ELSEVIER

Pattern Recognition Letters 22 (2001) 1233–1246

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Parameter-estimation in the auto-binomial model using the coding- and pseudo-likelihood method approached with simulated annealing and numerical optimization

Jan-Olof Johansson [a,b]

[a] *Mathematical Statistics, Centre for Mathematical Sciences, Lund University, Sweden*
[b] *Centre for Imaging Sciences and Technologies, Halmstad University, Lund, Sweden*

Abstract

In texture analysis, the Gibbs sampler constitutes an important tool in the generation of synthetic textures. The textures are modeled as distributions with specified parameters. In this paper, we study the estimation process of the parameters in such distributions and compare Besags coding method with a pseudo-likelihood method. We also compare simulated annealing with the Newton–Raphson method to find the global maximum of a likelihood or pseudo-likelihood function. For some textures, the two methods differ but in most case there are no important differences between them. The two maximization methods find the same maximum, but the Newton–Raphson method is much faster. However, the Newton–Raphson method cannot be applied in some cases when the location of the maximum differs too much from the starting points. Here, it is often possible to find the global maximum using simulated annealing. The methods have been used in an application with newsprint. © 2001 Published by Elsevier Science B.V.

Keywords: Auto-binomial model; Coding method; Pseudo-likelihood method

1. Introduction

Distributions of Gibbs type have important applications in texture analysis (Geman and Geman, 1989; Winkler, 1995). Several natural and technical textures can be well described by such distributions. By means of digital image analyses, they can be visualized as images, where the values of points of the texture are given colors or a grey-scale levels.

The background of this study originates in a work concerning modelling the surface structure of newsprint (Johansson, 2000). Fig. 1 is an example of a grey-scale image of such a surface. In that application, we used an auto-binomial model, which is of Gibbs type with parameters that characterize the local properties of the micro-texture.

In this paper, we focus on the estimation of the parameters. We have found that a common method, “The coding method”, does not always succeed in the estimation process. The failure may depend on the use of the Newton–Raphson method for finding the global maximum in the likelihood function. We suggest the use of simulated annealing for such structures as an alternative way and discuss and compare two estimation

E-mail address: jan_olof.johansson@ide.hh.se (J.-O. Johansson).

0167-8655/01/$ - see front matter © 2001 Published by Elsevier Science B.V.
PII: S 0 1 6 7 - 8 6 5 5 (0 1) 0 0 0 5 5 - 1

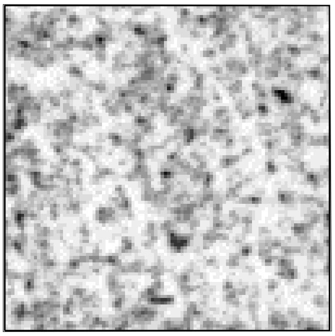

Fig. 1. An example of the surface structure of newsprint.

methods, the coding method with a pseudo-likelihood method and compare Newton–Raphson with simulated annealing for finding the global maximum.

We use simulated micro-textures or images, of auto-binomial models with specified parameters for evaluating the methods. These images consist of first- to fourth-order models and are generated using the Gibbs sampler (Cross and Jain, 1983). In the application of newsprint, a fourth-order model was the most proper one.

The paper is organized as follows. In Section 2, we introduce the auto-models and define Gibbs distribution and its relation to Markov random fields. We also briefly describe the simulation of the micro-textures. In Section 3 we give an overview of the coding method and a pseudo-likelihood method for estimation of the parameters and in the subsequent section, we describe two methods for finding the global maximum of a likelihood or pseudo-likelihood function. Finally, in Section 5 we present the results from the experiments of estimating auto-binomial models of first to and including fourth-order.

2. Images, micro-textures and random fields

In the literature, there is no generally accepted definition of micro-textures. Some authors use attribute such as coarseness, line-likeness, contrast, directionality, regularity and roughness (Haralick, 1984). Cross and Jain (1983), regard micro-textures as random fields and we adopt that point of view.

2.1. Definitions

We follow the notations and definitions given by Besag (1974) and Dubes and Jain (1989). A pixel is denoted with (i,j) or (k) where $1 \leqslant i \leqslant N_1$, $1 \leqslant j \leqslant N_2$ and $1 \leqslant k \leqslant N_1 \times N_2$. The intensity at a pixel is denoted with $X(i,j)$ or X_k.

Definition 1. A lattice, $\mathscr{L}$, is a square array of pixels, $\{(i,j) : 1 \leqslant i \leqslant N_1, 1 \leqslant j \leqslant N_2\}$.

Definition 2. A colouring of $\mathscr{L}$, $\boldsymbol{X} = \{X(i,j)\}$, with G levels, is a function from the pixels of $\mathscr{L}$ to the finite set $\Gamma = \{0, 1, 2, \ldots G-1\}$.

Definition 3. The configuration space of $\boldsymbol{X}$ is the set $\mathscr{C} = \Gamma^{N_1 \times N_2}$.

Definition 4. A pixel, (j), is defined as a neighbour to pixel (i) if it satisfies the symmetric, non-reflexive relation $(j)\mathscr{R}(i)$.

Definition 5. The neighbourhood to pixel (i) is the set $\{(j); (i)\mathscr{R}(j)\}$.

If $(j)\mathscr{R}(i)$, then $(i)\mathscr{R}(j)$ and the pixels (i) and (j) are called neighbours. No pixel (j) is a neighbour to itself. The order of a neighbour is defined as 1 if $r = \pm 1, \pm 2$, as 2 if $r = \pm 3, \pm 4$, as 3 if $r = \pm 5, \pm 6$ and as 4 if $r = \pm 7, \pm 8, \pm 9, \pm 10$. Fig. 2 shows neighbours to pixel (t) of the first to and including the fourth-order.

	$t : -7$	$t : -6$	$t : +8$	
$t : -9$	$t : -3$	$t : -2$	$t : +4$	$t : +10$
$t : -5$	$t : -1$	t	$t : +1$	$t : +5$
$t : -10$	$t : -4$	$t : +2$	$t : +3$	$t : +9$
	$t : -8$	$t : +6$	$t : +7$	

Fig. 2. The indices of the neighbours to pixel (t).

Definition 6. A clique is a set of pixels where all pairs of pixels are mutual neighbours or consisting of a single pixel.

Definition 7. A neighbourhood system is the ordered class $Q = \{Q_1, Q_2, \dots, Q_M\}$, where $M = N_1 \times N_2$ and Q_i denotes the set of cliques containing pixel (i).

2.1.1. The Gibbs distribution and the auto-binomial model

The Gibbs distribution is defined by the probability mass function $P(\boldsymbol{X} = \boldsymbol{x}) = \mathrm{e}^{-U(\boldsymbol{x})}/Z$, where $U(\boldsymbol{x})$ is the energy function, $\boldsymbol{x}$ is a colouring of the lattice and Z is a normalizing constant, the partition function. A clique function, $V_\mathrm{c}(\boldsymbol{x})$, for a given image $\boldsymbol{x}$, is associated with each clique and the energy function is a sum of clique functions,

$$U(\boldsymbol{x}) = \sum_{c \in Q} V_\mathrm{c}(\boldsymbol{x}).$$

We parameterize the Gibbs distribution by the choice of clique functions.

An auto-model is a model where the energy function for a pairwise interaction model can be written:

$$\begin{aligned} U(\boldsymbol{x}) &= \sum V_\mathrm{c}(\boldsymbol{x}) \\ &= \sum_{t=1}^{M} F(x_t) + \sum_{t=1}^{M} \sum_{r=-K}^{K} H(x_t, x_{t:+r}), \end{aligned} \tag{1}$$

where K is the size of the neighbourhood, F and H are potential functions. We focus our interest on auto-binomial models (Besag, 1974; Winkler, 1995). In this case, the functions $F(\cdot)$ and $H(\cdot,\cdot)$ can be written as

$$F(x_t) = \alpha x_t - \ln \binom{G-1}{x_t} \tag{2}$$

and

$$H(x_t, x_{t:+r}) = \beta_{k,l} x_t x_{t:+r}, \tag{3}$$

which is a global description. The indexing of β is explained in (6)–(9). The local description is the conditional distribution, $X \,|\, X_{\delta t}$, given the neighbourhood, δt, to pixel (t),

$$X \,|\, X_{\delta t} \in \mathrm{Bin}(G-1, \Theta(t)), \tag{4}$$

where

$$\Theta(t) = \frac{\exp(T)}{1 + \exp(T)} \tag{5}$$

and

$$T = \alpha + \beta_{11}(x_{t:+1} + x_{t:-1}) + \beta_{12}(x_{t:+2} + x_{t:-2}) \tag{6}$$

for a first-order model.

For a second-, third- and fourth-order model, we successively add to T the terms

$$\beta_{21}(x_{t:+3} + x_{t:-3}) + \beta_{22}(x_{t:+4} + x_{t:-4}) \tag{7}$$

$$\beta_{31}(x_{t:+5} + x_{t:-5}) + \beta_{32}(x_{t:+6} + x_{t:-6}) \tag{8}$$

$$\begin{aligned} &\beta_{41}((x_{t:+7} + x_{t:-7}) + (x_{t:+9} + x_{t:-9})) \\ &\quad + \beta_{42}((x_{t:+8} + x_{t:-8}) + (x_{t:+10} + x_{t:-10})). \end{aligned} \tag{9}$$

The process is a Markov random field. This relation between the Gibbs distribution and the Markov random field is stated in the Hammersley–Clifford theorem (Besag, 1974).

2.2. Generation of micro-textures, the Gibbs sampler

To investigate the estimators we need stochastic images with known parameters. Such images of size $M = 100 \times 100$ and with eight grey levels, are generated using Gibbs sampler (Geman and Geman, 1989). This sampler is a special case of the single-component Metropolis–Hasting's algorithm and is widely used to generate stochastic images (Winkler, 1995).

The parameters of a fourth-order model are

$$\alpha, \beta_{11}, \beta_{12}, \beta_{11}, \beta_{21}, \beta_{22}, \beta_{31}, \beta_{32}, \beta_{41}, \beta_{42}.$$

The largest value of the first figure of the indices is the order of the model. The β-parameters have been set according to the desired local properties and in a fourth-order model the parameter α equals

$$\begin{aligned} &-7(\beta_{11} + \beta_{12} + \beta_{11} + \beta_{21} + \beta_{22} + \beta_{31} + \beta_{32} \\ &\quad + 2(\beta_{41} + \beta_{42})). \end{aligned}$$

For first to third-order models α is set analogous.

The special choice of the parameter α implies the binomial parameter θ, in average, to be close to 0.5. This, in turn, gives images with the mean pixel value close to 3.5, i.e., in the middle of the grey scale interval. The structure of such images are more perceivable than those where the mean pixel value is close to 0 or 7.

With infinitely many iterations these images can be regarded as realizations of Gibbs sampler. We have iterated only 200 times but this seems to be sufficient for the images to stabilize.

3. Estimation

In this section, we discuss the estimation of the parameters of Gibbs distributions. As in earlier sections, we concentrate on auto-binomial models up to and including the fourth-order.

At least four different estimators are known from the literature, coding method (Besag, 1974), "least squares error method" (Derin and Elliot, 1984), "logit model fit method" (Chen and Dubes, 1989) and "minimum logit χ^2 method" (Dubes and Jain, 1989). The last method is based on Berkson procedure (Berkson, 1955), but is used only for binary variables and cannot be applied here. The least squares error method and logit model fit method both solve a system of linear equations where the number of equations is equal to the number of grey scales raised to the number of neighbours, in our case 8^{20}, a too large number for computations. Beside these methods also a pseudo-likelihood method can be used. In this study, we investigate the coding method and a pseudo-likelihood method and compare the Newton–Raphson method and simulated annealing for finding global maximum.

3.1. The coding method

This method, proposed by Besag, is probably also the most common one, (Derin and Elliot, 1984). It is based on a coding scheme of the image where the pixels are given different codes. The coding depends on the the order of the system. For a third- and fourth-order model the vertical and horizontal directions between pixels of the same code must be larger than two, which results conditionally independent values of the pixels belonging to the same coding set. This is a consequence of the Markov property. Fig. 3 is an example of the coding of a third- and fourth-order model, which contain nine coding sets.

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

Fig. 3. Coding pattern for third- and fourth-order models.

The estimator $\hat{\boldsymbol{\beta}}$ is defined as

$$\hat{\boldsymbol{\beta}} = \frac{1}{9}\sum_{i=1}^{9}\hat{\boldsymbol{\beta}}_i, \tag{10}$$

where $\hat{\boldsymbol{\beta}}_i$ is the parameter vector that maximizes $L(\boldsymbol{\beta}_i)$ in coding i.

$$\begin{aligned} L(\boldsymbol{\beta}_i) &= \sum_{(t)\in \text{coding } i} \ln P(X_t = x_t | X_{\delta t} = x_{\delta t}) \\ &= \sum_{(t)\in \text{coding } i} \left(\ln \binom{G-1}{x_t} \right. \\ &\qquad \left. + Tx_t - (G-1)\ln(1+\mathrm{e}^T) \right). \end{aligned} \tag{11}$$

Besag uses the Newton–Raphson method to find the maximum. We compare this method with simulated annealing.

3.2. The pseudo-likelihood method

This method is computationally very similar to the coding method. Instead of partitioning the image into different codings all pixels are considered and the log pseudo-likelihood function is

$$\begin{aligned} \ln PL = &\sum_{(i)\in \mathscr{L}} \left(\ln \binom{G-1}{x_i} \right. \\ &\qquad \left. + Tx_i - (G-1)\ln(1+\mathrm{e}^T) \right), \end{aligned} \tag{12}$$

where the summation is taken over the hole image, $\mathscr{L}$. The parameter values that maximizes the log-pseudolikelihood function are the maximum pseudo-likelihood estimates. This method was introduced by Besag (1975).

4. Maximization

4.1. Newton–Raphson method

The maximum values of the cost function have partial derivatives equal to zero. Therefore, we seek the zero of the derivative of $f(\boldsymbol{\beta}, \boldsymbol{x})$ with respect to $\boldsymbol{\beta}$ for a given image $\boldsymbol{x}$. The Newton–Raphson method implies solving a system of non-linear equations of the partial derivatives,

$$\frac{\partial f(\boldsymbol{\beta}, \boldsymbol{x})}{\partial \beta_{i,j}} = 0 \quad 1 \leqslant i \leqslant 4, \ 1 \leqslant j \leqslant 2, \tag{13}$$

for a fourth-order system.

In a neighbourhood of a solution, $\boldsymbol{\beta}_s$, each partial derivative can be expanded in a Taylor series, giving

$$\frac{\mathrm{d}f(\boldsymbol{\beta}_s + \delta\boldsymbol{\beta}, \boldsymbol{x})}{\mathrm{d}\boldsymbol{\beta}} = \frac{\mathrm{d}f(\boldsymbol{\beta}_s, \boldsymbol{x})}{\mathrm{d}\boldsymbol{\beta}} + \boldsymbol{J} \cdot \delta\boldsymbol{\beta} + \mathrm{O}(\delta\boldsymbol{\beta}^2), \tag{14}$$

where $\boldsymbol{J}$ is the Jacobian matrix

$$J_{(ij),(i'j')} = \frac{\partial^2 f(\boldsymbol{\beta}_s, \boldsymbol{x})}{\partial \beta_{ij} \beta_{i'j'}} \tag{15}$$

for $1 \leqslant i, i' \leqslant 4, \ 1 \leqslant j, j' \leqslant 2$.

If terms of order $\delta\boldsymbol{\beta}^2$ and higher are neglected and if

$$\frac{\mathrm{d}f(\boldsymbol{\beta}_s + \delta\boldsymbol{\beta}, \boldsymbol{x})}{\mathrm{d}\boldsymbol{\beta}}$$

is assumed to be zero, we obtain a system of linear equations for the correction vector $\delta\boldsymbol{\beta}$,

$$\boldsymbol{J} \cdot \delta\boldsymbol{\beta} = -\frac{\mathrm{d}f(\boldsymbol{\beta}, \boldsymbol{x})}{\mathrm{d}\boldsymbol{\beta}} \tag{16}$$

and a iterative scheme

$$\boldsymbol{\beta}_{\mathrm{new}} = \boldsymbol{\beta}_{\mathrm{old}} + \delta\boldsymbol{\beta} \tag{17}$$

The convergence of the method is certain if the Jacobian, $\boldsymbol{J}$, is non-singular, the second-order derivatives are continuous in a neighbourhood of the solutions and if the starting values are sufficiently close to the solutions (Ostrowski, 1966). The first two conditions are satisfied but the last one is difficult to verify. There are, in fact, problems involved with the convergence of the method.

In our experiments, we have set the start vector equal to 0 and used LU decomposition to solve the system of linear equations (Strang, 1988). The iterative process is stopped when the sum of the absolute values of the components of the correction vector are less than 0.000001.

4.2. Simulated annealing

Simulated annealing is a method based on Monte Carlo techniques which can be used to find global extremes (Aarst and Korst, 1989). We specify our maximization problem by a set of problem instances that are formalized as a pair (S, f), where S is the space of possible solutions and $f = f(\boldsymbol{\beta}, x)$ is the function,

$$f : S \to \boldsymbol{R}$$

that we want to maximize. A solution $i_{\mathrm{opt}} \in S$ satisfies

$$f(i_{\mathrm{opt}}) \geqslant f(i), \quad \forall i \in S. \tag{18}$$

We apply the Metropolis algorithm to generate a sequence of solutions to the maximization problem and use the following algorithm.

Let i and j be two solutions to the problem generated by the Metropolis algorithm and $f(i)$ and $f(j)$ the associated functions. Accept j by the probability

$$P_{\mathrm{c}}(\text{accept } j) = \begin{cases} 1 & \text{if } f(j) \geqslant f(i), \\ \mathrm{e}^{(f(j) - f(i)/c_k)} & \text{if } f(j) < f(i), \end{cases} \tag{19}$$

where $\{c_k\}$; $k = 0, 1, \ldots, n$ is a sequence of decreasing real values.

In our maximization problem S, is the space spanned by the parameter vector $\boldsymbol{\beta}$ and the function f is defined as

$$\begin{aligned} f(\boldsymbol{\beta},x) &= \ln\left(\prod_{(i)\in\mathscr{L}}\binom{G-1}{x_i}\frac{\mathrm{e}^{Tx_i}}{(1+\mathrm{e}^{T})^{G-1}}\right)\\ &= \sum_{(i)\in\mathscr{L}}\left(\ln\binom{G-1}{x_i}\right.\\ &\qquad \left. + Tx_i - (G-1)\ln(1+\mathrm{e}^{T})\right), \end{aligned} \tag{20}$$

where T is defined in Eqs. (6)–(9), G is the number of grey levels, $\mathscr{L}$ is the whole lattice and x_i is the observed grey level of pixel (i).

The expression

$$\sum_{(i)}\ln\binom{G-1}{x_i}$$

is constant for a given image and can therefore be eliminated. We finally have to maximize

$$\sum_{(i)\in\mathscr{L}} Tx_i - (G-1)\ln(1+\mathrm{e}^{T}). \tag{21}$$

We use an algorithm given in Aarst and Korst (1989), with a starting value of c large enough to allow all transitions to be accepted. The decrement of the control parameter, c_k, is slow and the final value c_n so small that the function f remains unchanged for a number of consecutive chains.

In our applications we have set the initial value of c to 50,000 and used the formula

$$c_j = \frac{c_{j-1}}{1+0.1\ln(c_{j-1}+3)}. \tag{22}$$

A large number of experiments have shown that this gives a cooling scheme that works well. For every c_k, we run the Markov chain 20 times and change the parameters with steps equal to 0.001.

The number δ is a small number. We have chose 0.001 because differences in parameter values less than this number does not seem to visually influence the images. The number of iterations for given c_k is found by experiments. Our aim have been to minimize computer time without loss of accuracy. With 20 iteration for every value of c we use about 30 min. of computer time for a SPARC-station 5 to estimate a fourth-order model. We have tried to increase the convergence speed by taking larger jumps but then it is necessary to increase the number of iterations at every jump to retain the accuracy and it results only in small improvements.

5. Properties of the estimators

In this section, we discuss the properties of the coding Method and the pseudo-likelihood method. We also compare the two maximization methods, simulated annealing and Newton–Raphson. All images, except one, have been generated using the Gibbs sampler and the algorithm described in section 4. The exception is the first fourth-order model in the experiments, which has been generated using the exchange algorithm, see Section 5.2.2 and Cross and Jain (1983).

For each parameter vector, we generate 25 images which we estimate with the coding method and the pseudo-likelihood method using both Newton–Raphson and simulated annealing for the optimization. We calculate the means, biases and standard deviations of the 25 estimated parameter values and present the results in tables, grey scale maps of images and in graphs of the biases and standard deviations. For each estimation we have used start values equal to zero.

5.1. First- and second-order models

First-order models do only take account of the nearest neighbours in the horizontal and vertical directions. This fact restricts the number of possible textures to those that are coarse in these two directions. The second-order models also depend on the nearest neighbours in the two diagonal directions which increase the number of possible textures. We have performed several experiments with first- and second-order models but present here, only the results for one first-order and one second-order model. These examples therefore represent the set of first- and second-order models with small parameter values.

5.1.1. First-order model

The chosen first-order model can be described as small-coarse with no line structures. The parameter β_{11} and β_{12} are equal which gives a symmetrical image. The results from the estimations

Table 1
Results from experiments with a first-order model

Parameter	β_{11}	β_{12}	β_{21}	β_{22}	β_{31}	β_{32}	β_{41}	β_{42}
Specified values	0.100	0.100	0.000	0.000	0.000	0.000	0.000	0.000
Coding method using simulated annealing								
Mean	0.099467	0.099200	−0.000644	0.000200	0.000022	−0.002022	0.000622	0.000156
Bias	−0.000533	−0.000800	−0.000644	0.000200	0.000022	−0.002022	0.000622	0.000156
S.D.	0.000050	0.000043	0.000041	0.000041	0.000042	0.000031	0.000023	0.000006
Pseudo-likelihood using simulated annealing								
Mean	0.100476	0.098858	0.000133	0.000791	−0.001102	−0.001027	0.000289	−0.000480
Bias	0.000476	−0.001142	0.000133	0.000791	−0.001102	−0.001027	0.000289	−0.000480
S.D.	0.000053	0.000046	0.000045	0.000023	0.000041	0.000020	0.000013	0.000011
Coding method using Newton–Raphson								
Mean	0.099921	0.099020	−0.000646	0.000238	−0.000528	−0.000921	0.000592	0.000489
Bias	−0.000079	−0.000980	−0.000646	0.000238	−0.000528	−0.000921	0.000592	0.000489
S.D.	0.000044	0.000045	0.000048	0.000039	0.000036	0.000035	0.000026	0.000009
Pseudo-likelihood using Newton–Raphson								
Mean	0.099726	0.098815	−0.000715	0.000232	−0.000566	−0.001063	0.000605	0.000440
Bias	−0.000274	−0.001185	−0.000715	0.000232	−0.000566	−0.001063	0.000605	0.000440
S.D.	0.000043	0.000045	0.000047	0.000038	0.000035	0.000032	0.000026	0.000010

Table 2
Results from experiments with a second-order model

Parameter	β_{11}	β_{12}	β_{21}	β_{22}	β_{31}	β_{32}	β_{41}	β_{42}
Specified values	0.100	−0.100	0.150	0.000	0.000	0.000	0.000	0.000
Coding method using simulated annealing								
Mean	0.100889	−0.101733	0.149222	0.001800	−0.000511	−0.002644	−0.000200	0.000556
Bias	0.000889	−0.001733	−0.000778	0.001800	−0.000511	−0.002644	−0.000200	0.000556
S.D.	0.000048	0.000036	0.000018	0.000057	0.000025	0.000027	0.000016	0.000009
Pseudo-likelihood using simulated annealing								
Mean	0.100600	−0.101000	0.148800	0.001200	−0.000600	−0.002200	−0.000200	0.000200
Bias	0.000600	−0.001000	−0.001200	0.001200	−0.000600	−0.002200	−0.000200	0.000200
S.D.	0.000055	0.000042	0.000021	0.000055	0.000025	0.000024	0.000019	0.000007
Coding method using Newton–Raphson								
Mean	0.101369	−0.101766	0.149839	0.001836	−0.000648	−0.002161	−0.000240	0.000526
Bias	0.001369	−0.001766	−0.000161	0.001836	−0.000648	−0.002161	−0.000240	0.000526
S.D.	0.000050	0.000042	0.000018	0.000056	0.000026	0.000026	0.000017	0.000008
Pseudo-likelihood using Newton–Raphson								
Mean	0.101026	−0.101399	0.149176	0.001802	−0.000594	−0.002219	−0.000206	0.000373
Bias	0.001026	−0.001399	−0.000824	0.001802	−0.000594	−0.002219	−0.000206	0.000373
S.D.	0.000049	0.000039	0.000018	0.000056	0.000025	0.000026	0.000015	0.000008

are shown in Tables 1 and 2 and the grey-scale images in Fig. 4. Graphs showing the behaviour of the biases and standard deviations are shown in Fig. 9.

We only show realizations of images with the specified parameter values and images generated using estimated parameter values from the pseudo-likelihood method procedure.

As can be seen from the images, the structures are very similar and so are the numerical results. There are no important differences among the four methods concerning the numerical results or the

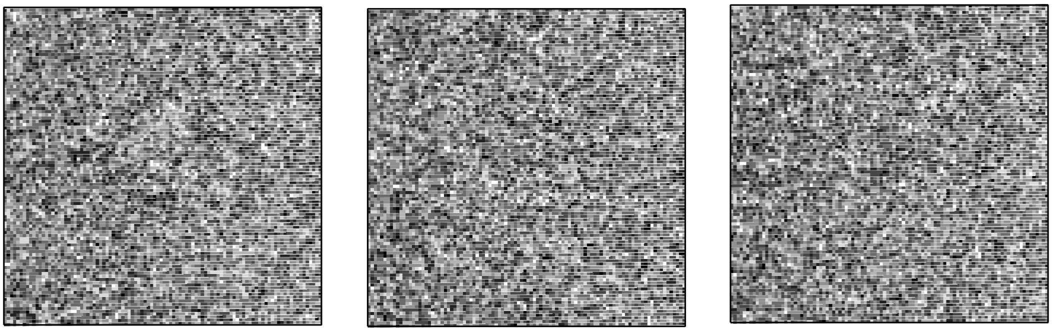

Fig. 4. A first-order model. The left image is generated using specified parameter values, the image in the middle using parameters obtained using pseudo-likelihood and Newton–Raphson maximization procedure and the right image with parameters obtained using pseudo-likelihood and simulated annealing.

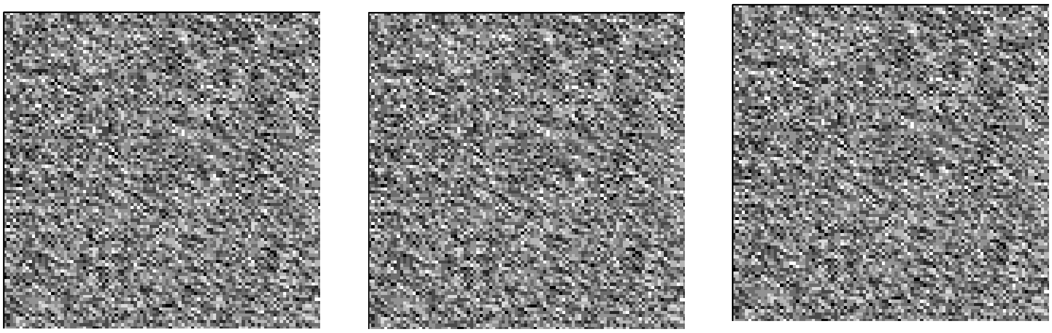

Fig. 5. A second-order model. The left image is generated using specified parameter values, the image in the middle with parameters obtained using the coding method and Newton–Raphson maximization procedure and the right image with parameters obtained using coding method and simulated annealing.

images. The biases and the standard deviations are small for all methods. The computing time, however, differs much and Newton–Raphson is roughly 50–100 times faster than simulated annealing.

5.1.2. Second-order model

The chosen example of the second-order model shows some coarse structures in the North–West direction which is a consequence of the positive β_{11} and β_{21} parameters, cf. Fig. 5.

The results and conclusions from the estimation of the second-order models are very similar those of the first-order model. We can note that the biases and the standard deviations are of the same magnitudes and that the the biases have the same sign for all four methods. The grey-scale plots of the images are very similar.

5.2. Third- and fourth-order models

Third and fourth-order models take the nearest and the next-nearest neighbours into account. This gives further possibilities for modelling a texture. We present the results from the estimation of one third-order model and two fourth-order models. One fourth-order model has been generated using a different variant of the Gibbs sampler which make it possible to decide the final marginal distribution of the intensities. The method is described in section 5.2.2 We have also chosen one fourth-order model with specified parameter values that differs much from zero, i.e., the starting point for Newton–Raphson maximization. This is an example of an image that gives singular Jacobians and Newton–Raphson fails to find the maximum.

In Tables 3–5, we list the specified parameter values and the results from the experiments. The images are shown in Figs. 6 and 7.

5.2.1. Third-order model

The third-order model produces a coarse image with faint directional structures in the two diagonal directions.

Both the biases and the standard deviations are large for this third-order model compared with the first- and second-order models. However, the four different estimation methods have similar biases and standard deviations where the parameters β_{31} and β_{32} have the largest bias. This model is an example of an image with parameters

Table 3
Results from experiments with a third-order model

Parameter	β_{11}	β_{12}	β_{21}	β_{22}	β_{31}	β_{32}	β_{41}	β_{42}
Values	0.050	0.050	0.200	0.200	−0.100	−0.100	0.000	0.000
Coding method using simulated annealing								
Mean	0.049489	0.045000	0.190178	0.191356	−0.091578	−0.090444	0.003822	0.003311
Bias	−0.000511	−0.005000	−0.00922	−0.008644	0.008422.	0.009556	0.003822	0.003311
S.D.	0.000072	0.000067	0.000042	0.000026	0.000015	0.000034	0.000007	0.000008
Pseudo-likelihood using simulated annealing								
Mean	0.050480	0.046120	0.189800	0.190640	−0.091000	−0.090240	−0.006280	−0.006640
Bias	0.000480	−0.003880	−0.010200	0.009360	−0.009000	0.009760	−0.006280	−0.006640
S.D.	0.000078	0.000073	0.000043	0.000025	0.000017	0.000026	0.000005	0.000005
Coding method using Newton–Raphson								
Mean	0.054639	0.048977	0.200390	0.199085	−0.082871	−0.081524	0.017989	0.018559
Bias	0.004639	−0.001023	0.000390	−0.000915	0.017129	0.018476	0.017989	0.018559
S.D.	0.000072	0.000053	0.000076	0.000037	0.000038	0.000027	0.000071	0.000097
Pseudo-likelihood using Newton–Raphson								
Mean	0.054639	0.048977	0.200390	0.199085	−0.082871	−0.081524	0.017989	0.018559
Bias	0.004639	−0.001023	0.000390	−0.000915	0.017129	0.018476	0.017989	0.018559
S.D.	0.000082	0.000067	0.000068	0.000036	0.000034	0.000026	0.000106	0.000106

Table 4
Results from experiments with a fourth-order model

Parameter	β_{11}	β_{12}	β_{21}	β_{22}	β_{31}	β_{32}	β_{41}	β_{42}
Values	0.070	0.070	0.070	0.070	0.050	0.000	0.030	0.000
Coding method using simulated annealing								
Mean	0.066445	0.068044	0.067111	0.069289	0.044156	−0.009844	0.020956	−0.010467
Bias	−0.003555	−0.001956	−0.002889	−0.000711	−0.005844	−0.009844	−0.009044	−0.010467
S.D.	0.000034	0.000029	0.000050	0.000043	0.000037	0.000029	0.000015	0.000020
Pseudo-likelihood using simulated annealing								
Mean	0.065600	0.068000	0.066600	0.069000	0.043800	−0.010400	0.021400	−0.010200
Bias	−0.004400	−0.002000	−0.003400	−0.001000	−0.006200	−0.010400	−0.008600	−0.010200
S.D.	0.000041	0.000042	0.000057	0.000062	0.000045	0.000030	0.000019	0.000019
Coding method using Newton–Raphson								
Mean	0.066971	0.068015	0.069997	0.073174	0.051910	−0.000408	0.031383	0.001722
Bias	−0.003029	−0.001985	−0.000003	0.003174	0.001910	−0.000408	0.001383	0.001722
S.D.	0.000030	0.000029	0.000040	0.000049	0.000038	0.000046	0.000101	0.000126
Pseudo-likelihood using Newton–Raphson								
Mean	0.066999	0.068031	0.069921	0.072555	0.051729	−0.000311	0.031139	0.001999
Bias	−0.003001	−0.001969	−0.000079	0.002555	0.001729	−0.00031	0.001139	0.00199
S.D.	0.000032	0.000028	0.000040	0.000049	0.000040	0.000042	0.00009	0.000114

Table 5
Results from experiments with a fourth-order model

Parameter	β_{11}	β_{12}	β_{21}	β_{22}	β_{31}	β_{32}	β_{41}	β_{42}
Values	0.450	0.450	0.450	0.450	−0.450	−0.450	−0.450	−0.450
Coding method using simulated annealing								
Mean	0.456066	0.458133	0.456511	0.452622	−0.455200	−0.455889	−0.455466	−0.455155
Bias	0.006066	0.008133	0.006511	0.002622	−0.005200	−0.005889	−0.005466	−0.005155
S.D.	0.000637	0.000767	0.000405	0.000221	0.000498	0.000305	0.000142	0.000068
Pseudo-likelihood using simulated annealing								
Mean	0.448600	0.447200	0.449800	0.447400	−0.448400	−0.448000	−0.448400	−0.448600
Bias	−0.001400	−0.002800	−0.000200	−0.002600	0.001600	0.002000	0.001600	0.001400
S.D.	0.000465	0.000430	0.000369	0.000186	0.000299	0.000182	0.000111	0.000075

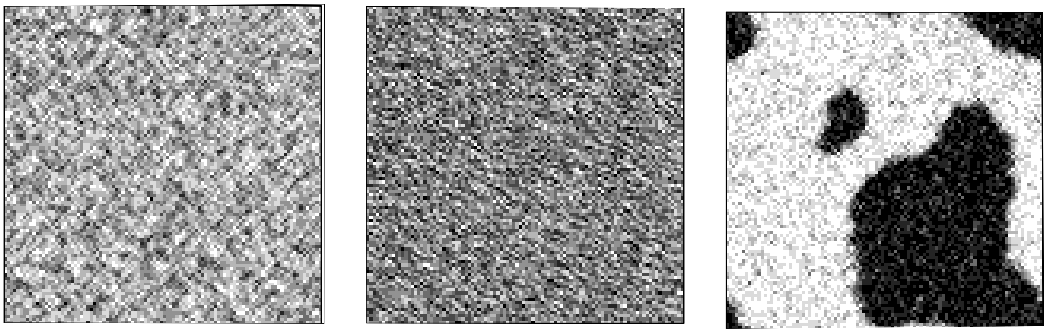

Fig. 6. A third-order model. The left image using specified parameters, the image in the middle with parameters obtained with the coding method and Newton–Raphson maximization procedure and the right image with parameters obtained using the coding method and simulated annealing.

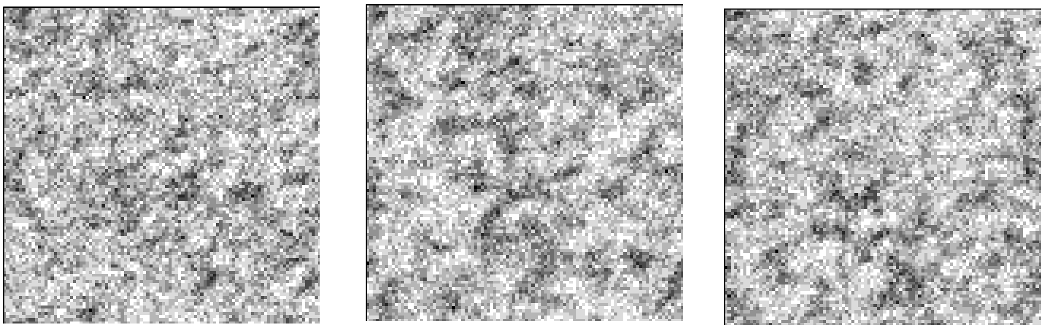

Fig. 7. A fourth-order model generated using a binomial marginal density of the intensities. The left image using specified parameters, the image in the middle with parameters obtained using the pseudo-likelihood method with Newton–Raphson maximization and the right image with parameters obtained using the coding method and simulated annealing.

that are very difficult to estimate and where the values of the parameters are critical for the texture. The estimates obtained using the coding method with Newton–Raphson maximization exhibit a phase transition phenomenon. This is a state where the short-term correlation develops into long-term correlation, see Pickard (1987) and the corresponding image loses its fine texture structure, cf. Fig. 6. The image corresponding to the estimates obtained using the coding method and maximization using simulated annealing does not degenerate but it is darker than the original image. It has, however, a similar texture structure.

5.2.2. *A fourth-order model with constaint*

We have generated a fourth-order model using a different Gibbs sampler where the given marginal distribution of the intensities remains unchanged during the simulation process. Instead of assigning a pixel a value according to the values of its neighbours, two pixels are chosen and their values are exchanged according to the values of their neighbours. In our experiments, we have chosen the initial distribution of the intensities binomial, Bin (8, 0.25). As in earlier subsections, we show only three images due to the very similar numerical estimates using the coding method and the pseudo-likelihood method. The images have large clusters but no directions and the mean value of the intensities is equal to 2.0.

The bias is of the same magnitude as for the third-order model for most parameters. However, parameter β_{32}, β_{41} and β_{42} are rather large with the maximization using simulated annealing and the bias are of magnitude 0.01. Newton–Raphson maximization method gives bias of magnitude 0.001 for these parameters. The grey-scale plots are similar the original image, cf. Fig.7.

5.2.3. *A fourth-order model with large specified parameter values*

The second fourth-order model is chosen with parameter values far from zero. This is an example of an image where the Newton–Raphson method fails to find the maxima. The image corresponding to the parameter values has short line segments in the two diagonal directions and looks like a labyrinth, cf. Fig. 8.

The numerical maximum obtained using simulated annealing is close to the specified value and the standard deviation is small. The biases are systematically positive for the coding method where the pseudo-likelihood method produces negative biases and vice versa. We also note that for this example the biases are smaller for Newton–Raphson than for simulated annealing.

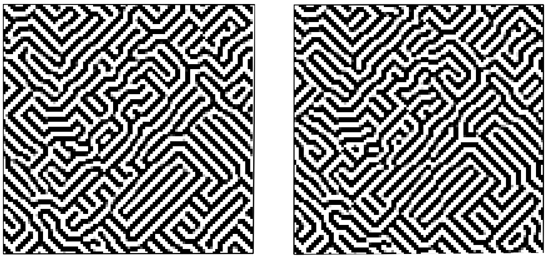

Fig. 8. A fourth-order model with parameter values far from zero. The left image is generated using specified parameter values and the right image using parameters estimated using the coding method and maximization with simulated annealing.

6. An application

The surface structure of newsprint can be regarded as an engineered stochastic structure (Deng and Dodson, 1994). The quality of the print on the surface depends highly on this structure and therefore it is important to describe different newsprint qualities in terms of numbers. A parametric model where the parameters can be related to the quality of the newsprint can be a solution to this problem.

There are different ways to measure the surface of newsprint. Much depends on which property is in focus of interest. Is the physical variation on the surface the interesting property, a small needle can be used to sample the surface. Optical measurements are often used. In our applications an optical instrument has been used to measure the reflected light from a laser illuminating small areas on a rectangular lattice.The results from such measurements are recorded in two-dimensional arrays and can be visualized as grey-scale images showing the texture of the surface. These images have been modelled as an fourth-order auto-binomial model.

6.1. *Estimating the model parameters*

The parameters of the model have been estimated with both the coding method and the pseudo-likelihood method. We have also used both Newton–Raphson and simulated annealing for the optimization. The results are given in Table 6. The size of the measured piece of

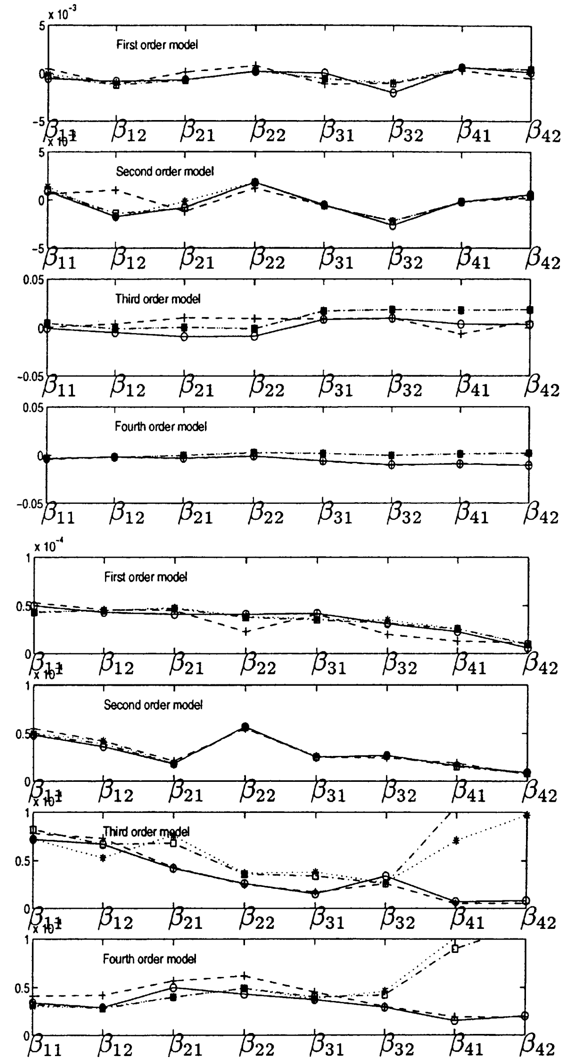

Fig. 9. Graphs showing bias and standard deviations of first, second, third and fourth-order models. The solid lines with the circles are used for the coding method and simulated annealing, the dashed lines with the pluses for the pseudo-likelihood and simulated annealing, the dotted lines with the stars for the coding method and Newton–Raphson and the dot-dashed lines with the squares for pseudo-likelihood and Newton–Raphson. The upper four figures show the biases and the lower four show the standard deviations. Note the different scales.

newsprint is 5×5 cm^2 and 100×100 samples are taken on a regular quadratic lattice.

Based on the value of these estimates, 25 images have simulated and a goodness-of-fit test have been performed.

6.2. Testing goodness-of-fit

To evaluate the appropriateness of the model to the given image we consider goodness-of-fit tests. The null hypothesis is

H_0: The image $\boldsymbol{x}$ is a realization of a fourth-order auto-binomial model.

A ranking test for testing goodness-of-fit of spatial point processes was proposed by Diggle (1979). This test is based on a vector formed by spatial features of the image. From n realizations of the model n different feature vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ are obtained. From these feature vectors and the feature vector of the given image, $\boldsymbol{v}_0$, averages of the vectors, excluding the jth vector, are defined by

$$\overline{\boldsymbol{v}}_j = \frac{1}{n} \sum_{\substack{0 \leqslant i \leqslant n \\ i \neq j}} \boldsymbol{v}_i$$

for $j = 0, 1, \ldots, n$. The distance between the vector $\boldsymbol{v}_j$ and the average omitting $\boldsymbol{v}_j$ is $u_j = \|\boldsymbol{v}_j - \overline{\boldsymbol{v}}_j\|$, where $\|\cdot\|$ denotes Euclidean norm. Finally, these distances are ranked in ascending order $u_{(0)} \leqslant u_{(1)} \leqslant \cdots \leqslant u_{(n)}$. The hypothesis that the given images comes from the same model as the n realizations is rejected at the significance level $100(1 - t/(n+1))$ if $u_0 > u_{(t)}$.

Based on the (unnormed) covariances

$$c_1 = \sum_{(i)\in\mathscr{L}} (x_i - \bar{x})(x_{i:+1} - \bar{x}) = \sum_{(i)\in\mathscr{L}} x_i(x_{i:+1} - \bar{x}),$$

$$c_2 = \sum_{(i)\in\mathscr{L}} x_i(x_{i:+2} - \bar{x}),$$

$$c_3 = \sum_{(i)\in\mathscr{L}} x_i(x_{i:+3} - \bar{x}),$$

$$c_4 = \sum_{(i)\in\mathscr{L}} x_i(x_{i:+4} - \bar{x}),$$

$$c_5 = \sum_{(i)\in\mathscr{L}} x_i(x_{i:+5} - \bar{x}),$$

$$c_6 = \sum_{(i)\in\mathscr{L}} x_i(x_{i:+6} - \bar{x}),$$

$$c_7 = \sum_{(i)\in\mathscr{L}} x_i(x_{i:+7} + x_{i:+9} - 2\bar{x})$$

and $$c_8 = \sum_{(i)\in\mathscr{L}} x_i(x_{i:+8} + x_{i:+10} - 2\bar{x}),$$

$$\left(\text{for } \bar{x} = \sum_{(i)\in\mathscr{L}} x_i/M\right),$$

Table 6
Parameter-estimates of newsprint using four different approaches

Parameter	β_{11}	β_{12}	β_{21}	β_{22}	β_{31}	β_{32}	β_{41}	β_{42}
Coding method using simulated annealing	0.17	0.21	0.07	0.01	−0.02	−0.02	−0.02	−0.01
Pseudo-likelihood using simulated annealing	0.19	0.22	0.07	0.03	0.00	−0.03	−0.03	0.00
Coding method using Newton–Raphson	0.19	0.21	0.05	0.02	−0.01	−0.03	−0.02	−0.02
Pseudo-likelihood using Newton–Raphson	0.18	0.20	0.06	0.03	0.01	−0.01	−0.02	0.00

where M is the number of pixels, correlations are formed by dividing with the (unnormed) variance

$$c_0 = \sum_{(i)\in\mathscr{L}} (x_i - \bar{x})^2.$$

The feature vectors considered in the present case are

$$(c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8)^t / c_0.$$

The test is based on 25 different realizations of the model and the given image.

The test has been performed for the four different methods of the images and none of them resulted in rejection of the null hypothesis at the 5% significance level.

7. Conclusions

In most cases, there are no important differences between the coding method and the pseudo-likelihood method. They produce similar estimates and use the about the same computing time. However, some parameter-vectors are difficult to estimate and only small differences among the parameter values can result in phenomenon similar to those of phase transition. For such images it is very difficult to chose the best method.

The main differences between the two methods to find the global maximum, Newton–Raphson and simulated annealing, are the computing time and the fact that simulated annealing is able to find global maximum in some cases where Newton–Raphson fails.

In applications where auto-binomial models have been assumed but the order of the model not is known or specified, an appropriate method for estimating the parameter is to start with the coding method or pseudo-likelihood method, with Newton–Raphson for the optimization. If no solution is found optimization with simulated annealing should be tried.

Acknowledgements

The author is grateful to two referees for their detailed comments and suggestions on earlier versions of the paper.

References

Aarst, E., Korst, J., 1989. Simulated Annealing and Boltzmann Machines. Wiley, Chichester.

Berkson, J., 1955. Maximum likelihood and minimum χ^2 estimates of the logistic function. J. Amer. Statist. Assoc. 50, 130–162.

Besag, J., 1974. Spatial interaction and the statistical analysis of lattice systems. J. Roy. Statist. Soc. Ser. B 36, 192–236.

Besag, J., 1975. Statistical analysis of non-lattice data. Statistician 24, 179–195.

Chen, C.C., Dubes, R.C., 1989. Experiments in fitting discrete Markov random fields to textures. In: Proc. Computer Vision and Pattern Recognition Conference, San Diego.

Cross, G.R., Jain, A.K., 1983. Markov random fields texture models. IEEE Trans. Pattern Anal. Machine Intell. 5, 25–39.

Deng, M., Dodson, C.T.J., 1994. Paper – an Engineered Stochastic Structure. Tappi Press, Atlanta.

Derin, H., Elliot, H., 1984. Modeling and segmentation of noisy and textured images using Gibbs random fields. IEEE Trans. Pattern Anal. Machine Intell. 9, 39–55.

Diggle, P.J., 1979. On parameter estimation and goodness-of-fit testing for spatial point processes. Biometrics 5, 87–101.

Dubes, R.C., Jain, A.K., 1989. Random fields models in image analysis. J. Appl. Statist. 16, 131–164.

Geman, S., Geman, D., 1989. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Machine Intell. 6, 721–741.

Haralick, R.M., 1984. Statistical and structural approaches to texture. IEEE Trans. Pattern Anal. Machine Intell., 5.

Johansson, J.O., 2000. Modelling the surface structure of newsprint. J. Applied Statist. 27, 425–438.

Ostrowski, A., 1966. Solution of Equations and System of Equations. Academic Press, New York.

Pickard, D.K., 1987. Inference for discrete Markov fields: the simplest nontrivial case. J. Amer. Statist. Assoc. 82, 90–96.

Strang, G., 1988. Linear Algebra and its Applications. Harcourt Brace Jovanovich College Publisher, Orlando.

Winkler, G., 1995. Image Analysis, Random Fields and Dynamic Monte Carlo Methods, a Mathematical Introduction. Springer, New York.