ICM for object recognition

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

Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

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

Abstract

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 sets. We develop analogues of Besag's ICM algorithm and present relationships with existing techniques like the Hough transform and the erosion operator.

1 Introduction

Many image analysis tasks can be formulated as statistical parameter estimation problems. Examples are the application of the E–M algorithm to tomographic reconstruction [25], stochastic annealing and ICM algorithms for segmentation, classification, edge detection and de-blurring [4, 5, 9], and deformable template annealing methods for shape recognition [13, 18, 21].

This paper develops a similar approach to object recognition, 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. Applications include document reading and robot vision. Thus the input $\mathbf{y}$ is a digital image, and the desired output $\mathbf{x}$ is a graphical pattern in continuous space, such as a line drawing, list of filled polygons, or a wire frame model. We follow the template approach: objects are assumed to be specified by a few continuous real-valued parameters that determine size, shape and location. However the number of objects, their position, orientation and spatial relations are not fixed in advance, and the objects may overlap.

The natural prior probability models for $\mathbf{x}$ come from stochastic geometry and spatial statistics [7, 19, 26]. The role played by Markov random fields is taken over by Markov random patterns [2, 20].

The deformable template models of Grenander and Keenan [13] and Ripley et al. [18, 21] describe a single object, composed of jointed pieces, with the angles and lengths of the joints forming a Markov chain. In our approach the Markov model describes the relative spatial positioning of objects (for example, it controls the probability of overlap).

In the present paper we study deterministic algorithms which are formally analogous to Besag's ICM [5]. A sequel will consider stochastic algorithms analogous to stochastic annealing [9]. We show that the popular Hough transform [3, 6, 8, 14, 15, 24] is a special case of a likelihood ratio technique and the erosion operator of mathematical morphology [22, 23] is the MLE for a specific noise model. We show that pre-processing the grey level image before searching for objects is equivalent to assuming another model for the pixel noise.

In the next section we establish some notation. The likelihood approach is developed in section 3, with some iterative algorithms in section 4. The Bayesian approach is then introduced in section 5. Section 6 compares the peformance of different recognition algorithms.

2 Setup and notation

We follow the general formalism of [5, 9, 10, 11, 12]. Assume the observed image $\mathbf{y}$ depends on the 'true' image $\mathbf{x}$ through a known probability distribution. Then the objective is to estimate the unknown $\mathbf{x}$ given observation of $\mathbf{y}$.

2.1 Data image y

The observed image $\mathbf{y}$ is digitized on a finite pixel lattice T ('image space'), and y_t denotes the observed pixel value at pixel $t \in T$. Let V be the set of possible pixel values for the data image $\mathbf{y}$: typically it is $\mathbb{R}$, or the integers 0 to 255, or $\{0,1\}$. If $\mathbf{y}$ is binary ($V = \{0,1\}$) we identify 1 with 'black' and 0 with 'white', and write $Y = \{t \in T : y_t = 1\}$ for the set of black pixels.

2.2 Object configuration x

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

An object configuration is an unordered list of objects

$$\mathbf{x} = \{x_1, \cdots, x_n\}$$

where

$$x_i \in U, \ i = 1, \cdots, n, \quad n \geq 0.$$

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.

We often associate the list $\mathbf{x}$ with the 'silhouette' scene $S(\mathbf{x})$ formed by taking the union of all the objects in the list,

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

and the corresponding binary image

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

2.3 Existing methods

A standard numerical criterion for template matching is the *Hough transform* [3, 6, 8, 14, 15, 24]. This is a function on object space

$$H_{\mathbf{y}}(u) = \sum_{t \in R(u)} y_t \ , \quad u \in U \tag{1}$$

where $\mathbf{y}$ is the data image. The Hough transform is often interpreted as a 'vote-counting' operation: each pixel in the data image 'votes' for all the objects that contain that pixel. The optimal match is located typically by finding local maxima of the Hough transform in object space.

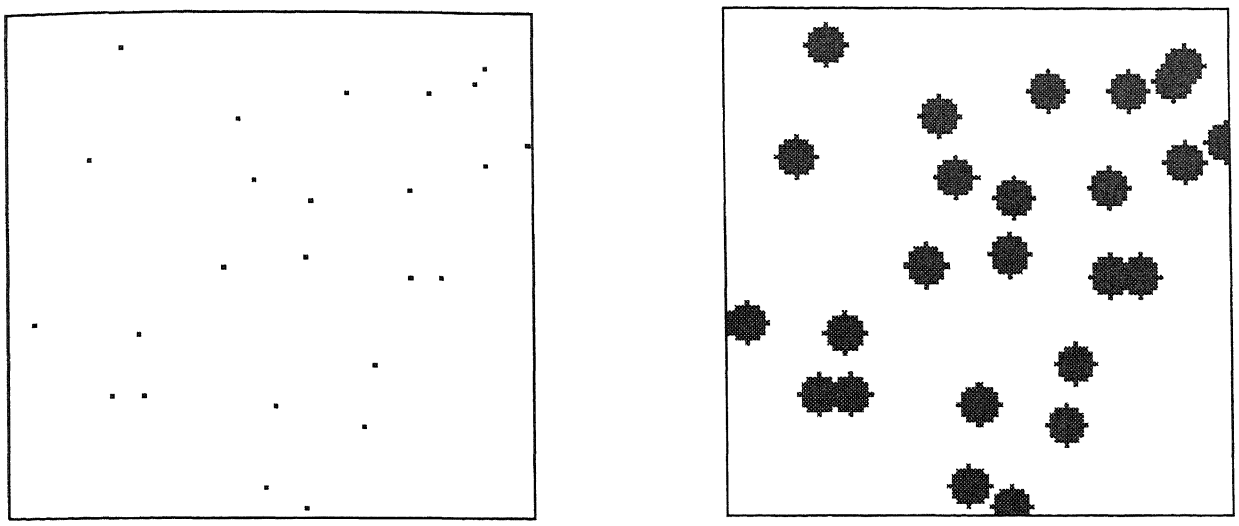

Figure 1: Binary image representing centre points of discs with radius 4, digitized on a 98 × 98 square grid with its induced silhouette.

An alternative for binary images is to perform an erosion with respect to the template. If Y is the set of black pixels define the generalized erosion of Y by

$$\begin{aligned} E_R(Y) &= \{u : R(u) \subseteq Y\} \\ &= \{u : y_t = 1 \text{ for all } t \in R(u)\} \end{aligned} \tag{2}$$

i.e. accept only those positions u where every pixel in the template is white. This is a generalization [23] of the classical erosion operator $Y \ominus \check{R}$ [22].

3 Likelihood approach

In the likelihood approach [25] the true image $\mathbf{x}$ is assumed fixed (though unknown) and the probability distribution of the observed image $\mathbf{y}$ has a density $f(\mathbf{y}|\mathbf{x})$ that depends only on $\mathbf{x}$. Given observation of $\mathbf{y}$, the *maximum likelihood estimator* of $\mathbf{x}$ is

$$\hat{\mathbf{x}} = \text{argmax}_{\mathbf{x}} f(\mathbf{y}|\mathbf{x}). \tag{3}$$

if it exists. In this section we present several examples of stochastic models $f(\mathbf{y}|\mathbf{x})$ and methods for finding the MLE.

We consider simple models which consist of a deterministic deformation of the image $\mathbf{x}$ into a 'blurred signal' $\theta^{(\mathbf{x})}$ followed by a noise process which is statistically independent between pixels, though not necessarily additive. Recall T is the pixel grid and V the set of possible pixel values.

Definition 1 *An* **independent noise model** *is a stochastic model for* $\mathbf{y}$ *given* $\mathbf{x}$ *in which pixel values* y_t *are conditionally independent given* $\mathbf{x}$*, with joint probability density*

$$f(\mathbf{y}|\mathbf{x}) = \prod_{t \in T} g(y_t|\theta^{(\mathbf{x})}(t)). \tag{4}$$

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

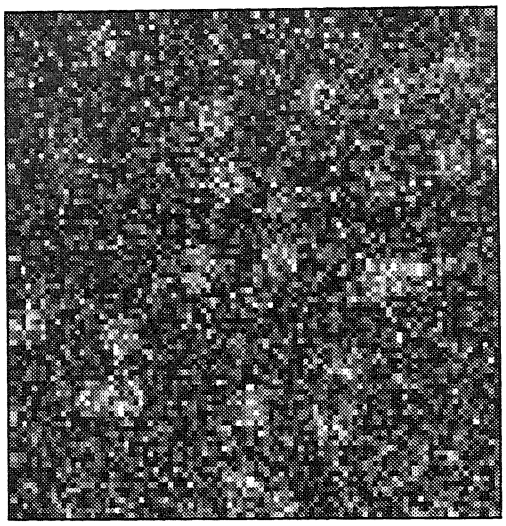

Figure 2: Realization from model 1 with $\sigma = 50$, $\theta_1 = 150$ and $\theta_0 = 100$ conditional on Figure 1.

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.

For brevity, we will only discuss models which are **blur-free**,

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

i.e. the distribution of y_t depends only on whether t belongs to $S(\mathbf{x})$.

Model 1: additive Gaussian white noise

The pixel distribution is Gaussian with mean $\mu = \theta^{(\mathbf{x})}(t)$ 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. A simulation is shown in Figure 2.

Model 2: additive double exponential noise

The pixel distribution is double exponential with mean $\mu = \theta^{(\mathbf{x})}(t)$ 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 $S(\mathbf{x})$ 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

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

with $0 < p < 1$ fixed.

Model 4: binary image, pepper noise

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

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

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

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

This is not practically useful because of the combinatorial and geometric complexity of the functions $\theta^{(\mathbf{x})}(t)$.

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

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

where $\triangle$ *denotes the symmetric set difference ('exclusive-or') and* $|\cdot|$ *denotes number of pixels ('area').*

Computing the MLE is thus equivalent to an L^p optimization problem where $p = 1$ for models 2 and 3 , and $p = 2$ for model 1 . This should be compared to recent arguments in the literature [16, 17] in favour of using L^1 filtering except when the noise is Gaussian.

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

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

$$\hat{\mathbf{x}}_{max} = E_R(Y) = \{u \in U : R(u) \subseteq Y\},$$

the generalized erosion defined in (2). This is the largest solution of the ML equations; the other solutions are the subsets $\mathbf{x} \subseteq \hat{\mathbf{x}}_{\max}$ *with the same silhouette,* $S(\hat{\mathbf{x}}) = S(\hat{\mathbf{x}}_{max})$.

4 Iterative maximization of likelihood

4.1 General add-and-delete algorithms

Iterative maximization techniques can 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 $\mathbf{x}$, yielding $\mathbf{x} \cup \{u\}$, if the log likelihood ratio

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

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

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

is sufficiently large. Henceforth, $w \geq 0$ is a chosen threshold value.

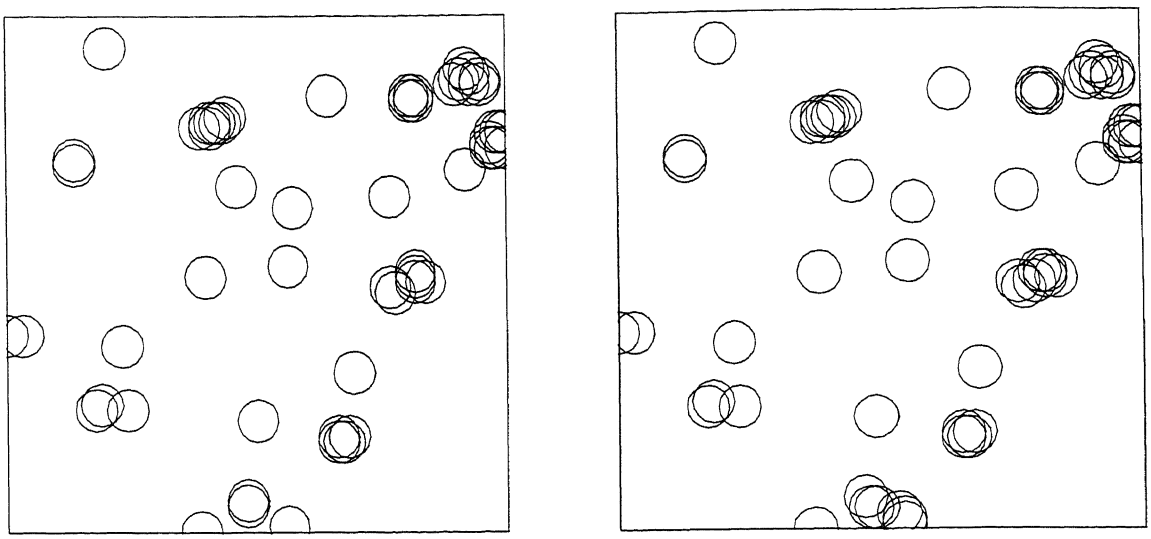

Figure 3: MLE reconstructions using coordinatewise ascent with the local extrema of the Hough transform as initial state. Left: Algorithm 1; right: Algorithm 3.

Algorithm 1 (Coordinatewise optimization) *Initialize* $\mathbf{x}^{(0)} = \emptyset$ *or some other chosen initial state. When the current reconstruction is* $\mathbf{x}^{(m-1)}$*, visit every* $u \in U$ *sequentially in a predetermined order. If* $u \notin \mathbf{x}^{(m-1)}$ *and* $L(\mathbf{x}^{(m-1)} \cup \{u\} \,;\, \mathbf{y}) - L(\mathbf{x}^{(m-1)} \,;\, \mathbf{y}) > w$*, then add* u *to the configuration, yielding* $\mathbf{x}^{(m)} = \mathbf{x}^{(m-1)} \cup \{u\}$*. If* $u = x_i \in \mathbf{x}$ *and* $L(\mathbf{x}^{(m-1)} \setminus \{x_i\} \,;\, \mathbf{y}) - L(\mathbf{x}^{(m-1)} \,;\, \mathbf{y}) > w$*, then delete* x_i *yielding* $\mathbf{x}^{(m)} = \mathbf{x}^{(m-1)} \setminus \{x_i\}$*. Update recursively until one complete scan of the image yields no changes.*

Algorithm 2 (Steepest ascent) *Initialize* $\mathbf{x}^{(0)} = \emptyset$ *or some other chosen initial state. Given* $\mathbf{x}^{(k-1)}$*, compute*

$$a = \max_{x_i \in \mathbf{x}^{(k-1)}} \left\{ L(\mathbf{x}^{(k-1)} \setminus \{x_i\} \,;\, \mathbf{y}) - L(\mathbf{x}^{(k-1)} \,;\, \mathbf{y}) \right\}$$

and

$$b = \sup_{u \in U} \left\{ L(\mathbf{x}^{(k-1)} \cup \{u\} \,;\, \mathbf{y}) - L(\mathbf{x}^{(k-1)} \,;\, \mathbf{y}) \right\}.$$

If $\max\{a, b\} < w$*, then stop. Otherwise, if* $b \geq a$*, add the corresponding object, while if* $a > b$*, delete the corresponding object.*

There is a very strong analogy between the coordinatewise optimization rule and Besag's [5] ICM algorithm, since our rule effectively updates the state of each point of U by maximizing the conditional probability given information about all other points of U and given the data.

Clearly these algorithms increase the likelihood at each step, $f(\mathbf{y} \mid \mathbf{x}^{(k+1)}) \geq f(\mathbf{y} \mid \mathbf{x}^{(k)})$. As there are only a finite number of possible configurations, convergence of $f(\mathbf{y} \mid \mathbf{x}^{(k)})$ 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. The local maximum obtained will depend on the initial configuration $\mathbf{x}^{(0)}$.

4.2 Example

For our synthetic example (Figure 1-2), Figure 3 shows the reconstructions obtained by the coordinatewise optimization algorithm with $w = 0$. The pixels were scanned in row major order and for the initial state we took the local maxima of $\log f(\mathbf{y} \mid \{u\}) - \log f(\mathbf{y} \mid \emptyset)$ where this was

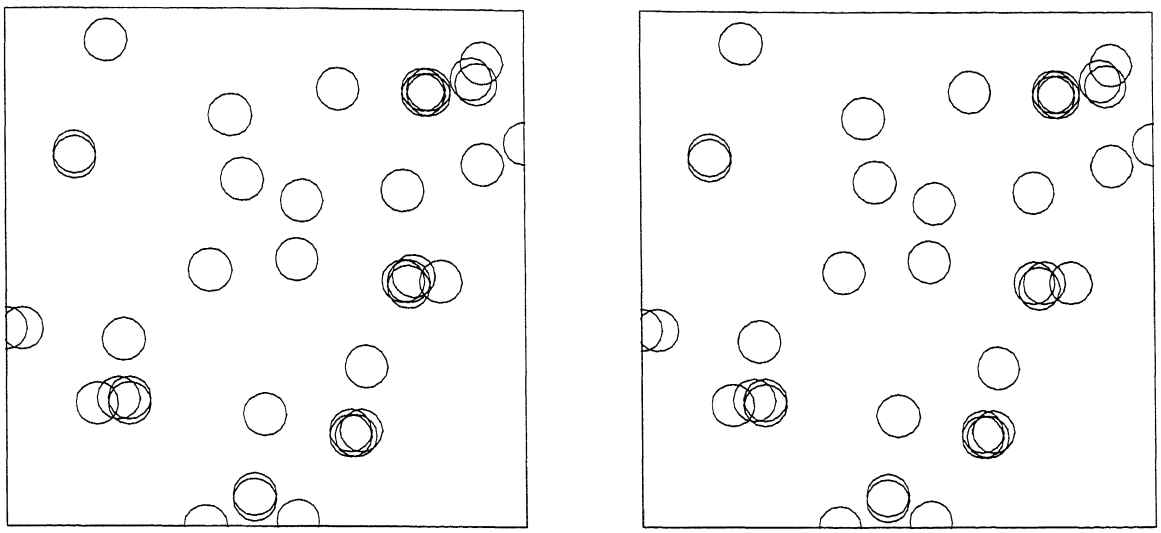

Figure 4: MLE reconstructions using steepest ascent with the empty list as initial state. Left: Algorithm 2; right: Algorithm 4.

non-negative. The algorithms perform reasonably well in regions with isolated objects, but fail when the discs overlap each other.

Figure 4 shows reconstructions obtained using steepest ascent with $w = 0$ and empty initial state. Here it seems better to stop short of convergence, since when all the objects that are really present have been detected, the method keeps adding spurious ones. This can be counteracted by taking a higher threshold value.

4.3 Connection with Hough transform

The log likelihood ratios in Algorithms 1 and 2 can be interpreted as the differences in 'goodness-of-fit' attained by altering the list $\mathbf{x}$. 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:*

$$L(\mathbf{x} \cup \{u\}\,;\,\mathbf{y}) - L(\mathbf{x}\,;\,\mathbf{y}) = \sum_{t \in R(u) \setminus S(\mathbf{x})} h(y_t, \theta_0, \theta_1) \tag{5}$$

where

$$h(y_t, \theta, \theta') = \log \frac{g(y_t|\theta')}{g(y_t|\theta)}$$

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

$$L(\{u\}\,;\,\mathbf{y}) - L(\emptyset\,;\,\mathbf{y}) = \sum_{t \in R(u)} h(y_t, \theta_0, \theta_1). \tag{6}$$

The right hand side of (6) is analogous to the Hough transform (1) in that it is a sum of pixel votes over the object $R(u)$. Here, however, pixels can cast fractional or negative votes, a generalization which has been suggested ad hoc by some authors.

The more general expression (5) may be regarded as a generalization of the Hough transform that calls for conditional summation only inside the mask $T \setminus S(\mathbf{x})$. However it is also expressible as the straightforward Hough transform

$$L(\mathbf{x} \cup \{u\}\,;\,\mathbf{y}) - L(\mathbf{x}\,;\,\mathbf{y}) = \sum_{t \in R(u)} e_t$$

of the 'residual likelihood' image $e_t = (1 - 1_{S(\mathbf{x})}(t))h(y_t, \theta_0, \theta_1) = \log g(y_t|\theta_1) - \log g(y_t|\theta^{(\mathbf{x})}(t))$.

4.4 Relation to preprocessor filters

Pre-processing of the data image before applying the Hough transform is typically equivalent to modifying the noise model. 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

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

where $|\cdot|$ again denotes number of pixels. Thus the algorithms 1 and 2 will add an object u iff the mean of

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

over $R(u) \setminus S(\mathbf{x})$ exceeds the threshold $A(\theta_0) - A(\theta_1)$. In particular for model 1

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

that is, the algorithm will add an object when the average $\mathbf{y}$ value over the masked object $R(u) \setminus S(\mathbf{x})$ exceeds the average intensity $(\mu_0 + \mu_1)/2$. Similarly, for model 3 with p fixed, the algorithm adds an object when more than half the pixels in $R(u) \setminus S(\mathbf{x})$ are black.

It can also be shown that clipping the data image values to a range $[a, b]$ before applying the Hough transform is equivalent to assuming a double exponential model for the original image data.

4.5 General add-delete-shift algorithms

Another form of iterative adjustment is to change an existing object by moving, rotating or expanding it slightly. This should achieve a better optimum, and should accelerate convergence, since throwing away an incorrect object and replacing it by the right one can then be carried out in one single step. Write

$$M(\mathbf{x}, x_i, u) = \mathbf{x} \cup \{u\} \setminus \{x_i\}$$

for the configuration obtained from $\mathbf{x}$ by moving the element $x_i \in \mathbf{x}$ to a new position u. Let $Q(\mathbf{x}, x_i)$ be the set of all object points u for which this operation is permitted. Typically u will be required to be close to x_i but not equal to any x_j, say $Q(\mathbf{x}, x_i) = N(x_i) \setminus \mathbf{x}$ where $N(x_i)$ is some neighbourhood of x_i.

Then Algorithms 1 and 2 can be modified by allowing the following transitions from configuration $\mathbf{x}$:

- deleting x_i from $\mathbf{x}$ yielding $\mathbf{x} \setminus \{x_i\}$;
- if $u \notin \mathbf{x}$, adding u to $\mathbf{x}$ giving $\mathbf{x} \cup \{u\}$;
- if $u \in Q(\mathbf{x}, x_i)$ for some $x_i \in \mathbf{x}$, moving x_i to u giving $M(\mathbf{x}, x_i, u)$.

The corresponding loglikelihood ratios are

- $L(\mathbf{x} \setminus \{x_i\} \; ; \; \mathbf{y}) - L(\mathbf{x} \; ; \; \mathbf{y})$
- $L(\mathbf{x} \cup \{x_i\} \; ; \; \mathbf{y}) - L(\mathbf{x} \; ; \; \mathbf{y})$
- $L(M(\mathbf{x}, x_i, u) \; ; \; \mathbf{y}) - L(\mathbf{x} \; ; \; \mathbf{y})$

respectively. In **Algorithm 3 (Coordinatewise optimization with shifts)** we visit every $u \in U$, consider every possible transition, and update whenever the log likelihood ratio exceeds the threshold w. In **Algorithm 4 (Steepest ascent with shifts)** we scan all possible transitions from a given $\mathbf{x}$, and take that transition which has the maximum log likelihood ratio.

These algorithms again increase the likelihood at every step and and are guaranteed to converge to a local maximum in finite time if $w > 0$. Here a 'local maximum' of the likelihood is a state $\mathbf{x}$ such that no neighbouring configuration $\mathbf{x} \cup \{u\}$ or $\mathbf{x} \setminus \{x_i\}$ or $M(\mathbf{x}, x_i, u)$ has larger likelihood. This is a more stringent definition than for the previous algorithms, and one expects the results to be better.

Interpretation of the algorithms is similar to the previous cases. The log likelihood ratio for a shift can be represented as

$$L(M(\mathbf{x}, x_i, u) \; ; \; \mathbf{y}) - L(\mathbf{x} \; ; \; \mathbf{y}) = [L(\mathbf{z} \cup \{u\} \; ; \; \mathbf{y}) - L(\mathbf{z} \; ; \; \mathbf{y})] - [L(\mathbf{z} \cup \{x_i\} \; ; \; \mathbf{y}) - L(\mathbf{z} \; ; \; \mathbf{y})]$$

where $\mathbf{z} = \mathbf{x} \setminus \{x_i\}$. This is a difference of two values of the generalized Hough transform (Lemma 4) for the configuration with x_i deleted.

Reconstructions obtained by add-delete-shift algorithms are shown in Figures 3–4.

5 Bayesian approach

5.1 General

In the Bayesian approach to image reconstruction [5, 9, 13, 18], the true image $\mathbf{x}$ is assumed to have been generated by a prior probability distribution with density $p(\mathbf{x})$. Then the posterior distribution for $\mathbf{x}$ after observing data $\mathbf{y}$ is $p(\mathbf{x}|\mathbf{y}) \propto f(\mathbf{y}|\mathbf{x})p(\mathbf{x})$ and the *maximum a posteriori* (MAP) estimator of $\mathbf{x}$ is

$$\tilde{\mathbf{x}} = \mathrm{argmax}_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}) = \mathrm{argmax}_{\mathbf{x}} f(\mathbf{y}|\mathbf{x})p(\mathbf{x}). \tag{7}$$

Thus $p(\mathbf{x})$ can also be regarded as a smoothing penalty attached to the optimization of f, and $\tilde{\mathbf{x}}$ as a penalized maximum likelihood estimator. Suitable choices for $p(\mathbf{x})$ will be discussed below.

A strong motivation for Bayesian methods in our context is the experience (e.g. Figures 3 – 4) that maximum likelihood solutions $\hat{\mathbf{x}}$ tend to contain clusters of almost identical objects. This phenomenon is undesirable if the number of objects is important, 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, MLE methods exhibit oversensitivity to the data and to the scanning order in image space.

5.2 Iterative algorithms for MAP

In our context (7) is an optimization over variable-length lists $\mathbf{x}$ of parameter points in the continuous space U. For example, in model 1 with prior $p(\mathbf{x})$, the MAP equations require minimizing

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

for model 4 , MAP requires constrained minimization of

$$|S(\mathbf{x})|\log p - \log p(\mathbf{x})$$

subject to $S(\mathbf{x}) \subseteq Y$. We shall use iterative algorithms similar to those in section 4.

Algorithm 3 (ICM in configuration space) *Apply Algorithms 1, 2, 3 or 4 with $f(\mathbf{y}|\mathbf{x})$ replaced by the posterior probability $p(\mathbf{x}|\mathbf{y})$. Thus we iteratively*

- *add object u to list $\mathbf{x}$ iff* $\log f(\mathbf{y} \mid \mathbf{x}\cup\{u\})\ p(\mathbf{x}\cup\{u\}) - \log f(\mathbf{y} \mid \mathbf{x})\ p(\mathbf{x}) > w$;
- *delete existing object x_i iff* $\log f(\mathbf{y} \mid \mathbf{x}\setminus\{x_i\})\ p(\mathbf{x}\setminus\{x_i\}) - \log f(\mathbf{y} \mid \mathbf{x})\ p(\mathbf{x}) > w$;
- *if permitted, shift $x_i \in \mathbf{x}$ to u iff $u \in Q(\mathbf{x}, x_i)$ and* $\log f(\mathbf{y} \mid M(\mathbf{x}, x_i, u)) - \log f(\mathbf{y} \mid \mathbf{x}) > w$.

Similar statements about convergence hold 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 completely analogous to Besag's ICM algorithm [5].

5.3 Prior model

The appropriate analogues of Markov random fields are nearest-neighbour Markov random sets [2], generalizations of the Markov point processes of Ripley and Kelly [20]. Their essential property, that replaces the local interaction property of Markov random fields [5, 9], is that $p(\mathbf{x}\cup\{u\})/p(\mathbf{x})$ depends only on local information.

For brevity we discuss only one prior model, the **Strauss overlapping object process**. This is a generalization of the Strauss point process [2, 7, 20, 26], with density

$$p(\mathbf{x}) = \alpha\beta^{n(\mathbf{x})}\gamma^{r(\mathbf{x})} \tag{8}$$

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

If object space U is discrete, then $p(\mathbf{x})$ is simply the probability of configuration x. In general p is a density with respect to the Poisson process on U of unit rate.

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 has a spatial Markov property

$$\log\frac{p(\mathbf{x}\cup\{u\})}{p(\mathbf{x})} = \log\beta + r(\mathbf{x}, u)\log\gamma \tag{9}$$

where $r(\mathbf{x}, u) = r(\mathbf{x}\cup\{u\}) - r(\mathbf{x})$ is the number of $x_i \in \mathbf{x}$ such that $R(x_i) \cap R(u) \neq \emptyset$. This depends only on the added object u and on those existing objects x_i that overlap it.

5.4 Relation to Hough transform

If the Strauss model (8) is used as the prior, its parameter γ controls the tradeoff between goodness-of-fit to the data and 'complexity' of the solution $\mathbf{x}$. 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 5 *For any blur-free independent noise model (Definition 1) with $g(\cdot|\cdot) > 0$, and the Strauss process prior (8) with $\gamma > 0$, 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:*

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

Taking $\mathbf{x} = \emptyset$, this shows that thresholding the Hough transform of $\mathbf{y}$ 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).

To recognise non-overlapping objects, for instance characters in text, a Strauss prior with interaction parameter $\gamma = 0$ ('hard core model') could be used. Then the log likelihood ratio is $\log\beta + \sum_{t \in R(u)} h(y_t, \theta_0, \theta_1)$ and a new object will be added iff the Hough transform exceeds some predetermined level and the candidate object does not overlap any existing one.

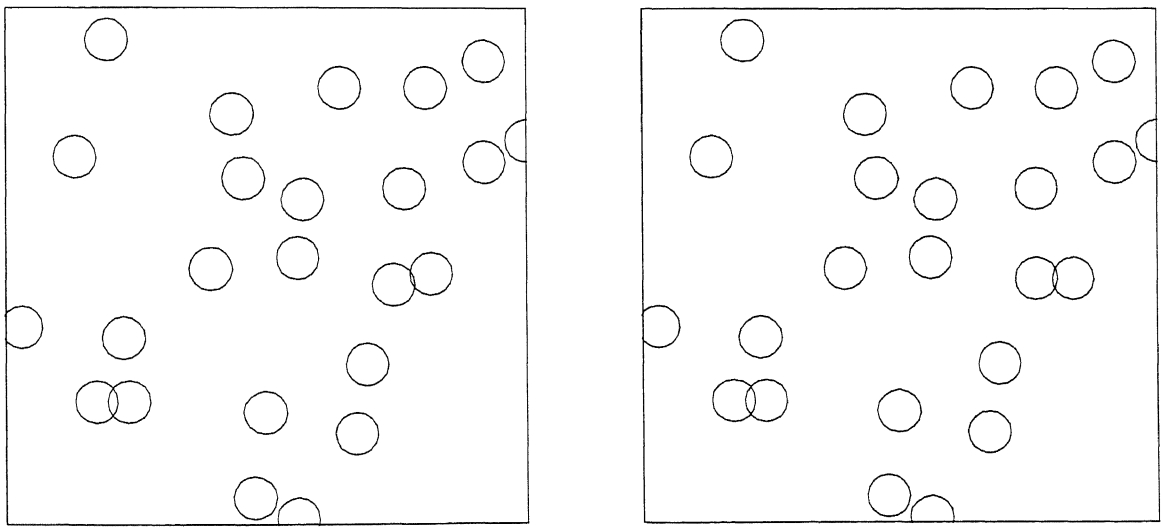

Figure 5: MAP reconstructions using coordinatewise ascent with the local extrema of the Hough transform as initial state. Left: only births and deaths; right: births, deaths and translations.

Algorithm 3 is illustrated in Figures 5 and 6 using coordinatewise optimization and steepest ascent respectively. The initial configuration was the set of local extrema of the Hough transform. A Strauss prior model with $\beta = .0025$ and $\gamma = .25$ was used. The steepest ascent version introduces fewer spurious discs.

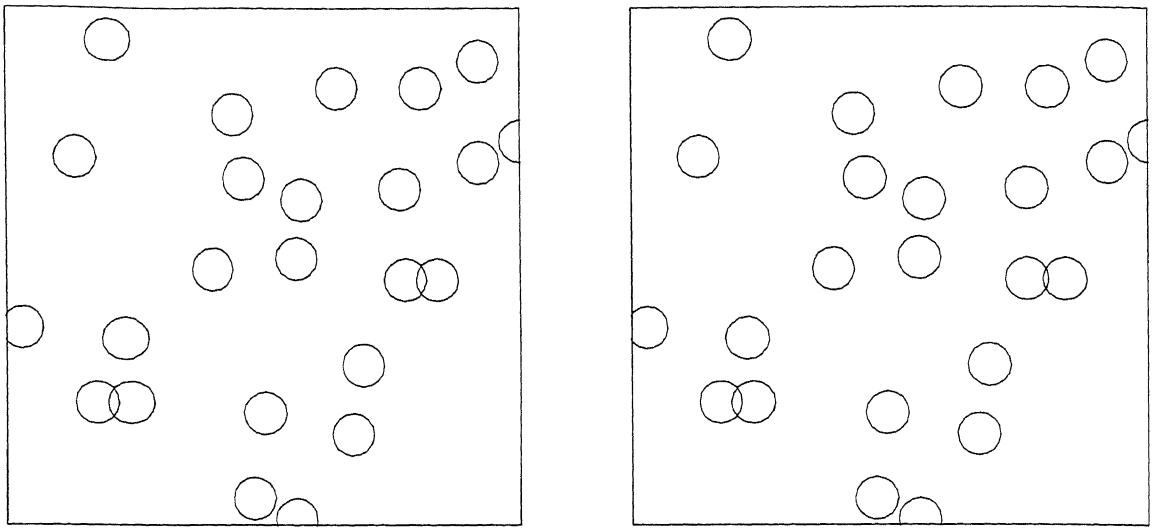

Figure 6: MAP reconstructions using steepest ascent with the empty list as initial state. Left: only births and deaths; right: births, deaths and translations.

6 Performance

Algorithms 1 – 3 all have a recursive structure in which the current reconstruction $\mathbf{x}$ determines the conditional Hough transform (5) which is then optimized to determine how $\mathbf{x}$ shall be updated. After $\mathbf{x}$ is updated the corresponding update of the conditional Hough transform is 'local', restricted to $R(u)$ where u is the object just added or deleted.

We tested Algorithms 1 – 3 on the simulated data of Figure 2 and measured performance using the log likelihood $L(\mathbf{x}^{(k)}\ ;\ \mathbf{y})$ itself and Pratt's figure of merit [1].

Figure 7 graphs the performance of coordinatewise optimization iteration-by-iteration, starting with the local extrema of the Hough transform. MAP is superior to the MLE, both subjectively and in terms of the figure of merit. Obviously the graph of log likelihood for MAP need not (should not) be monotone.

Steepest ascent behaves differently; see Figure 8. Steepest ascent from an empty initial image requires at least as many scans as there are objects in the image. New objects are added one-by-one, gradually improving the reconstruction quality, until all objects are detected; then the reconstructions deteriorate. This method however can yield more accurate reconstructions than coordinatewise optimization algorithms, especially in the non-Bayesian case.

The popular technique of finding local extrema of the Hough transform performed relatively badly, as can be seen from the graphs in Figure 7 where the y intercept is the performance of the Hough extrema operator.

The add-delete-shift algorithms are clearly superior to add-delete algorithms (Figure 8) and seem less sensitive to the initial state. Another advantage is that the number of scans needed for convergence (in steepest ascent) decreases.

Sensitivity to noise was studied by simulating model 1 for several values of σ^2. Reconstructions were obtained and the average quality calculated for Algorithms 1–3. The results are depicted in Figure 9. The MAP solutions are less sensitive to the noise variance than ML estimates. Steepest ascent is less sensitive than coordinatewise optimization.

7 Discussion

The results so far are mainly of theoretical interest, in that they unite various ad hoc techniques for object recognition under a single statistical framework. Although it might appear that applications are restricted to video images, the theory applies equally to other imaging modalities such as tomography and stereo vision. In real applications the 'true model' (object shapes, noise parameters) will be unknown; parameter estimation must be carried out simultaneously with recognition of objects. Maximum pseudolikelihood procedures suggested by Besag [5] can be adapted for use here.

Acknowledgements

The authors thank Adri Steenbeek for technical assistance, and R.D. Gill, F.C.A. Groen and J. Møller for helpful feedback. Algorithms were implemented in C++ using the image processing package `scilaim` [27].

References

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

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

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

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

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

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

[7] P. J. Diggle. *Statistical analysis of spatial point patterns.* Academic Press, London, 1983.

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

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

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

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

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

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

[14] P.V.C. Hough. Method and means for recognizing complex patterns. US Patent 3069654, 1962.

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

[16] J.-H. Lin, T. M. Sellke, and E. J. Coyle. Adaptive stack filtering under the mean absolute error criterion. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:938–954, 1990.

[17] P. Maragos. Optimal morphological approaches to image matching and object detection. In *Proceedings of the IEEE International Conference on Computer Vision 1988, Tampa, Florida*, pages 695 - 699. 1988.

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

[19] B. D. Ripley. *Statistical inference for spatial processes.* Cambridge University Press, 1988.

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

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

[22] J. Serra. *Image analysis and mathematical morphology.* Academic Press, London, 1982.

[23] J. Serra, editor. *Image analysis and mathematical morphology, volume 2: Theoretical advances.* Academic Press, London, 1988.

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

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

[26] D. Stoyan, W. S. Kendall, and J. Mecke. *Stochastic Geometry and its Applications.* John Wiley and Sons, Chichester, 1987.

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

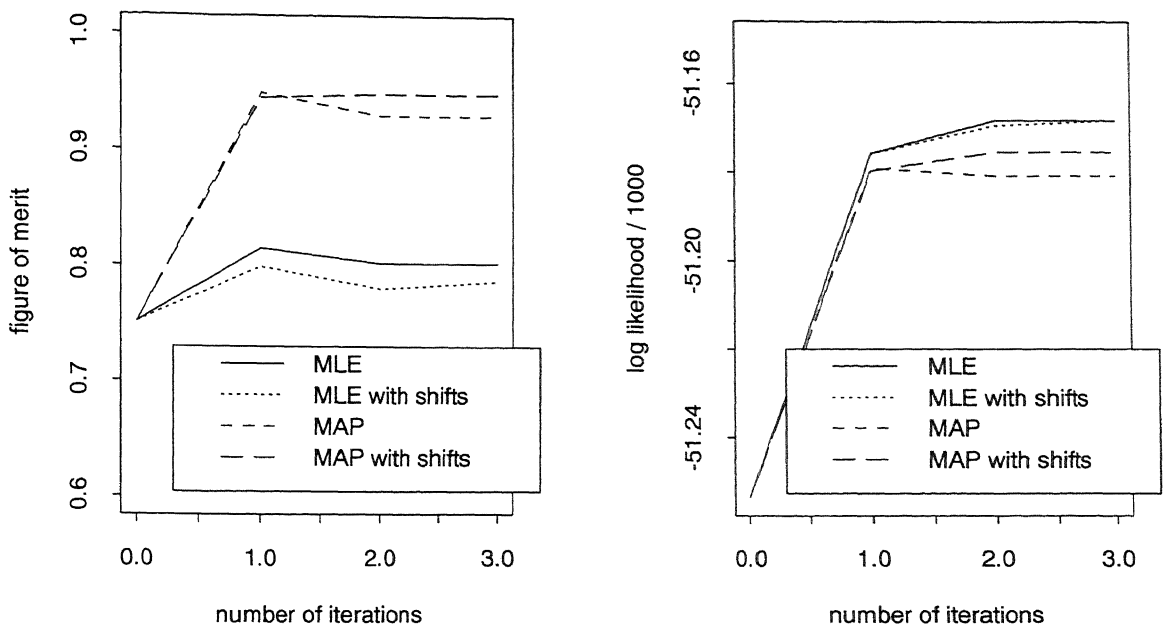

Figure 7: Reconstruction quality at successive steps of coordinatewise ascent starting with the local maxima of the Hough transform.

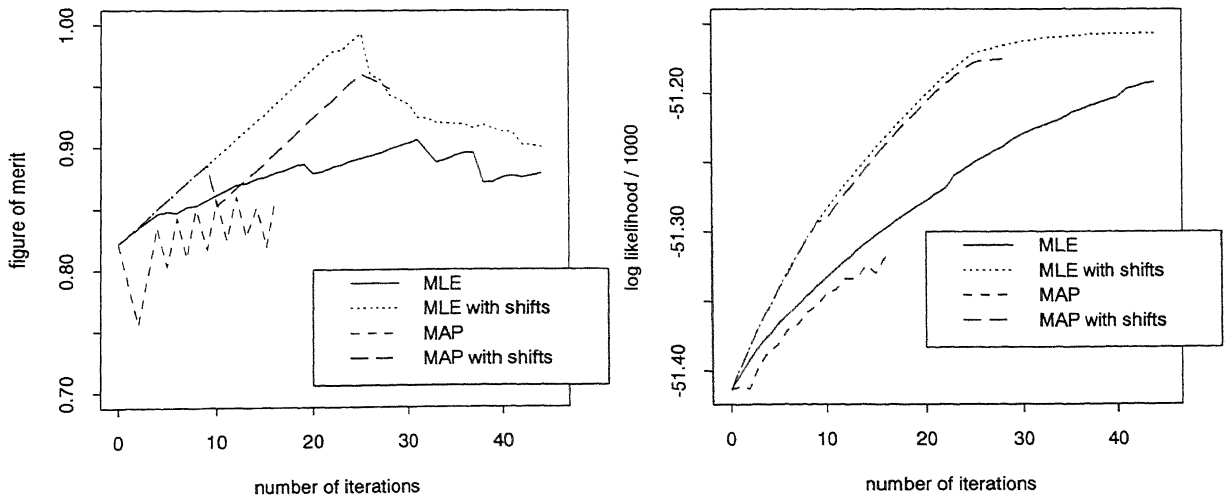

Figure 8: Reconstruction quality for the successive reconstructions obtained using steepest ascent starting with a diagonally translated copy of the true configuration.

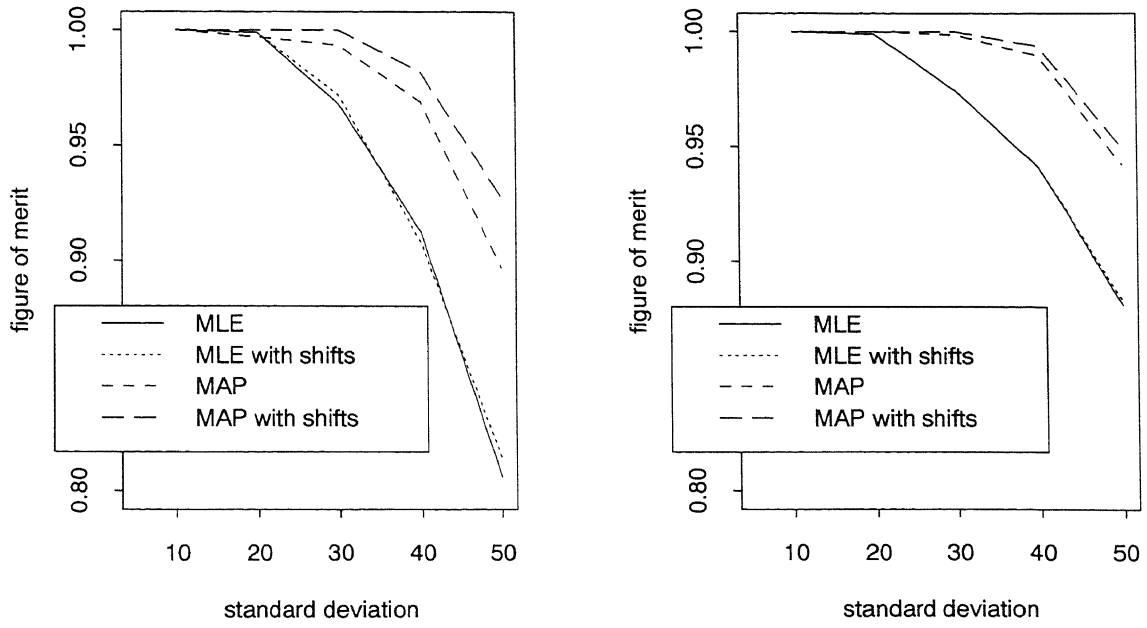

Figure 9: Average reconstruction quality for 10 independent realizations at different noise levels. Left: coordinatewise ascent; right: steepest ascent.