1991

A.J. Baddeley, M.N.M. van Lieshout

Recognition of overlapping objects using Markov spatial processes

Department of Operations Research, Statistics, and System Theory   Report BS-R9109   March

CWI, nationaal instituut voor onderzoek op het gebied van wiskunde en informatica

Recognition of Overlapping Objects using Markov Spatial Processes

A. J. Baddeley

CWI

Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

M. N. M. van Lieshout

Department of Mathematics and Computer Science, Free University
De Boelelaan 1081 a, 1081 HV Amsterdam, The Netherlands

CWI

Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

The Bayesian approach to image processing based on Markov random fields is adapted to image analysis problems such as object recognition and edge detection. Here the input is a grey-scale or binary image and the desired output is a graphical pattern in continuous space, such as a list of geometric objects or a line drawing. The natural prior models are Markov point processes and random patterns from stochastic geometry and spatial statistics. We develop analogues of Besag's ICM algorithm and the Geman-Geman simulated annealing algorithm and present relationships with existing techniques like the Hough transform and the erosion operator in mathematical morphology. The methods are demonstrated on a simple example.

AMS Mathematics Subject Classification (1985 Revision): 68U10, 68T10, 62F15, 60D05.
Key words & Phrases: Hough transform, ICM algorithm, likelihood ratio, maximum a posteriori estimation, mathematical morphology, nearest-neighbour Markov processes, simulated annealing, spatial birth-and-death process, stochastic relaxation, Strauss model, template matching.
Note: This paper will be submitted for publication.

1 Introduction

Grenander [19, 20, 21], Geman and Geman [16, 17] and Besag [8] showed that many image analysis tasks can be reformulated as problems of statistical parameter estimation. Maximum likelihood and Bayesian approaches then suggest iterative algorithms, which are usable and efficient if the underlying stochastic model has a Markov dependence structure. Examples are the application of the E–M algorithm to tomographic reconstruction [46], stochastic annealing and ICM algorithms for segmentation, classification, edge detection and de-blurring [7, 8, 16], and deformable template annealing methods for shape recognition [22, 31, 40].

In this paper we develop a similar approach to feature detection and object recognition,

Report BS-R9109
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

where a scene composed of overlapping objects is observed in the presence of blur and noise, and the task is to determine the number of objects and locate them. Thus the input is a digital image, and the desired output is a graphical pattern in continuous space, such as a line drawing, list of filled polygons or circles, or a three-dimensional wire frame model. The objects to be recognized are assumed to be representable by a small number of continuous real-valued parameters that determine size, shape and location. The number of objects, their position, orientation and spatial relations are not fixed in advance, and the objects may overlap. Applications might include robot vision, document reading, automated chip inspection, region-of-interest selection in medical imaging [9], astronomy [31, 40] and the microscopy of simple structures in material science.

Our approach to object recognition, described in section 2 below, follows the general Bayesian scheme conceived by Grenander [19, 20, 21], Geman and Geman [17, 16] and Besag [8]. A special feature is that the 'true' image x is a list of geometrical objects in continuous space, instead of a digital image. The natural prior probability models for x come from stochastic geometry and spatial statistics [13, 38, 48]. The role played by Markov random fields in stochastic annealing is taken over by Markov random patterns [3, 39].

The deformable template models of Grenander and Keenan [22] and Ripley et al. [31, 40] effectively describe a single object composed of jointed pieces, with Markov stochastic interactions controlling the angles and lengths of the joints. In our approach the objects have a fixed shape or belong to a simple class of shapes, and the Markov model describes the relative spatial positioning of objects (for example, it controls the probability of overlap).

Results of the paper include two new algorithms for object recognition based on iterative conditional maximum likelihood (similar to Besag's ICM) and on stochastic annealing. These are demonstrated on a tame example, but have not yet been tried on real images. Instead, the present paper concentrates on gaining a mathematical understanding. Thus we show that the popular Hough transform [4, 9, 12, 14, 26, 27, 29, 45] is a special case of a likelihood ratio technique (lemmas 4, 8 and 9). The erosion operator of mathematical morphology [43, 44] is the MLE for a specific noise model (cf. lemma 3). We also give examples (lemma 6) to argue that pre-processing the grey level image before searching for objects is simply equivalent to assuming another model for the image noise.

Our stochastic annealing algorithm bears an interesting resemblance to the 'blackboard' techniques popular in computer vision, in which a tentative reconstruction of the scene is formed by iteratively adding and deleting component objects. In our algorithm this is done stochastically implementing a spatial birth-and-death process [3, 30, 34, 36].

In the next section we establish some notation and describe existing methods for recognition of simple objects. The likelihood approach is developed in section 3, with some iterative algorithms in section 4. The Bayesian approach is then introduced in section 5. An analogue of simulated annealing is developed in section 6. Finally section 7 compares the behaviour of the different algorithms using quantitative error measures.

2 Notation and background

2.1 General notation

We follow the general formalism described in [8, 16]. Assume the observed image y depends on the 'true' image x through a known probability distribution. Then the objective is to estimate the unknown x given observations of y. In the likelihood approach (e.g. [46]), it is assumed that x is fixed (though unknown) and that the probability distribution of y has a density $f(y|x)$ that depends only on x. This density describes the 'forward problem' of image formation including both the stochastic noise inherent in observing y and the deterministic influence of x on y. Given observation of y, the relationship is reversed so that $f(y|x)$ is regarded as the 'likelihood' of x being the true image. The *maximum likelihood estimator* of x is

$$\hat{x} = \text{argmax}_x f(y|x). \tag{1}$$

Strictly this is the 'maximum likelihood equation' since the maximum need not exist nor be unique in general.

In the Bayesian approach [8, 16, 31, 22], x is assumed to have been generated at random from a prior probability distribution with density $p(x)$. This can be interpreted as expressing prior information about the image, e.g. that a certain configuration is impossible, or that smooth images are more likely than irregular ones. Then the posterior probability distribution for x after observing data y is found by Bayes' theorem,

$$p(x|y) \propto f(y|x)p(x)$$

and the *maximum a posteriori* (MAP) estimator of x is

$$\begin{aligned} \tilde{x} &= \text{argmax}_x p(x|y) \\ &= \text{argmax}_x f(y|x)p(x). \end{aligned} \tag{2}$$

Because of the latter expression, $p(x)$ can be regarded as a smoothing penalty attached to the optimization of f, and $\tilde{x}$ can be described as a penalized maximum likelihood estimator.

2.2 Assumptions specific to this paper

The observed image y is digitized on a finite pixel lattice T ('image space'), and y_t denotes the observed pixel value at pixel $t \in T$. The values y_t may be real, integer, binary or may belong to any arbitrary set. If y is a binary image ($y_t = 0$ or 1) we identify 1 with 'white' and 0 with 'black', and write

$$Y = \{t \in T : y_t = 1\} \tag{3}$$

for the set of white pixels.

The objects to be recognized are assumed to be representable by a finite number of real parameters that determine size, shape and location. Let U be the space of possible parameter values ('object space'), so that a single point u represents an object $R(u) \subseteq T$. For example, discs of fixed radius r can be identified by their centre points, so $U = \mathbf{R}^2$ would suit; discs of variable radius can be identified by pairs (x, r) where r is the radius and x the centre point, so $U = \mathbf{R}^2 \times (0, \infty)$; the location of an industrial robot can be specified by the position and orientation of the body and the attitude of each joint. We shall assume U is either a bounded open region of d-dimensional space $\mathbf{R}^d$ ('continuous case'), or a finite set of points in $\mathbf{R}^d$ ('discrete case').

An object configuration is an unordered list of such objects,

$$x = \{x_1, \cdots, x_n\}, \ x_i \in U, \ i = 1, \cdots, n, \quad n \geq 0.$$

Note that the length of the list is variable, and the empty list $\emptyset$ is allowed. The objects may be in any relation to each other.

In this paper we typically associate the list x with the 'silhouette' scene $S(x)$ formed by taking the union of all the objects in the list,

$$S(x) = \bigcup_{i=1}^{n} R(x_i)$$

and the corresponding binary image

$$s^{(x)}(t) = \begin{cases} 1 & \text{if } t \in S(x) \\ 0 & \text{else} \end{cases}$$

2.3 Existing methods

Object recognition techniques are surveyed in [5, 15, 33, 42]. They can be divided into methods like region growing [50] which detect an object of unspecified shape and size by characterising it as a region of homogeneous pixel intensity (etc.), and template matching methods which compare the data image with a translated and rotated copy of a reference shape and locate the optimal match [42]. Here we follow the template matching approach, which in general involves

(a) computing a numerical criterion for the degree of match;

(b) optimizing (a);

(c) smoothing or post-processing the result of (b).

A standard criterion for (a) is the *Hough transform* [4, 9, 12, 14, 26, 27, 29, 45]. In our notation, this is a function of the object parameter u defined by

$$H_y(u) = \sum_{t \in R(u)} y_t \, , u \in U \tag{4}$$

where y is the data image. If y is binary, $H_y(u)$ is the number of pixels inside object $R(u)$ that have value 1. In the optimization phase (b), objects are located typically by finding local maxima of the matching criterion, or by accepting all template positions where the match exceeds a threshold value: see [6, 49].

An alternative approach to (a)–(b) using mathematical morphology [43] is to perform an erosion with respect to the template. For example if y is a binary image and Y is the set of white pixels as in (3) define the generalized erosion of Y by

$$\begin{aligned} Y \ominus \check{R} &= \{u : R(u) \subseteq Y\} \\ &= \{u : y_t = 1 \text{ for all } t \in R(u)\} \end{aligned} \tag{5}$$

i.e. accept only those positions u where every pixel in the template is white. This is a generalized erosion operator, roughly as defined by Serra [43, 44]. On a discrete image lattice, the erosion is the set of u points where the Hough transform achieves its maximum possible value.

A third alternative to (a)–(b) is to match boundary edges, detected by a step-edge filter [11, 47] and subsequently peak-detected and smoothed, or by first dividing the image into regions of similar gradient orientation and then fitting one line in each region [10].

Post-processing or smoothing (c) is needed to remove impossible configurations (gaps in edges, isolated points), and for stability. In the presence of noise, template methods tend to reconstruct one object as a a cluster of many almost-identical objects. The number of objects is thus overestimated, and the reconstruction is unstable. This is particularly difficult when the objects themselves overlap.

Most smoothing methods involve (explicitly or implicitly) iterative optimization of a smoothness criterion. Here the experience of edge detection is valuable: although deterministic sequential edge-following techniques [23, 18] are quite successful, the effect of an error at an early stage can be severe. This can be alleviated by allowing backup i.e. stepping back to reconsider earlier decisions, but a better alternative is deterministic or stochastic relaxation [41, 24, 16].

3 Likelihood approach

Here we present several examples of stochastic models $f(y|x)$ and methods for solving the ML equation (1).

Let V be the set of possible pixel values for the data image y; typically V is the real line, or the integers 0 to 255, or the set $\{0, 1\}$.

Definition 1 *An* **independent noise model** *is a stochastic model for y given x which assumes pixel values y_t are stochastically independent given x, with joint probability density*

$$f(y|x) = \prod_{t \in T} g(y_t | \theta^{(x)}(t)) \tag{6}$$

where $\{g(\cdot|\theta) : \theta \in \Theta\}$ is a family of probability densities on V, indexed by a parameter θ in an arbitrary set Θ, and the 'signal' $\theta^{(x)}(t)$ is a function determined by x with values in Θ. Thus the distribution of y_t depends on x only through $\theta^{(x)}(t)$.
The signal is **blur-free** *if*

$$\theta^{(x)}(t) = \theta(s^{(x)}(t)) = \begin{cases} \theta_1 & \textit{if } t \in S(x) \\ \theta_0 & \textit{else} \end{cases}$$

i.e. if the distribution of y_t depends only on whether t belongs to $S(x)$. Then θ_0, θ_1 are called the **background** *and* **foreground parameters** *respectively.*

This model states that the noise is statistically independent between pixels. Note that no assertions are made about the way objects interact and that the model does not imply that the pixel values are (unconditionally) independent.

If $g(\cdot) > 0$ computation of the MLE is equivalent to maximizing the log-likelihood

$$L(x; y) = \log f(y|x) = \sum_{t \in T} \log g(y_t | \theta^{(x)}(t))$$

which is a sum of error terms associated with individual pixels.

In the blur-free case the MLE cannot be unique, because $\theta^{(x)}(t)$ depends on x only through $S(x)$; two solutions x with the same silhouette $S(x)$ have the same likelihood.

Model 1: additive Gaussian white noise

The pixel distribution is Gaussian with mean $\mu = \theta^{(x)}(t)$ and *fixed* standard deviation σ:

$$g(y_t|\mu) = (2\pi\sigma^2)^{-1/2} e^{-(y_t - \mu)^2/(2\sigma^2)}$$

This is equivalent to adding i.i.d. Gaussian noise to the signal.

Model 2: additive double exponential noise

The pixel distribution is double exponential with mean $\mu = \theta^{(x)}(t)$ and dispersion parameter λ fixed:

$$g(y_t|\mu) = (2\lambda)^{-1} e^{-\lambda|y_t - \mu|}.$$

Model 3: binary image, salt-and-pepper noise

Here we convert the silhouette $S(x)$ to a binary image and introduce noise by randomly flipping each pixel value with probability p independently of other pixels. Thus $V = \{0, 1\}$,

$$g(y_t|\theta) = \theta^{y_t}(1-\theta)^{(1-y_t)}$$

and

$$\theta^{(x)}(t) = \begin{cases} 1-p & \text{if } t \in S(x) \\ p & \text{else} \end{cases}$$

with $0 < p < 1$ fixed.

Model 4: binary case, pepper noise

This is similar to model 3 except that only background pixels are flipped,

$$\theta^{(x)}(t) = \begin{cases} 1 & \text{if } t \in S(x) \\ p & \text{else} \end{cases}$$

Lemma 1 *The MLE in Model 1 is the solution of the least squares regression of y on the class of functions $\{\theta^{(x)}(t) : x = \{x_1, \ldots, x_n\}, x_i \in U, n \geq 0\}$. In model 2 the MLE is the solution of a least absolute deviation regression on the same class.*

This is clear since both are 'location models' of the form

$$f(y|x) = \prod_{t \in T} h\left(y_t - \theta^{(x)}(t)\right)$$

and the log-likelihood is (model 1)

$$L(x; y) = -{}^1/_2 \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t \in T} \left(y_t - \theta^{(x)}(t)\right)^2$$

and (model 2)

$$L(x; y) = -\log 2\lambda - \lambda \sum_{t \in T} |y_t - \theta^{(x)}(t)|.$$

The result is not practically useful because of the combinatorial and geometric complexity of the functions $\theta^{(x)}(t)$. We return to this problem further below.

Lemma 2 *For model 3 with $0 < p < 1/2$, the MLE is*

$$\hat{x} = argmin_x |S(x) \triangle Y|$$

where $\triangle$ denotes the symmetric set difference ('exclusive-or').

Proof: The density is

$$f(y|x) = \prod_{t \in S(x)} p^{1-y_t}(1-p)^{y_t} \prod_{t \in T \setminus S(x)} p^{y_t}(1-p)^{(1-y_t)}$$

so that

$$\begin{aligned} L(x;y) &= |S(x) \setminus Y| \log p + |S(x) \cap Y| \log(1-p) \\ &\quad + |Y \setminus S(x)| \log p + (|T| - |Y \cup S(x)|) \log(1-p) \\ &= |T| \log(1-p) + |S(x) \bigtriangleup Y| \log \frac{p}{1-p} \end{aligned}$$

and for $p < 1/2$ the coefficient of $|S(x) \bigtriangleup Y|$ is negative. □

The next result shows a connection between maximum likelihood and mathematical morphology.

Lemma 3 *A maximum likelihood estimator for model 4 is*

$$\begin{aligned} \hat{x}_{max} &= Y \ominus \check{R} \\ &= \{u \in U : R(u) \subseteq Y\}, \end{aligned}$$

the generalized erosion defined in (5). This is the largest solution of the ML equations; the other solutions are the subsets $x \subseteq \hat{x}_{\max}$ with the same silhouette,

$$S(\hat{x}) = S(\hat{x}_{max}).$$

Proof: The density is nonzero only if $S(x) \subseteq Y$ and in this case equals

$$f(y|x) = \prod_{t \in T \setminus S(x)} p^{y_t}(1-p)^{(1-y_t)}.$$

The log-likelihood is then

$$\begin{aligned} L(x;y) &= \sum_{t \in T \setminus S(x)} [y_t \log p + (1-y_t) \log(1-p)] \\ &= |Y \setminus S(x)| \log p + (|T \setminus S(x)| - |Y \setminus S(x)|) \log(1-p) \\ &= |T| \log(1-p) + |Y| \log \frac{p}{1-p} - |S(x)| \log p \end{aligned}$$

where $|\cdot|$ denotes number of pixels. Hence the ML equations are

$$\hat{x} = \text{argmax}_{\ x: S(x) \subseteq Y} |S(x)|$$

and the result follows. □

4 Iterative maximization of likelihood

As we saw in the previous section, it is sometimes impossible to compute the maximum likelihood estimator explicitly. Iterative maximization techniques can then be used to find the MLE. The simplest form of iterative adjustment is to add or delete objects. We would thus add an object $u \in U$ to the list x, yielding $x \cup u$, if the log likelihood ratio

$$L(x \cup u; y) - L(x; y) = \log \frac{f(y|x \cup u)}{f(y|x)}$$

is sufficiently large; and we would delete one of the existing objects $x_i \in x$ to yield $x \setminus x_i$ if

$$L(x \setminus x_i; y) - L(x; y) = \log \frac{f(y|x \setminus x_i)}{f(y|x)}$$

is sufficiently large. Two iterative algorithms suggest themselves.

Algorithm 1 (Coordinatewise optimization)
Assuming that the parameter space is finite, order the points $u \in U$ in a visitation schedule $\{u_1, \cdots, u_M\}$.

step 0 : *Initialize $x^{(0)} = \emptyset$ or some other sensible initial state;*

step $k = 1, 2, \ldots$ **:**
Visit every $u \in U$ sequentially, and add or remove u_j if the likelihood ratio is sufficiently large. For $j = 1, \ldots, M$ and $m = j + (k-1)M$,

$$x^{(m+1)} = \begin{cases} x^{(m)} \cup u_j & \textit{if } u_j \notin x^{(m)} \textit{ and } L(x^{(m)} \cup u_j; y) - L(x^{(m)}; y) > w \\ x^{(m)} \setminus u_j & \textit{if } u_j \in x^{(m)} \textit{ and } L(x^{(m)} \setminus u_j; y) - L(x^{(m)}; y) > w \\ x^{(m)} & \textit{else} \end{cases}$$

where $w \geq 0$ is a chosen threshold value (in this section we take $w = 0$).

Algorithm 2 (steepest ascent)

step 0: *initialize $x^{(0)} = \emptyset$ or some other sensible initial state;*

steps $k = 1, 2, \ldots$**:**
Given $x^{(k)}$, compute

$$a = \max_{x_i \in x^{(k)}} \left\{ L(x^{(k)} \setminus x_i; y) - L(x^{(k)}; y) \right\}$$

and

$$b = \sup_{u \in U} \left\{ L(x^{(k)} \cup u; y) - L(x^{(k)}; y) \right\}.$$

Assume that these maxima are attained by objects x_i^* and u^* respectively. If $\max\{a, b\} < w$*, then stop. Otherwise, if $b \geq a$, add the corresponding object:*

$$x^{(k+1)} = x^{(k)} \cup u^*$$

while if $a > b$, delete the corresponding object:

$$x^{(k+1)} = x^{(k)} \setminus x_i^*.$$

Again $w \geq 0$ is some chosen threshold value.

Clearly these algorithms increase the likelihood at each step, $f(y|x^{(k+1)}) \geq f(y|x^{(k)})$. As there are only a finite number of possible configurations, convergence of $f(y|x^{(k)})$ is guaranteed and (if $w = 0$) we reach a local maximum of the likelihood function. At worst there is cycling between images of equal likelihood. However the algorithm does not necessarily yield the global maximum likelihood solution, and the local maximum obtained will depend on the initial configuration $x^{(0)}$. For the initial configuration we should therefore choose a sensible state, such as the empty list $\emptyset$, or the set of local maxima of

$$\frac{f(y|\{u\})}{f(y|\emptyset)}.$$

The log likelihood ratios in Algorithms 1 and 2 can be interpreted as the differences in 'goodness-of-fit' attained by altering the list x. The following result shows that these are related to the Hough transform.

Lemma 4 *For any blur-free independent noise model (Definition 1) with $g(\cdot|\cdot) > 0$, the log likelihood ratio depends only on pixels inside the added object $R(u)$:*

$$L(x \cup u; y) - L(x; y) = \sum_{t \in R(u) \setminus S(x)} h(y_t, \theta_0, \theta_1) \tag{7}$$

where

$$h(y_t, \theta_0, \theta_1) = \log \frac{g(y_t|\theta_1)}{g(y_t|\theta_0)}$$

is the difference in 'goodness of fit' at pixel t. In particular the log likelihood ratio of a single object u against an empty scene $\emptyset$ is

$$L(\{u\}|y) - L(\emptyset|y) = \sum_{t \in R(u)} h(y_t, \theta_0, \theta_1). \tag{8}$$

The proof is trivial since $s^{(x \cup u)}(t) \neq s^{(x)}(t)$ only for $t \in R(u) \setminus S(x)$.

The right hand side of (8) is a sum of pixel contributions over the object $R(u)$, analogous to the Hough transform $H_y(t)$ in (4). The more general expression (7) is a generalization of the Hough transform that calls for summation only inside the mask $T \setminus S(x)$.

Lemma 5 *Consider any blur-free independent noise model (Definition 1), where the density g is a one-parameter exponential family*

$$g(y_t|\theta) = \exp\{A(\theta) + B(y_t) + C(\theta) \cdot D(y_t)\}$$

Here A, B are arbitrary real-valued functions and C, D are arbitrary real-valued or vector-valued functions ($\cdot$ denoting inner product). Then

$$\begin{aligned} L(x \cup u; y) - L(x; y) &= (C(\theta_1) - C(\theta_0)) \cdot \sum_{t \in R(u) \setminus S(x)} D(y_t) \\ &\quad + (A(\theta_1) - A(\theta_0)) \, |R(u) \setminus S(x)| \end{aligned}$$

where $|\cdot|$ denotes number of pixels.

This is again trivial: for an exponential family

$$\log \frac{g(y_t|\theta_1)}{g(y_t|\theta_0)} = A(\theta_1) - A(\theta_0) + (C(\theta_1) - C(\theta_0)) \cdot D(y_t)$$

and summation over $R(u) \setminus S(x)$ gives the result.

Thus the algorithms 1 and 2 will add an object u when the average value of a function of y_t over the masked object $R(u) \setminus S(x)$ exceeds a threshold:

$$\text{Mean}\,(z_t, R(u) \setminus S(x)) > A(\theta_0) - A(\theta_1)$$

where

$$z_t = (C(\theta_1) - C(\theta_0)) \cdot D(y_t)$$

and

$$\text{Mean}(z_t, V) = \frac{1}{|V|} \sum_{t \in V} z_t$$

is the average of z_t over pixel subset V.

In particular for model 1

$$L(x \cup u; y) - L(x; y) = \frac{\mu_1 - \mu_0}{\sigma^2} \left\{ \sum_{t \in R(u) \setminus S(x)} y_t - \frac{\mu_1 + \mu_0}{2} |R(u) \setminus S(x)| \right\}$$

since for the Gaussian density with σ fixed we have $A(\mu) = -{}^1/_2 \log 2\pi\sigma^2 - \mu^2/(2\sigma^2)$, $B(y_t) = -y_t^2/(2\sigma^2)$, $C(\mu) = \mu/\sigma^2$ and $D(y_t) = y_t$. That is, the algorithm will add an object when the average y value over the masked object $R(u) \setminus S(x)$ exceeds the average intensity $(\mu_0 + \mu_1)/2$.

Similarly, for model 3 with p fixed,

$$L(x \cup u; y) - L(x; y) = -2 \, \log \frac{p}{1-p} \left\{ \sum_{t \in R(u) \setminus S(x)} y_t - \frac{1}{2} |R(u) \setminus S(x)| \right\}.$$

since $A(p) = \log(1-p)$, $B(y_t) = 0$, $C(p) = \log(p/(1-p))$, $D(y_t) = y_t$. That is, the algorithm adds an object when the average fraction of white pixels in the masked region $R(u) \setminus S(x)$ exceeds one half.

Note that the double exponential distribution does not form an exponential family, but we still have a similar result.

Lemma 6 *For model 2*

$$L(x \cup u; y) - L(x; y) = 2\lambda \left\{ \sum_{t \in R(u) \setminus S(x)} \mathrm{clip}(y_t, \mu_0, \mu_1) - \frac{\mu_1 + \mu_0}{2} |R(u) \setminus S(x)| \right\}$$

where

$$\mathrm{clip}(s, a, b) = \begin{cases} a & \textit{if } s < a \\ s & \textit{if } a \le s \le b \\ b & \textit{if } s > b \end{cases}$$

Proof: Observe that (with $a < b$)

$$\begin{aligned} |s - b| - |s - a| &= 1\{s < a\}(b - a) + 1\{a \le s \le b\}(a + b - 2s) + 1\{s > b\}(a - b) \\ &= a + b - 2s + 1\{s < a\}(-2a + 2s) + 1\{s > b\}(-2b + 2s) \\ &= a + b - 2\mathrm{clip}(s, a, b) \end{aligned}$$

and continue as before. □

This case is interesting because it shows that the use of a certain type of pre-processing, namely clipping, before applying the Hough transform, is equivalent to assuming a double exponential model for the un-processed data.

Figure 1 shows a tame example in which a pattern of discs of fixed radius has been observed after addition of Gaussian noise. Figures 2 and 3 show the reconstructions obtained respectively by the coordinatewise optimization and steepest ascent algorithm, taking threshold value $w = 0$. Note that the coordinatewise algorithm performs reasonably well in regions with isolated objects, but fails when the discs overlap each other (figure 2). The pixels were scanned in row major order and the initial configuration empty.

When using the steepest ascent algorithm, it is important to stop short of convergence. Typically, when all the objects that are really present have been detected, the method keeps adding spurious ones. Usually, provided that the data is not too noisy, the increase in likelihood drops when the 'best' reconstruction is reached. Figure 3a shows the best reconstruction for our example and the final one is given in figure 3b.

5 The Bayesian approach

Maximum likelihood solutions x tend to contain clusters of almost identical objects. This phenomenon is undesirable if the number of objects is an important consideration, or if it is known that objects cannot overlap, or if the number of objects is effectively fixed (say, if it is unlikely that there is more than one object). Further, the use of a 'hard' threshold criterion implies that likelihood ratio procedures are sensitive to changes in the data, and that the results of coordinatewise optimization may depend on the scan order in the object space.

Analogous to the introduction of Markov random fields as prior models or roughness penalties for the true image in segmentation problems [8, 16], we too introduce a prior model to solve this instability problem. Since x is no longer a discretized image but a variable-length list $x = \{x_1, \ldots, x_n\}$ of parameter points in a continuous space, the natural stochastic models come from stochastic geometry and spatial statistics. The prior probability $p(x)$ for image x will now be a probability density with respect to the Poisson process [3, 13, 36, 48]. Suitable choices for $p(x)$ will be discussed further below.

The posterior probability density for x given observation of image y is formally identical to that in section 2.1, $p(x|y) \propto f(y|x)p(x)$ and the maximum a posteriori (MAP) estimator is again defined by (2). However (2) is now an optimization over variable-length lists x of parameter points in continuous space.

For example, in model 1 with prior $p(x)$, the MAP equations require minimizing

$$\frac{1}{2\sigma^2} \sum_{t \in T} \left(y_t - \theta^{(x)}(t) \right)^2 - \log p(x);$$

for model 2

$$\lambda \sum_{t \in T} \left| y_t - \theta^{(x)}(t) \right| - \log p(x);$$

and for model 3 ,

$$|S(x) \bigtriangleup Y| \log \frac{1-p}{p} - \log p(x).$$

For model 4 , the MAP equations require maximizing

$$\log p(x) - |S(x)| \log p$$

subject to the constraint $S(x) \subseteq Y$. Thus it is generally not possible to solve the MAP equations directly. We shall use iterative algorithms similar to those developed in section 4.

Algorithm 3 (continuous ICM) *Apply Algorithms 1 or 2 with $f(y|x)$ replaced by the posterior probability $p(x|y)$. Thus we iteratively add object u to list x iff*

$$\frac{f(y|x \cup u)p(x \cup u)}{f(y|x)p(x)} > 1.$$

The convergence properties of Algorithms 1 and 2 remain valid for this new objective function. An alternative description of Algorithm 3 is that the static threshold value used in the likelihood ratio algorithms is replaced by one that depends on the current reconstruction and on a smoothing parameter.

Algorithm 3 is analogous to Besag's ICM algorithm [8]. In the discrete case, consider labelling each $u_j \in U$ with value

$$v_j = \begin{cases} 1 & \text{if } u_j \in x \\ 0 & \text{else} \end{cases}$$

The ICM approach would be to visit each object sequentially and update its label in the light of current estimates for the other objects: when the labelling is $\hat{v}$, the label of the current object u_j is updated to

$$k = \text{argmax } \mathbf{P}\left(v_j = k | y, (\hat{v}_i)_{i \neq j}\right).$$

But this is clearly equivalent to comparing

$$f(y|\hat{x} \setminus u_j) p(\hat{x} \setminus u_j)$$

against

$$f(y|(\hat{x} \setminus u_j) \cup u_j) p((\hat{x} \setminus u_j) \cup u_j)$$

where $\hat{x}$ is the current object list; and this is Algorithm 3.

Now we turn to the choice of $p(x)$. The appropriate analogues of Markov random fields are nearest-neighbour Markov point processes [3], generalizations of the Markov point processes of Ripley and Kelly [39]. Their essential property, that replaces the local interaction property of Markov random fields [8, 16], is that $p(x \cup u)/p(x)$ depends only on local information.

Definition 2 *A* **Markov overlapping object process** *is a stochastic point process on U having a probability density $p(x)$ with respect to the standard Poisson process on U, such that*

$$p(x \cup u) > 0 \Longrightarrow p(x) > 0;$$

and whenever $p(x) > 0$,

$$\frac{p(x \cup u)}{p(x)}$$

depends only on u and on those $x_i \in x$ that overlap u, $R(x_i) \cap R(u) \neq \emptyset$.

The following result is the analogue of the Hammersley-Clifford theorem, characterising the structure of interactions (see [3, 39]).

Lemma 7 *A stochastic point process on U is a Markov overlapping object process iff its density is of the form*

$$p(x) = \alpha \prod_{i=1}^{n} q_1(x_i) \prod_{i \neq j} q_2(x_i, x_j) \prod_{i,j,k \text{ distinct}} q_3(x_i, x_j, x_k) \ldots$$

where $\alpha > 0$ is a constant (normalizing the density) and $q_k : U \times \cdots \times U \to [0, \infty)$ are measurable functions such that $p(\cdot)$ is integrable, and such that $q_k(x_{i_1}, \cdots, x_{i_k}) \neq 1$ implies that all x_{i_j} are pairwise overlapping.

Here $q_k(u_1, \ldots, u_k)$ can be regarded as a measure of interaction between objects $u_1, \ldots, u_k$. Values of $q_k(\cdot)$ less than 1 indicate 'repulsion', i.e. a tendency for this configuration to be avoided; values equal to 1 indicate an absence of interaction; values greater than 1 indicate 'clustering' or a tendency for this configuration to be favoured. Equivalently, one may say the process imposes an additive penalty $\log q_k(u)$ for the presence of each object u, a penalty $\log q(u, v)$ for the conflict between each pair of objects u, v, and so on.

Prior Model 1: Strauss process

This is a generalization of the Strauss point process [3, 13, 39, 48], defined by constant penalties

$$\begin{aligned} q_1(u) &= \beta \\ q_2(u,v) &= \begin{cases} \gamma & \text{if } R(u) \cap R(v) \neq \emptyset \\ 1 & \text{otherwise} \end{cases} \\ q_k(\cdot) &\equiv 1 \text{ for } k \geq 3, \end{aligned}$$

where $\beta > 0$ and $0 \leq \gamma \leq 1$. The density is thus

$$p(x) = \alpha \beta^{n(x)} \gamma^{r(x)} \tag{9}$$

where $n(x)$ denotes number of objects in x and $r(x)$ the number of pairs of overlapping objects.

Interaction between objects is controlled by γ. If $\gamma < 1$, there is repulsion between objects; indeed, if $\gamma = 0$, no objects are permitted to overlap. If $\gamma = 1$ we get a Poisson process of intensity β. For $\gamma > 1$ the process is undefined since the density is not integrable.

The Strauss process is a Markov overlapping object process, since

$$\log \frac{p(x \cup u)}{p(x)} = \log \beta + r(x,u) \log \gamma \tag{10}$$

where

$$\begin{aligned} r(x,u) &= r(x \cup u) - r(x) \\ &= \text{number of } x_i \in x \text{ such that } R(x_i) \cap R(u) \neq \emptyset \end{aligned}$$

Prior Model 2: area interaction process

This has density

$$p(x) = \alpha \beta^{n(x)} \delta^{|S(x)|} \tag{11}$$

with parameters $\beta > 0$, $\delta \geq 1$ and where $\alpha > 0$ is the normalizing constant. As usual $n(x)$ is the number of objects in list x and $|S(x)|$ is the number of pixels in the silhouette. For $\delta \geq 1$ this density is bounded above and below (since $|S(x)|$ is bounded) and is thus integrable. The process is Markov since

$$\frac{p(x \cup u)}{p(x)} = \beta \delta^{|R(u) \setminus S(x)|}$$

depends only on those x_i which overlap u. The density can be written as a product of interaction terms using the inclusion/exclusion formula

$$|S(x)| = \sum_{i=1}^{n} |R(x_i)| - \sum_{i \neq j} |R(x_i) \cap R(x_j)| + \sum_{i,j,k \text{ distinct}} |R(x_i) \cap R(x_j) \cap R(x_k)| - \ldots$$

If the Strauss model is used as the prior, the parameter γ controls the tradeoff between goodness-of-fit to the data and 'complexity' of the solution x. Assume $\beta = 1$. For $\gamma = 1$ the MAP estimator is just the maximum likelihood estimator; while when $\gamma = 0$ the MAP estimator maximizes the likelihood subject to the constraint that no two objects overlap.

Lemma 8 *For any blur-free independent noise model (Definition 1) with $g(\cdot|\cdot) > 0$, and the Strauss process prior, the log posterior likelihood ratio depends only on data pixels inside the added object $R(u)$ and on the number of existing objects overlapping u:*

$$\log \frac{f(y|x \cup u)p(x \cup u)}{f(y|x)p(x)} = \log \beta + \sum_{t \in R(u) \setminus S(x)} h(y_t, \theta_0, \theta_1) + r(x,u) \log \gamma.$$

In particular, the log posterior likelihood ratio of a single object u against an empty scene $\emptyset$ is

$$\log \frac{f(y|\{u\})p(\{u\})}{f(y|\emptyset)p(\emptyset)} = \log \beta + \sum_{t \in R(u)} h(y_t, \theta_0, \theta_1)$$

which shows that thresholding the Hough transform of y at a fixed level is equivalent to performing for each possible object u a likelihood ratio test for $\{u\}$ against $\emptyset$ with a Poisson prior model (i.e. taking no interaction between objects).

Algorithm 3 is illustrated in figure 4 for the example of the previous section, using coordinatewise optimization. Again, the pixels were scanned in row major order and the initial configuration was empty.

Lemma 9 *For a blur-free independent noise model (Definition 1) with $g(\cdot|\cdot) > 0$, and the area-interaction prior (11), the log posterior likelihood ratio is*

$$\log \frac{f(y|x \cup u)p(x \cup u)}{f(y|x)p(x)} = \log \beta + \sum_{t \in R(u) \setminus S(x)} h(y_t, \theta_0, \theta_1) + |R(u) \setminus S(x)| \log \delta.$$

For a single object u against an empty scene $\emptyset$,

$$\log \frac{f(y|\{u\})p(\{u\})}{f(y|\emptyset)p(\emptyset)} = \log \beta + \sum_{t \in R(u)} h(y_t, \theta_0, \theta_1) + |R(u)| \log \delta$$

which shows that linear modifications of the Hough transform are equivalent to imposing an area-interaction prior model.

6 Stochastic annealing

An alternative method to approximate the MAP estimator is simulated annealing [17, 16]. For every $H > 0$ define the probability distribution $p_H(\cdot|y)$ by

$$p_H(x|y) \propto \{f(y|x)p(x)\}^{1/H}.$$

As $H \rightarrow 0$, this converges to a uniform distribution on the set of maxima of the posterior distribution (at least when U is discrete). If we could simulate from each distribution $p_H(\cdot)$, we could generate a sequence of approximations to the MAP estimator. This is almost possible, in that for each H we can construct a Markov process with equilibrium distribution $p_H(\cdot|y)$. The stochastic annealing algorithm simulates these processes consecutively with H gradually dropping to zero.

In the case of discrete Markov random fields [8, 17, 16] the required Markov process is the Metropolis algorithm, Gibbs sampler or a similar state-flipping mechanism [25, 32, 37]. In the case of a Markov point process it is a *spatial birth-and-death process* [3, 34, 39]. In our context this is a continuous time, pure jump Markov process with states in the space of all finite object configurations x, with the property that the only transitions are the birth of a new object (a transition from x to $x \cup u$) or the death of an existing one (transition from x to $x \setminus u$). The process is said to have birth rate $B(x, F)$ and death rate $D(x, u)$, for measurable $F \subseteq U$ and $u \in U$, when it can be described as follows:

- given the state x at time t, the waiting time to the first transition after time t is independent of the history of the process and exponentially distributed with parameter $D(x) + B(x)$, where
$$\begin{array}{rcl} D(x) & = & \sum_{x_i \in x} D(x \setminus x_i, x_i) \\ B(x) & = & B(x, U). \end{array}$$
- the next transition is a birth with probability $B(x)/[D(x) + B(x)]$ and a death with probability $D(x)/[D(x) + B(x)]$.
- given that the next transition is a birth, the new point belongs to the measurable subset F with probability $B(x, F)/B(x)$.
- given that the next transition is a death, x_i is deleted from x with probability $D(x \setminus x_i, x_i)/D(x)$.

Under certain conditions discussed below (see [3, 34]) the birth and death rates uniquely specify the process.

If $B(x, \cdot)$ is absolutely continuous with respect to the reference measure ν on U, its Radon-Nikodym density will be denoted by $b(x, \cdot)$. In our context, define a spatial birth-and-death process on the object space U by its birth rate

$$b_H(x, u) = \begin{cases} \left(\frac{f(y|x \cup u)p(x \cup u)}{f(y|x)p(x)}\right)^{1/H} & \text{if } f(y|x)p(x) > 0 \\ 0 & \text{if } f(y|x)p(x) = 0 \end{cases} \tag{12}$$

for $u \notin x$ and death rate

$$D(x \setminus u, u) = 1. \tag{13}$$

The choice of constant death rate follows Ripley [35]. The total transition rate out of state x associated with deaths is then simply $n(x)$, the number of elements in list x; the total birth rate is

$$B_H(x) = \int_U b_H(x, u) d\nu(u).$$

Thus the total transition rate out of state x is $n(x) + B_H(x)$.

Lemma 10 *Let y and H be fixed. Assume $f(y|x) > 0$ for all x. For each $n = 0, 1, \ldots$ define*

$$\kappa_n = \sup_{n(x)=n} B_H(x).$$

Assume either (a) *that $\kappa_n = 0$ for all sufficiently large $n \geq 0$,* or (b) *that $\kappa_n > 0$ for all $n \geq 0$ and both the following hold:*

$$\begin{aligned} \sum_{n=1}^{\infty} \frac{\kappa_0 \cdots \kappa_{n-1}}{n!} &< \infty \\ \sum_{n=1}^{\infty} \frac{n!}{\kappa_0 \cdots \kappa_n} &= \infty \end{aligned}$$

Then there exists a unique spatial birth-and-death process for which $b_H(\cdot)$ and $D(\cdot)$ as above are the transition rates; this process has unique equilibrium distribution $p_H(\cdot|y)$; and it converges weakly to the equilibrium distribution from any initial state x_0 satisfying $f(y|x_0)p(x_0) > 0$.

This is just a restatement of Theorem 2.10 in [3] which is derived from Proposition 5.1 and Theorem 7.1 of [34]. The assumption $f(y|x) > 0$ is needed so that the class $K = \{x : f(y|x)p(x) > 0\}$ is hereditary: $x \in K$ implies $x \setminus u \in K$. Therefore all states in K can be reached from one another with positive probability. Moreover, starting in K, the process never leaves it. We conclude that the birth-and-death process restricted to K is irreducible, so any equilibrium distribution concentrated on K is necessarily unique.

Given that the spatial birth-death processes for each fixed H are well defined and converge in distribution to the corresponding $p_H(\cdot|y)$, the method will be to run an 'annealing schedule' involving a combination of these processes with $H \downarrow 0$. For the moment we concentrate on fixed $H > 0$.

Lemma 11 *For any blur-free independent noise model with $g(\cdot|\cdot) > 0$, and a pairwise interaction prior with 'pure inhibition' $q(\cdot) \leq 1$, the associated spatial birth-and-death process satisfies the conditions of Lemma 10.*

Proof: Use the representation of log likelihood ratios in Lemma 4. Since T is finite, we have upper and lower bounds on the goodness of fit, say $|h(y_t, \theta_0, \theta_1)| \leq a$ for all t. Hence

$$|L(x \cup u; y) - L(x; y)| \leq a|R(u)| \leq a|T|.$$

For $p(\cdot)$ we have

$$\begin{aligned}\frac{p(x \cup u)}{p(x)} &= \beta \prod_{x_i \in x} q(u, x_i) \\ &\leq \beta\end{aligned}$$

so that

$$\begin{aligned}b_H(x,u) &= \left(\frac{f(y|x \cup u)}{f(y|x)} \frac{p(x \cup u)}{p(x)}\right)^{1/H} \\ &\leq \exp((a|T| + \log \beta)/H) \\ &= \kappa \text{ (say).}\end{aligned}$$

Thus $\kappa_n \leq \kappa \nu(U)$ and it is easy to check that the two conditions of Lemma 10 hold. □

From now on, we assume that the object space is finite, say $U = \{u_1, \ldots, u_M\}$. The total birth rate is then simply

$$B_H(x) = \sum_j b_H(x, u_j) \nu(u_j).$$

To simulate the birth-death process, note that the sequence of states of the process after each transition is a (discrete-time) Markov chain which however does **not** have the same equilibrium distribution as the continuous-time process. The (continuous) time spent in each state is exponentially distributed with a mean sojourn time depending on the current state. It is necessary to simulate these sojourn times as well.

The process can be represented as a sequence $(X^{(k)}, T^{(k)})$, $k = 1, 2, \ldots$ of random variables, where $X^{(k)}$ are the successive states and $T^{(k)}$ is the sojourn time in state $X^{(k)}$. The sojourn time $T^{(k)}$ is exponentially distributed with mean $1/(n(x) + B_H(x))$, independent of other sojourn times and of past states. The next state transition is a death with probability $n(x)/(n(x) + B_H(x))$, obtained by deleting one of the existing points with equal probability; otherwise the transition is a birth generated by choosing one of the points $u_j \notin X^{(k)}$ with probability

$$\frac{b_H(x, u_j)}{B_H(x)}$$

and adding u_j to the state. This representation is implemented directly by the following algorithm.

Algorithm 4 *An initial state $x^{(0)}$ is given.*
For each $k = 0, 1, 2, \cdots$

- *Compute $n = n(x^{(k)})$ and $B = B_H(x^{(k)})$;*

- *Generate an exponentially distributed waiting time by setting $t^{(k)} := \frac{-\log V}{n+B}$ where V is a random variable uniformly distributed on $(0, 1)$;*

- *with probability $n/(n + B)$, generate a death $x^{(k+1)} := x^{(k)} \setminus x_I$ by deleting one of the points of $x^{(k)}$ at random with equal probability;*

- *otherwise generate a birth* $x^{(k+1)} := x^{(k)} \cup u_J$ *by choosing a point* $u_j \in U$ *with probability* $\mathbf{P}\{J = j\} = b_H(x^{(k)}, u_j)/B$.

All random steps are to be independent of each other and of the state of the process. To simulate from the equilibrium distribution, the process is run for a 'large' time period C, *and we take* $x^{(K)}$ *where*

$$K = \min\{k = 0, 1, 2, \dots \mid \sum_{i=0}^{k} t^{(i)} > C\}.$$

Since the birth rate $b_H(x,u)$ is an exponential function of the Hough transform, it tends to have sharp peaks as a function of u when H is small or when x is far from the MAP solution. There is then a high probability that the next transition will add a new object u at one of the locations that is close to maximal for $b_H(x,u)$.

This implies that rejection sampling methods are not a workable alternative to Algorithm 4, since the waiting times would be very long. Calculation of $B_H(x)$ seems unavoidable.

To avoid numerical instability or overflow one could first compute

$$c_H(x,u) = \log b_H(x,u) = H^{-1}\left(L(x \cup u; y) - L(x;y) + \log \frac{p(x \cup u)}{p(x)}\right)$$

for each $u = u_j$, to find

$$m(x) = \max_j c_H(x, u_j)$$

then to compute

$$B_H(x) = e^{m(x)} \sum_j \exp\left[c_H(x,u_j) - m(x)\right] \nu(u_j)$$

where the exponential term lies between 0 and 1. The conditional probability of a birth at u_j is computable from the same summands,

$$\frac{b_H(x,u_j)}{B_H(x)} = \exp\{c_H(x,u_j) - m(x)\} \exp\{m(x) - \log B_H(x)\}$$

The reconstruction obtained using Algorithm 4 on the data of figure 1 is displayed in figure 5. Here the temperature decreases at a geometrical rate.

For very small H the following would be more efficient.

Algorithm 5 *Let* $\epsilon > 0$ *be given. For each* $k = 0, 1, 2, \dots$

- *Compute* $n = n(x^{(k)})$ *and for each* u_j *compute* $c_j = \log b_H(x^{(k)}, u_j)$;

- *Find* $m(x^{(k)}) = \max_j c_j$;

- *Take* $v = v(H, x^{(k)}) \geq 0$ *so large that*

$$\nu(U) \exp\{m(x^{(k)}) - v\} \leq \epsilon;$$

- *Rank all the values c_j for $m(x^{(k)}) - v \leq c_j \leq m(x^{(k)})$ to give a list of values $m(x^{(k)}) = m_1 > m_2 > \ldots > m_L > m(x^{(k)}) - v$, and compute the 'multiplicities'*

$$k_\ell = \nu(\{u_j : c_j = m_\ell\});$$

- *Approximate $B_H(x^{(k)})$ by*

$$B = \sum_{\ell=1}^{L} k_\ell e^{m_\ell};$$

- *Proceed as in Algorithm 4, except that to generate a birth, the procedure is to choose one of the values $\ell = 1, \ldots, L$ with probability $k_\ell e^{m_\ell}/B$; then one of the k_ℓ points u_j such that $c_j = m_\ell$ is chosen with respective probabilities $\nu(u_j)/k_l$ yielding $x^{(k+1)} = x^{(k)} \cup u_j$.*

The choice of v is arranged so that B is an approximation to $B_H(x)$ with error at most ϵ:

$$0 \leq B_H(x) - B = \sum_{c_j \leq m(x)-v} e^{c_j}\nu(u_j) \leq \nu(U)\exp\{m(x) - v\} \leq \epsilon.$$

For very small H, when $x^{(k)}$ is not close to the MAP solution, Algorithm 5 amounts to choosing **at random** one of the u_j that maximizes the conditional posterior likelihood ratio $c_H(x, u)$. This is in a sense comparable to the behaviour of Algorithm 3 (steepest ascent version) but we note that this description ceases to hold as the solution approaches optimality.

7 Performance evaluation

7.1 Complexity

When comparing the progress of different algorithms as plotted below, note that one 'step' of each algorithm corresponds to one scan through the discretised parameter space. The number of 'transitions' (additions or deletions of objects) per scan may vary: steepest ascent and stochastic annealing generate one transition per scan, while coordinatewise optimization yields a variable number of transitions (depending on the data and the scanning order).

Every scan (in each of the three algorithms) requires $O(M)$ computations, where M is the number of parameter points. Thus 'steps' are roughly equivalent to total computer time, although this ignores the possibility of parallelism: steepest ascent and stochastic annealing could be implemented in parallel computation, but coordinatewise optimization is inherently sequential.

7.2 Typical behaviour

We have tested Algorithms 1,2,3 and 4 on simulated data and measured the performance quantitatively using Pratt's figure of merit [1] and the Δ_2 metric [2] for binary images.

Consider once again the model of discs of fixed radius, degraded by additive white Gaussian noise. To see what happens to the reconstructions when iteration proceeds, we computed the Δ_2 distance and the figure of merit for the true image and the reconstruction at successive iteration steps. Starting with an empty image, the results for noise of standard deviation 50 are given in figure 6. The pixels were scanned in row major order.

Note that coordinatewise optimization and steepest ascent behave in entirely different ways. Coordinatewise optimization needs only a few scans through the image. In the first step a lot of objects are added, clustered around the correct positions. In subsequent steps these groups are thinned out. In contrast, steepest ascent requires at least as many scans as there are objects in the image. One at a time, new objects are added, gradually improving the reconstruction quality, until all objects are detected; then the reconstructions become worse. This method however yields more accurate reconstructions than coordinatewise optimization algorithms, especially in the non-Bayesian case.

Using other initial configurations does not change this pattern. Coordinatewise optimization converges in a few steps; steepest ascent slowly improves the reconstruction by removing incorrect objects and replacing them by the right ones. Indeed, the comparison is only more favourable to the coordinatewise optimization algorithm if the initial pattern is poor. The number of iteration steps required by this method is nearly insensitive to the initial state, but steepest ascent needs to delete more and more spurious objects.

Figure 7 shows the reconstruction quality as a function of time for the simulated annealing algorithm. The temperature decreases at a geometrical rate.

7.3 Initial state influence

In the previous section we discussed the initial state influence on typical behaviour. In this section the influence on reconstruction quality will be analysed. To this purpose we computed the Δ_2 distance (tables 1 – 3) and Pratt's figure of merit (tables 4 – 6) for the final reconstructions obtained using our methods with several initial states. Note that both similarity measures display roughly the same pattern.

Comparing these tables, we see that the achieved reconstruction quality fluctuates less for algorithm 4 than for other methods. This is in agreement with the fact that the annealing schedule converges to global maxima of the posterior distribution, regardless of the starting configuration (cf. Lemma 10), while all other methods reach only a local maximum and consequently do depend on the initial state.

Not surprisingly, the best reconstructions were obtained when initializing with the true image, but a perfect match is not guaranteed. This reflects the fact that the truth is not necessarily a solution of the MAP equations.

A widely used technique for choosing the initial state is to compute the Hough transform and find its local minima in the foreground region. However, in noisy images this procedure performs very badly. Even when used only as a first state in our algorithms, better reconstructions could be obtained in most cases with an empty image as initial state. Again, if the true objects are translated over a small distance and this is taken as the starting image, the final reconstruction quality is poorer than when starting with an empty image. A possible explanation is that any algorithm based on addition and deletion of objects will require extra time to throw away incorrect estimates and replace them with better ones.

Therefore, as a rule of thumb, we would recommend using the empty list as a starting state for the algorithms, unless additional information about the objects to be detected is available. Even more so as steepest ascent will waste a lot of time throwing away incorrect objects (see the remarks in subsection 7.2). Another advantage is that no preprocessing (e.g. computing the Hough transform) of the data is required.

7.4 Noise influence

It is also of interest to investigate the influence of the noise variance. To do so we generated ten independent realisations of model 1 for several values of σ. Reconstructions were obtained using algorithm 1 and the coordinatewise version of algorithm 3. For each σ the average reconstruction quality was calculated. The results are shown in figure 8. As could be expected, the reconstructions become poorer when more and more noise is added. Note that if there is little to no noise both algorithms yield a perfect reconstruction. Steepest ascent is less sensitive to noise; we still obtained a perfect reconstruction for noise of standard deviation as large as 40. A similar remark holds for simulated annealing.

7.5 Discussion

In this paper we have indicated how the Bayesian approach to image segmentation could be adapted to recognize configurations of overlapping objects. Instead of using Markov random fields, we chose prior models from the class of nearest-neighbour Markov processes. Objects were defined to be neighbours iff their intersection is non-void. Several algorithms were developed and evaluated using synthetic data. Their behaviour on real image data however is a subject open to future research.

Acknowledgements

The authors express their gratitude to Adri Steenbeek for his technical assistance. Algorithms were implemented in C++ using the image processing package `scilaim` [28].

References

[1] Abdou, I.E. and Pratt, W.K. (1979) Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE* **67**, 753–763.

[2] Baddeley, A. (1991) An error metric for images. Manuscript submitted for publication.

[3] Baddeley, A. and Møller, J. (1989) Nearest-neighbour Markov point processes and random sets. *International Statistical Review* **57**, 89–121.

[4] Ballard, D.H. (1981) Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* **13**, 111 – 122.

[5] Ballard, D.H. and Brown, C.M. (1982) *Computer vision* Englewood Cliffs: Prentice-Hall.

[6] Barnea, D.I. and Silverman, H.F. (1972) A class of algorithms for fast digital image registration. *IEEE Transactions on Computing* **21**, 179 – 186.

[7] Besag, J. (1983) Discussion of paper by P. Switzer. *Bulletin of the International Statistical Institute* **50**, 422 – 425.

[8] Besag, J. (1986) On the statistical analysis of dirty pictures (with discussion). *Journal of the Royal Statistical Society, Series B* **48**, 259 – 302.

[9] Blokland, J.A.K., Vossepoel, A.M., Bakker, A.R. and Pauwels, E.K.J. (1989) Automatic assignment of elliptical ROIs: First results in planar scintigrams of the left ventricle. *European Journal of Nuclear Medicine* **15**, 87 – 92.

[10] Burns, J.B., Hanson, A.R. and Riseman, E.M. (1986) Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**, 425 – 455.

[11] Canny, J.F. (1986) Finding edges and lines in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**, 679 – 700.

[12] Cohen, M. and Toussaint, G.T. (1977) On the detection of structures in noisy pictures. *Pattern Recognition* **9**, 95 – 98.

[13] Diggle, P.J. (1983) *Statistical analysis of spatial point patterns.* London: Academic Press.

[14] Duda, R.O. and Hart, P.E. (1972) Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM* **15**, 11 – 15.

[15] Duda, R.O. and Hart, P.E. (1973) *Pattern classification and scene analysis.* New York: John Wiley and Sons.

[16] Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721 – 741.

[17] Geman, D. Random fields and inverse problems in imaging. Saint - Flour lectures. To appear in *Lecture Notes in Mathematics.*

[18] Greanias, E.C., Meagher, P.F., Norman, R.J. and Essinger, P. (1963) The recognition of handwritten numerals by contour analysis. *IBM Journal of Research and Development* **7**, 14 – 21.

[19] Grenander, U. (1976) *Lectures on Pattern Theory.* Vol. 1: *Pattern Synthesis.* New York-Berlin: Springer-Verlag, Applied Mathematical Sciences vol. 18.

[20] Grenander, U. (1978) *Lectures on Pattern Theory.* Vol. 2: *Pattern Analysis.* New York-Berlin: Springer-Verlag, Applied Mathematical Sciences vol. 24.

[21] Grenander, U. (1981) *Lectures on Pattern Theory.* Vol. 3: *Regular Structures.* New York-Berlin: Springer-Verlag, Applied Mathematical Sciences vol. 33.

[22] Grenander, U. and Keenan, D.M. (1989) A computer experiment in pattern theory. *Communications in Statistics - Stochastic Models* **5**, 531 – 553.

[23] Grimsdale, R.L., Sumner, F.H., Tunis, C.J. and Kilburn, T. (1959) A system for the automatic recognition of patterns. *Proceedings of the IEE* **106B**, 210 – 221.

[24] Hancock, E.R. and Kittler, J. (1990) Edge-labeling using dictionary-based relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**, 165 – 181.

[25] Hastings, W.K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97 – 109.

[26] Hough, P.V.C. (1962) *Method and means for recognizing complex patterns.* US Patent 3,069,654.

[27] Illingworth, J. and Kittler, J. (1987) The adaptive Hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **9**, 690 – 698.

[28] ten Kate, T.K., van Balen, R., Smeulders, A.W.M., Groen, F.C.A., and den Boer, G.A. (1990) SCILAIM: a multi-level interactive image processing environment. *Pattern Recognition Letters* **11**, 429 – 441.

[29] Kimme, C. , Ballard, D. and Sklansky, J. (1975) Finding circles by an array of accumulators. *Communications of the ACM* **18**, 120 – 122.

[30] Møller, J. (1989) On the rate of convergence of spatial birth-and-death processes. *Annals of the Institute of Statistical Mathematics* **41**, 565 – 581.

[31] Molina, R. and Ripley, B.D. (1989) Using spatial models as priors in astronomical image analysis. *Journal of Applied Statistics* **16**, 193 – 206.

[32] Peskun, P.H. (1973) Optimum Monte Carlo sampling using Markov chains. *Biometrika* **60**, 607 – 612.

[33] Pratt, W.K. (1978) *Digital Image Processing.* New York: Wiley-Interscience.

[34] Preston, C.J. (1977) Spatial birth-and-death processes. *Bulletin of the International Statistical Institute* **46**, 371 – 391.

[35] Ripley, B.D. (1977) Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society, Series B* **39**, 172 – 212.

[36] Ripley, B.D. (1981) *Spatial statistics.* New York: John Wiley and Sons.

[37] Ripley, B.D. (1987) *Stochastic simulation.* New York: John Wiley and Sons.

[38] Ripley, B.D. (1988) *Statistical inference for spatial processes.* New York: John Wiley and Sons.

[39] Ripley, B.D. and Kelly, F.P. (1977) Markov point processes. *Journal of the London Mathematical Society* **15**, 188 – 192.

[40] Ripley, B.D. and Sutherland, A.I. (1990) Finding spiral structures in images of galaxies. *Philosophical Transactions of the Royal Society of London, Series A* **332**, 477 – 485.

[41] Rosenfeld, A., Hummel, R.A. and Zucker, S.W. (1976) Scene labeling by relaxation operations. IEEE Transactions on Systems, Man and Cybernetics **6**, 420 – 433.

[42] Rosenfeld, A. and Kak, A.C. (1982) *Digital picture processing (second edition).* New York: Academic Press.

[43] Serra, J. (1982) *Image analysis and mathematical morphology.* London: Academic Press.

[44] Serra, J., ed. (1988) *Image analysis and mathematical morphology, Volume 2: Theoretical Advances.* London: Academic Press.

[45] Shapiro, S.D. (1978) Feature space transforms for curve detection. *Pattern Recognition* **10**, 129 – 143.

[46] Shepp, L.A. and Vardi, Y. (1982) Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging* **1**, 113 – 122.

[47] Spacek, L.A. (1986) Edge-detection and motion detection. *Image and Vision Computing* **4**, 43 – 56.

[48] Stoyan, D., Kendall, W.S. and Mecke J. (1987) *Stochastic geometry and its applications.* Chichester: John Wiley and Sons.

[49] Turin, G.L. (1976) An introduction to digital matched filters. *Proceedings of the IEEE* **64**, 1092 – 1112.

[50] Zucker, S.W. (1976) Region growing: childhood and adolescence. *Computer Graphics and Image Processing* **5**, 382 – 399.

Appendix A

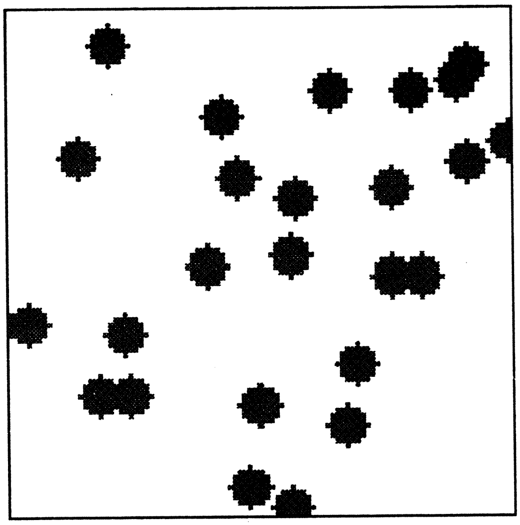


Figure 1A

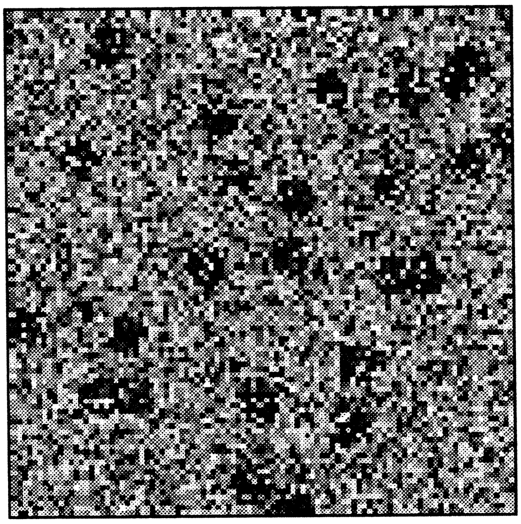


Figure 1B

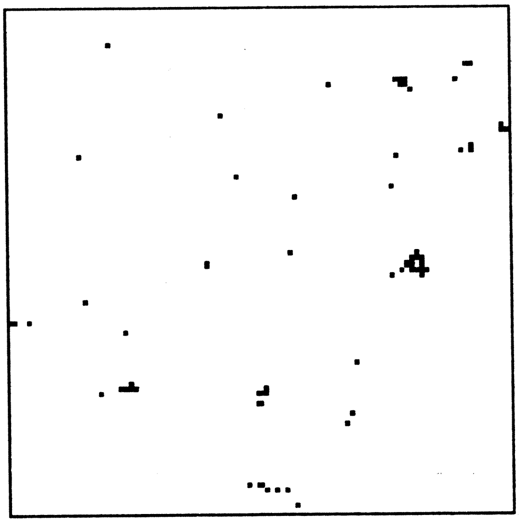


Figure 2

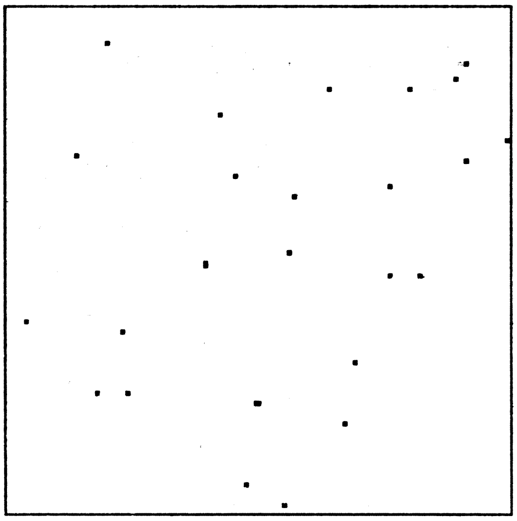


Figure 3A

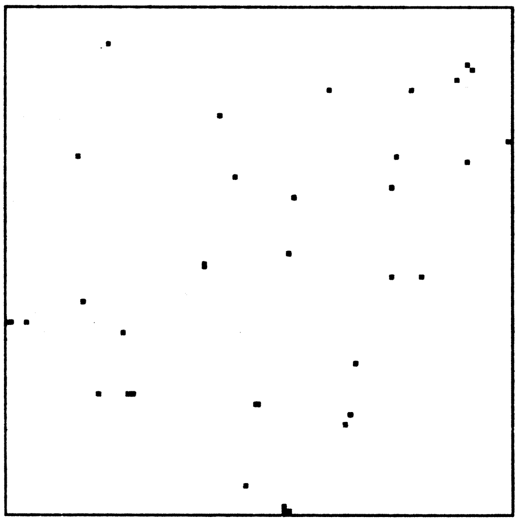


Figure 3B

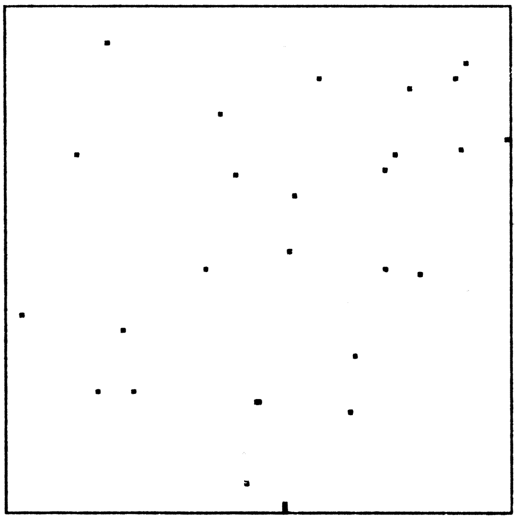


Figure 4

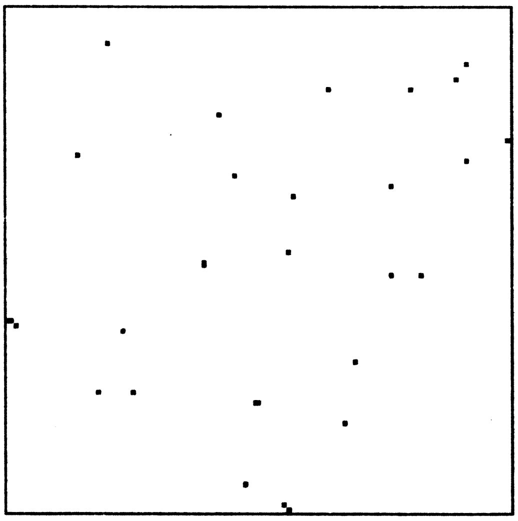


Figure 5

Figure 6A: Reconstruction quality as iteration proceeds

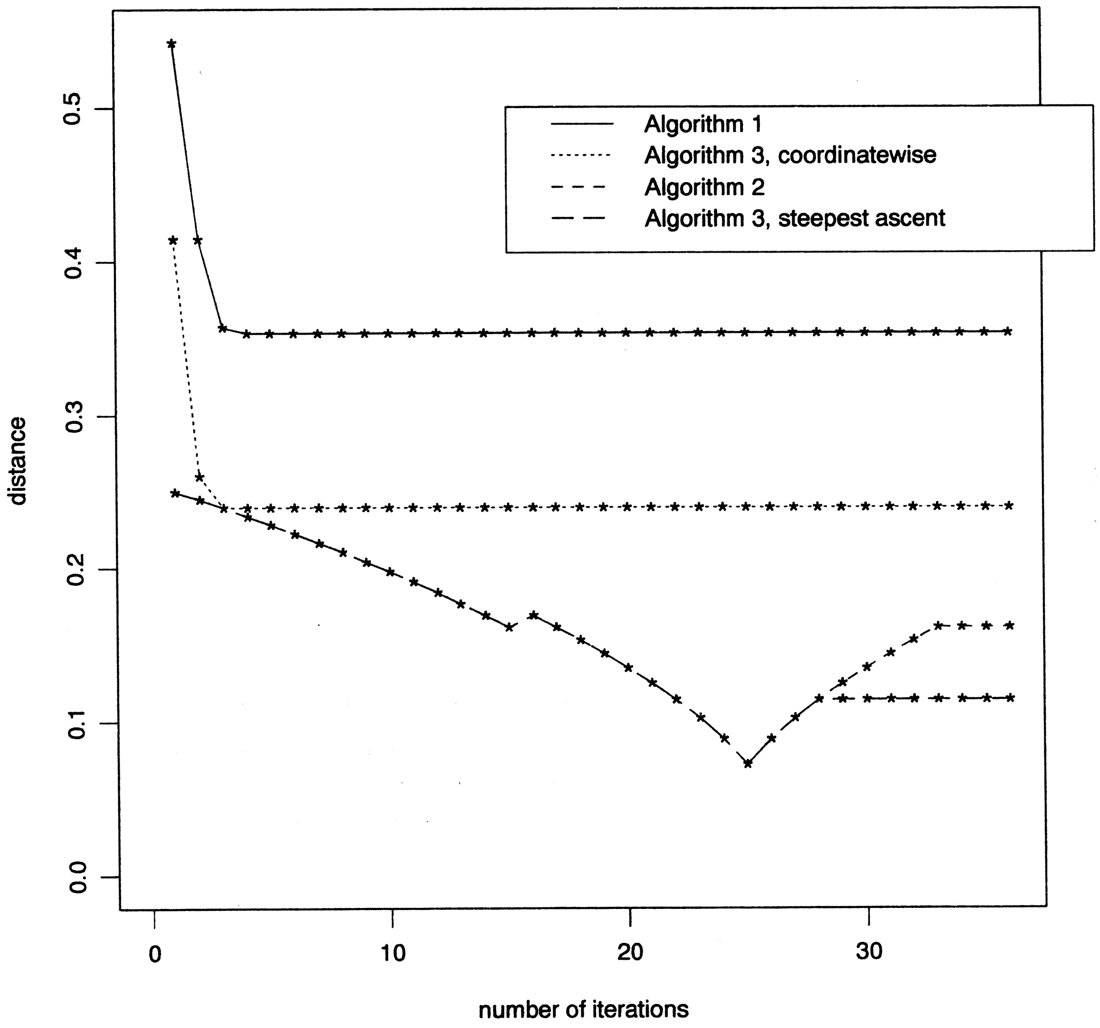


Figure 6B: Reconstruction quality as iteration proceeds

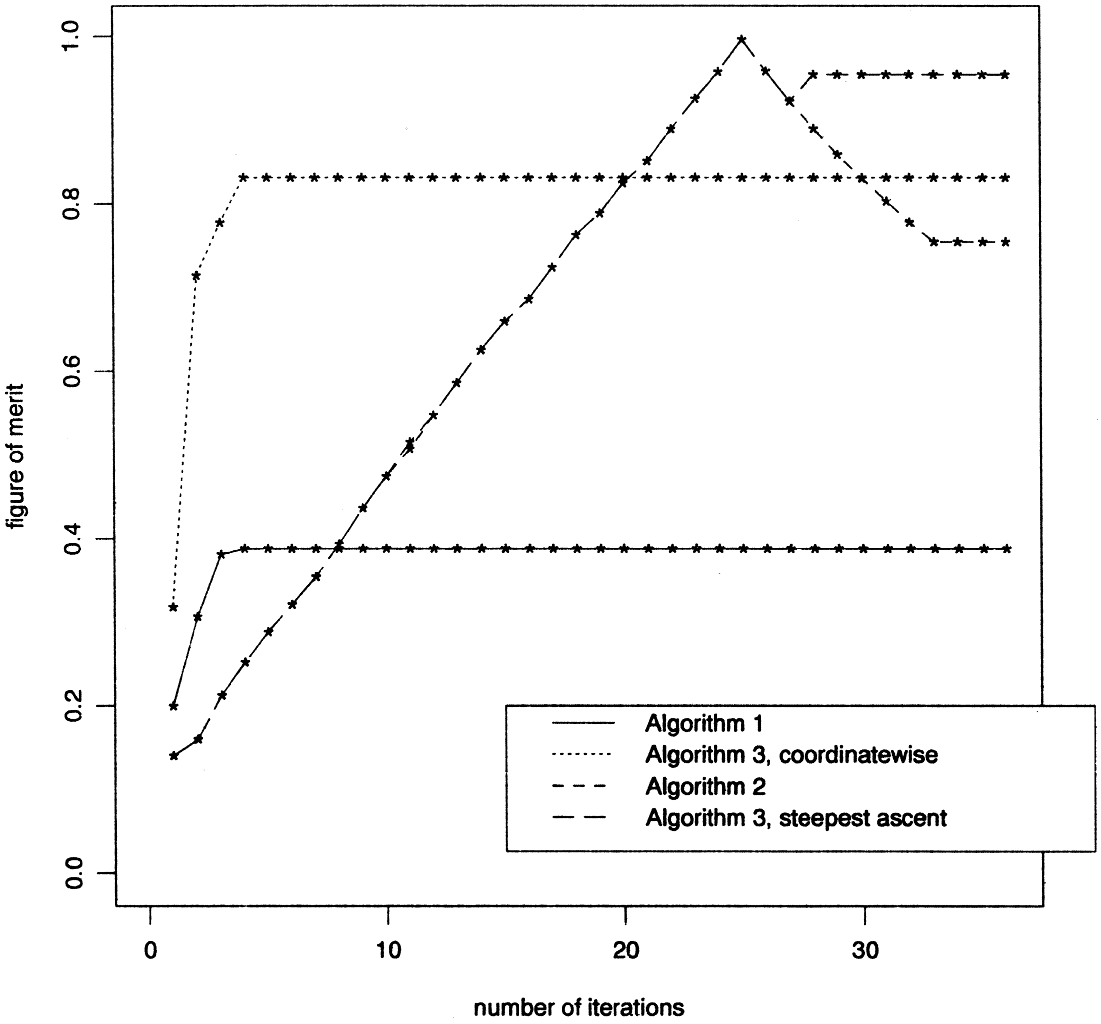


Figure 7: Simulated annealing

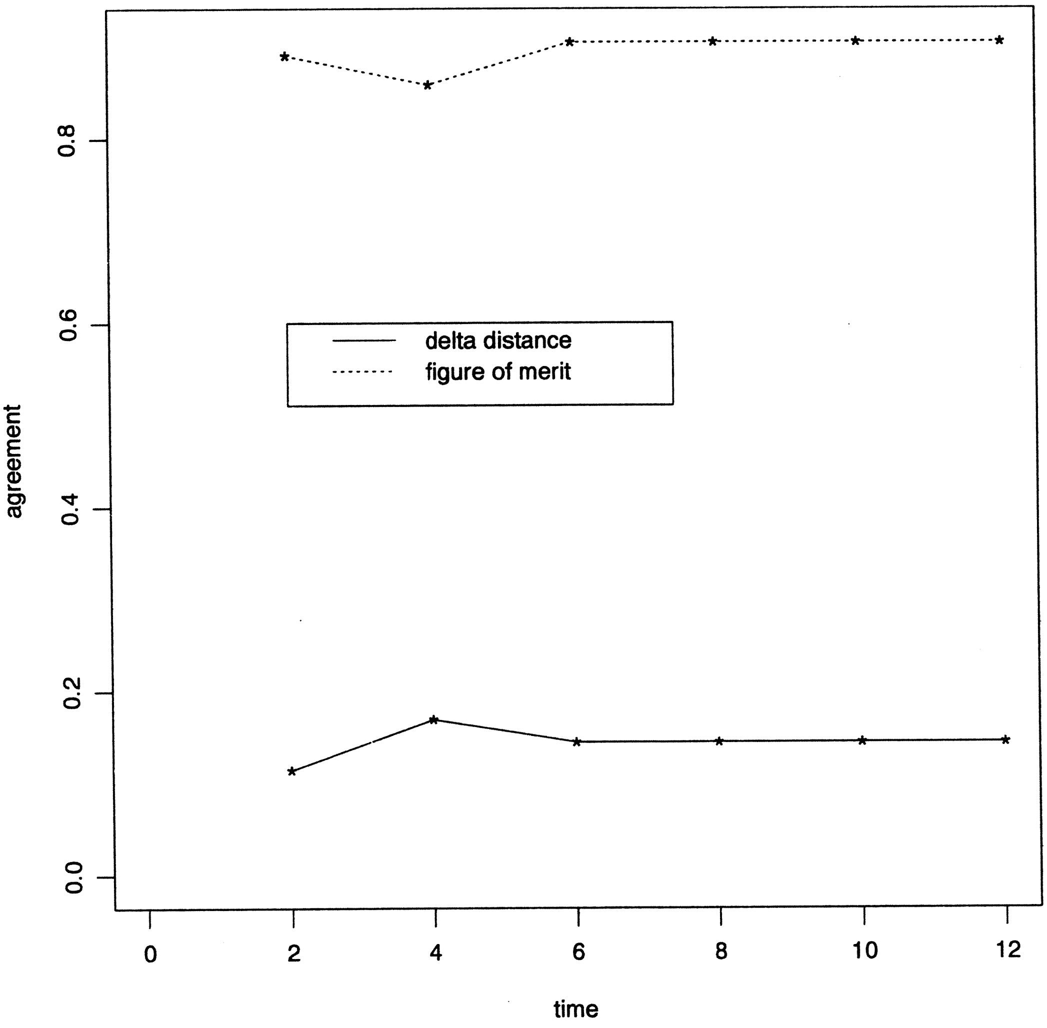


Figure 8A: Influence of noise variance on reconstruction quality

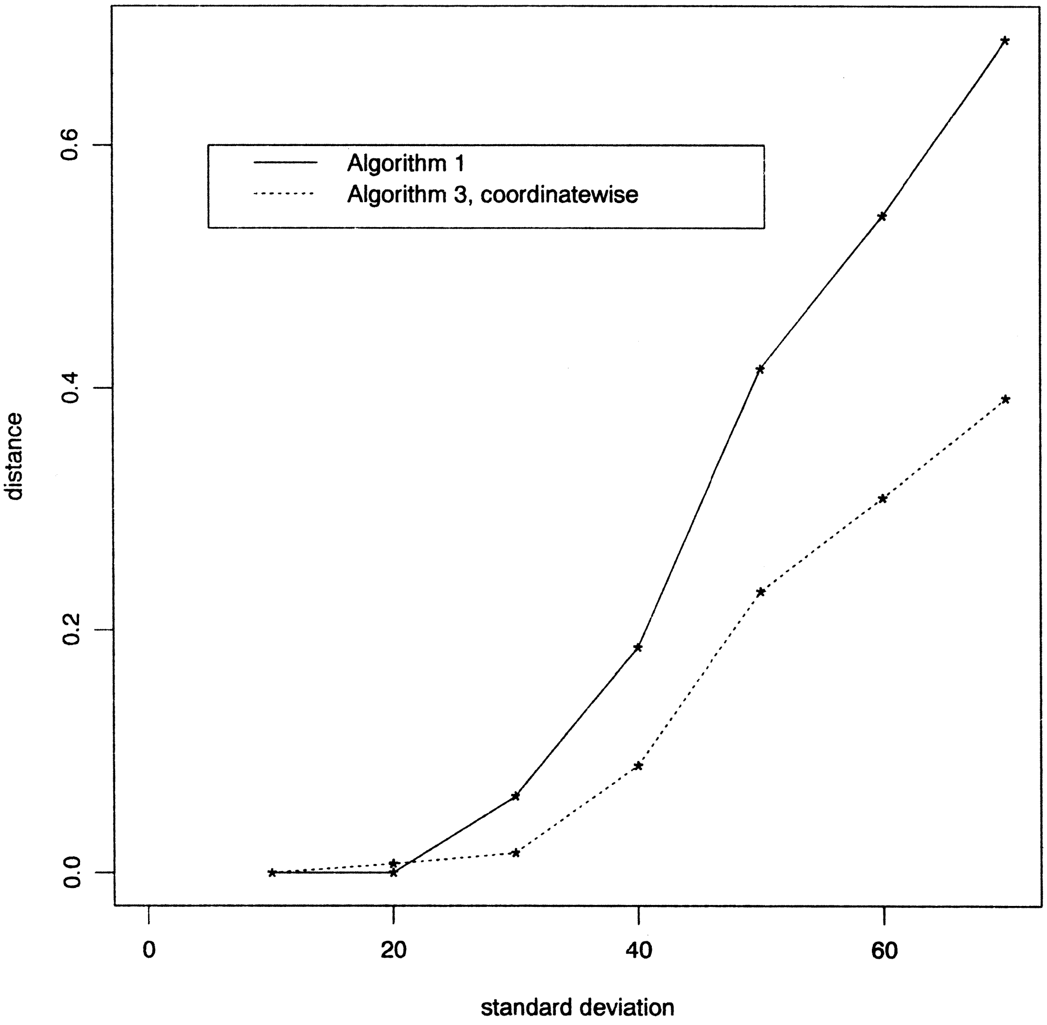
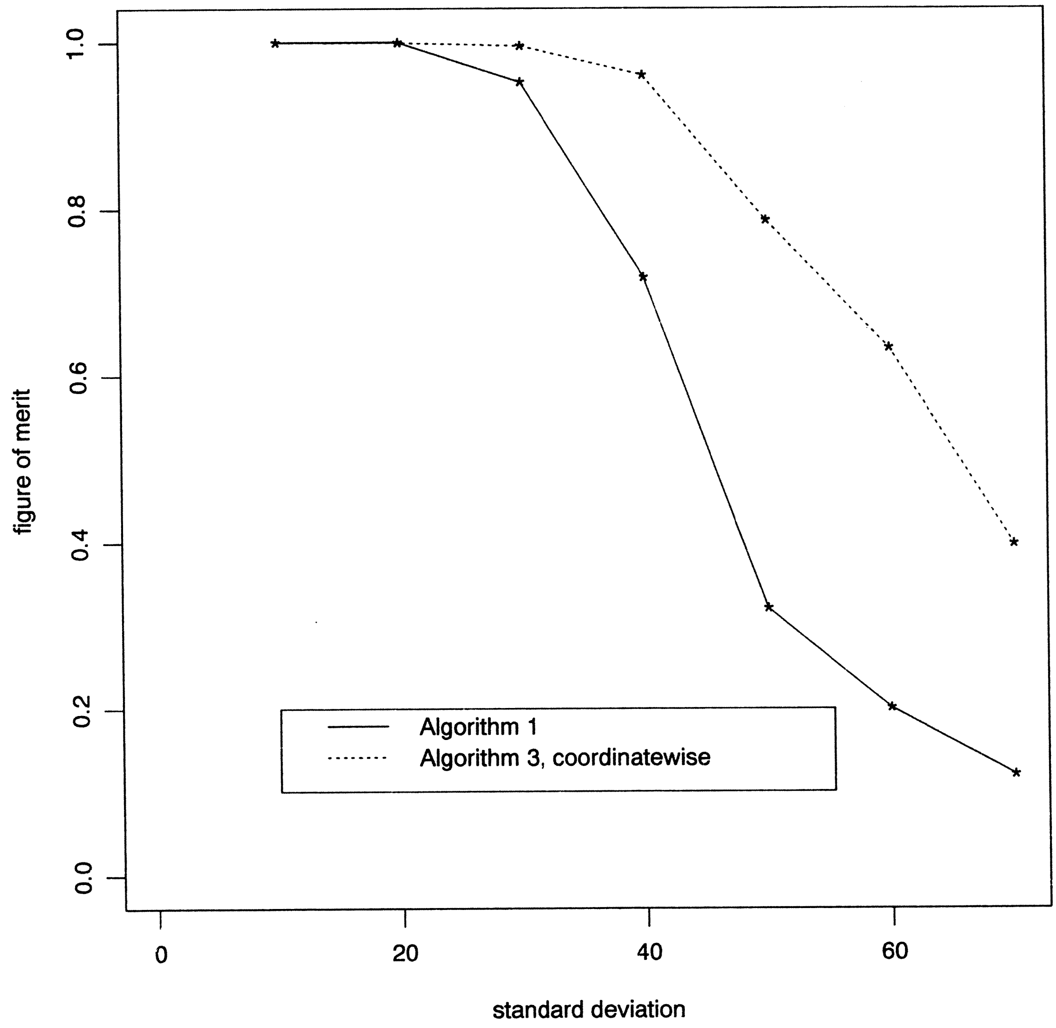


Figure 8B: Influence of noise variance on reconstruction quality



Appendix B

Delta distance

Table 1

Initial pattern	Algorithm 1	Algorithm 3 (coordinatewise)
empty	.353	.239
true	.161	.072
shifted	.255	.260
Hough extrema	.368	.216

Table 2

Initial pattern	Algorithm 2 best	Algorithm 2 at convergence	Algorithm 3 best (steepest ascent)	Algorithm 3 at convergence (steepest ascent)
empty	.072	.161	.072	.114
true	.000	.161	.000	.072
shifted	.234	.234	.210	.210
Hough extrema	.135	.198	.102	.114

Table 3

Initial pattern	Algorithm 4
empty	.144
true	.114
shifted	.114
Hough extrema	.144

Figure of merit

Table 4

Initial pattern	Algorithm 1	Algorithm 3 (coordinatewise)
empty	.387	.831
true	.755	.996
shifted	.582	.782
Hough extrema	.367	.849

Table 5

Initial pattern	Algorithm 2 best	Algorithm 2 at convergence	Algorithm 3 best (steepest ascent)	Algorithm 3 at convergence (steepest ascent)
empty	.996	.755	.996	.954
true	1.000	.755	1.000	.996
shifted	.821	.704	.907	.907
Hough extrema	.942	.683	.992	.954

Table 6

Initial pattern	Algorithm 4
empty	.904
true	.954
shifted	.954
Hough extrema	.904