TECHNICAL REPORT: 95-10

"Gray Level Texture Generation by Binary Markov Random Field Model with Morphological Operations"

Osman Burçhan Bayazıt
Levent Erkök
Okan Uludağ
Fatoş Yarman Vural

E-mail: {burc, erkok, vural}@ceng.metu.edu.tr
okan@rorqual.cc.metu.edu.tr

Department of Computer Engineering
Middle East Technical University

July 10, 1995

Abstract

In this report a new method for gray level texture generation which utilizes a binary texture generation model together with some morphological operations is proposed and discussed. A software that implements these ideas is described and the experimental results are given.

Keywords : Binary and gray level textures, Texture generation and modeling, Markov Random Field, Metropolis algorithm, Blurring, Histogram Equalization.

Contents

List of Figures

1 Introduction

Texture is an important characteristic for the analysis of many types of images. There is no universally accepted definition of texture but it can be considered to be a stochastic, possibly periodic, two dimensional image field.

Generating and modeling textures which resemble the real ones is very useful in computer science. Markov Random Field is a well known method used for texture generation and modelling. Since Ising's thesis [1], there has been a huge amount of work that has been published. Although, it is basically related to physics, it has been successfully used in texture modeling of nature. Markov random field model is established on the relationship of single points to its neighbors, where the definition of the neighborhood relation is a major characteristic of the model. This relationship is a stochastic process and it is usually assumed to be the autobinomial distribution, which is first formulated by Besag [2].

As explained by Cross et al. [3], for the generation of synthetic textures using MRF model, an algorithm invented by Metropolis et al. [4] can be used. The algorithm uses a number of parameters which defines the order of the texture. This process is reversible, i.e. given an unknown texture that has been generated by MRF model, or that fits the MRF model, it is possible to estimate the original parameters. The parameters can, also, be used to classify the textures, as explained in [5, 6].

A major problem of the algorithm is that, although it needs only 5 parameters for binary texture generation, for gray level textures the computational complexity of the problem becomes too high for all practical purposes. In this report we propose a new method for gray level texture generation that will utilize the binary MRF model, thus avoiding the complexity of gray level texture generation.

The proposed method, first, applies binary MRF model and then uses some morphological operations that will obtain the binary textures to gray level ones without any need to other parameters.

The following sections describe and illustrate these ideas. Section 2, gives the theoretical background for MRF texture model. Section 3 introduces the gray level texture model. Section 4, presents the experimental results and Section 5 conclude our findings.

2 Markov Random Field Model

The brightness level at a point in a textured image is highly dependent on the brightness levels of the neighboring points unless the image is simply random noise. Markov Random Field is a precise model of this dependence.

A formal discussion follows from that of Besag [2] and Bartlett [7].

The brightness level at a point i, j on the $N \times N$ lattice L is denoted by $X(i,j)$. For simplicity, labeling of $X(i,j)$ is translated to $X(i)$ where $i = 1, 2, \ldots, M$ and $M = N^2$.

Definition 1: Let L be a lattice. A *coloring* of L (or a coloring of L with G levels) denoted X is a function from the points of L to the set $\{0, 1, \ldots, G-1\}$. The notation **O** denote the function that assigns each point of the lattice to 0.

Definition 2: The point j is said to be a neighbor of point i, if

$$p(X(i) \mid X(1), X(2), \ldots, X(i-1), X(i+2), \ldots, X(M))$$

depends on $X(j)$.

Definition 3: A *Markov Random Field* is a joint probability density on the set of all possible colorings X of the lattice L, subject to following conditions:

1. *Positivity*: $p(X) > 0$ for all X.
2. *Markovianity*:

   $$p(X(i) \mid \text{all points in the Lattice except } i) = p(X(i) \mid \text{neighbors of } i)$$

3. *Homogeneity*: $p(X(i) \mid \text{neighbors of } i)$ depends only on the configuration of neighbors and is translation invariant.

2.1 Neighborhood

Usually neighborhood is defined as in Figure 1. In this definition point j is chosen to be a neighbor of any point i if they are close enough. The order of the MRF is related to the neighborhood definition used; this project concentrates on second order models as defined in the next section.

	o1	m	q1	
o2	v	u	z	q2
l	t	X	t'	l'
q1'	z'	u'	v'	o1'
	q2'	m'	o2'	

Figure 1: Neighbors of the point X

2.2 Order Dependence

The probability $p(X = k \mid \text{neighbors})$ is binomial with parameter $\theta(T)$ and number of tries $G - 1$ where G is the number of gray levels. The value of T determines the order of the texture.

The value of T depends on the neighborhood and the parameters of the model. For a second order model the corresponding relations are:

$$T = a + b_{11}(t + t') + b_{12}(u + u') + b_{21}(v + v') + b_{22}(z + z')$$

and,

$$\theta(T) = \frac{\exp(T)}{1 + \exp(T)}$$

The formal definition of order can be given as follows:

Definition 4: The *order* of a Markov random field process on a lattice is the largest value of i such that b_{i1} or b_{i2} is nonzero.

2.3 Metropolis Algorithm

As mentioned before, the Metropolis algorithm (see Figure 2) can be used to produce textures according to the MRF model. The properties of the algorithm can be stated as follows:

- The convergence to unit distribution is unaffected by the choice of the initial configuration; only the rate at which equilibrium is reached depends on the choice of the initial configuration.

- Given a coloring X, we choose the next coloring Y to be the same as X except that the gray values of two randomly selected points are interchanged. If the probability of next coloring is greater than the current one, the exchange is accepted, otherwise rejected.
- A theorem of Hammersley [8] guarantees that the application of the metropolis algorithm in Figure 2 will converge to a lattice with the desired distribution.

The metropolis algorithm can be stated as follows:

```
{ Algorithm for generating MRF with joint probability
 function p(X). The coloring Y is obtained from
 the coloring X by switching the values of the
 points X(1) and X(2).
}


while not STABLE do
     begin
         choose sites X(1), X(2) with X(1) <> x(2);
         r := P(Y)/P(X);
         if r >= 1
              then switch X(1), X(2)
              else
                  begin
                       u := uniform random on [0,1];
                       if r > u
                          then switch X(1), X(2)
                          else retain X
                  end
     end;
```

Figure 2: Metropolis Algorithm

3 Generating Gray Level Textures from Binary Counterparts

In principle, the metropolis algorithm given in the previous section can be generalized to produce gray level textures. But this makes computational complexity of the problem too high to be used in efficient systems. On the other hand, some morphological operations might be used to transform the binary texture into a gray level one as described in this section. This will have the effect of reducing the complexity to those of binary texture generation.

3.1 Basic Idea

Following block diagram represents the process pictorially:

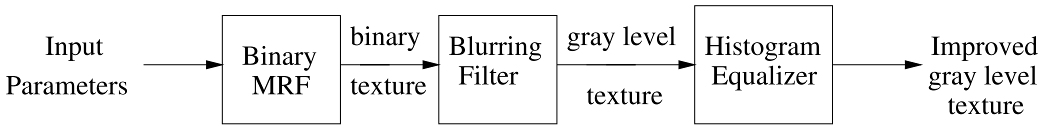

Figure 3: Block diagram representation of generating gray level textures

In this model, we propose morphological operations to be carried out in order to obtain gray level textures. The input parameters are the binary Markov Random Field model, the output is fed to a blurring filter in order to obtain the desired gray level texture. After this stage, a histogram equalizer is employed to improve the resulting gray level texture. The following sections discuss these steps.

3.2 Binary MRF phase

In the first phase for gray level texture generation, we have used binary markov random field model to generate binary textures. The algorithm is the previously described metropolis algorithm and its source can be seen at the end of this report.

The binary case MRF, where the point variables have range $\{0, 1\}$, is a special case of the Besag's autobinomial model. The conditional probability of x is expressed as:

$$p(X = x \mid T) = \frac{\exp(xT)}{1 + \exp(T)}$$

This expression is substituted in the Metropolis algorithm (Figure 2) when calculating $p(Y)/p(X)$. As a result the formula becomes:

$$\frac{p(Y)}{p(X)} = \prod_{i=1}^{M} \frac{p(X(i) = y(i) \mid X(1), X(2), \ldots, X(i-1), Y(i+1), \ldots, Y(N))}{p(X(i) = x(i) \mid X(1), X(2), \ldots, X(i-1), Y(i+1), \ldots, Y(N))}$$

Number of iterations to guarantee the stability condition in the metropolis algorithm has been found to be 3 by experimentation for a 64×64 image. This process takes only a few seconds on a Sparc-10 workstation.

3.3 Morphological Operations

After the original binary textures has been created by the binary MRF model described in the previous sections, we have applied some morphological operations to obtain the gray level correspondents. These operations are convolving the image by the blurring filter and histogram equalization.

3.3.1 Blurring

The basic idea behind blurring is averaging each points gray level value with its neighbors. If a white pixel is expressed as decimal 255 and black pixel as 0, after a blurring operation the pixel values takes values between 0 and 255, depending on the neighboring pixel values.

There are many ways for performing averaging operation, which depends on the definition of the convolution matrix and the neighboring function.

For this operation, we convolved the entire binary texture with the following matrix in figure 4, which simply performs arithmetic mean operation.

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Figure 4: Blurring Matrix

The first blurring operation creates a gray level image which has 12 to 15 different colors. It is possible to reapply the same operation on the resulting image to get more colors and this can go several times. It should be noted that the number of application is very important since as it increases the resulting images become distorted and the texture property gets lost. In this project we applied blurring only once.

The source code for the blurring operation can be seen at the end of this document.

3.3.2 Histogram Equalization

In general, the output of the blurring operation contains gray levels that are close to each other. For instance, the gray level value range turns out to be 50-90. So it is a good idea to equalize the histogram so that the resulting gray levels are mapped to the 0-255 range with equal distances from each other.

To do this, the algorithm first determines the number of colors then sorts them in increasing order. The minimum color is then mapped to zero and the maximum one to 255. The intervening values are, also, mapped according to a linear distribution. This results in more realistic textures.

The source code for the histogram equalization operation can be seen at the end of this document.

4 Test Results

We have applied the ideas mentoined in the previous sections and generated five classes of textures. Each class has 5 samples. The following sections contains the parameters used and the textures generated.

In the figures that show the textures, the first row shows the binary textures, and the second row contains the corresponding gray level ones.

The parameters given in each class effect the shape of the texture. For example, in a second order model positive values of the clustering parameter b_{11} cause clustering in the horizontal direction whereas b_{12} controls clustering in the vertical direction. The b_{21} and b_{22} control clustering in the diagonal directions. Positive values of these parameters cause attraction; negative values result in repulsion or a checkerboard effect.

4.1 Class 1: Clustering Effect

The class parameters are:

$$a = 0, b_{11} = 0, b_{12} = 0, b_{21} = -0.3, b_{22} = -0.3$$

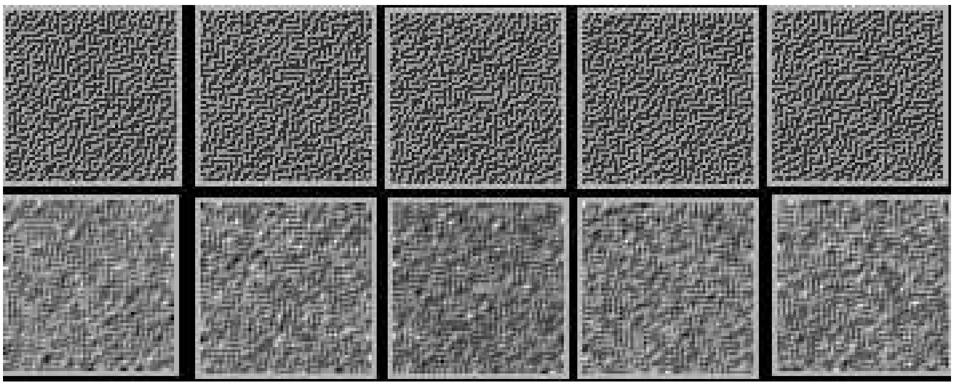

Figure 5: Clustering Effect

4.2 Class 2: Attraction Repulsion Effect

The class parameters are:

$$a = 0, b_{11} = 0.5, b_{12} = 0.5, b_{21} = -0.8, b_{22} = -0.8$$

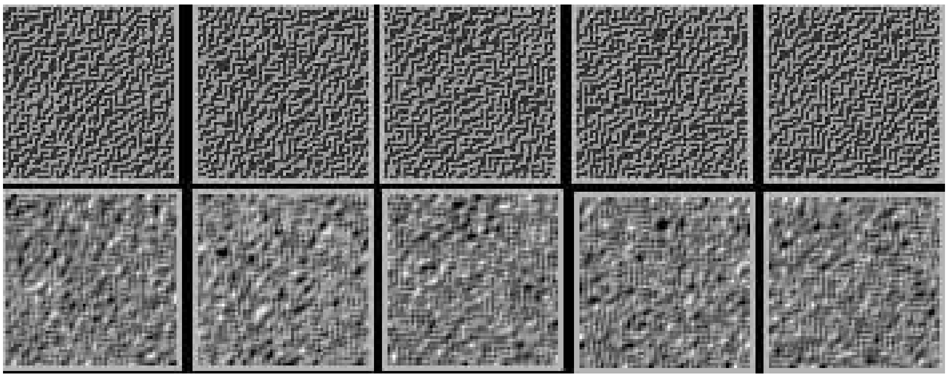

Figure 6: Attraction Repulsion Effect

4.3 Class 3: Horizontal, vertical and diagonal effect

The class parameters are:

$$a = -0.1, b_{11} = 0.3, b_{12} = 0.3, b_{21} = 0.4, b_{22} = -0.4$$

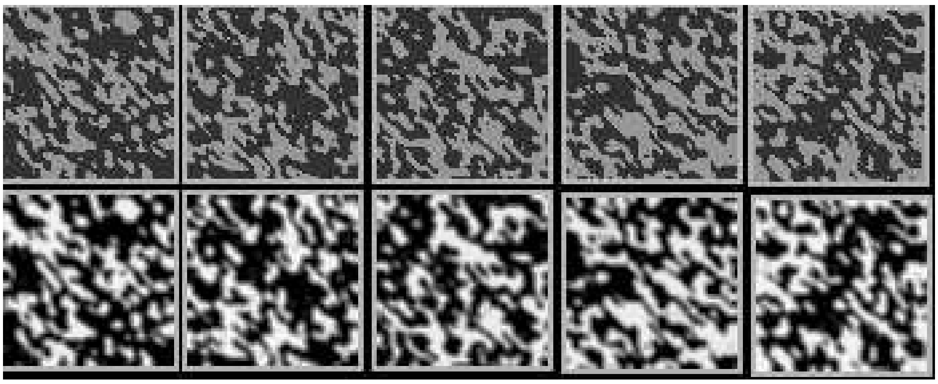

Figure 7: Horizontal, vertical and diagonal effect

4.4 Class 4: Vertical Effect

The class parameters are:

$$a = -0.26, b_{11} = -2, b_{12} = 2.1, b_{21} = 0.13, b_{22} = 0.015$$

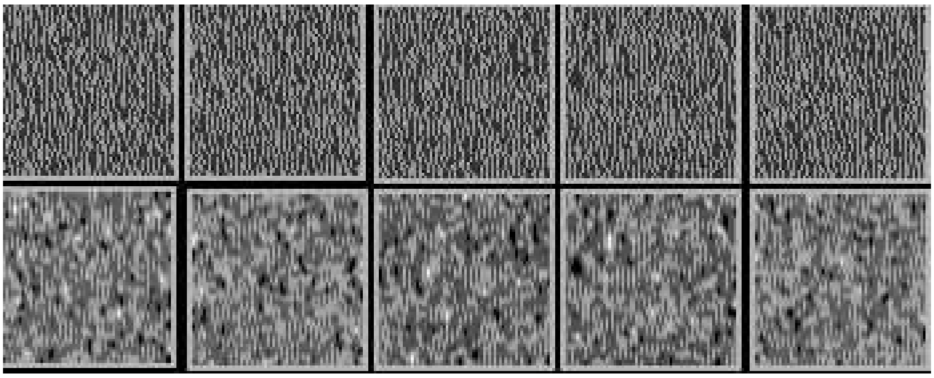

Figure 8: Vertical Effect

4.5 Class 5: Diagonal Effect

The class parameters are:

$$a = -1.9, b_{11} = -0.1, b_{12} = 0.1, b_{21} = 1.9, b_{22} = -0.075$$

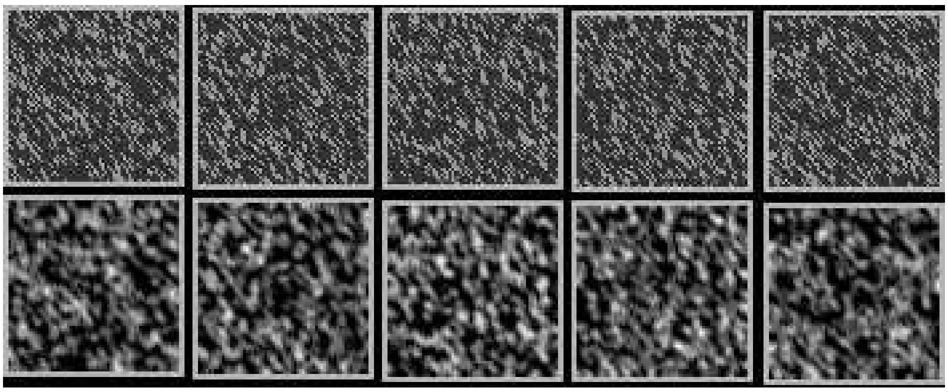

Figure 9: Diagonal Effect

5 Conclusion

In this study, a new method for gray level texture generation has been proposed. The idea is to post process the binary textures to generate gray scale textures via morphological operations.

Autologistic binary MRF model generates textures using 5 parameters for the textures to be generated. When the method is improved for generating gray level textures the complexity of the algorithm becomes too high that can not be afforded for efficiency reasons. The morphological operations that we propose, on the other hand, avoids this requirement and solves the problem in time comparable to those of binary texture generation.

The process described here has been applied and the generated textures has been presented in section 4. The results are quite satisfactory, the newly generated textures have 14 to 16 colors and they preserve the same texture with the original ones. This demonstrates that the resulting gray level textures are good models of MRF, in spite of the fact that it uses the same number of parameters as in the binary MRF model.

The proposed method can be fully used for modelling the gray level textures by using binary MRF models. Therefore the following future work should be accomplished:

- The resulting textures should be formally investigated to show that they satisfy the conditions given in section 2 of this report, i.e. it should be shown that they have MRF property.
- Another work should be done to show that this process is reversible, i.e. given an MRF gray level texture, is it possible to obtain the parameters of the corresponding binary texture? Intuitively the answer to this question seems to be yes, involving the use of trial-error that will find the original binary image. A better algorithm can be found after mathematical treatment of the problem.

As a final note, it should be emphasized that, this is a preliminary study for texture generation and modeling problem. Further work is required to prove efficiency and convergence of the processes described in this report.

References

[1] E. Ising, Zeitschrihft Physik vol. 31, p.253, 1925.

[2] J. Besag, *"Spatial Interaction And The Statistical Analysis Of Lattice Systems (with discussion)"*, J. Royal Statis. Soc., series V, vol. 36, pp. 192-326, 1974.

[3] G. R. Cross and A. K. Jain, *"Markov Random Field Texture Models"*, IEEE Transac. on Pattern Anal. and Machine Int., vol. PAMI-5, no. 1, pp. 25-39, 1983.

[4] N. Metropolis, A. W. Rosenblatt, M. N. Rosenblatt, A. H. Teller, E. Teller, *"Equations of state calculations by fast computing machines"*, J. Chan. Phys., vol. 21, pp. 1087-1091, 1953.

[5] R. Chellappa, S. Chatterjee, *"Classification of Textures Using Gaussian Markov Random Fields"*, IEEE Transac. on Acoustics, speech and signal proc., vol. ASSP-33, no. 4, pp. 959-963, 1985.

[6] O. Şener, *"Application of Markov Random Field To Textured Images"*, Master Thesis, METU, 1990.

[7] M. S. Bartlett, *"The Statistical Analysis Of Spatial Pattern"*, London: Chapman and Hall, 1976.

[8] J. M. Hammersley and D. C. Handscomb, *"Monte Carlo Methods"*, London: Methuen and Company, 1964.