Least squares image texture analysis and synthesis

Trygve Randen and John Håkon Husøy
Rogaland University Center
Department of Electrical and Computer Engineering
P.O. Box 2557 Ullandhaug, N-4004 Stavanger, Norway

Abstract

Stochastic texture modelling is a useful tool for revealing the mechanisms underlying the generation of natural textures. This paper presents a description of the calculation of the stochastic texture parameters, when we assume an autoregressive model. The parameters are estimated with a least squares estimator. The paper also presents an approach to texture synthesis using the estimated parameters. Experiments and comparisons of the results from both a causal and a non-causal model are presented.

1 Introduction

Texture modelling is a useful tool in several applications, such as segmentation/classification [1, 2], image coding [3] and texture synthesis [3, 4, 5]. See Rosenfeld [6] for a review article on image modelling.

Haralick [7] and Van Gool [8] divide the techniques for texture description into two main categories: statistical and structural. Most natural textures can not be described by any structural placement rule, therefore the statistical methods are usually the methods of choice. One possible approach to reveal many of the statistical texture properties is by modelling the texture as an autoregressive (AR) stochastic process, using least squares parameter estimation. Letting s and r be coordinates in the 2-D coordinate system, a general causal or non-causal auto-regressive model may be written

$$y(s) = \sum_{r \in N} \theta_r y(s-r) + e(s), \tag{1}$$

where $y(s)$ is the image, θ_r are the model parameters, $e(s)$ is the prediction error process, and N is a neighbor set. The usefulness of this modelling is demonstrated with experiments showing that it is possible to create synthetic textures with visual properties similar to natural textures.

Several papers have approached the analysis and synthesis of textures, using least squares methods [2, 3, 4, 9]. This paper gives a formulation of the analysis and synthesis from a digital signal processing point of view, using notations from linear algebra where applicable.

The concept of causality is tightly connected to the concept of time. Thus causality is not obvious for still-images. However, the synthesis with causal models is conceptually simpler than with non-causal models. Most articles on image modelling restrict their treatment to non-causal models, and no comparison with causal models is made. In this paper we put special emphsis on the comparison of causal and non-causal models, and their ability to generate synthetic textures.

2 Least squares parameter estimate

The first task in modelling the random field $y(s)$ is the determination of the model parameters θ_r. Several approaches have been presented, including a coding scheme [10], maximum likelihood [11, 12] and least squares [2, 3, 4, 13]. The least squares method will be examined further in this section.

2.1 The least squares formulation for 1-D signals

We will first give a brief review of the vector/matrix based formulation of the least squares criterion. The error in estimating the current sample, $y(n)$, with a 1-D non-causal model is given by

$$e(n) = y(n) - \sum_{l=-M, l\neq 0}^{M} \theta_l y(n-l),$$

here the neighborhood is $N = \{-M, \cdots, -1, 1, \cdots, M\}$. If we substitute some values for n, we get

$$\begin{aligned}
e(0) &= y(0) - \sum_{l=-M}^{M} \theta_l y(0-l), \\
e(1) &= y(1) - \sum_{l=-M}^{M} \theta_l y(1-l), \\
e(2) &= y(2) - \sum_{l=-M}^{M} \theta_l y(2-l), \\
&\vdots \\
e(L-1) &= y(L-1) - \sum_{l=-M}^{M} \theta_l y(L-1-l),
\end{aligned}$$

which may be written in vector/matrix form as $\mathbf{e} = \mathbf{y} - \mathbf{Y}\boldsymbol{\theta}$, where

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_{-M} & \cdots & \theta_{-1} & \theta_1 & \cdots & \theta_M \end{bmatrix}^T,$$

$$\mathbf{y} = \begin{bmatrix} y(0) & y(1) & y(2) & \cdots & y(L-1) \end{bmatrix}^T,$$

and

$$\mathbf{Y} = \begin{bmatrix}
y(M) & \cdots & y(1) & y(-1) & \cdots & y(-M) \\
y(M+1) & \cdots & y(0) & y(-2) & \cdots & y(-M+1) \\
y(M+2) & \cdots & y(-1) & y(-3) & \cdots & y(-M+2) \\
y(M+3) & \cdots & y(-2) & y(-4) & \cdots & y(-M+3) \\
\vdots & & & & & \vdots \\
y(M+L-1) & \cdots & y(-L+2) & y(-L) & \cdots & y(-M+L-1)
\end{bmatrix} \tag{2}$$

The mean square error can then be found by minimizing $\mathbf{e}^T\mathbf{e}$. Finding the $\boldsymbol{\theta}$ which yields the minimum squared error entails the solution of

$$\frac{\partial \mathbf{e}^T\mathbf{e}}{\partial \boldsymbol{\theta}} = 0,$$

where T denotes transpose. The solution is

$$\mathbf{Y}^T\mathbf{Y}\boldsymbol{\theta} = \mathbf{Y}^T\mathbf{y},$$

which is commonly known as the *normal equations*. The normal equations can be solved by Cholesky decomposition or other methods [14]. When implementing this method for a finite dataset, we extend the signal $y(n)$ periodically.

2.2 Extension to 2-D signals

The error signal for a two-dimensional signal and model is defined as

$$e(s) = y(s) - \sum_{r\in N} \theta_r y(s-r), \tag{3}$$

where s and r are two-dimensional coordinates. Substituting for different values of s, in the same way as for 1-D signals, we define $\boldsymbol{\theta}$ to be a vector of all the θ_r's supported by the neighbor set. For an image of $I \times J$ pixels $\boldsymbol{y}$ is defined as the vector obtained by lexicographically ordering the rows of the image $y(s)$,

$$\mathbf{y} = \left[\begin{array}{cccccccc} y(1,1) & y(1,2) & \cdots & y(1,J) & \cdots & y(I,1) & y(I,2) & \cdots & y(I,J) \end{array} \right]^T. \tag{4}$$

Remember that since the image has finite duration, we use periodic extension in the forming of $\mathbf{y}$.

Unfortunately, the generation of $\mathbf{Y}$ is not obvious. In the one-dimensional case, Eq. (2), with neighborhood $N = \{-M, \cdots, -1, 1, \cdots, M\}$, $\mathbf{Y}$ were formed as the matrix where $\mathbf{y}$ shifted by $-M, \cdots, -1, 1, \cdots, M$ were the columns. Each column then corresponds to one parameter, e.g. the column with $\mathbf{y}$ shifted by k corresponds to parameter θ_k.

For a two-dimensional model, we have to form $\mathbf{Y}$ in a similar way. Thus, the column corresponding to the parameter θ_r, where $r = (i,j)$ will be made of the periodic extension $\mathbf{y}$ starting with element (i,j).

The solution to the optimization problem, the $\boldsymbol{\theta}$ that yields the minimum squared error, can now be found by solving the normal equations

$$\mathbf{Y}^T\mathbf{Y}\boldsymbol{\theta} = \mathbf{Y}^T\mathbf{y}. \tag{5}$$

Solving this equation set by Cholesky factorzation has complexity $\mathcal{O}\left(|N|^3\right)$, where $|N|$ is the size of the neighbor set.

This approach to the least squares parameter estimation is basically the same as the approaches by Chellappa et.al. [2, 3, 4] and Kok et.al. [13]. Kok et.al. [13] gives a comprehensive optimization procedure to the solution of the normal equations (5).

3 Texture synthesis

The generation of a synthetic texture based on the parameters $\boldsymbol{\theta}$ is called *synthesis*. Thus the synthesis is the process of generating an image with pixels $y(s)$ which obey the equation

$$y(s) = \sum_{r \in N} \theta_r y(s-r) + e(s). \tag{6}$$

As mentioned in the introduction, the synthesis of causal textures is conceptually simpler than the synthesis of non-causal textures. Causal textures may be generated by space-domain filtering of a white noise image through the AR model, using an all-pole two-dimensional filter. Because of the non-causality, this is not possible for non-causal models. Therefore, we have to apply a different method for the synthesis of non-causal models.

Before we proceed, we have to make a couple of assumptions which will be necessary later. We assume our neighborhood to be symmetric, which yields symmetric AR parameters [10]. We further assume $e(s)$ to be zero mean correlated (colored) noise with

$$E\{e(r)e(s)\} = \begin{cases} \sigma_e^2 & \text{for} r = s \\ -\theta_r\sigma_e^2 & \text{for} r - s \in N \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

We also assume wide sense stationarity (WSS) and get

$$\begin{aligned} R_{ye}(q) &= E\{y(s)e(s+q)\} \\ &= E\left\{\sum_{r \in N} \theta_r y(s-r)e(s+q) + e(s)e(s+q)\right\} \\ &= \sum_{r \in N} \theta_r R_{ye}(q-r) + R_{ee}(q). \end{aligned}$$

If we take the z-transform of this, we get

$$S_{ye}(z) = \sum_{r \in N} \theta_r z^{-r} S_{ye}(z) + S_{ee}(z)$$

or

$$S_{ye}(z) = \frac{S_{ee}(z)}{1 - \sum_{r \in N} \theta_r z^{-r}}.$$

Using Eq. (7) we get

$$S_{ee}(z) = \sigma_e^2 \left[1 - \sum_{r \in N} \theta_r z^{-r} \right].$$

Thus

$$S_{ye}(z) = \sigma_e^2 \frac{1 - \sum_{r \in N} \theta_r z^{-r}}{1 - \sum_{r \in N} \theta_r z^{-r}} = \sigma_e^2$$

and

$$R_{ye}(q) = \begin{cases} \sigma_e^2 & \text{for } q = (0,0) \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

which is identical to the autocorrelation of a white noise variable, $x(s)$, with variance σ_e^2.

With these relations established, we may proceed. If we put

$$y'(s) = \sum_{r \in N} \theta_r y(s - r),$$

we may rewrite Equation (6) as

$$y(s) = y'(s) + e(s).$$

Thus the auto-correlation is

$$\begin{aligned} R_{yy}(q) &= E\{y(s)y(s+q)\} \\ &= E\{y(s)[y'(s+q) + e(s+q)]\} \\ &= \sum_{r \in N} \theta_r E\{y(s)y(s+q-r)\} + E\{y(s)e(s+q)\} \\ &= \sum_{r \in N} \theta_r R_{yy}(q-r) + R_{ye}(q). \end{aligned}$$

We define

$$\theta'_r = \begin{cases} 1 & \text{for } r = (0,0) \\ -\theta_r & \text{otherwise} \end{cases}$$

and get

$$\sum_{r \in N \cup (0,0)} \theta'_r R_{yy}(q - r) = R_{ye}(q). \tag{9}$$

Remember that the two-dimensional z-transform of the auto-correlation function, evaluated at the unit sphere (unit circle in one dimension) gives the power spectral density. If we take the z-tranform of Equation (9) we get

$$\Theta'(z) S_{yy}(z) = S_{ye}(z)$$

or

$$S_{yy}(z) = \frac{S_{ye}(z)}{\Theta'(z)},$$

where z denotes the frequency variable, $z = (z_1, z_2)$ in two dimensions. Remember also that an estimate of $S_{yy}(z)$ is $\hat{S}_{yy}(z) = |Y(z)|^2$, where $Y(z)$ is the two-dimensional Fourier-transform of

$y(s)$. Letting $X(z)$ be the z-transform of a white noise variable $x(s)$ with variance σ_e^2, this suggests that

$$Y(z) = \frac{X(z)}{\sqrt{\Theta'(z)}},$$

which is verified by

$$\hat{S}_{yy}(z) = |Y(z)|^2 = \left| \frac{X(z)}{\sqrt{\Theta'(z)}} \right|^2 = \frac{|X(z)|^2}{\Theta'(z)} = \frac{\hat{S}_{ye}(z)}{\Theta'(z)}.$$

Consequently we get the synthetic image as the inverse Fourier-transform of $Y(z)$.

For simulations on a computer it is necessary to use the discrete Fourier-transform (DFT). To be able to use this transform, we have to approximate $y(s)$ as a periodic signal. In conclusion, our algorithm for synthesis becomes:

1. Generate a Gaussian white noise image $\mathbf{x}$.
2. Create $\mathbf{X} = \text{DFT}(\mathbf{x})$.
3. $\mathbf{\Theta}'$=DFT($\boldsymbol{\theta}'$)
4. $\mathbf{Y} = \frac{\mathbf{X}}{\sqrt{\Theta'}}$.
5. Generate the synthetic texture by $\mathbf{y} = \text{IDFT}(\mathbf{Y})$.

where $\boldsymbol{\theta}$ is the transformation matrix, given by

$$\boldsymbol{\theta}' = \begin{bmatrix}
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
 & & & & & & \vdots & & & & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & \cdots & 0 & -\theta_{(-N_x,-N_y)} & \cdots & -\theta_{(-N_x,-1)} & -\theta_{(-N_x,0)} & -\theta_{(-N_x,1)} & \cdots & -\theta_{(-N_x,N_y)} & 0 & \cdots & 0 \\
 & & & & & & \vdots & & & & 0 & \cdots & 0 \\
0 & \cdots & 0 & -\theta_{(-1,-N_y)} & \cdots & -\theta_{(-1,-1)} & -\theta_{(-1,0)} & -\theta_{(-1,1)} & \cdots & -\theta_{(-1,N_y)} & 0 & \cdots & 0 \\
0 & \cdots & 0 & -\theta_{(0,-N_y)} & \cdots & -\theta_{(0,-1)} & 1 & -\theta_{(0,1)} & \cdots & -\theta_{(0,N_y)} & 0 & \cdots & 0 \\
0 & \cdots & 0 & -\theta_{(1,-N_y)} & \cdots & -\theta_{(1,-1)} & -\theta_{(1,0)} & -\theta_{(1,1)} & \cdots & -\theta_{(1,N_y)} & 0 & \cdots & 0 \\
 & & & & & & \vdots & & & & 0 & \cdots & 0 \\
0 & \cdots & 0 & -\theta_{(N_x,-N_y)} & \cdots & -\theta_{(N_x,-1)} & -\theta_{(N_x,0)} & -\theta_{(N_x,1)} & \cdots & -\theta_{(N_x,N_y)} & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
 & & & & & & \vdots & & & & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0
\end{bmatrix},$$

where the neighbor set N is within a symmetric $(2N_x + 1) \times (2N_y + 1)$ range. The number of zeros padded in $\boldsymbol{\theta}$ must be large enough to make it the same size as the wanted synthetic texture.

4 Experimental results

The causal and the non-causal neighbor-set we have applied in our experiments are given in Figure 1.

Five different real world textures, grass lawn (D9), calf leather (D24), wood (D68), pigskin (D92) and plast bubbles (D111) were modeled. All real world images were digitized from Brodatz [15], and the wood texture were inverted and reduced in size by two, to match images given in other articles.

The textures and their synthetic counterparts are shown in Figures 2 and 3. These results compare well to results published in other articles, but it takes a good imagination to accept that

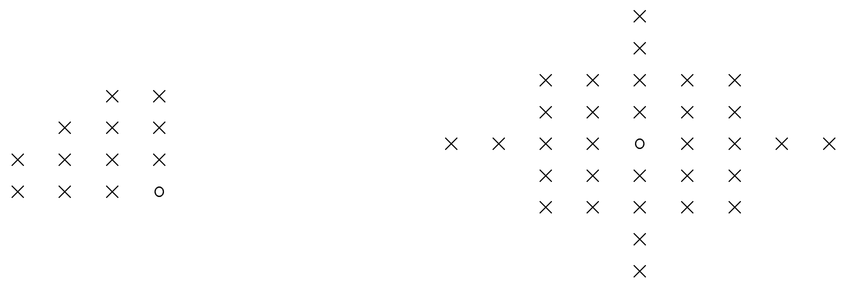
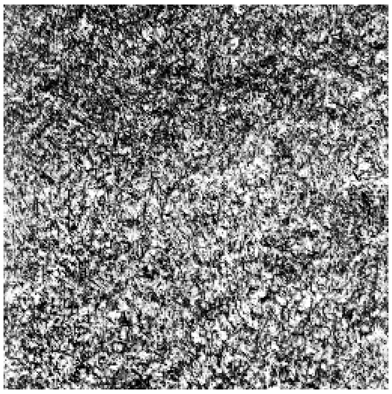


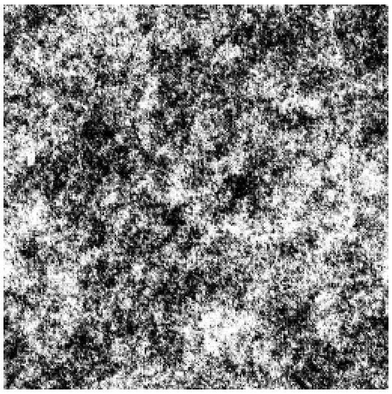
Figure 1: The causal and the non-causal neighbor sets

Coord.	D9	D24	D68	D68s	D92	D111
(1, 0)	0.3366	0.4395	0.5197	0.5229	0.2686	0.2974
(0, 1)	0.2019	0.1137	0.1280	0.1276	0.3987	0.4243
(1, 1)	0.0716	0.0995	0.0023	0.0068	0.0482	0.0065
(-1, 1)	-0.0447	-0.0718	-0.0753	-0.0813	-0.0906	-0.0853
(0, 2)	-0.0190	-0.0196	-0.0088	-0.0141	-0.0819	-0.1004
(2, 0)	-0.0484	-0.0753	-0.0309	-0.0362	-0.0357	-0.0506
(2, 2)	0.0113	0.0302	0.0102	0.0072	0.0150	0.0215
(2,-2)	0.0058	0.0059	-0.0026	-0.0043	-0.0042	0.0023
(3, 0)	0.0142	0.0302	0.0177	0.0179	0.0052	0.0173
(0, 3)	0.0099	0.0104	-0.0008	-0.0003	0.0229	0.0277
(4, 0)	0.0036	0.0008	-0.0064	-0.0045	-0.0020	-0.0021
(0, 4)	0.0040	-0.0007	0.0010	0.0006	0.0005	-0.0071
(-2, 1)	0.0116	0.0279	-0.0066	-0.0017	0.0159	0.0062
(1,-2)	0.0133	0.0147	0.0135	0.0166	0.0215	0.0177
(1, 2)	-0.0294	-0.0657	-0.0120	-0.0053	-0.0521	-0.0356
(2, 1)	-0.0430	-0.0405	-0.0491	-0.0519	-0.0302	-0.0400
(-2,-1)	-0.0430	-0.0405	-0.0491	-0.0519	-0.0302	-0.0400
(-1,-2)	-0.0294	-0.0657	-0.0120	-0.0053	-0.0521	-0.0356
(-1, 2)	0.0133	0.0147	0.0135	0.0166	0.0215	0.0177
(2,-1)	0.0116	0.0279	-0.0066	-0.0017	0.0159	0.0062
(0,-4)	0.0040	-0.0007	0.0010	0.0006	0.0005	-0.0071
(-4, 0)	0.0036	0.0008	-0.0064	-0.0045	-0.0020	-0.0021
(0,-3)	0.0099	0.0104	-0.0008	-0.0003	0.0229	0.0277
(-3, 0)	0.0142	0.0302	0.0177	0.0179	0.0052	0.0173
(-2, 2)	0.0058	0.0059	-0.0026	-0.0043	-0.0042	0.0023
(-2,-2)	0.0113	0.0302	0.0102	0.0072	0.0150	0.0215
(-2, 0)	-0.0484	-0.0753	-0.0309	-0.0362	-0.0357	-0.0506
(0,-2)	-0.0190	-0.0196	-0.0088	-0.0141	-0.0819	-0.1004
(1,-1)	-0.0447	-0.0718	-0.0753	-0.0813	-0.0906	-0.0853
(-1,-1)	0.0716	0.0995	0.0023	0.0068	0.0482	0.0065
(0,-1)	0.2019	0.1137	0.1280	0.1276	0.3987	0.4243
(-1, 0)	0.3366	0.4395	0.5197	0.5229	0.2686	0.2974

Table 1: Parameters from the least squares estimator. The column D68s gives the parameters estimated from the synthetic texture generated from the parameters from D68 – we see that the synthetic texture has parameters very close to the real world texture.

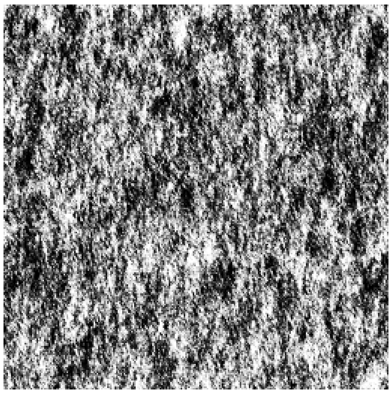


D9 – a

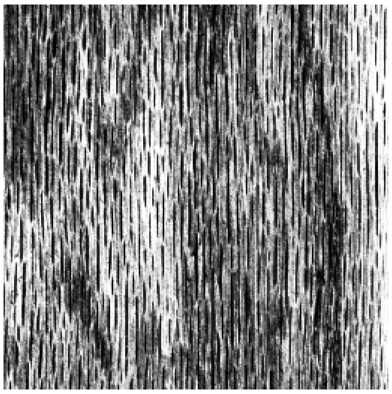


D9 – b

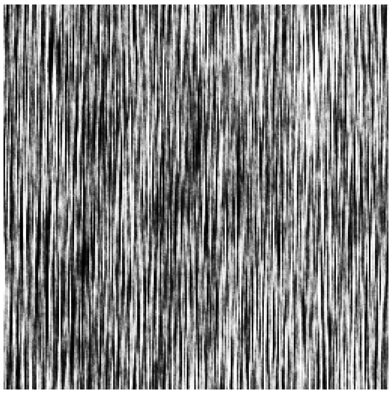
D24 – a



D24 – b



D68 – a



D68 – b

Figure 2: Real world textures (a) and their synthetic counterparts (b). The textures are D9, D24 and D68 respectively.

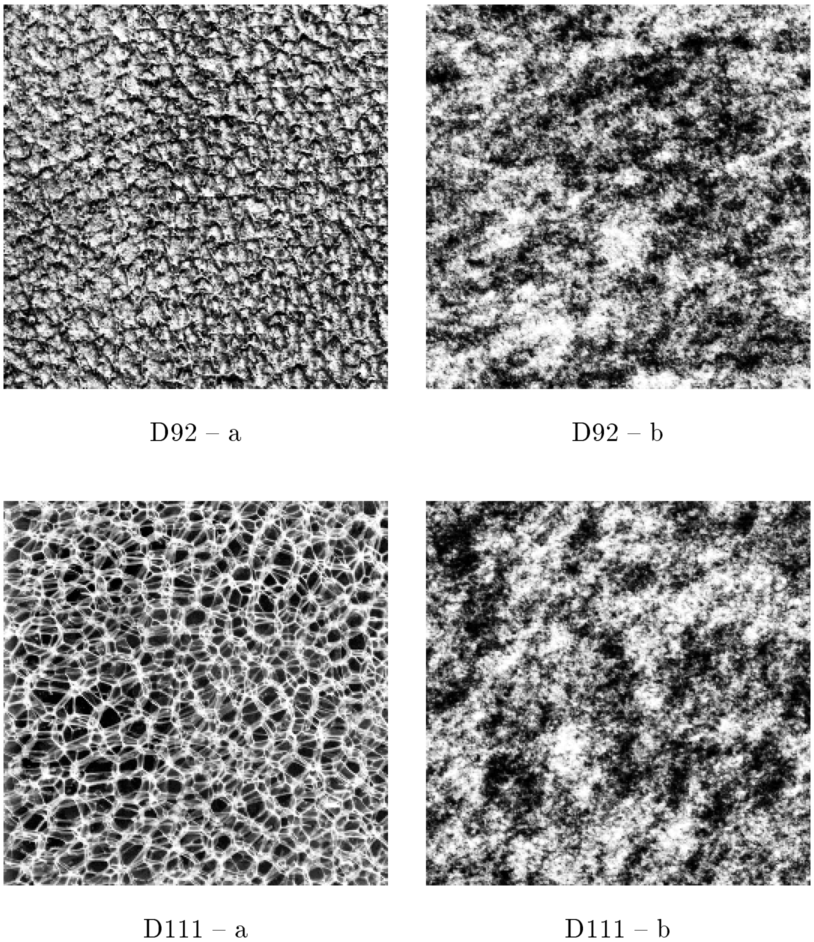
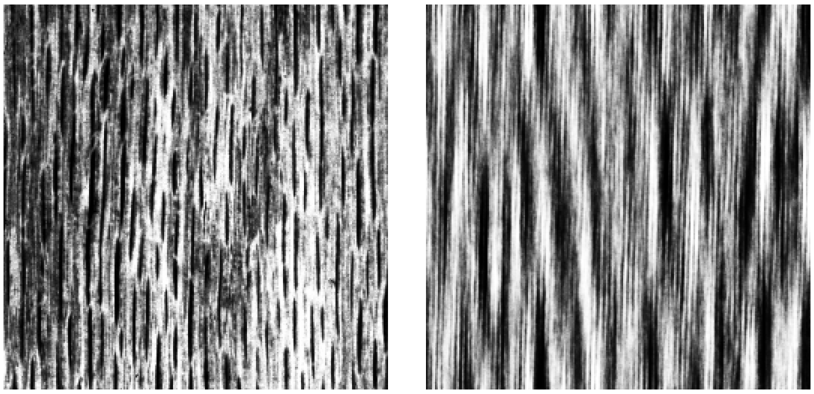


Figure 3: Real world textures (a) and their synthetic counterparts (b). The textures are D92 and D111 respectively.

		0.0378	-0.0086
	0.0358	-0.1012	-0.0608
0.0216	-0.0082	-0.1019	1.0223
-0.0054	-0.0221	0.1899	•

Table 2: Least squares estimated parameters for D68 using a causal model.



a b

Figure 4: The effect of modelling a too coarse real world texture. This experiment shows that when the model size is small compared to the coarseness of the texture, we get a poor synthesizing. Real world texture D68 at the double size (a) and its synthetic counterpart (b)

the real world and the synthetic textures are the same. The best synthetic texture was the one generated based on wood, Figure 2, D68. The estimated parameters for these models are given in Table 1. We see that the synthetic textures retain some of the characteristics of the real world textures. The synthesized textures compares well with synthesized textures presented in other papers [2, 3].

These examples show that this setup gives fairly good synthesized textures, but does the synthesized textures really have the desired properties? To address this question, the least squares parameter estimation procedure were applied to one synthetic texture, the synthetic texture based on D68. Table 1, column marked 'D68s', shows the parameters estimated from the synthetic texture. Since these parameters are close to the parameters the synthetic texture was generated from, we conclude that the setup is capable of generating textures with the desired properties.

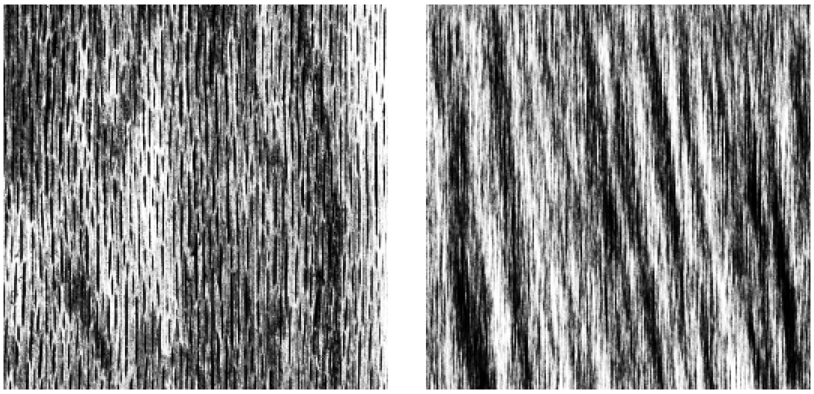
If our neighbor set is small compared to the coarsness of the image, the synthesized textures are poor. One good example of this is given in Figure 4. Here we model the same texture as in Figure 2, but scanned with the double resolution. In this example, we see that D68 which was fairly good modelled in Figure 2 is poorly modelled when we use double resolution.

What is the effect of using causal versus non-causal models? To examine this question, experiments with a causal model were carried out. The estimated parameters for D68 are given in Table 2, and the synthesized images for D68 and D92 are given in Figure 5. For the wood texture, D68, we see that the causal synthesized texture has approximately the same quality as the non-causal synthesized texture, while having a much lower number of parameters. For the pigskin, D92, however, the causal synthesized texture is poorer. The results were the same even if we increased the neighborhood, N, to four times the size used in these experiments.

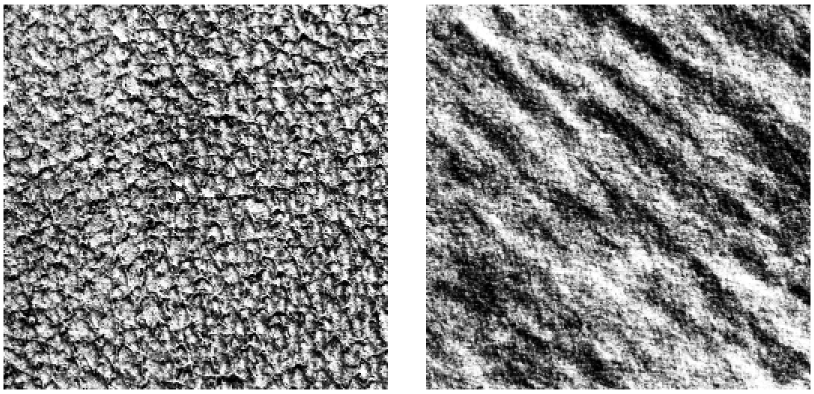
5 Summary and conclusions

This paper has presented some examples which illustrate the usefulness of texture modelling using both causal and non-causal models and least squares parameter estimation. Examples illustrate that it is possible to regenerate synthetic textures from parameters extracted from several real world textures, and obtain some similarities in the visual characteristics.

An example illustrating that the synthetic textures have approximately the same statistical characteristics as the real world textures is also given. This verifies that the synthesis process



a b



a b

Figure 5: Real world texture D68 and D92 (a) and its synthetic counterpart (b), synthesized with a causal model and least squares methods.

generates textures with similar stochastic parameters as the real world textures.

References

[1] J. H. Husøy, T. Randen, and T. O. Gulsrud, "Image texture classification with digital filter banks and transforms," in *Proc. SPIE International symposium on optical instrumentation and applied science: Applications of Digital Image Processing XVI*, (San Diego), pp. 260 – 271, July 1993.

[2] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, no. 4, pp. 959–963, 1985.

[3] R. Chellappa, S. Chatterjee, and R. Bagdazian, "Texture synthesis and compression using Gaussian-Markov random field models," *IEEE Trans. Syst., Man, Cyb.*, vol. 15, pp. 298–303, 1985.

[4] R. Chellappa and R. L. Kashyap, "Texture synthesis using 2-d noncausal autoregressive models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, no. 1, pp. 194–203, 1985.

[5] R. L. Kashyap, "Characterization and estimation of two-dimensional ARMA models," *IEEE Trans. Inform. Theory*, vol. 30, pp. 736–745, Sept. 1984.

[6] A. Rosenfeld, "Image modelling during the 1980s: a brief overview," in *Markov random fields* (R. Chellappa and A. K. Jain, eds.), pp. 1–10, Academic Press, 1993.

[7] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, pp. 786–804, May 1979.

[8] L. V. Gool, P. Dewaele, and A. Oosterlinck, "Texture analysis anno 1983," *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 336–357, 1985.

[9] R. C. Dubes and A. K. Jain, "Random field models in image analysis," *Journal of applied statistics*, vol. 16, no. 2, pp. 131–164, 1989.

[10] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society, series B*, no. 6, pp. 192–236, 1974.

[11] J. Besag, "Statistical analysis of non-lattice data," *The statistician*, no. 24, pp. 179–195, 1975.

[12] J. Besag, "Efficiency of pseudolikelihood estimation for simple Gaussian fields," *Biometrica*, no. 64, pp. 616–618, 1977.

[13] A. L. Kok, D. G. Manolakis, and V. K. Ingle, "Symmetric noncausal spatial models for 2-D signals with applications in stochastic texture modeling," *Mulitdimensional systems and signal processing*, vol. 4, pp. 125–149, 1993.

[14] G. H. Golub and C. F. Van Loan, *Matrix computations.* Baltimore, Maryland, USA: The Johns Hopkins University Press, 2 ed., 1989.

[15] P. Brodatz, *Textures: A photographic album for artists and designers.* NY: Dover, 1966.