model we can compute for a given σ^2 the theoretical variogram as a function of the Poisson parameter τ or checkerboard square size b. Given a variogram of a real texture, these parameters can be obtained using fitting techniques.

Some other models [56] of this class, as for example the Johnson-Mehl model and the bombing model are not mentioned here, because they do not fulfill our criterion to have solved the analytical step (parameter estimation) too.

If a texture can be divided into fixed-size windows, regularly replicated over a whole image array, we obtain a deterministic mosaic model (periodic tessellated model).

4 Syntactic models

In the structural approach a texture is considered to be defined by subpatterns which occur repeatedly according to a set of well-defined placement rules within the overall pattern. A texture pattern is composed of fixed-sized windows as subpatterns. Each window is represented by a tree of which each node corresponds to a pixel in the window. A pattern primitive is the label representing the gray level of each pixel. A tree grammar is then used to characterize the windowed texture pattern. Some examples of this approach can be found in [24, 25]. In order to model the image of interest, it is necessary to have the stochastic tree grammar actually inferred from the available image samples. Such an inference procedure requires the inference of both the tree grammar and its production probabilities. Unfortunately, a general inference procedure for stochastic tree grammars does not exist and is still a subject of research [25]. Only some very special cases can be treated by existing inference algorithms, like for example [25]:

a) Windowed patterns are grouped into clusters using a similarity measure.

b) Each cluster consists of a finite set of trees having the same structure. A stochastic tree grammar is then inferred for each cluster.

c) The final texture grammar is the union of placement rules and all stochastic tree grammars.

Using such an algorithm it is still difficult to solve grammar inference for other then very simple structures. It is therefore difficult to model realistic textures, and the algorithm is computationally demanding due to employing large set searching procedures.

5 Fractal models

Fractals are sets, whose Hausdorff-Besicovitch dimension, which in general is a real number, is strictly larger than their topological dimension [3, 49]. We limit ourselves to fractals with two additional properties, the similarity of each segment to all others and statistical invariance over wide transformation of scale. The primary control that fractal techniques provide over the resulting object is by the value of a single parameter determining the fractal dimension. The only examples of fractal objects being used to model natural phenomena are based on fractional Brownian functions. A random function Y_r is a fractional Brownian function (surface for 2D) if for all r and d:

$$P(\frac{Y_{r+d} - Y_r}{||d||^H} < t) = F(t), \tag{8}$$

where $F(t)$ is a cumulative distribution function. If Y_r is scalar, then the fractal dimension of Y_r is $D = 2 - H$.

If $H = 0.5$ and $F(t)$ is the standardized Gaussian distribution, then Y_r is the classical Brownian function. A surface can be accurately approximated by a single fractal function, if the fractal dimension is stable over a wide range of scales. The texture is defined to be fractal [50] if

$$E\{|Y_{r+d} - Y_r|\} \parallel d \parallel^{-H} = const. \tag{9}$$

Several useful properties of the fractal Brownian function were proven in [50]. The fractal dimension of a fractal Brownian function is invariant under transformation of scale. A 3D surface with a spatially isotropic fractal Brownian shape produces an image whose intensity surface is fractal Brownian and whose fractal dimension is identical to that of components of the surface normal, given a Lambertian surface reflectance function and constant illumination and albedo. The opposite proposition is also valid. The fractal textures can be synthesized using one of the following methods [18] (5.1-5.3), combined with an analytical method (5.4,5.5).

5.1 The Fourier transformation method

A random Gaussian field is spatially filtered to generate well-defined second-order connections between pixels. The phase of the Fourier transformed random field is unaltered, but the modulus is forced to assume a form f^{-P}. The parameter P can be altered to cause the inverse Fourier transformation of this modified field to have any particular fractal dimension between 2 and 3. Experiments on

fractal textures generated using this method with identical first order statistics show [18], that the minimal resolvable difference in fractal dimension is 0.06.

5.2 The cylindrical integration method
A path within a random Gaussian field of zero mean painted on a cylinder is integrated over to produce a Brownian variation in image brightness. The path in cylindrical coordinates, is defined as $x\cos(\theta) + y\sin(\theta), \theta$, where $r = (x, y)$ is the rectangular coordinate of the cell in the Brownian field, and θ is the variable of integration. Although this method is able to produce true fractals, it is unable to generate a range of different dimension fractals.

5.3 The midpoint displacement method
The brightness at the center of a square is interpolated from the four corner brightness values. This value is perturbed by an amount related to the size of the square. The four subsquares are treated similarly and so on down to the resolution of the image. The result was shown not to be truly fractal, but the method is quick and efficient to implement.

5.4 The Peleg method
The fractal dimension of Brownian function can be measured directly or from Fourier power spectrum $P(z)$ of Y_r, as the spectral density of fractal Brownian function is proportional [50] to z^{-2H-1}. The brightness field of a texture is imagined to form a surface with brightness represented by height. The method [49] is based on the volume v , occupied by all points distant ϵ or less from the surface. The surface area is obtained by dividing the volume by 2ϵ and is found to be a function of ϵ. The method uses the increase in apparent area $A(\epsilon)$ at each increment of ϵ to obtain a signature for the texture from which the relation (10) can be fitted to obtain H:

$$A(\epsilon) = F\epsilon^{(H-1)}. \tag{10}$$

5.5 The Pentland method
If the texture is fractal, then (9) must hold so we can estimate the fractal dimension from relation (11):

$$E\{|Y_{r+d} - Y_r|\}||d||^{-H} = E\{|Y_{r+\tilde{d}} - Y_r|\}, \tag{11}$$

where $||\tilde{d}|| = 1$. Using (11) and a least square regression for different d, we can estimate H. If the relation (11) holds, then the viewed surface must be a 3D fractal Brownian surface and can be modeled using this fractal model.

6 Second order statistical models
This texture model generates a random field which has second-order statistics similar to the natural texture we are trying to simulate. It is assumed, that although Julesz conjecture [38] (that second-order statistics are sufficient in terms of human visual texture discrimination) was proved not to be generally valid, such an approximation is still fairly accurate. The set of second-order joint density functions $p_{s,\tilde{s}}(Y_1, Y_2)$, necessary for model analysis, can be estimated from a

natural texture field using occurrence matrices computed from a given texture; first order statistics are estimated from the corresponding histogram.

6.1 The algebraic reconstruction technique

We assume that a reconstructed random field is homogeneous (translation invariant). The problem is to find a joint density function $p(Y_1, \ldots, Y_{MN})$ from known marginals $p(Y_s, Y_r)$. This problem corresponds to a linear programming problem with K^{MN} linear equations and inequalities. The use of standard techniques is limited by the large dimensionality of the problem. The ART method starts [22] with generating all simulated pixels from a uniform distribution:

$$p^{(0)}(Y_1, \ldots, Y_{MN}) = \frac{1}{K^{MN}}. \tag{12}$$

In each iteration the sum of the differences between the actual and the reconstructed marginal is computed and evenly divided amongst the K^{MN-2} reconstruction elements:

$$p^{(n+1)}(Y_1, \ldots, Y_{MN}) = \max\{0, p^{(n)}(Y_1, \ldots, Y_{MN}) + tc^{(n)}(Y)\}, \ \forall\{Y\}, \tag{13}$$

$$c^{(n)}(Y) = \frac{1}{K^{MN-2}\binom{MN}{2}} \sum_{r=1}^{MN-1} \sum_{s=r+1}^{MN} (p(Y_r, Y_s) - p^{(n)}(Y_r, Y_s)). \tag{14}$$

The convergence of the ART algorithm depends on the choice of constant t.

6.2 Second order spatial averages

The principle of this method is to generate a homogeneous random field with the second order spatial averages

$$\tilde{p}_d(\omega_i, \omega_j) = \frac{1}{n} \sum_{r=1}^{n} \delta(Y_r - \omega_i)\delta(Y_{r+d} - \omega_j), \tag{15}$$

equal to those of a given texture field. The first step of the method is synthesizing a texture field, which is a realization of a homogeneous white noise and whose histogram is equal to the desired ($\tilde{p}^*$) histogram [26]:

$$\tilde{p}(\omega_i) = \tilde{p}^*(\omega_i) \qquad \forall i = 1, \ldots, K.$$

In the second step, the texture field is modified point by point to minimize the mean square error between the desired second order spatial averages and current second order spatial averages. It was experimentally verified [26], that the selection of modified points should be random, otherwise the resulting texture is not homogeneous.

6.3 The autocovariance and histogram model

This second order statistical model is defined by a set of histogram and autocovariance parameters (16):

$$B(d) = \frac{1}{MN\sigma^2} \sum_{r=1}^{MN} (Y_r - \mu)(Y_{r+d} - \mu). \tag{16}$$

The first synthesis step is equal to the first step of 6.2. The second one is also random location pixel modification so as to minimize the mean square error between the template feature vector and the feature vector computed on the synthesized texture field, but the feature vector is now concatenated from the texture histogram and the autocovariance function [4].

7 Autoregressive models

Autoregressive or simultaneous autoregressive (SAR) models from the class of simultaneous models have several favourite properties, which cause their wide application in image analysis. Simultaneous models are basically generalizations of 1D time series models into 2D. Such a generalization is not quite straightforward because unlike in 1D time series, where the existence of a preferred direction (time) is assumed, no such preferred ordering exists on the discrete lattice. Other groups from the class of simultaneous models are simultaneous moving average (SMA) and simultaneous autoregressive and moving average (SARMA) models. These other models have complicated parameter estimation. For example, estimation of ARMA model parameters is a nonlinear problem even in 1D, and some approximations have to be used. The advantage of SAR models is their close relation with Markov random field models (MRF) see Section 8. For every SAR model there exists a unique conditional MRF with an equivalent spectral density function, but the converse is not always true (except for the Gaussian case [6]). The conditional MRF model is generally characterized by more parameters than the equivalent SAR model (if it exists). SMA and SARMA models are not subsets of MRF models. The SAR model can be written as follows:

$$Y_r = \sum_{s \in I_r} a_s Y_{r-s} + e_r, \tag{17}$$

where e_r is a white Gaussian noise with zero mean and a constant variance σ^2, and is uncorrelated with data from I_r. If I_r is a symmetric neighborhood, we assume that symmetrically opposite neighbors have equal parameters. Otherwise, the parameters may not be identifiable [6]. We can rewrite a SAR model in the matrix form:

$$Y_r = \gamma X_r + e_r, \tag{18}$$

where

$$\gamma = [a_1, \ldots, a_s], \tag{19}$$

and X_r is a corresponding vector of Y_{r-s}. If we replace Y_r in (17) by $\tilde{Y}_r$ ($\tilde{Y}_{i,j} = Y_{i,j} - \mu$), the joint probability density of process (17) can be written as

$$p(Y) = (2\pi\sigma^2)^{-MN/2} |\Psi| \exp\{-1/2\sigma^{-2}(Y-U)^T \Psi^T \Psi (Y-U)\}, \tag{20}$$

where Ψ is defined in (49) and is for a torus lattice I a block circulant matrix, each block being again a circulant matrix and U is defined in (48). We can synthesize an image described by SAR from the equation

$$Y = A^{-1}(E - B), \tag{21}$$

where the noise vector

$$E = [e_1, \ldots, e_{MN}]^T \tag{22}$$

has an arrangement corresponding with Y. B is a vector with border conditions and A is an $MN \times MN$ block Toeplitz type matrix of a_s coefficients with unit diagonal. If we assume zero boundary values ($B = 0$), symmetric coefficients ($a_{r-s} = a_{r+s}$) and regularity of A, then A is regular, symmetric and positive definite. Under this condition the solution of (21) can be obtained iteratively by the conjugate gradient method [14, 33] (Fletcher-Reeves version of the method).

7.1 Woods iterative synthesis

Woods has proved [62], that a random field which can be described by (17), where $\{e_r\}$ are independent random variables with bounded absolute moments (possibly non-Gaussian), can be generated using the following iterative procedure (23),(24):

$$Y_r^{(n+1)} = \sum_{s \in I_r} a_s Y_{r-s}^{(n)} + e_r, \tag{23}$$

$$Y_r^{(n+1)} = Y_r, \tag{24}$$

and the stability assumption $\sum_{\forall s \in I_r} |a_s| < 1$. Equation (24) is used if r is from the boundary area. Equation (23) for the non-boundary area is initialized by $Y_r^{(0)} = e_r$. This iterative procedure converges to the mean absolute solution of (17), with the boundary condition $Y_r = Y_r^{boundary}$.

7.2 The autoregressive model with conditional expectations

Let us assume now to have chosen a partition of a neighborhood I_r, we will denote by I_r^i the i-th such a subset of I_r and l the number of these subsets. Now the autoregressive model with conditional expectations can be put in the following form [21]:

$$Y_r = \sum_{i=1}^{l} a_i E\{Y_r | Y_s \forall s \in I_r^i\} + e_r, \tag{25}$$

or in the matrix form (18),(19) , where X_r is now

$$X_r = [E\{Y_r | Y_s \forall s \in I_r^1\}, \ldots, E\{Y_r | Y_s \forall s \in I_r^l\}]^T. \tag{26}$$

The main disadvantage of synthesis model (25) is the necessity to identify and store a large number of parameters. If we denote the cardinality of I_r^i as q_i, then the number of model parameters is $\sum_{i=1}^{l} K^{q_i} + l$.

7.3 Optimal model selection

The selection of an appropriate autoregressive (AR) model is important to obtain good results in modelling a given random field. If the contextual neighborhood is too small we cannot capture all details of the random field. On the contrary, larger than necessary contextual neighborhoods can introduce problems with numerical accuracy and can be too time consuming. Apart from visual

comparison, which is subjective, we can choose the appropriate model using pairwise hypothesis testing [6, 13], Akaike's information criterion (AIC) [39], or the Bayesian approach. The main disadvantage of the hypothesis testing approach is that the resulting decision rule is not transitive and also not consistent. The AIC method gives transitive decision rules but not consistent ones [40]. The approximate Bayesian decision rule for the AR model was given in [40]: choose the AR model k if

$$k = \arg\min_j \{D_j\}, \tag{27}$$

where

$$D_j = -\sum_{r \in I} \ln(1 - \gamma_k C_r + \gamma_k Q_r Q_r^T) + MN \ln \sigma_k^2 + m_k \ln(MN), \tag{28}$$

γ_k, and σ_k^2 are iterative estimations of corresponding parameters, C_r is equal to S_r except for the replacement of sin by cos, m_k is the number of neighbors in the k-th model, $Q_r = S_r S_r^T + C_r C_r^T$, and S_r is defined as follows:

$$S_r = [\sin(2\pi(MN)^{-1/2}(r-1)^T s), \ldots]^T \qquad \forall s \in I_r. \tag{29}$$

7.4 Autoregressive model parameter estimation

Similar to the Gaussian Markov random field model (GMRF) case (Section 8.6.1.), the maximum likelihood (ML) estimates require numerical optimization methods due to the log-likelihood function, which is nonquadratic in γ. The ML estimate for σ^2 is also (33). To avoid computationally expensive numerical methods, it is possible to use the least square method (LS), the maximum pseudo-likelihood (MPL) method or some approximation method like 7.4.2.

7.4.1 The prediction error variance minimization. Let us define model prediction as $\tilde{Y}_r = E\{Y_r\}$, then

$$\tilde{Y}_r = \gamma X_r. \tag{30}$$

The parameters are now adjusted to minimize the variance of the prediction error:

$$E\{(Y_r - \tilde{Y}_r)^2\}. \tag{31}$$

The LS solution can be found in the form:

$$\hat{\gamma} = E\{X_r X_r^T\}^{-1} E\{X_r Y_r\} = [\sum X_r X_r^T]^{-1} \sum X_r Y_r, \tag{32}$$

$$\sigma^2 = \frac{1}{MN} \sum_r (Y_r - \hat{\gamma} X_r)^2. \tag{33}$$

The drawback of the LS estimator (32) is its inconsistency for nonunilateral neighborhoods [40].

7.4.2 The iterative estimation method. The principle of this method [40] is to avoid numerical optimization of the nonquadratic ML function using quadratic

approximation of the determinant in (20) and the toroidal lattice assumption. The method yields estimates close to asymptotically consistent and efficient ML estimates.

7.4.3 Estimation of the SAR model with conditional expectations. To estimate parameters in model (25), we can use the prediction error variance minimization. The solution [21] is similar to (32),(33):

$$\hat{\gamma} = [E\{X_r X_r^T\}]^{-1} E\{X_r Y_r\} = W^{-1} V, \tag{34}$$

where

$$w_{ij} = E\{E\{Y_r | Y_s \forall s \in I_r^i\} E\{Y_r | Y_s \forall s \in I_r^j\}\}$$

$$= \sum_{Y_s \forall s \in I_r^i \cup I_r^j} E\{Y_r | Y_s \forall s \in I_r^i\} E\{Y_r | Y_s \forall s \in I_r^j\} p(Y_s \forall s \in I_r^i \cup I_r^j), \tag{35}$$

and

$$v_i = E\{E\{Y_r | Y_s \forall s \in I_r^i\} Y_r\}$$

$$= \sum_{Y_r, Y_s \forall s \in I_r^i} Y_r E\{Y_r | Y_s \forall s \in I_r^i\} p(Y_r, Y_s \forall s \in I_r^i). \tag{36}$$

The only complication in comparison with (32),(33) is knowledge of joint density function in (35),(36), which has to be estimated from given real texture data. Conditional expectations are computed as follows:

$$E\{Y_r | Y_s \forall s \in I_r^i\} = \sum_{Y_r=0}^{K-1} Y_r p(Y_r | Y_s \forall s \in I_r^i). \tag{37}$$

Parameter estimation is computationally time-consuming, especially for larger contextual neighborhoods. Also, the memory requirements of model (25) are higher than for the simpler model (17).

8 Markov random field models

The Markov random field (MRF) is a family of random variables with a joint probability density on the set of all possible realizations Y of the lattice I, subject to following conditions:

$$p(Y) > 0, \qquad \forall Y, \tag{38}$$

(positivity condition), and

$$p(Y_r | Y_s \forall s \in I) = p(Y_r | Y_s \forall s \in I_r), \tag{39}$$

(Markovianity in a strict sense [41], or local Markov property). We limit ourselves here to homogeneous and symmetric neighborhood systems I_r. A hierarchy of MRF models can be defined using such a neighborhood system, as in the following figure, where numbers on the place of contextual neighbors indicate the order of the model (up to fifth order) relative to x [6]:

5	4	3	4	5
4	2	1	2	4
3	1	x	1	3
4	2	1	2	4
5	4	3	4	5

The neighborhood systems I_r of the n-th order MRF model then contain all neighbors of x with position numbers $1, 2, \ldots, n$. We define a clique to be a set of points that consists either of a single point, or has the property that each point in the clique is a neighbor of all the others. For example for the first-order MRF there are cliques $\{(i,j)\}, \{(i,j-1),(i,j)\}$, and $\{(i-1,j),(i,j)\}$. A corresponding sufficient statistic for the MRF requires $2(K-1)^2+K-1$ independent parameters (an isotropic MRF requires $(K-1)^2+K-1$ parameters).

The Hammersley-Clifford theorem [6] states that Y is a MRF with strictly positive distribution and neighborhood system I_r, if and only if the distribution of Y can be written as a Gibbs distribution with cliques induced by the neighborhood system I_r. The function $G_{r,s,\ldots,u}$ in (42) may be non-null if and only if the sites $r, s, \ldots, u$ form a clique. Subject to this restriction the G-functions may be chosen arbitrarily. For a given neighborhood system a Gibbs density function can be expressed in the form

$$p(Y) = \exp\{-Q(Y)\}/Z, \tag{40}$$

where the normalization constant is

$$Z = \sum_{Y} \exp\{-Q(Y)\}. \tag{41}$$

The global energy function (or Gibbs energy function) is, according to the Hammersley-Clifford theorem:

$$Q(Y) = \sum_{\forall cliques \in I_r} Q_j$$

$$= Q(0) + \sum_{r} G_r Y_r + \sum_{r,s} G_{r,s} Y_r Y_s + \ldots + G_{r,s,\ldots,u} Y_r, \ldots, Y_u. \tag{42}$$

Local conditional densities, for example for the first-order model, can be determined from (42):

$$p(Y_r | Y_{(r)}) = \exp\{-Y_r G_r - \sum_{s \in I_r} G_{rs} Y_r Y_s\}/\tilde{Z}, \tag{43}$$

$$\tilde{Z} = \sum_{r} \exp\{-Y_r G_r - \sum_{s \in I_r} G_{rs} Y_r Y_s\}. \tag{44}$$

Boundary pixels in a finite image have less neighbors than the interior pixels. This missing data problem is solved using two approximations. The free boundary condition assumes the interaction potential between a boundary pixel and its missing neighbors to be zero. The second approximation assumes a toroidal

lattice that results in a random field which is wrapped around in a torus structure [12]. Unlike 1D time series, where we can differentiate between past and future data and accordingly between causal and non-causal models respectively, no preferred direction is assumed in 2D data. Therefore 2D causal models seem somehow artificial, but their advantage lies in the possibility of recursive parameter estimation for some models.

8.1 The Gaussian Markov random field model

If the local conditional density of the MRF model (45) is Gaussian, we obtain the GMRF model:

$$p(Y_r|Y_s \forall s \in I_r) = (2\pi\sigma^2)^{-1/2} \exp\{-1/2\sigma^{-2}(Y_r - \tilde{\mu}_r)^2\}, \tag{45}$$

where the mean value is

$$E\{Y_r|Y_s \forall s \in I\} = \tilde{\mu}_r = \mu_r + \sum_{s \in I_r} a_s(Y_{r-s} - \mu_{r-s}), \tag{46}$$

and the joint probability density function is

$$p(Y) = (2\pi\sigma^2)^{-MN/2}|\Psi|^{1/2} \exp\{-[(Y-U)^T\Psi(Y-U)]/(2\sigma^2)\}, \tag{47}$$

where the $(MN \times 1)$ mean vector consists of single location means

$$E\{Y\} = U = [\mu_{1,1}, \ldots, \mu_{M,N}]^T, \tag{48}$$

and the matrix Ψ has unit diagonal elements [6] and off diagonal elements are coefficients a_r. Ψ is symmetric, but is also required to be positive definite. Coefficients of symmetric sites have to be equal i.e. $a_{r-s} = a_{r+s}$. The covariance matrix is defined by

$$E\{(Y-U)^T(Y-U)\} = \sigma^{-2}\Psi^{-1}. \tag{49}$$

The argument in the exponent of $p(Y)$ can be put in the following form:

$$Q(Y) = \sum \tilde{Y}_{i,j}^2 + a_{1,0} \sum \tilde{Y}_{i,j}\tilde{Y}_{i+1,j} + a_{0,1} \sum \tilde{Y}_{i,j}\tilde{Y}_{i,j+1} + \ldots, \tag{50}$$

where

$$\tilde{Y}_{i,j} = Y_{i,j} - \mu. \tag{51}$$

The GMRF is parametrized by $\gamma = \{\mu, \sigma^2, a_{1,0}, a_{0,1}, \ldots\}$. The GMRF can be expressed as a stationary noncausal 2D autoregressive process described by the difference equation

$$Y_{i,j} = \mu_{i,j} + \sum_{k,l \in I_{i,j}} a_{k,l}(Y_{i-k,j-l} - \mu_{i-k,j-l}) + e_{i,j}, \tag{52}$$

where $a_{k,l} = a_{-k,-l}$, $\mu_{i,j}$ is the mean value of $Y_{i,j}$, $e_{i,j}$ is a stationary Gaussian noise with zero mean and autocorrelation given by

$$R_e(k,l) = \begin{cases} \sigma^2 & \text{if } (k,l) = (0,0), \\ -\sigma^2 a_{k,l} & \text{if } (k,l) \in I_{i,j}, \\ 0 & \text{otherwise.} \end{cases} \tag{53}$$

The power spectral density associated with $\tilde{Y}_{i,j}$ is [20]:

$$S_{\tilde{Y}}(i,j) = \sigma^2/[1 - 2 \sum_{(k,l)\in I^*_{i,j}} a_{k,l} \cos\{2\pi(ki/M + lj/N)\}], \tag{54}$$

where $I^*_{i,j}$ is the nonsymmetric half-plane of $I_{i,j}$ and 2D Fourier transformation of $\tilde{Y}_{i,j}$:

$$\mathcal{F}\{\tilde{Y}_{i,j}\} = \sum_{\forall(k,l)\in I} \sum \tilde{Y}_{k,l} \exp\{-(-1)^{1/2} 2\pi(ki/M + lj/N)\}. \tag{55}$$

It can be shown for toroidal lattices [12, 62] that the discrete Fourier transformation $\mathcal{F}\{\tilde{Y}\}$ is a white Gaussian field.

$$p(\mathcal{F}\{\tilde{Y}\}) = \prod_{i,j} [2\pi MN S_{\tilde{Y}}(i,j)]^{-1/2}$$

$$\times \exp\{-\sum_{i,j} \parallel \mathcal{F}\{\tilde{Y}\} \parallel^2 [2MN S_{\tilde{Y}}(i,j)]^{-1}\}. \tag{56}$$

The GMRF can be generated from $Y = \mathcal{F}^{-1}\{\hat{Y}\} + U$, where the mean vector U is defined in (48) and $\hat{Y}$ is generated from the Gaussian generator $\mathcal{N}(0, NMS_{\tilde{Y}}(i,j))$.

A modified possibility for generating a GMRF on a toroidal matrix $M \times M$ can be found in [11]. For GMRF models which have an equivalent AR model, we can use Woods iterative algorithm 7.1. GMRF models which do not have an equivalent AR model, can be generated by modified Woods algorithm, but nothing is known about its convergence properties, because the convergence theorem [62] is not valid any more. For the optimal GMRF model selection we can use a modification of the Bayesian method 7.3., where m_k is now a number of asymmetrical half neighborhood parameters, and

$$D_k = -\sum_{r\in I} \ln(1 - 2\gamma_k C_r) + MN \ln \sigma_k^2 + m_k \ln(MN).$$

8.2 The mutually compactible Gibbs random field model

We shall consider a contextual window consisting of three neighbors $I^*_{i,j} = \{(i,j),(i-1,j),(i-1,j-1),(i,j-1)\}$ and a constant mean global energy function

$$E\{Q\} = \sum_Y Q(Y)p(Y) = E_c, \tag{57}$$

where the global energy function is defined as

$$Q(Y) = \sum_{i=1}^{M} \sum_{j=1}^{N} Q_{ij}(Y_{i,j}, Y_{i-1,j}, Y_{i-1,j-1}, Y_{i,j-1}). \tag{58}$$

It is convenient to define the local transfer function (LTF) $\sigma_{i,j}$ as

$$\sigma_{i,j}(Y_l, Y_m, Y_n, Y_o) = \exp\{-Q_{i,j}(Y_l, Y_m, Y_n, Y_o)\}. \tag{59}$$

DEFINITION 1. A set of sites A is called the primary sublattice of lattice I if $A \subseteq I, \{(m,n) : 1 \leq m \leq M_A, n = 1\} \subseteq A, \{(m,n) : m = 1, 1 \leq n \leq N_A\} \subseteq A$ for some (M_A, N_A) with $1 \leq M_A \leq M, 1 \leq N_A \leq N$, and if $(m,n) \in A$, then $\{(i,k),(i-1,k),(i-1,k-1),(i,k-1)\} \subseteq A$.

DEFINITION 2. The GRF defined over a rectangular lattice I, and whose joint probability measure is given by eqs. (40),(58),(59) is called a mutually compactible Gibbs random field (MC-GRF) [32], if its restriction Y_A over any primary sublattice A of lattice I is also a GRF with a joint probability measure

$$p(Y_A) = 1/Z_A \prod \prod_{(i,j)\in A} \sigma_{i,j}(Y_{i,j}, Y_{i-1,j}, Y_{i-1,j-1}, Y_{i,j-1}), \tag{60}$$

where

$$Z_A = \sum_{Y_A} \prod \prod_{(i,j)\in A} \sigma_{i,j}(Y_{i,j}, Y_{i-1,j}, Y_{i-1,j-1}, Y_{i,j-1}). \tag{61}$$

THEOREM 1. *A necessary and sufficient condition for a GRF to be a MC-GRF is*

$$\sum_x \sigma_{ij}(x,t,z,y) = k_{ij}, \tag{62}$$

for $1 \leq i \leq M, 1 \leq j \leq N$ *and for every triplet* (t,z,y), *where* k_{ij} *is a constant independent of* (t,z,y).

THEOREM 2. *A GRF whose LTF satisfies (62) for every* (i,j), *is statistically equivalent to a unilateral MRF. A random field is statistically equivalent to a MC-GRF if and only if it is statistically equivalent to a unilateral MRF.*

DEFINITION 3. The GRF Y is called a GRF with a homogeneous LTF if

$$\sigma_{i,1}(Y_{i,1}, Y_{i-1,1}) = \sigma_v(Y_{i,1}, Y_{i-1,1}), \qquad for \quad i = 2,3,\ldots,M \tag{63}$$

$$\sigma_{1,j}(Y_{1,j}, Y_{1,j-1}) = \sigma_h(Y_{1,j}, Y_{1,j-1}), \qquad for \quad j = 2,3,\ldots,N \tag{64}$$

and

$$\sigma_{i,j}(Y_{i,j}, Y_{i-1,j}, Y_{i-1,j-1}, Y_{i,j-1}) = \sigma(Y_{i,j}, Y_{i-1,j}, Y_{i-1,j-1}, Y_{i,j-1}),$$

$$for \quad i = 2,3,\ldots,M, \qquad and \quad j = 2,3,\ldots,N. \tag{65}$$

THEOREM 3. *A GRF defined over a rectangular lattice* I *is a translation invariant GRF if and only if it is a MC-GRF with a homogeneous LTF such that:*

$$\sum_z \sigma_{11}(z)\sigma_h(v,z) = \sigma_{11}(v), \tag{66}$$

$$\sum_z \sigma_{11}(z)\sigma_v(y,z) = \sigma_{11}(y), \tag{67}$$

$$\sum_y \sigma_v(y,z)\sigma(x,t,z,y) = \sigma_v(x,t), \text{ for every } x,t,z \in I_A \text{ and} \tag{68}$$

$$\sum_t \sigma_h(t,z)\sigma(x,t,z,y) = \sigma_h(x,y). \tag{69}$$

Simulation of a translation invariant GRF is therefore based on simulation of a translation invariant MC-GRF. Simulation of a MC-GRF is a simplified task due to the causal type of model, the free boundary condition and the special form of global energy function (58). Therefore we can lexicographically generate a MC-GRF from a set of known LTFs. Simulation starts from specified values of σ_v, σ_h. The initial probability σ_{11} is computed from (66),(67), then the probability $\sigma(x,t,z,y)$ is calculated from (68),(69). It has to solve $2(K-1)K^2$ equations with $(K-1)^3$ unknowns. In the case of a translation invariant and isotropic GRF, the following additional constraints should be satisfied:

$$\sigma_v(v,z) = \sigma_h(v,z) = \sigma_*(v,z), \tag{70}$$

$$\sigma_*(v,z) = \sigma_*(z,v), \quad \textit{and} \tag{71}$$

$$\sigma(x,t,z,y)\sigma_*(t,z) = \sigma(t,x,y,z)\sigma_*(x,y) = \sigma(t,z,y,x)\sigma_*(x,y). \tag{72}$$

8.3 The binomial Markov random field model

We shall assume the conditional probability of point r having class k to be binomial:

$$p(Y_r = k | Y_s \forall s \in I_r) = \binom{K-1}{k} \theta^k (1-\theta)^{K-1-k}, \tag{73}$$

where

$$\theta = \exp(T)/(1+\exp(T)). \tag{74}$$

In the binary case ($K = 2$) the conditional probability is:

$$p(Y_r = k | Y_s \forall s \in I_r) = \exp(kT)/(1+\exp(T)). \tag{75}$$

A first-order model has the following form for T:

$$T_1 = a_1 + a_2(Y_{m,n-1} + Y_{m,n+1}) + a_3(Y_{m-1,n} + Y_{m+1,n}). \tag{76}$$

A second-order model is

$$T_2 = T_1 + a_4(Y_{m-1,n-1} + Y_{m+1,n+1}) + a_5(Y_{m-1,n+1} + Y_{m+1,n-1}), \tag{77}$$

etc.

Simulation of BMRF can be done for example using the Metropolis algorithm see Section 8.4. First we generate a random field from a uniform random generator, then by the Metropolis algorithm the random field will converge to our specified BMRF.

8.4 The Metropolis algorithm
Given the state Y^t of a MRF, another random configuration X is chosen. The ratio $\alpha = p(X)/p(Y^t)$ is computed. If $\alpha > 1$, then $Y^{t+1} = X$, otherwise the transition is made with probability α. A variable ξ is selected from a standardized uniform distribution, if $\xi \leq \alpha$ then $Y^{t+1} = X$, otherwise $Y^{t+1} = Y^t$. A modification of the algorithm is the Exchange algorithm [13], where X is obtained from Y^t by exchanging values of two randomly chosen pixels. The disadvantage of this method is its sensitivity to the initial configuration [12]. In the 'single-flip' [30] algorithm, X is obtained from Y^t by changing the value of one randomly chosen pixel.

8.5 The Gibbs sampler
The Gibbs sampler [30] generates realizations from a given MRF using a relaxation technique similar to the Metropolis algorithm. The stationary configuration Y^0 is arbitrary. By repeatedly visiting all sites (for example by raster scanning) we always replace one pixel with a value generated from the local characteristic of Gibbs distribution:

$$p(Y^t) = p(Y_r^t | Y_s^{t-1} \forall s \neq r) p(Y_s^{t-1} \forall s \neq r), \tag{78}$$

where

$$p(Y_r^t | Y_s^{t-1} \forall s \neq r) = \exp\{-Q_r^t(Y)\}/\tilde{Z}. \tag{79}$$

Convergence of the algorithm is assured by the relaxation theorem.

THEOREM. *Assume that for each site from a lattice the visit sequence contains this site infinitely often. Then for every starting configuration Y^0 and every configuration Y,*

$$\lim_{t \to \infty} p(Y^t = Y | Y^0) = \exp\{-Q(Y)\}/Z.$$

For a proof, see [30].

8.6 MRF parameter estimation
ML parameter estimation of the MRF model is complicated by the difficulty associated with computing the normalization constant Z. Generally we have K^{MN} possible realizations for which $\exp\{-Q\}$ has to be computed. Therefore, except for trivial tasks, it is necessary to use some approximation method, like for example the coding method. To overcome the difficulties caused by mutual correlation of lattice pixels, the coding method [6] codes lattice pixels to obtain independent lattice pixels in a given contextual neighborhood. Such a coding for a first-order scheme for example, is:

```
. × .
× . ×
. × .
```

Using such a coding system, variables associated with × and (.)-sites are according to the Markov assumption mutually independent. A disadvantage of this method is that the estimates thus obtained are not efficient [6, 40] due to

a partial utilization of the data. The different coding estimates for the same parameter can in some cases considerably differ.
ML estimate for (.)-sites is obtained from

$$\max_{\gamma} \prod_{r\in .} p(Y_r|\forall Y_s \in \times), \tag{80}$$

or

$$\max_{\gamma}\{-(\log \tilde{Z} + Y_r G_r + \sum_{s\in I_r} G_{r,s} Y_s Y_r)\}. \tag{81}$$

This maximization is nonlinear and an iterative solution is needed. For the case of a small neighborhood, few possible levels and a low order model, it is possible to approximate (81) by a set of linear equations, if we approximate $p(Y_r|Y_{(r)})$ by its frequency of occurrence.

8.6.1 GMRF parameter estimation. A normalization constant (41) is easy obtainable for a GMRF model, so we can obtain a ML estimate maximizing either (58) or (56). The variance ML estimate for $U = 0$ is:

$$\sigma^2 = \frac{1}{MN} \tilde{Y}^T \Psi \tilde{Y}. \tag{82}$$

To obtain Ψ estimation we have to solve

$$\min_{\Psi}\{-\frac{1}{MN} \ln|\Psi| + \ln(\tilde{Y}^T \Psi \tilde{Y})\}.$$

This estimator is nonlinear (determinant), so an iterative solution is needed. Another computationally attractive possibility is the coding method (83), where $\tilde{M}, \tilde{N}$ is the number of independent rows and columns, respectively. The likelihood function can be simplified:

$$L = p(Y) = \prod_{r=1}^{\tilde{M}\tilde{N}} p(Y_r|Y_s \forall s \in I_r)$$

$$= (2\pi)^{-\tilde{M}\tilde{N}/2} \sigma^{-\tilde{M}\tilde{N}} \exp\{\frac{-1}{2\sigma^2} \sum_{r=1}^{\tilde{M}\tilde{N}} (Y_r - \mu - \sum_{s\in I_r} a_s (Y_{r-s} - \mu))^2\}. \tag{83}$$

Parameters are determined from the following equations:

$$\frac{\delta \log L}{\delta a_s} = 0, \quad \text{and}$$

$$\frac{\delta \log L}{\delta \sigma^2} = 0.$$

From this equations we obtain:

$$\frac{1}{\tilde{M}\tilde{N}} \sum_{r=1}^{\tilde{M}\tilde{N}} \tilde{Y}_r \tilde{Y}_{r-s} = \sum_{s\in I_r} a_s \frac{1}{\tilde{M}\tilde{N}} \sum_{r=1}^{\tilde{M}\tilde{N}} \tilde{Y}^2_{r-s}, \tag{84}$$

$$\sigma^2 = \frac{1}{\tilde{M}\tilde{N}} \sum_{r=1}^{\tilde{M}\tilde{N}} (\tilde{Y}_r - \sum_{s \in I_r} a_s \tilde{Y}_{r-s})^2, \tag{85}$$

$$\mu = \mu_s = \frac{1}{\tilde{M}\tilde{N}} \sum_{r=1}^{\tilde{M}\tilde{N}} \tilde{Y}_r. \tag{86}$$

The final estimates of the parameters is the average value over all the possible codings (i.e. first-order process two codings, second-order process four codings, third and fourth-order nine codings, etc.) Another possibility is the pseudo-likelihood estimator

$$\max_{\gamma} \prod_{\forall r \in I} p(Y_r | Y_{(r)}). \tag{87}$$

The pseudolikelihood estimate for a_s parameters has the form

$$\gamma_a = [a_{10}, a_{01}, \ldots] = [\sum_{\forall r \in I} X_r^T X_r]^{-1} \sum_{\forall r \in I} X_r^T \tilde{Y}_r, \tag{88}$$

where

$$X_r = [\tilde{Y}_{i-1,j} + \tilde{Y}_{i+1,j}, \tilde{Y}_{i,j-1} + \tilde{Y}_{i,j+1}, \ldots]. \tag{89}$$

Under a torus structure, the likelihood function of Y is given [20] by

$$p(\tilde{Y}|\gamma) = \prod_{r \in I} (1/2MNS_{\tilde{Y}}(r))^{1/2} \exp\{-\sum_{r \in I} |\mathcal{F}\{\tilde{Y}_r\}|^2 / 2MNS_{\tilde{Y}}(r)\}. \tag{90}$$

The ML estimates of GMRF parameters are obtained by maximizing likelihood function (90), where $\mathcal{F}\{\tilde{Y}_r\}$ is defined in (55) and $S_{\tilde{Y}}(r)$ in (54).

Woods [62] has given an estimate for an $\bar{M} \times \bar{N}$ sublattice $\tilde{I}$:

$$I_B = \{r : r \in I \wedge \exists s \in I_r (r - s) \notin I\}, \qquad \tilde{I} = I - I_B,$$

$$\gamma_a = [\sum_{\forall r \in \tilde{I}} X_r^T X_r]^{-1} \sum_{\forall r \in \tilde{I}} X_r^T \tilde{Y}_r, \tag{91}$$

$$\sigma^2 = \frac{1}{\bar{M}\bar{N}} \sum_{r=1}^{\bar{M}\bar{N}} (\tilde{Y}_r - \gamma_a X_r^T)^2. \tag{92}$$

The estimate (91) was shown [40] to be asymptotically consistent.

8.6.2 ML estimation of MC-GRF parameters. The ML estimation of the LTF of a MC-GRF with a homogeneous LTF is [32]:

$$\sigma_h(i,j) = \frac{\nu_{i,j}}{\sum_{i=1}^{K} \nu_{i,j}}, \tag{93}$$

$$\sigma_v(i,j) = \frac{\epsilon_{i,j}}{\sum_{i=1}^{K} \epsilon_{i,j}}, \tag{94}$$

$$\sigma(i,j,k,l) = \frac{\eta_{i,j,k,l}}{\sum_{i=1}^{K} \eta_{i,j,k,l}}, \tag{95}$$

where $\nu_{i,j}$ is the number of observed transitions in the first picture row from state j to state i, $\epsilon_{i,j}$ is the corresponding number for the first column and $\eta_{i,j,k,l}$ is the number of observed transitions from states (j,k,l) to state i on remaining pixels. Estimations (93)-(95) are consistent, efficient and asymptotically Gaussian.

8.6.3 ML parameter estimation of binomial MRF. Using the coding technique, the log-likelihood function can be written in the following form:

$$L = \sum_{r=1}^{\tilde{M}\tilde{N}} \log p(Y_r | Y_s \forall s \in I_r)$$

$$= \sum_{r=1}^{\tilde{M}\tilde{N}} \{\log \binom{K-1}{i} + i(T - \log(1 - \exp T)) \tag{96}$$

$$+(K-1-i)\log(1 - \frac{\exp T}{1+\exp T})\},$$

$$\frac{\delta L}{\delta a_i} = \sum_{r=1}^{\tilde{M}\tilde{N}} \{i(\frac{\delta T}{\delta a_i} + \frac{\exp T}{1-\exp T}\frac{\delta T}{\delta a_i}) - (K-1-i)\frac{\exp T}{1+\exp T}\frac{\delta T}{\delta a_i}\}. \tag{97}$$

Maximalization of (97) is a demanding task using numerical methods. Another possibility is the MPL method, then $\tilde{M} = M$, $\tilde{N} = N$.

9 Experimental results and conclusions

Several possibilities exist for generating a given random field, but not all of them are equally suitable for implementation in a graphical system.

The main disadvantage of random mosaic models for generating a given random field is that they produce geometrical shapes (straight lines, rectangles, circles) which are not appropriate to model natural textures. They can be used for limited modelling of certain man-made textures. Periodic tessellated models can also be useful mainly for man-made regular structure generation. It is also possible to fill single cells with data from a random field generator. Syntactic models at the present state of art are not possible to use for general texture generation because of unsolved grammar inference problems. Fractal models have the advantage that they can be approximated by a single fractal function over a range of scales and the second-order statistics are described by a single parameter. Such properties sound promising, so further study would be necessary to find how credible natural textures modelled by fractals are, and in what range real fractals are scale invariant.

Second order models were based on the assumption [38] (later proven to be wrong) that the second order statistics are sufficient for a texture description. Although such an approximation is good enough for large amount of textures, the synthesis step is done iteratively and is time-consuming. Autoregressive models

(AR) are highly flexible to model natural textures; their advantage over equivalent Gaussian Markov random field models (GMRF) is their usually smaller number of parameters. On the contrary, GMRF are more general: for some GMRF models a finite order AR model does not exist. It is also possible to model textures using a causal autoregressive model, which can be generated directly and for which we can easily find ML or Bayesian parameter estimates, but in such a case we have to limit spatial correlation and introduce causal direction, which is in most 2D cases unnatural. AR models on a toroidal lattice can be generated using a fast Fourier transformation. There is a possibility to use more complicated ARMA models for texture generation, but their main problem is a highly nonlinear parameter estimation. A disadvantage of AR models with conditional expectations is a large number of parameters, which gives problems with identifying and storing them, as well as time consuming synthesis. Markov random fields are the most flexible (it can describe the most complex random fields from our surveyed models) from all described possibilities to model realistic textures. Problems of general MRF are difficulties with evaluation of a normalizing constant, iterative synthesis, and non-linear parameter estimation. An exception is the GMRF, whose normalizing constant is easy to evaluate and if we assume the toroidal lattice, then we can use a rapid FFT based synthesis algorithm. MC-MRF has the advantage of easy synthesis and analysis; its disadvantage is a large number of parameters.

The analytical step of a texture synthesizer is not critical, because in graphical applications it can be solved off-line. It is assumed that a graphical system will have a database of different textures parameters and will be able to use or modify textures from this basic sortiment. On the other hand, the synthesis step is done in real time, and it is therefore necessary to generate textures as quickly as possible. Iteration-based algorithms, like for example algorithms of the Metropolis type are not very handy, especially for larger data fields, because they are rather time-consuming. Another disadvantage is that although these algorithms converge to a specified field, little is known about their convergence rate. This problem of limited texture sortiment for a graphical system can be solved experimentally. The GMRF generator as well as the AR generator, based on the fast Fourier transformation, is especially advantageous because it needs only one inverse FFT to generate the desired random field, which is much faster than any other iterative generator. Another advantage of this method is the possibility to use a special FFT processor to build a real time GMRF or AR generator.

Although time is not important for the analytical step of a synthesizer, ML parameter estimation needs numerical optimization in most of the AR and MRF models, so we have no guarantee to reach a global optimum. In our experimentation the MPL method (or Woods method) appears to be a rapid and sufficient alternative, at least for lower order models. The coding method of Besag uses only a fraction of the available data and offers no clear advantage over MPL. Similarly, the iterative estimation method (7.4.2.) was slowly converging in several experiments.

We have tested [33] Gaussian Markov random field generators and simultaneous autoregressive model generators of relatively low order (up to fifth order)

with several exceptions of ninth order Gaussian Markov random fields to generate natural textures from Brodatz album. Our limited experimentation confirms our belief of large flexibility and usefulness for natural texture modelling of both models.

Figures 1-3 show different realizations of the ninth order GMRF, all were synthesized using the Gibbs sampler algorithm. Figure 1 is wood grain and Figure 2 calf-leather both modelled using parameters from digitized Brodatz album of textures [20]. Figure 4 is modelled using the third order GMRF model and the FFT synthesis method. Comparing our different simulations, we found that the convergence rate is different for each random field. While some models were already near their limiting random fields after 30 iterations, others needed 70 iteration steps and for others even this number was not sufficient. When we compared GMRF results with non causal symmetric AR fields generated using the Woods algorithm and with similar corresponding parameters we could see their mutual similarity demonstrating close relation between both models.

The texture synthesis methods we have discussed so far are possible building blocks for a realistic texture synthesizer, but any single mentioned algorithm will seldom present a complete solution of our problem. Realistic modelling of natural surfaces needs color texture synthesis and combining several texture synthesis methods. It is possible to model macro features using periodic tessellated models or simple syntactic models with micro textures generated using random field models. Another tool for combining macro and micro features is Fourier transformation. The resulting texture will finally be mapped onto a 3D surface. This is possible by determining the correspondence between each sample point on the 3D surface and its projection into a planar image. There are different techniques [4, 29] to solve this problem. Some of them have the drawback to introduce a non-controlled spatial distortion, others are limited to certain regular 3D surfaces, or need numerical optimization techniques.

References

1. N. Ahuja, A. Rosenfeld (1981). Mosaic models for textures. *IEEE Trans. Pattern Anal. Mach. Int. PAMI-3*, 1-11.
2. N. Ahuja, B.J. Schachter (1983). *Pattern Models*, J.Wiley.
3. M. Barnsley (1988). Fractals Everywhere, Academic Press.
4. C. Bennis, A.Gagalowicz (1989). Hierarchical texture synthesis on 3-D surfaces. W.Hansmann, F.R.A. Hopgood, W.Strasser (eds.). *Eurographics '89*, 257-269.
5. C. Bennis, A. Gagalowicz (1989). 2-D macroscopic texture synthesis, *Computer Graphics Forum 8*, 291-300.
6. J. Besag (1974). Spatial interaction and the statistical analysis of lattice systems. *J. Royal Stat. Soc. B-36*, 192-236.
7. J. Besag (1986). On the statistical analysis of dirty pictures. *J. Royal Stat. Soc. B-48*, 259-302.
8. C. Bouman, B. Liu (1991). Multiple resolution segmentation of textured images. *IEEE Trans. Pattern Anal. Mach. Int. PAMI-13*, 99-113.
9. D. Cano, T.H. Minh (1988). Texture synthesis using hierarchical linear transformation. *Signal Processing SP-15*, 131-148.

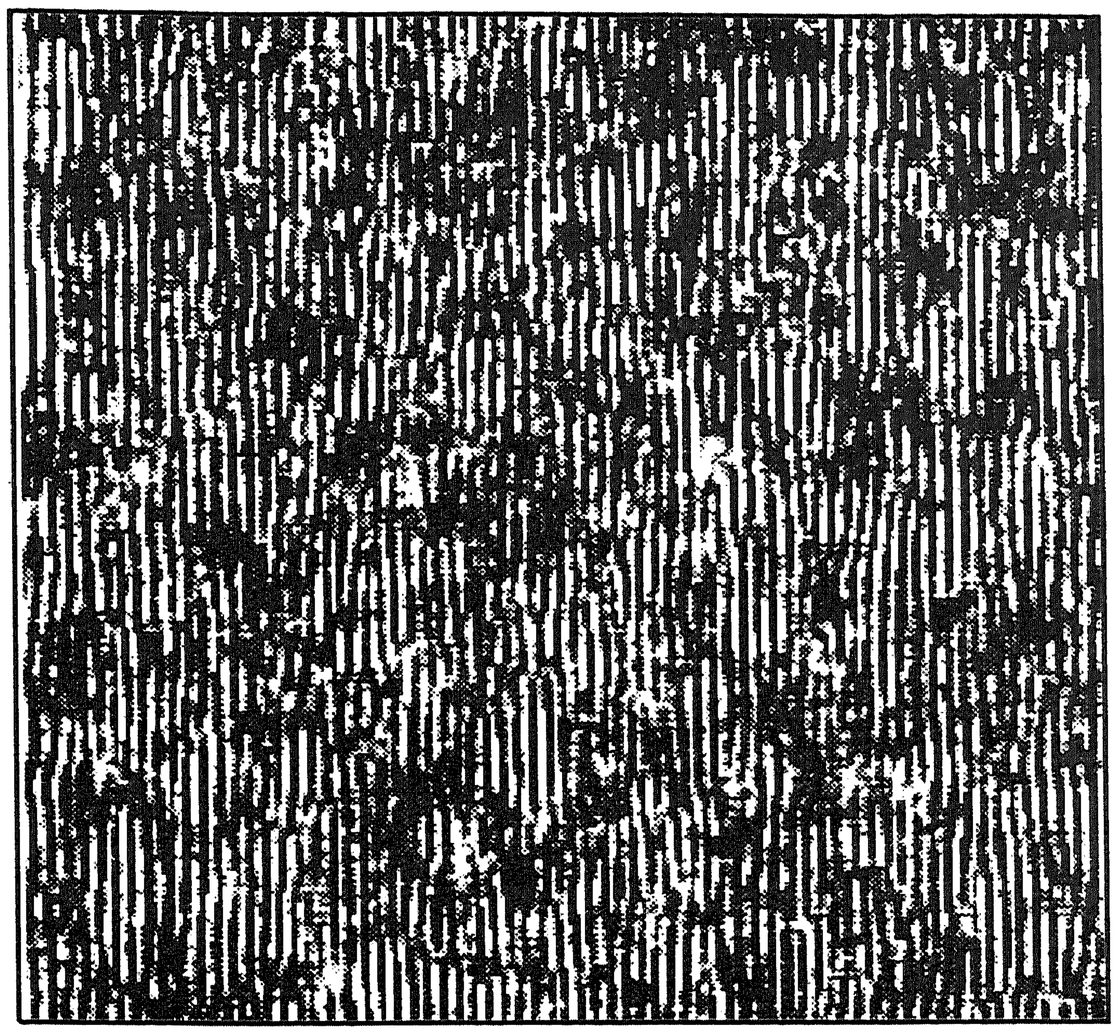

FIGURE 1

10. R. CHELLAPPA, R.L. KASHYAP (1982). Texture synthesis using spatial interaction models. *Proc. of IEEE Comp. Soc. Conf. on Pattern Rec., Inf. Proc.*, Las Vegas, 226-230.
11. R. CHELLAPPA (1985). Two-dimensional discrete Gaussian Markov random field models for image processing. L.N. KANAL, A. ROSENFELD (eds.). *Progress in Pattern Recognition 2*, Elsevier, North-Holland.
12. F.S. COHEN (1986). Markov random fields for image modelling and analysis. U.B. DESAI (ed.). *Modeling and Application of Stochastic Processes*, Kluwer Academic Publishers, Boston.
13. G.R. CROSS, A.K. JAIN (1983). Markov random field texture models. *IEEE Trans. Pattern Anal. Mach. Int. PAMI–5*, 25-39.
14. K. DEGUCHI, I. MORISHITA (1982). Two-dimensional auto-regressive model for the representation of random image fields. *Proc. 6-th Int.Conf. on Pattern Recognition*, Munich, 90-93.
15. S. DEMCO, L. HODGES, B. NAYLOR (1985). Construction of fractal objects with iterated function systems. *Computer Graphics 19*, 271-278.
16. H. DERIN, A.W. COLE (1986). Segmentation of textured images using Gibbs random fields. *Comp. Graphics Image Proc. CGIP-35*, 72-97.
17. H. DERIN, H. ELLIOT (1987). Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Trans. Pattern Anal. Mach. Int. PAMI-9*, 39-55..
18. N. DODD (1987). Multispectral texture synthesis using fractal concepts. *IEEE Trans. Pattern Anal. Mach. Int. PAMI-9*, 703-707.
19. G. ENGLERT, G. SAKAS (1989). A model for description and synthesis of

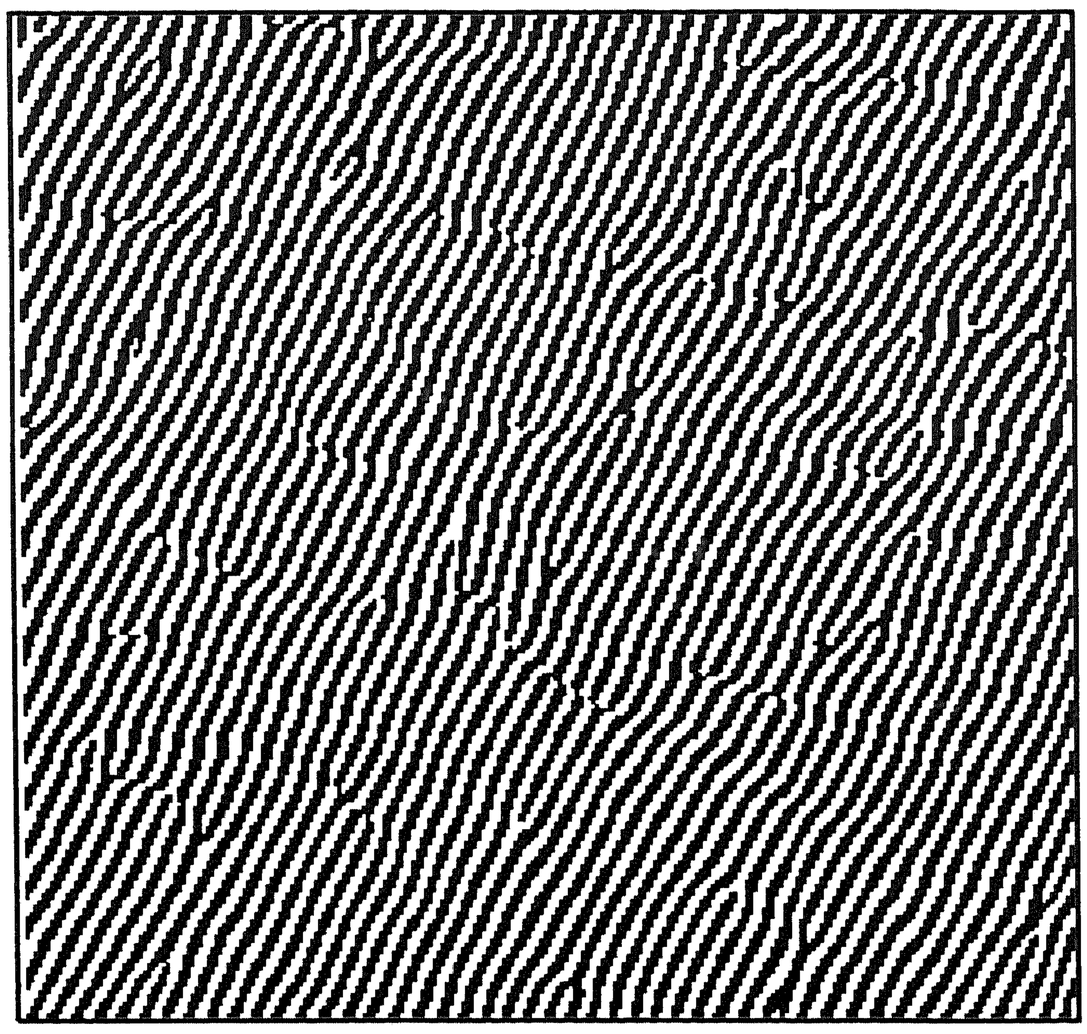

FIGURE 2

heterogeneous textures. W.HANSMANN, F.R.A. HOPGOOD, W.STRASSER (eds.). *Eurographics '89*, 245-255.

20. Z. FAN, F.S. COHEN, M.A. PATEL (1991). Rotated and scaled textured images using Markov random field models. *IEEE Trans. Pattern Anal. Mach. Int. PAMI-13*, 191-202.
21. O.D. FAUGERAS (1980). Autoregressive modeling with conditional expectations for texture synthesis. *Proc. 5-th Int. Conf. on Pattern Recognition*, Miami Beach, 782-792.
22. O.D. FAUGERAS, O.D. GARBER (1980). Algebraic reconstruction techniques for texture synthesis. *Proc. 5-th Int. Conf. on Pattern Recognition*, Miami Beach, 782-792.
23. J.M. FRANCOS, A.Z. MEIRI (1988). A unified structural-stochastic model for texture analysis and synthesis. *Proc. 9-th Int. Conf. on Pattern Recognition*, Washington, 41-46.
24. K.S. FU, S.Y. LU (1978). Computer generation of texture using a syntactic approach. *Computer Graphics 12*, 147-152.
25. K. FU (1980). Syntactic image modelling using stochastic tree grammars. *Comp. Graphics Image Proc. CGIP-12*, 136-152.
26. A. GAGALOWICZ, S. MA (1982). Synthesis of natural textures. *Proc. 6-th Int. Conf. on Pattern Recognition*, Munich, 1081-1086.
27. A. GAGALOWICZ, S. MA (1985). Sequential synthesis of natural textures. *Comp. Graphics Image Proc. CGIP-30*, 289-315.
28. A. GAGALOWICZ (1981). A new method for texture fields synthesis: some

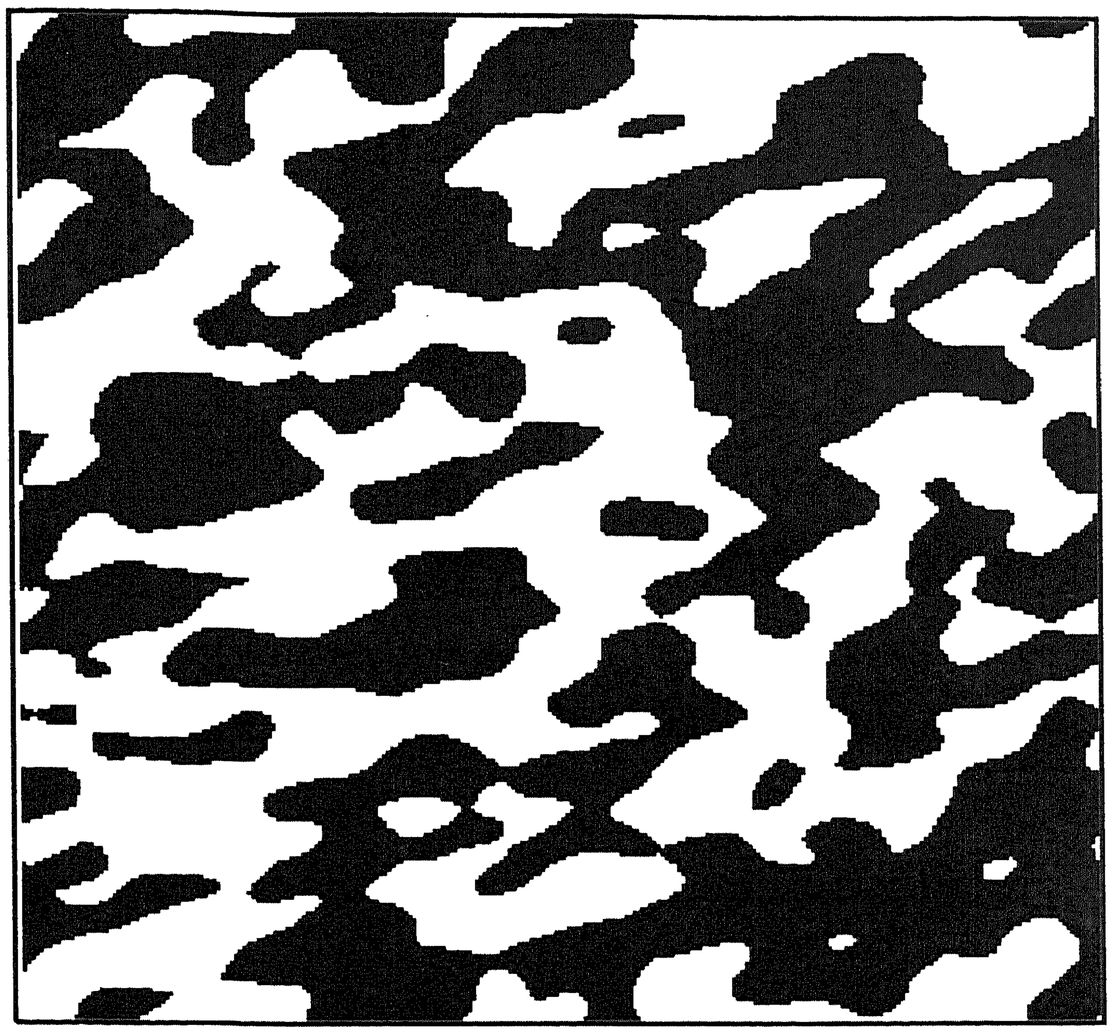

FIGURE 3

applications to the study of human vision. *IEEE Trans. Pattern Anal. Mach. Int. PAMI-3*, 520-533.

29. A. GAGALOWICZ (1987). Texture modelling applications. *Visual Computer 3*, 186-200.
30. S. GEMAN, D. GEMAN (1984). Stochastic relaxation, Gibbs distributions and Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Int. PAMI-6*, 721-741.
31. B. GIDAS (1989). A renormalization group approach to image processing problems. *IEEE Trans. Pattern Anal. Mach. Int. PAMI-11*, 164-180.
32. J. GOUTSIAS (1988). Mutually compactible Gibbs images: properties, simulation and identification. *Proc. ICASSP 88*, New York, vol. 2, 1020-1023.
33. M. HAINDL (1991). *Texture Synthesis*, *CWI Report CS-R9139*, Amsterdam.
34. J.M. HAMMERSLEY, D.C. HANDSCOMB (1964). *Monte Carlo Methods*, Methuen, London.
35. M. HASSNER, J. SKLANSKY (1980). The Use of Markov Random Fields as Models of Texture. *Comp. Graphics Image Proc. CGIP–12*, 357-370.
36. P.S. HECKBERT (1986). Survey of texture mapping. *IEEE Computer Graphics and Applications 11*, 56-67.
37. A. JAIN (1981). Advances in mathematical models for image processing. *Proc. IEEE 69*, 502-528.
38. B. JULESZ (1981). Textons, the elements of texture perception and their interactions. *Nature 290*, 91-97.
39. H. KANEKO, E. YODOGAWA (1982). A Markov random field application

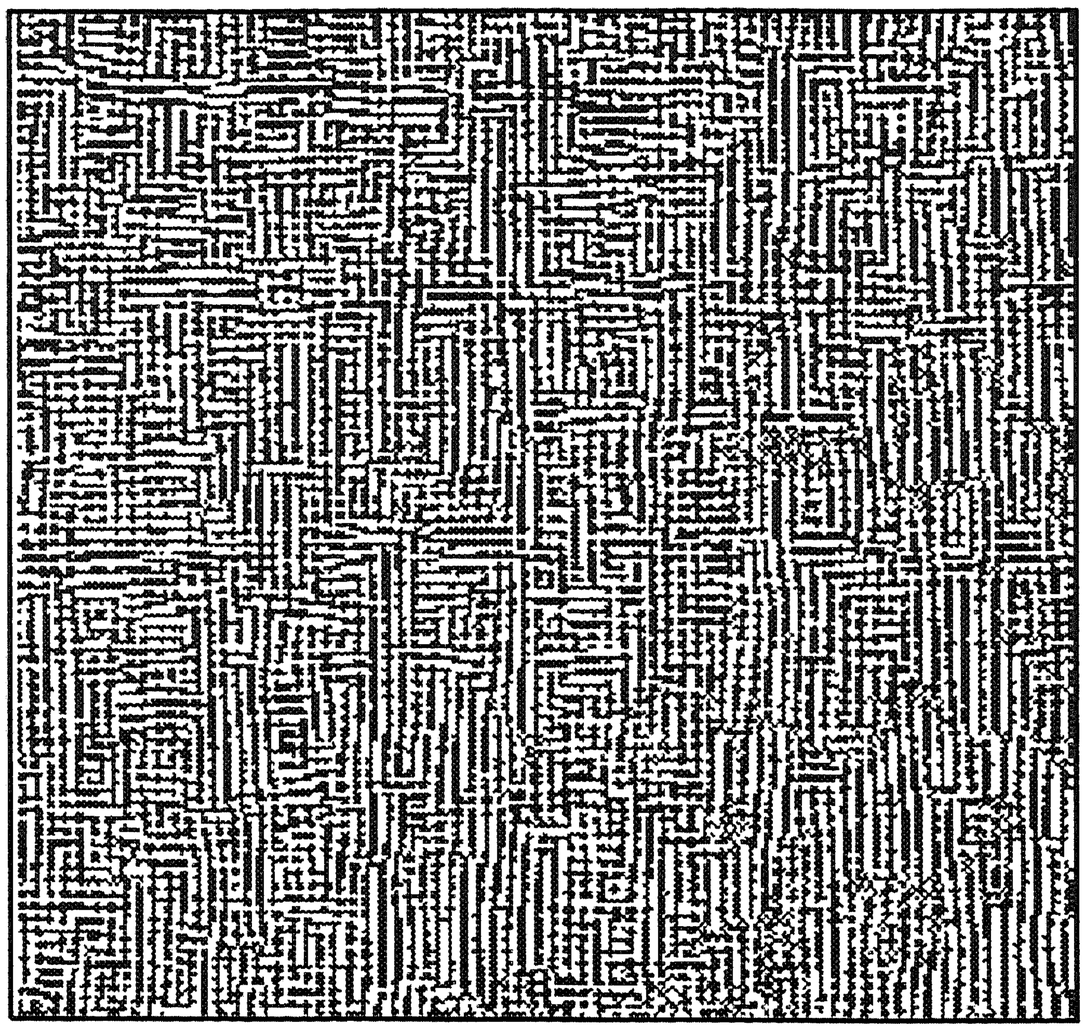

FIGURE 4

to texture classification. *Proc. IEEE Conf. Pattern Recognition and Image Processing*, Las Vegas.

40. R.L. KASHYAP, R. CHELLAPPA (1983). Estimation and choice of neighbors in spatial-interaction models of images. *IEEE Trans. Inf. Theory IT-29*, 60-72.
41. R.L. KASHYAP (1981). Analysis and synthesis of image patterns by spatial interaction models. L.N. KANAL, A.ROSENFELD (eds.). *Progress in Pattern Recognition 1*, Elsevier, North-Holland.
42. A. KAUFMAN (1988). TSL - a Texture Synthesis Language. *Visual Computer 4*, 148-158.
43. A. KAUFMAN, S. AZARIA (1985). Texture synthesis techniques for computer graphics. *Computers and Graphics 9*, 139-145.
44. R. KINDERMANN, J.L. SNELL (1980). *Markov Random Fields and Their Applications*, American Mathematical Society , Providence.
45. S.Y. LU, K.S. FU (1978). A syntactic approach to texture analysis. *Comp. Graphics Image Proc. CGIP-7*, 303-330.
46. N. MAGNENAT-THALMANN, D. THALMANN (1987). *Image Synthesis*, Springer, Tokyo.
47. B.B. MANDELBROT (1982). *The Fractal Geometry of Nature*, Freeman, San Francisco.
48. D.R. PEACHEY (1985). Solid texturing of complex surfaces. *Computer Graphics 19*, 279-286.
49. S. PELEG, J. NAOR, R. HARTLEY, D. AVNIR (1984). Multiple resolution texture analysis and classification. *IEEE Trans. Pattern Anal. Mach. Int.*

PAMI-6, 518-523.
50. A.P. PENTLAND (1984). Fractal-based description of natural scenes. *IEEE Trans. Pattern Anal. Mach. Int. PAMI-6*, 661-674.
51. K. PERLIN (1985). An image synthesizer. *Computer Graphics 19*, 287-296.
52. D.K. PICKARD (1977). A curious binary lattice process. *J. Appl. Probability 14*, 717-731.
53. W.K. PRATT, O.D. FAUGERAS, A. GAGALOWICZ (1981). Application of stochastic texture field models to image processing. *Proc. IEEE 69*, 542-551.
54. CH.J. PRESTON (1974). *Gibbs States on Countable Sets*, Cambridge Univ. Press, London.
55. W. QIAN, D.M. TITTERINGTON (1991). Multidimensional Markov chain models for image textures. *J. Royal Stat. Soc. B-53*, 661-674.
56. B.J. SCHACHTER, A. ROSENFELD, L.S. DAVIS (1978). Random mosaic models for textures. *IEEE Trans. Syst. Man Cyb. SMC-8*, 694-702.
57. B.J. SCHACHTER, N. AHUJA (1979). Random pattern generation processing. *Comp. Graphics Image Proc. CGIP-10*, 95-114.
58. J. SERRA (1989). *Image Analysis and Mathematical Morphology*, Academic Press.
59. D. STOYAN, W.S. KENDALL, J. MECKE (1989). *Stochastic Geometry and its Applications*, J. Wiley.
60. R.F. VOSS (1985). Random fractal forgeries. R.A. EARNSHAW (ed.). *Fundamental Algorithms for Computer Graphics*, NATO ASI Series F, vol. 17, Springer, Heidelberg.
61. J.J. VAN WIJK (1991). Spot noise texture synthesis for data visualization. *Computer Graphics 25.*
62. J.W. WOODS (1972). Two-dimensional discrete Markovian fields. *IEEE Trans. Inf. Theory IT-18*, 232-240.