ELSEVIER

Image and Vision Computing 20 (2002) 725–738

image AND vision COMPUTING

www.elsevier.com/locate/imavis

An expectation–maximisation framework for segmentation and grouping

Antonio Robles-Kelly*, Edwin R. Hancock

Department of Computer Science, University of York, York Y01 5DD, UK

Received 10 June 2001; accepted 7 February 2002

Abstract

This paper casts the problem of perceptual grouping into an evidence combining setting using the apparatus of the EM algorithm. We are concerned with recovering a perceptual arrangement graph for line-segments using evidence provided by a raw perceptual grouping field. The perceptual grouping process is posed as one of pairwise relational clustering. The task is to assign line-segments (or other image tokens) to clusters in which there is strong relational affinity between token pairs. The parameters of our model are the cluster memberships and the pairwise affinities or link-weights for the nodes of a perceptual relation graph. Commencing from a simple probability distribution for these parameters, we show how they may be estimated using the apparatus of the EM algorithm. The new method is demonstrated on line-segment grouping problems where it is shown to outperform a non-iterative eigenclustering method.

Keywords: Perceptual grouping; Pairwise clustering; EM algorithm; Image segmentation

1. Introduction

Perceptual grouping is an important process which permits low-level features to be organised into higher level relational structures that can be subsequently used for scene understanding and object recognition. Broadly speaking, the available literature on perceptual grouping can be divided into three areas according to the level of abstraction at which they operate. At the lowest level of abstraction, the available algorithms are concerned with computation of the grouping field. There are several contributions which deserve special mention. Heitger and von der Heydt [6] have shown how to model the line extension field using directional filters whose shapes are motivated by studies of the visual field of monkeys. Williams et al. [25,26] have taken a different approach using the stochastic completion field. Here the completion field of curvilinear features is computed using Monte Carlo simulation of particle trajectories between the end-points of contours. At the intermediate level of abstraction, several authors have investigated the use of iterative relaxation style operators for edgel grouping. This approach was pioneered by Shashua and Ullman [21] and later refined by Guy and Medioni [5] among others. Parent and Zucker have shown how co-circularity can be used to gauge the compatibility of neighbouring edges [15]. Matas and Kittler [10] have shown how Waltz's dictionary of line-label configurations can be used for junction re-enforcement. At the highest level of abstraction, the available algorithms pose the grouping problem as that of recovering a graph which represents the relational arrangement of segmental entities previously extracted from raw image data. One of the most popular methods here is to use ideas from spectral graph theory to locate the salient relational structure. For instance, both Sarkar and Boyer [20] and Perona and Freeman [17] have used the eigenstructure of a perceptual affinity matrix to find disjoint subgraphs that represent the main arrangements of segmental entities. Finally, it is worth mentioning that several authors have used similar algorithms based on eigenvalues of an affinity matrix to iteratively segment image data. One of the best known is the normalised cut method of Shi and Malik [22]. Recently, Weiss [24] has shown how this and other closely related methods can be improved using a normalised affinity matrix.

The observation underpinning this paper is that although considerable effort has been expended at intermediate level to develop algorithms for combining evidence from a raw grouping field, the higher-level graph-based methods use static affinity relationships or relational abstraction as input.

* Corresponding author.
E-mail addresses: arobkell@cs.york.ac.uk (A. Robles-Kelly), erh@cs.york.ac.uk (E.R. Hancock).

0262-8856/02/$ - see front matter
PII: S0262-8856(02)00062-8

The aim in this paper is to develop a different approach to the problem which poses the recovery of the perceptual arrangement graph in an evidence combining framework. We pose the problem as one of pairwise clustering which is parameterised using two sets of indicator variables. The first of these are cluster membership variables which indicate to which perceptual cluster a segmental entity belongs. The second set of variables is link-weights which convey the strength of the perceptual relations between pairs of nodes in the same cluster. Our contribution is to show how to estimate both sets of indicator variables using the apparatus of the EM algorithm.

Although there have been some attempts at using probabilistic methods for grouping elsewhere in the literature [19], our method has a number of unique features which distinguish it from these alternatives. Early work by Dickson [4] has used Bayes nets to develop a hierarchical framework for splitting and merging groups of lines. Cox, Rehg and Hingorani [8] have developed a grouping method which combines evidence from the raw edge attributes delivered by the Canny edge detector. Leite and Hancock [12] have pursued similar objectives with the aim of fitting cubic splines to the output of a bank of multiscale derivative of Gaussian filters using the EM algorithm. Castaño and Hutchinson [19] have developed a Bayesian framework for combining evidence for different graph-based partitions or groupings of line-segments. The method exploits bilateral symmetries. It is based on a frequentist approach over the set of partitions of the line-segments and is hence free of parameters. Recently, Crevier [3] has developed an evidence combining framework for extracting chains of colinear line-segments. Our work differs from this work in a number of important ways. We use a probabilistic characterisation of the grouping-graph based on a matrix of link-weights. The goal of computation is to iteratively recover the maximum likelihood elements of this matrix using the apparatus of the EM algorithm.

The outline of this paper is as follows. In Section 2 we develop a mixture model for the grouping problem. Section 3 shows how the parameters of this mixture model, namely the cluster membership probabilities and the pairwise link-weights can be estimated using the EM algorithm. We also describe how the number of components of the mixture model can be set using the eigenmodes of the link-weight matrix. We then provide two applications of the grouping algorithm. In Section 4 we address the problem of line-segment grouping. To commence, we describe a simple model which can be used to initialise the link-weights. We then evaluate the line-grouping method in a sensitivity study on synthetic data. We also furnish some examples on real-world images. Section 5 presents a similar study for the problem of grey-scale image segmentation. Finally, Section 6 concludes the paper by summarising our contributions and offering directions for future research.

2. Maximum likelihood framework

We pose the problem of perceptual grouping as that of finding the pairwise clusters which exist within a set of objects segmented from raw image data. These objects may be point-features such as corners, lines, curves or regions. However, in this paper we focus on the specific problems of grey-level image segmentation and grouping line-segments. The process of pairwise clustering is somewhat different to the more familiar one of central clustering. Whereas central clustering aims to characterise cluster-membership using the cluster mean and variance, in pairwise clustering it is link-weights between nodes which are used to establish cluster membership. Although less well studied than central clustering, there has recently been renewed interest in pairwise clustering aimed at placing the method on a more principled footing using techniques such as mean-field annealing [7].

We abstract the problem in the following way. The raw perceptual entities are indexed using the set V. Our aim is to assign each node to one of a set of pairwise clusters which are indexed by the set Ω. To represent the state of organisation of the perceptual relation graph, we introduce some indicator variables. First, we introduce a cluster membership indicator which is unity if the node i belongs to the perceptual cluster $\omega \in \Omega$ and is zero otherwise, i.e.

$$s_{i\omega} = \begin{cases} 1 & \text{if } i \in \omega \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

The second model ingredient is the link-weight $A_{i,j}$ between distinct pairs of nodes $(i,j) \in V \times V - \{(i,i)|i \in V\}$. When the link-weights become binary in nature, they convey the following meaning

$$A_{i,j} = \begin{cases} 1 & \text{if } i \in \omega \text{ and } j \in \omega \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

When the link-weights satisfy the above condition, then the different clusters represent disjoint subgraphs.

Our aim is to find the cluster membership variables and the link-weights which partition the set of raw perceptual entities into disjoint pairwise clusters. We commence by assuming that there are putative edges between each distinct pair of nodes (i,j) belonging to the Cartesian self-product $\Phi = V \times V - \{(i,i)|i \in V\}$. Further suppose that $p(A_{i,j})$ is the probability density for the link-weight appearing on the pair of nodes $(i,j) \in \Phi$. Our aim is to locate disjoint subgraphs by updating the link-weights until they are either zero or unity. Under the assumption that the link-weights on different pairs of nodes are independent of one-another, then the likelihood function for the observed arrangement of perceptual entities can be factorised over the set of putative edges as

$$P(A) = \prod_{(i,j)\in\Phi} P(A_{i,j}). \tag{3}$$

We are interested in partitioning the set of perceptual entities into pairwise clusters using the link-weights between them. We must therefore entertain the possibility that each of the Cartesian pairs appearing under the above product, which represent putative perceptual relations, may belong to each of the pairwise clusters indexed by the set Ω. To make this uncertainty of association explicit, we construct a mixture model over the perceptual clusters and write

$$P(A_{i,j}) = \sum_{\omega\in\Omega} P(A_{i,j}|\omega)P(\omega). \tag{4}$$

According to this mixture model, $P(A_{i,j}|\omega)$ is the probability that the nodes i and j are connected by an edge with link-weight $A_{i,j}$ which falls within the perceptual cluster indexed ω. The total probability mass associated with the cluster indexed ω is $P(\omega)$. In most of our experiments, we will assume that there are only two such sets of nodes; those that represent a foreground arrangement, and those that represent background clutter. However, for generality we proceed under the assumption that there are an arbitrary number of perceptual clusters. As a result, the probability of the observed set of perceptual entities is

$$P(A) = \prod_{(i,j)\in\Phi} \sum_{\omega\in\Omega} P(A_{i,j}|\omega)P(\omega). \tag{5}$$

To proceed, we require a model of the probability distribution for the link-weights. Here we adopt a model in which the observed link structure of the pairwise clusters arises through a Bernoulli distribution. The parameter of this distribution is the link-probability $A_{i,j}$. The idea behind this model is that any pair of nodes i and j may connect to each with a link. This link is treated as a Bernoulli variable. The probability that this link is correct is $A_{i,j}$ while the probability that it is in error is $1 - A_{i,j}$. To gauge the correctness of the link, we check whether the nodes i and j belong to the same pairwise cluster. To test for cluster-consistency we make use of the quantity $s_{i\omega}s_{j\omega}$. This is unity if both nodes belong to the same cluster and is zero otherwise. Using this switching property, the Bernoulli distribution becomes

$$p(A_{i,j}|\omega) = A_{i,j}^{s_{i\omega}s_{j\omega}}(1 - A_{i,j})^{1-s_{i\omega}s_{j\omega}}. \tag{6}$$

This distribution takes on is largest values when either the link-weight $A_{i,j}$ is unity and $s_{i\omega} = s_{j\omega} = 1$, or if the link-weight $A_{i,j} = 0$ and $s_{i\omega} = s_{j\omega} = 0$.

3. Expectation–maximisation

Our aim is to find the cluster-membership weights and the link-weights which maximize the likelihood function appearing in Eq. (5). One way to locate the maximum likelihood perceptual relation graph is to update the binary cluster and edge indicators. This could be effected using a number of optimisation methods including simulated annealing and Markov Chain Monte Carlo. However, here we use the apparatus of the EM algorithm originally developed by Dempster, Laird and Rubin [1]. Our reason for doing this is that the cluster-membership variables $s_{i\omega}$ must be regarded as quantities whose distribution is governed by the link-weights $A_{i,j}$. Since at the outset we know neither the associations between nodes and clusters nor the strength of the link-weights within clusters, this information must be treated as hidden data. In other words, we must use the EM algorithm to estimate them.

The idea underpinning the EM algorithm is to recover maximum likelihood solutions to problems involving missing or hidden data by iterating between two computational steps. In the E (or expectation) step we estimate the a posteriori probabilities of the hidden data using maximum likelihood parameters recovered in the preceding maximisation (M) step. The M-step in-turn aims to recover the parameters which maximise the expected value of the log-likelihood function. It is the available a posteriori probabilities from the E-step which allows the weighting of log-likelihood required in the maximisation-step.

3.1. Expected log-likelihood function

For the likelihood function appearing in Eq. (5), the expected log-likelihood function is defined as

$$Q\left(A^{(n+1)}|A^{(n)}\right) = \sum_{\omega\in\Omega}\sum_{(i,j)\in\Phi} P\left(w\middle|A_{i,j}^{(n)}\right)\ln p\left(A_{i,j}^{(n+1)}\middle|\omega\right) \tag{7}$$

where $p(A_{i,j}^{(n+1)}|\omega)$ is the probability distribution for the link-weights at iteration $n+1$ and $P(w|A_{i,j}^{(n)})$ is the a posteriori probability that the pair of nodes with link-weight $A_{i,j}^{(n)}$ belong to the cluster indexed ω at iteration n of the algorithm. When the probability distribution function from Eq. (6) is substituted, then the expected log-likelihood function becomes

$$\begin{aligned} Q(A^{(n+1)}|A^{(n)}) = \sum_{\omega\in\Omega}\sum_{(i,j)\in\Phi} \zeta_{i,j,\omega}^{(n)}\Big\{ & s_{i\omega}^{(n+1)}s_{j\omega}^{(n+1)}\ln A_{i,j}^{(n+1)} \\ & +\left(1 - s_{i\omega}^{(n+1)}s_{j\omega}^{(n+1)}\right)\ln\left(1 - A_{i,j}^{(n+1)}\right)\Big\} \end{aligned} \tag{8}$$

where we have used the shorthand $\zeta_{i,j,\omega}^{(n)} = P(w|A_{i,j}^{(n)})$ for the a posteriori cluster membership probabilities. After some algebra to collect terms, the expected log-likelihood function simplifies to

$$\begin{aligned} Q(A^{(n+1)}|A^{(n)}) = \sum_{\omega\in\Omega}\sum_{(i,j)\in\Phi} \zeta_{i,j,\omega}^{(n)}\Big\{ & s_{i\omega}^{(n+1)}s_{j\omega}^{(n+1)}\ln\frac{A_{i,j}^{(n+1)}}{1 - A_{i,j}^{(n+1)}} \\ & + \ln\left(1 - A_{i,j}^{(n+1)}\right)\Big\}. \end{aligned} \tag{9}$$

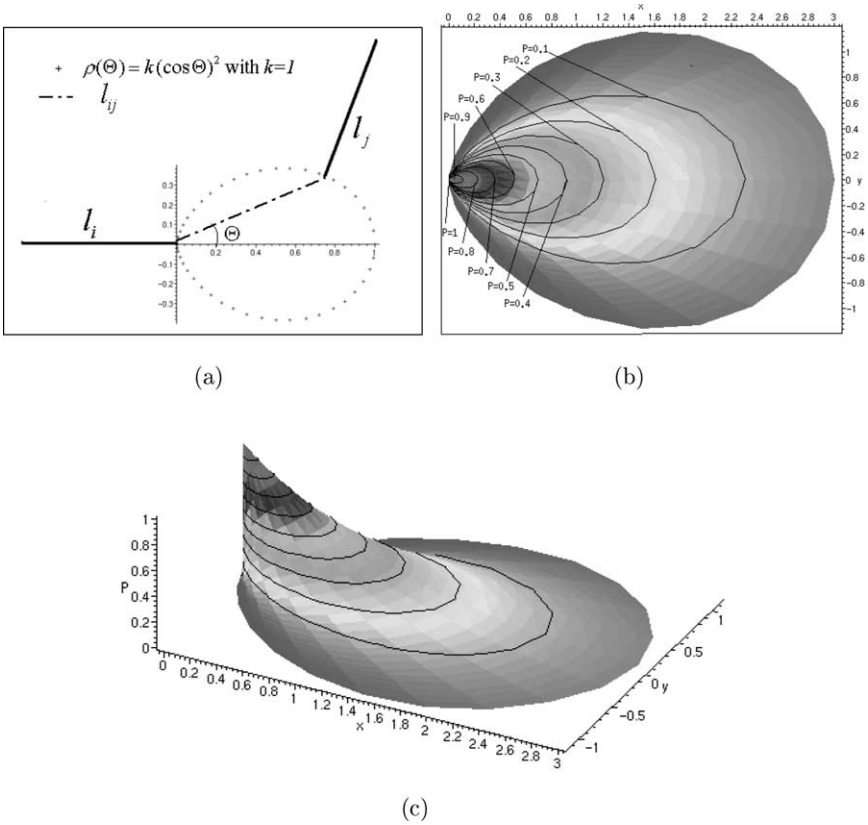

Fig. 1. (a) Geometric meaning of the parameters used to obtain $P_{i,j}$. (b) Plot showing the level curves. (c) 3D plot showing $A_{i,j}$ on the z axis.

3.2. Maximisation

In the maximisation step of the algorithm, we aim to recover the cluster and edge parameters $s_{i\omega}$ and $A_{i,j}$. The edge parameters are found by computing the derivatives of the expected log-likelihood function

$$\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial A_{i,j}^{(n+1)}} = \sum_{\omega\in\Omega}\zeta_{i,j,\omega}^{(n)}\left\{s_{i\omega}^{(n+1)}s_{j\omega}^{(n+1)}\frac{1}{A_{i,j}^{(n+1)}}\left(1-A_{i,j}^{(n+1)}\right)-\frac{1}{1-A_{i,j}^{(n+1)}}\right\} \tag{10}$$

and solving the equations

$$\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial A_{i,j}^{(n+1)}}=0. \tag{11}$$

As a result, the updated link-weights are given by

$$A_{i,j}^{(n+1)}=\frac{\sum_{\omega\in\Omega}\zeta_{i,j,\omega}^{(n)}s_{i\omega}^{(n+1)}s_{j\omega}^{(n+1)}}{\sum_{\omega\in\Omega}\zeta_{i,j,\omega}^{(n)}} \tag{12}$$

where $i\in V$, $j\in V$ and $i\neq j$. In other words, the link-weight for the pair of nodes (i,j) is simply the average of the product of individual node cluster memberships over the different perceptual clusters.

To compute the cluster membership variables, we again compute the derivative of the expected log-likelihood function

$$\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial s_{i\omega}^{(n+1)}}=\sum_{j\in V}\zeta_{i,j,\omega}^{(n)}s_{j\omega}^{(n+1)}\ln\frac{A_{i,j}^{(n+1)}}{1-A_{i,j}^{(n+1)}}. \tag{13}$$

However, the associated equations are not tractable in closed form. Instead, we use the soft-assign ansatz of Bridle [2] to update the cluster membership assignment variables. This involves exponentiating the partial derivatives of the expected log-likelihood function in the following manner

$$s_{i\omega}^{(n+1)}=\frac{\exp\left[\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial s_{i\omega}^{(n+1)}}\right]}{\sum_{\omega\in\Omega}\exp\left[\frac{\partial Q(A^{(n+1)}|A^{(n)})}{\partial s_{i\omega}^{(n+1)}}\right]}. \tag{14}$$

As a result, the update equation for the cluster

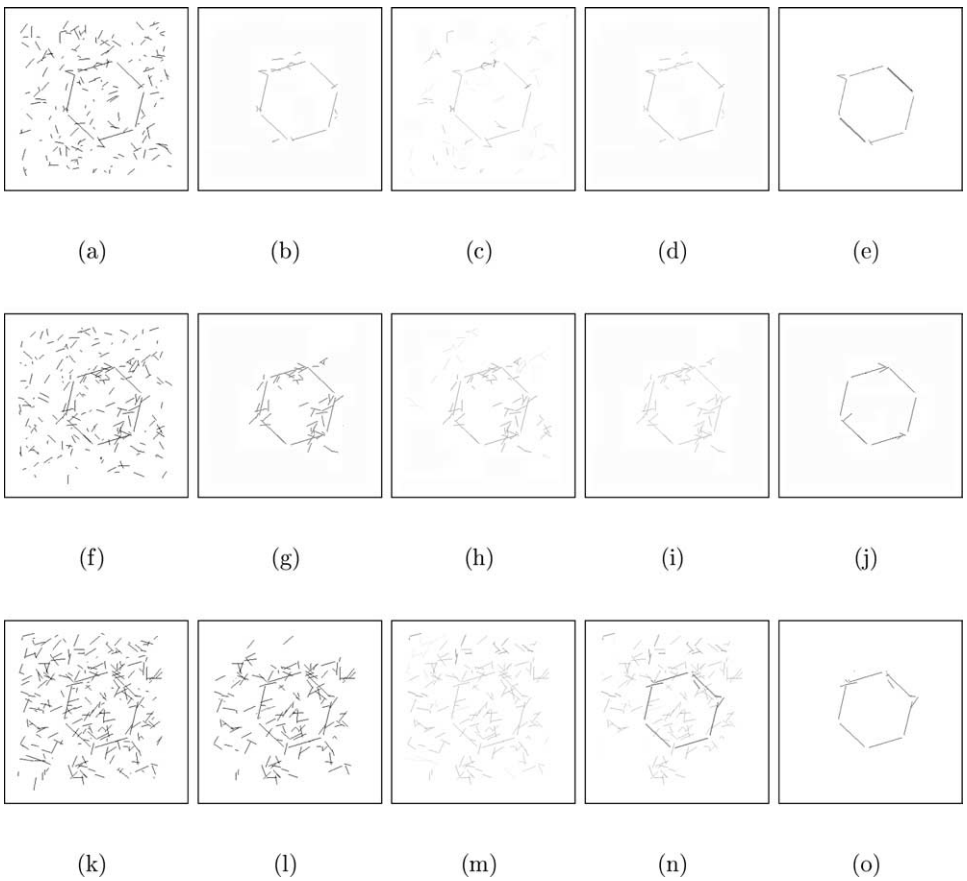

Fig. 2. Results of grouping for a sequence of noisy synthetic images.

membership indicator variables is

$$s_{i\omega}^{(n+1)} = \frac{\exp\left[\sum_{j\in V} \zeta_{i,j,\omega}^{(n)} s_{j\omega}^{(n)} \ln \frac{A_{i,j}^{(n+1)}}{1-A_{i,j}^{(n+1)}}\right]}{\sum_{\omega\in\Omega} \exp\left[\sum_{j\in V} \zeta_{i,j,\omega}^{(n)} s_{j\omega}^{(n)} \ln \frac{A_{i,j}^{(n+1)}}{1-A_{i,j}^{(n+1)}}\right]} = \frac{\prod_{j\in V}\left\{\frac{A_{i,j}^{(n+1)}}{1-A_{i,j}^{(n+1)}}\right\}^{\zeta_{i,j,\omega}^{(n)} s_{j\omega}^{(n)}}}{\sum_{\omega\in\Omega}\prod_{j\in V}\left\{\frac{A_{i,j}^{(n+1)}}{1-A_{i,j}^{(n+1)}}\right\}^{\zeta_{i,j,\omega}^{(n)} s_{j\omega}^{(n)}}} \tag{15}$$

where $i \in V$ and $\omega \in \Omega$.

3.3. Expectation

The a posteriori probabilities are updated in the expectation step of the algorithm. The current estimates of the parameters $s_{i\omega}^{(n)}$ and $A_{i,j}^{(n)}$ are used to compute the probability densities $p(A_{i,j}^{(n)}|\omega)$ and the a posteriori probabilities are updated using the formula

$$P\left(\omega\middle|A_{i,j}^{(n)}\right) = \frac{p\left(A_{i,j}^{(n)}\middle|\omega\right)\alpha^{(n)}(\omega)}{\sum_{\omega\in\Omega} p\left(A_{i,j}^{(n)}\middle|\omega\right)\alpha^{(n)}(\omega)} \tag{16}$$

where $\alpha^{(n)}(\omega)$ is the available estimate of the class-prior $P(\omega)$. This is computed using the formula

$$\alpha^{(n)}(\omega) = \frac{1}{|V|^2}\sum_{(i,j)\in\Phi} P(\omega|A_{i,j}^{(n)}). \tag{17}$$

Upon substituting for the probability density from Eq. (6), the updated a posteriori probabilities are given by

$$P(\omega|A_{i,j}^{(n+1)}) = \zeta_{i,j,\omega}^{(n+1)} = \frac{A_{i,j}^{(n)s_{i\omega}^{(n)}s_{j\omega}^{(n)}}\left(1-A_{i,j}^{(n)}\right)^{1-s_{i\omega}^{(n)}s_{j\omega}^{(n)}}\alpha^{(n)}(\omega)}{\sum_{(i,j)\in\Phi} A_{i,j}^{(n)s_{i\omega}^{(n)}s_{j\omega}^{(n)}}\left(1-A_{i,j}^{(n)}\right)^{1-s_{i\omega}^{(n)}s_{j\omega}^{(n)}}\alpha^{(n)}(\omega)}. \tag{18}$$

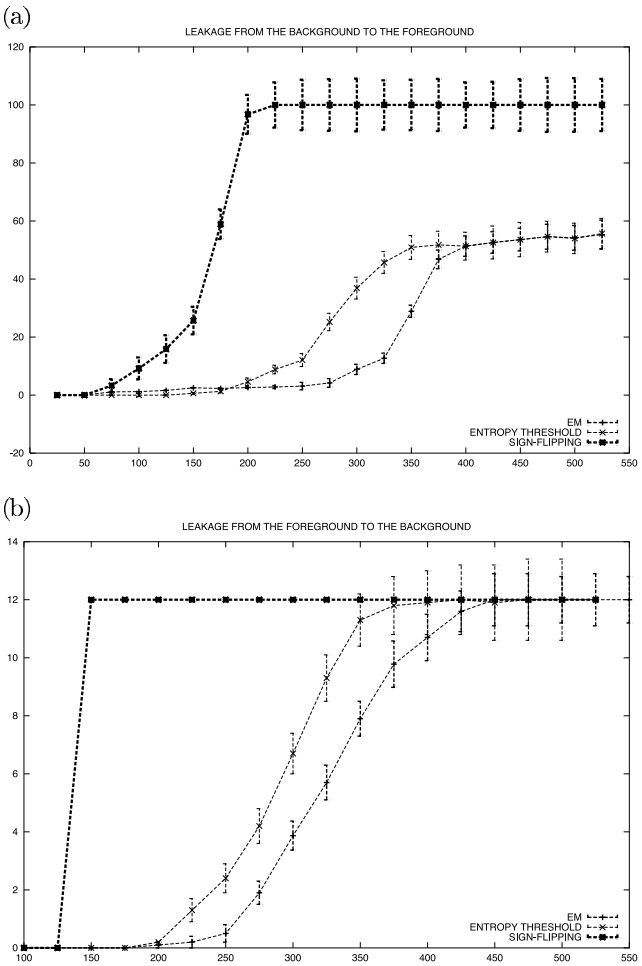

Fig. 3. Comparison between the EM algorithm and eigendecomposition.

3.4. Selecting the number of modes

One of the practical difficulties associated with using a mixture model is that of selecting the number of components. There are several well established ways of addressing this problem. These frequently involve removing or splitting components so as to optimise a measure of model-order complexity [13,14,23]. There are several possible choices of utility measures including the minimum description length [18] and the Akaike information criterion [27].

However, here we take a different route and use the modal structure of the initial affinity matrix to set the number of mixing components. Here we use a result due to Sarkar and Boyer [20] who have shown how matrix factorisation methods can be used to locate the set of edges which partition the nodes into distinct perceptual clusters. One way of viewing this is as the search for the permutation matrix which re-orders the elements of A into non-overlapping blocks. Unfortunately, when the elements of the matrix A are not binary in nature, then this is not a straightforward task. However, Sarkar and Boyer [20] have shown how the positive eigenvectors of the matrix of link-weights can be used to assign nodes to perceptual clusters. Using the Rayleigh–Ritz theorem, they observe that the scalar quantity $\underline{x}^t A^{(0)} \underline{x}$, where $A^{(0)}$ is the initial weighted adjacency matrix, is maximised when $\underline{x}$ is the leading eigenvector of A. Moreover, each of the subdominant eigenvectors corresponds to a disjoint perceptual cluster. They confine their attention to the same-sign positive eigenvectors (i.e. those whose corresponding eigenvalues are real and positive, and whose components are either all positive or are all negative in sign). If a component of a positive eigenvector is non-zero, then the corresponding node belongs to the perceptual cluster associated with the associated eigenmodes of the weighted adjacency matrix. The eigenvalues $\lambda_1, \lambda_2, \ldots$ of $A^{(0)}$ are the solutions of the equation $|A^{(0)} - \lambda I| = 0$ where I is the $|V| \times |V|$ identity matrix. The corresponding eigenvectors $\underline{x}_{\lambda_1}, \underline{x}_{\lambda_2}, \ldots$ are found by solving the equation $A^{(0)} \underline{x}_{\lambda_i} = \lambda_i \underline{x}_{\lambda_i}$. Let the set of positive same-sign eigenvectors be represented by $\Omega = \{\omega | \lambda_\omega > 0 \wedge [(\underline{x}^*_\omega(i) > 0 \forall i) \vee \underline{x}^*_\omega(i) < 0 \forall i])\}$. Since the positive eigenvectors are orthogonal, this means that there is only one value of ω for which $\underline{x}^*_\omega(i) \neq 0$. In other words, each node i is associated with a unique cluster. We denote the set of nodes assigned to the cluster with modal index ω as $V_\omega = \{i | \underline{x}^*_\omega(i) \neq 0\}$. Hence each positive same-sign eigenvector is associated with a distinct mixing component. We use the eigenvectors of the initial affinity matrix to initialise the cluster membership variables. This is done using the magnitudes of the modal coefficients and we set

$$s^{(0)}_{i\omega} = \frac{|\underline{x}^*_\omega(i)|}{\sum_{i \in V_\omega} |\underline{x}^*_\omega(i)|}. \tag{19}$$

We use the eigenvectors of the initial affinity matrix to initialise the cluster membership variables.

At this point it is worth pausing to review the way in which Sarkar and Boyer [20] use the eigenvectors of the adjacency matrix to assign nodes to clusters. In an attempt to render the clustering method robust to noise, they allow up to 5% of the components of the eigenvectors to flip sign. To be more formal let $U^+_\omega = \{i | \lambda_\omega > 0 \wedge \underline{x}^*_\omega(i) > 0\}$ be the set of components of $\underline{x}_\omega$ that are of positive sign and let $U^-_\omega = \{i | \lambda_\omega > 0 \wedge \underline{x}^*_\omega(i) < 0\}$ be the set of components that have negative sign. The set of eigenmodes used by Sarkar and Boyer to represent the set of perceptual clusters is

$$\Omega_{\mathrm{SB}} = \left\{\omega \middle| \frac{\left|U^+_\omega\right|}{|V|} > T \vee \frac{\left|U^-_\omega\right|}{|V|} > T\right\}.$$

In other words the positive eigenvectors are acceptable as modal clusters provided that the fraction of same-sign components exceeds the threshold T. In practice 5% of the components are allowed to flip sign, i.e. $T = 0.95$. One of the problems introduced by this procedure is that by allowing flipping, nodes may be ambiguously assigned to more than one cluster.

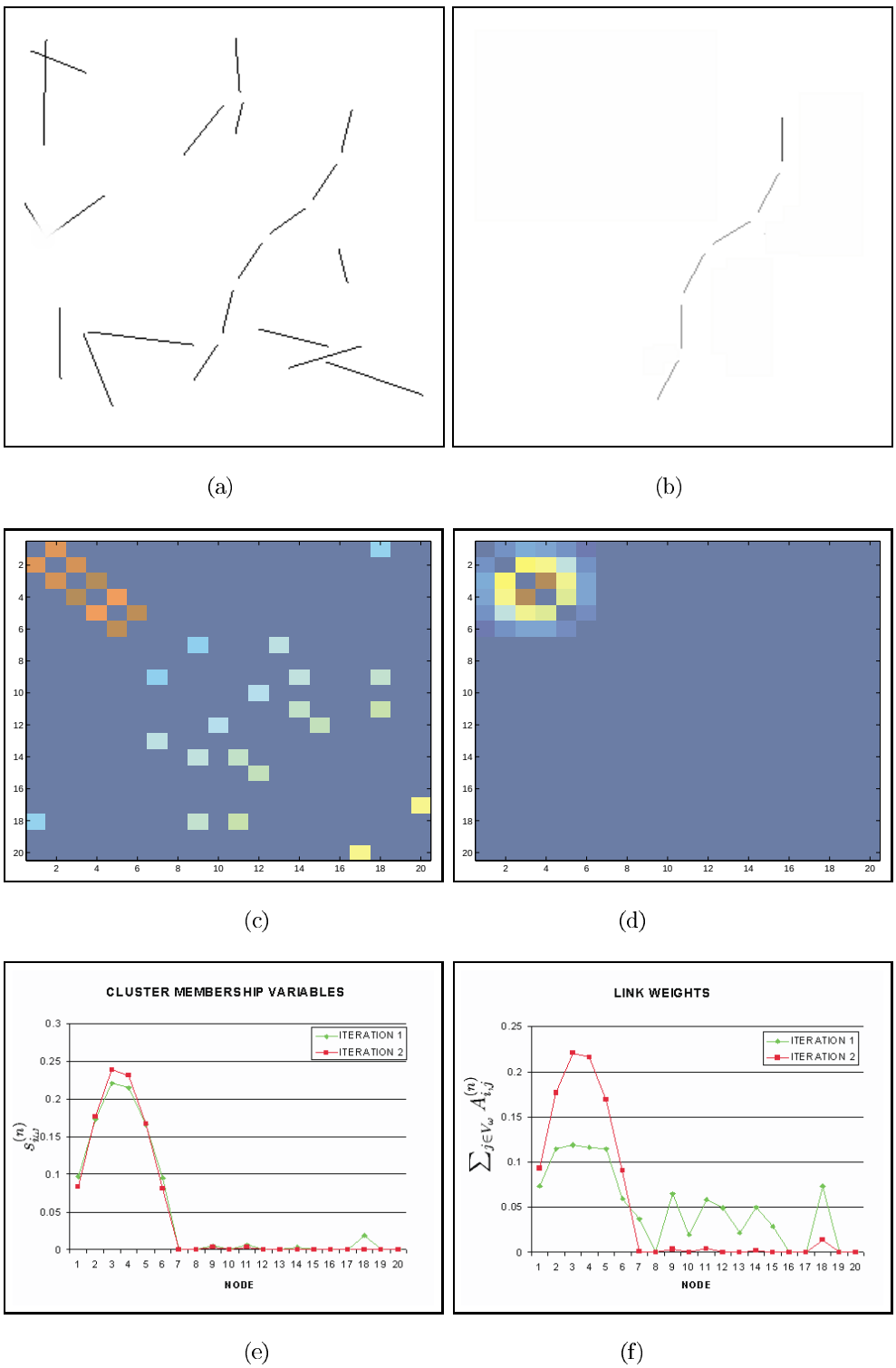

Fig. 4. Input pattern (a); output pattern (b); initial adjacency matrix (c); final adjacency matrix (d); distribution of the cluster membership variables (e); distribution of the link-weight mass (f).

We have recently developed an alternative way of rendering the non-iterative eigendecomposition method robust to noise [11]. This involves thresholding the link-weight matrix. Prior to performing eigendecomposition, we set insignificant elements, i.e. those which are smaller than 0.001, to zero.

In our experiments, we will compare the results delivered by the EM algorithm with those obtained

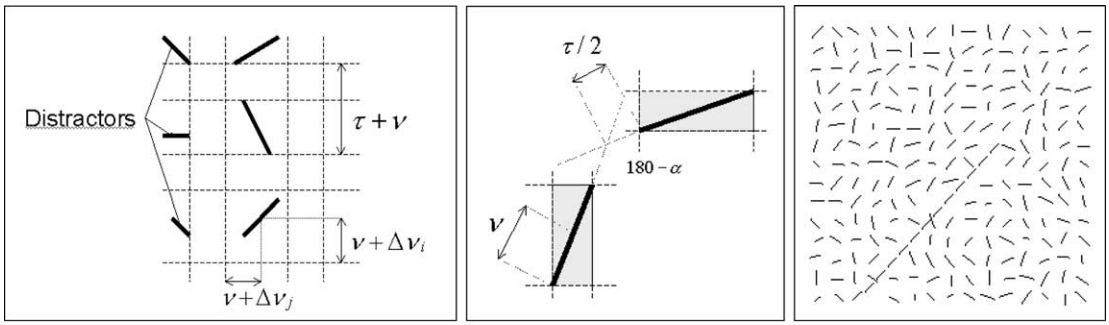

Fig. 5. Geometric meaning of the parameters α, υ and τ in the line pattern generation (a) and in the distractors generation (b); (c) sample test pattern ($\beta = 1.5$ and $\alpha = 0°$).

with both sign-flipping and the thresholded adjacency matrix.

3.5. Algorithm description

Finally, to summarise, the iterative steps of the algorithm are as follows:

- (1) *Initialisation*. Compute the eigenvectors of the initial current link-weight matrix $A^{(0)}$. Each same-sign eigenvector whose eigenvalue is positive is used to seed a different component of the mixture model.
- (2) *Expectation*. Compute the updated cluster-membership variables using the E-step.
- (3) *Maximisation*. Update the link-weights using the M-step to compute the updated link-weight matrix $A^{(n)}$.
- Repeat steps (2) and (3) until convergence is reached.

4. Line grouping

In this section, we provide the first example application of our new clustering method. This involves the grouping or linking of line-segments.

4.1. Initial line-grouping field

We are interested in locating groups of line-segments that exhibit strong geometric affinity to one-another. In this section, we provide details of a probabilistic linking field that can be used to gauge geometric affinity. To be more formal suppose we have a set of line-segments $L = \{\Lambda_i; i = 1, \ldots, n\}$. Consider two lines Λ_i and Λ_j drawn from this set. Their respective lengths are l_i and l_j. Our model of the linking process commences by constructing the line $\Gamma_{i,j}$ which connects the closest pair of endpoints for the two lines. The geometry of this connecting line is represented using its length $\rho_{i,j}$ and the polar angle $\theta_{i,j}$ of the line $\Gamma_{i,j}$ with respect to the base-line Λ_i. We measure the overall scale of the arrangement of lines using the length of the shorter line $\hat{\rho}_{i,j} = \min[l_i, l_j]$.

The relative length of the gap between the two line-segments is represented in a scale-invariant manner using the dimensionless quantity $\xi_{i,j} = \rho_{i,j}/\hat{\rho}_{i,j}$.

Following Heitger and von der Heydt [6] we model the linking process using an elongated polar grouping field. To establish the degree of geometric affinity between the lines we interpolate the end-points of the two lines using the polar lemniscate

$$\xi_{i,j} = k \cos^2 \theta_{i,j}. \tag{20}$$

The value of the constant k is used to measure the degree of affinity between the two lines. For each linking line, we compute the value of the constant k which allows the polar locus to pass through the pair of endpoints. The value of this constant is

$$k = \frac{\rho_{i,j}}{\hat{\rho}_{i,j} \cos^2 \theta_{i,j}}. \tag{21}$$

The geometry of the lines and their relationship to the interpolating polar lemniscate is illustrated in Fig. 1(a). It is important to note that the polar angle is defined over the interval $\theta_{i,j} \in (0, \pi]$ and is rotation invariant.

We use the parameter k to model the linking probability for the pair of line-segments. When the lemniscate envelope is large, i.e. k is large, then the grouping probability is small. On the other hand, when the envelope is compact, then the grouping probability is large. To model this behaviour, we assign the linking probability using the exponential

Table 1
Details of the results shown in Fig. 10

Image	Iterations	Number of clusters	k_g
(a)	1.4	7	3
(c)	1.8	5	3
(e)	4.5	16	3
(g)	4.8	7	3

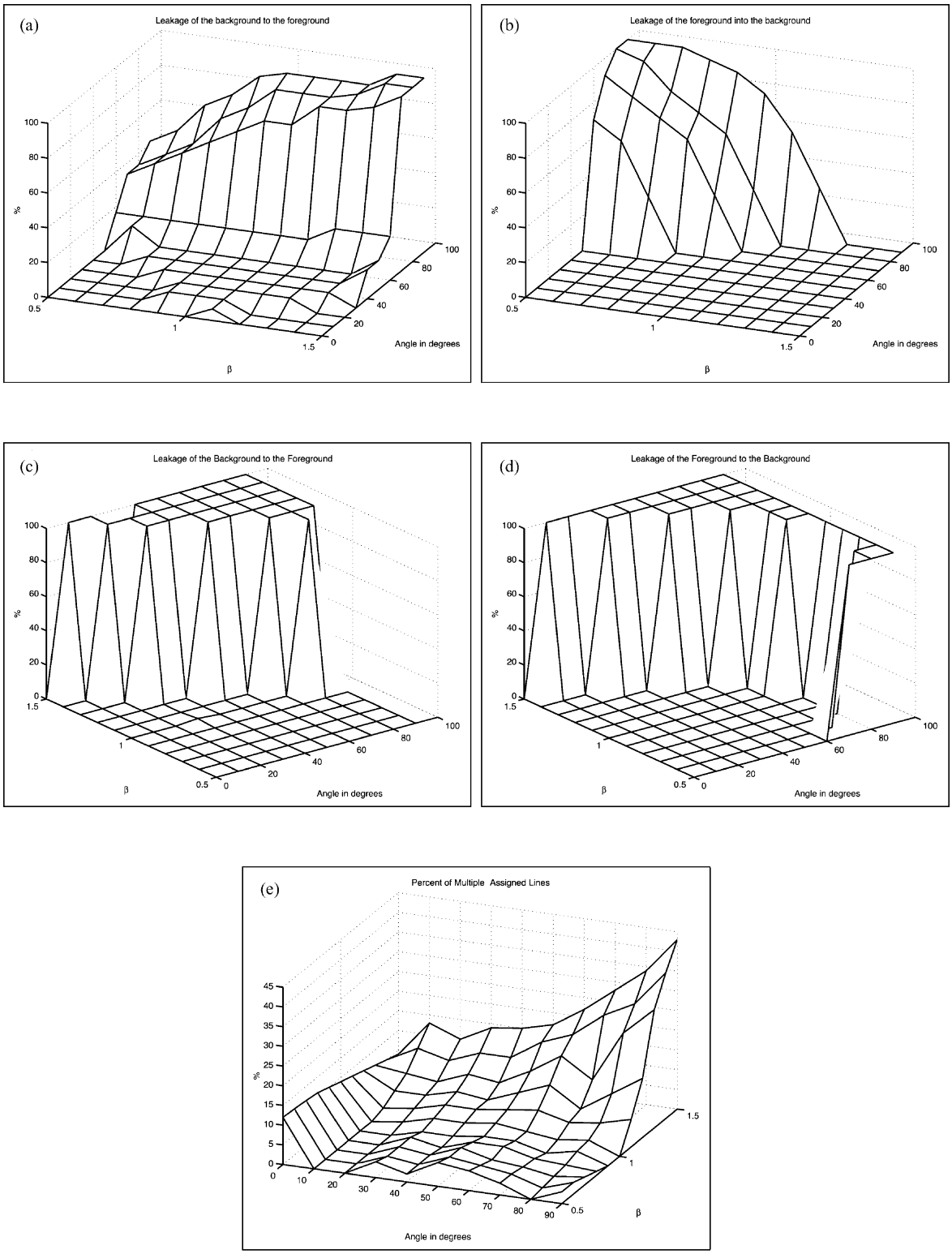

Fig. 6. Sensitivity study. (a) Background to foreground leakage (EM). (b) Foreground to background leakage (EM). (c) Background to foreground leakage (Sarkar and Boyer). (d) Foreground to background leakage (Sarkar and Boyer). (e) Fraction of multiply assigned lines (Sarkar and Boyer).

distribution

$$A_{i,j}^{(0)} = \exp[-\lambda k] \tag{22}$$

where λ is a constant whose best value has been found empirically to be unity. As a result, the linking probability is large when either the relative separation of the endpoints is small, i.e. $\rho_{i,j} \ll \hat{\rho}_{i,j}$ or the polar angle is close to zero or π, i.e. the two lines are

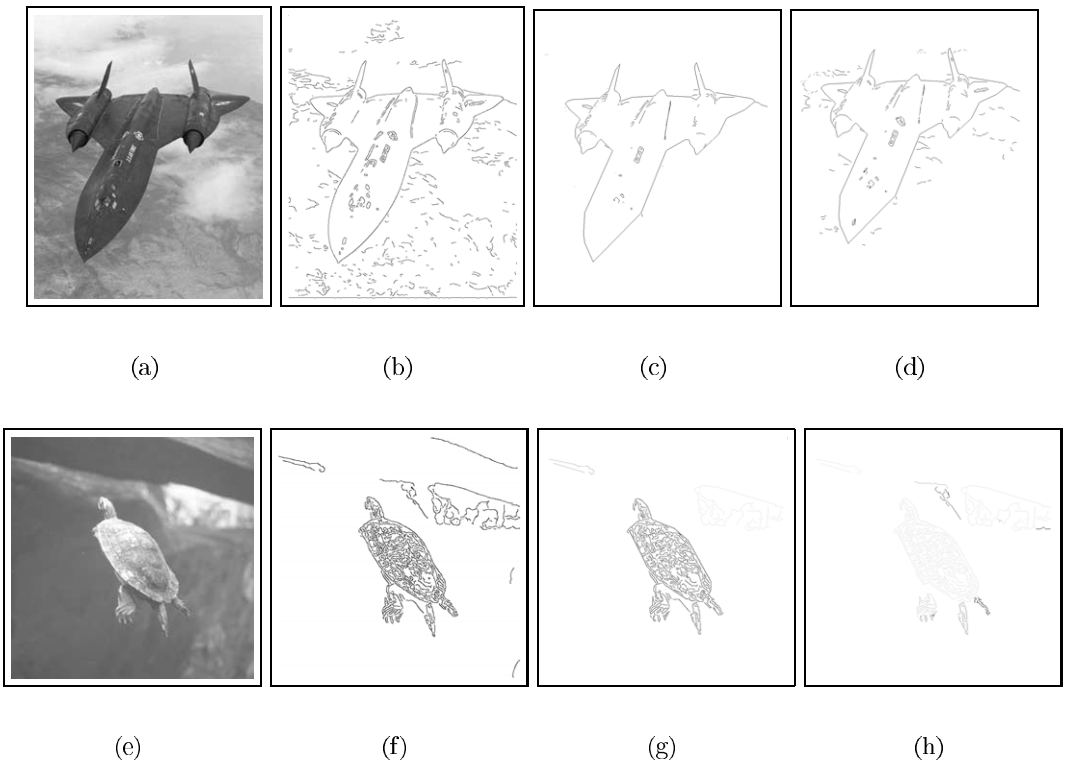

Fig. 7. Real-world images in (a) and (e); (b) and (f) result of the canny edge detector; (c), (g) Resultant clustered images with the EM algorithm; (d), (h) Result of the entropy threshold approach.

colinear or parallel. The linking probability is small when either the relative separation of the endpoints is large, i.e. $\rho_{i,j} \gg \hat{\rho}_{i,j}$ or the polar angle is close to $\pi/2$, i.e. the two lines are perpendicular. In Fig. 1(b) and (c) we show a plot of the linking probability as a function of $\rho_{i,j}/\hat{\rho}_i, j$ and $\theta_{i,j}$.

4.2. Experiments

In this section we provide some experiments to illustrate the utility of our new perceptual grouping method. There are two aspects to this study. We commence by providing some examples for synthetic images. Here we investigate the sensitivity of the method to clutter and compare it with an eigen-decomposition method. The second aspect of our study focuses on real-world images with known ground-truth.

4.2.1. Synthetic images

The first sequence of synthetic images is shown in Fig. 2. Here the foreground structure is an approximately circular arrangement of line-segments. In the first column of Fig. 2 we show the arrangement of lines with increasing amounts of added clutter. In the subsequent columns, we show the results of grouping. The second column shows the pattern of foreground line-segments extracted by applying the eigendecomposition method described in Ref. [11] to the grouping field already detailed in Section 4.1. The third, fourth and fifth columns show the results obtained with the first, second and third iterations of the EM algorithm. Here the line-segments are coded according to the value of the cluster membership weights $s_{i\omega_f}$ where ω_f is the foreground cluster label. As the EM algorithm iterates, the foreground cluster weights tend to unity. In each case, the foreground cluster located by the EM algorithm contains less noise contamination than the result delivered by non-iterative eigendecomposition. Moreover, none of the line-segments leaks into the background.

We have repeated the experiments described above for a sequence of synthetic images in which the density of distractors increases. For each image in turn we have computed the number of distractors merged with the foreground pattern and the number of foreground line-segments which leak into the background. Fig. 3(a) and (b), respectively, shows the fraction of nodes merged with the foreground and the fraction of nodes which leak into the background as a function of the number of distractors. Each plot contains three curves. In both cases, the dotted (leftmost) curve is the result of applying the Sarkar and Boyer [20] method to the raw link-weight matrix. This method assigns lines to clusters based on the same-sign eigenvectors as described in Section 3.3. However, the method attempts to accommodate noise by allowing 5% of the components to flip sign. The dashed (middle) curve is the result of applying the Sarkar and Boyer method to the thresholded link-weight matrix [11]. Here we set the

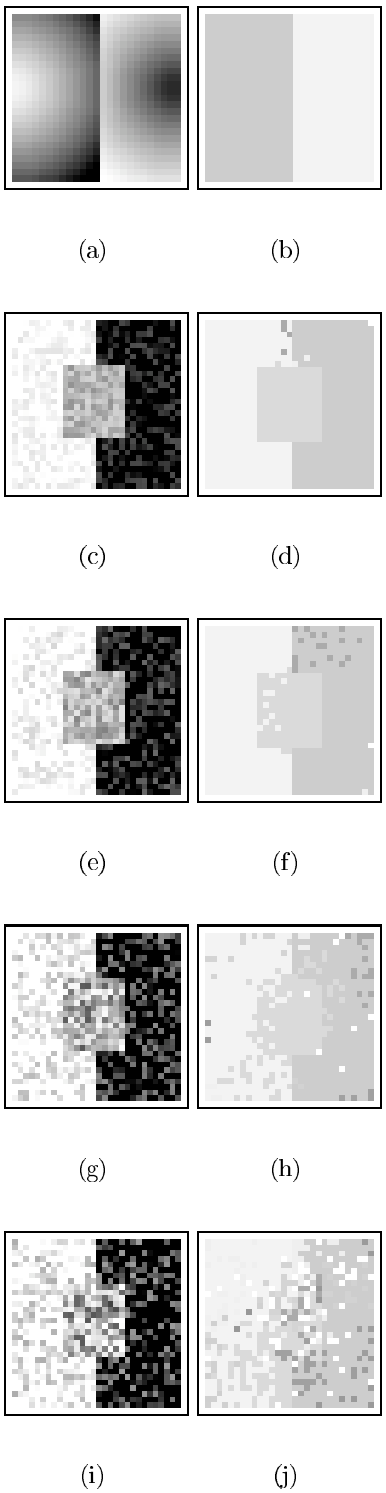

Fig. 8. Results obtained on synthetic images using the EM algorithm.

elements of the matrix to zero if they fall below an entropy threshold. Finally, the dot-dashed (rightmost) curve is the result obtained on convergence of the EM algorithm (in practice 3–4 iterations). In both cases, the shoulder response curve for the EM algorithm occurs at a significantly higher error rate than those for the eigen-decomposition method of Sarkar and Boyer when applied with either sign-flipping tolerance or thresholding of the link-weight matrix.

Next, we turn our attention to the link-weights $A_{i,j}$. To illustrate how the link-weights behave under iteration of the EM algorithm, we have taken an arrangement of line-segments with distractors of variable length and orientation. In Fig. 4(a) we show the pattern of line-segments used as input to the EM algorithm. Fig. 4(b) shows the foreground cluster obtained after two iterations of the algorithm. Fig. 4(c) and (d) shows the initial and final link-weight matrices. Here the foreground line-segments occupy the top left-hand corner of the matrix. Initially, there are a significant number of non-zero entries outside this corner of the matrix. After the EM algorithm has converged, these entries are zero.

In Fig. 4(e) and (f), we illustrate how the distributions of cluster membership variables and link-weights change with iteration number. In the two plots, the green curve shows the initial distribution and the red curve shows the distribution after one iteration of the algorithm. It is clear that in both cases the algorithm sharpens the distributions. However, re-distribution of link-weight is more strongly marked.

Finally, we provide some sensitivity analysis for our grouping algorithm. These experiments have been performed using the distractor grid proposed by Field [9]. The algorithm has been tested by embedding a line pattern consisting of 12 tokens in a background of 220 distractors. The distractor pattern was generated using a grid of 15×15 pixel squares each of which contained a randomly orientated line. The centre point of the distractor lines were positioned as illustrated in Fig. 5(a), where $\Delta\upsilon_{i,j}$ is a random positional jitter in the centre-point position which is sampled from the interval $[0, \tau - \upsilon - 1]$. Here 2τ is the distance from the closest endpoint of the line to the intersection point, 2υ is the line-length and α is the complement of the opening angle. The parameter υ was fixed to 12 pixels and τ was varied between 8 and 24 pixels. The 12 line foreground pattern was created using the geometric parameters τ, α and υ as shown in Fig. 5(b). If an endpoint of a distractor line fell inside a square of the grid occupied by the centre-point of another distractor, then it was deleted. Fig. 5c shows an example pattern.

The results of the performance analysis are shown in Fig. 6. Here we have investigated the effect of varying the opening angle α of the foreground pattern and increasing the gap-length τ. We measure the gap-length using the ratio $\beta = \nu/\tau$. Each point in the plot is averaged over four randomly generated distractor patterns. Fig. 6(a) and (b) shows the results obtained with the EM algorithm. In Fig. 6(a) we show the fraction of background contamination of the foreground as a function of the gap-length ratio β and the line opening angle α. The contamination is greatest for large opening angles and large gap-lengths (i.e. small values of β). The contamination becomes severe once the opening angle exceeds 60°. In Fig. 6(b), we show the fraction of foreground lines that leak into the background. The leakage is again shown as a function of the opening

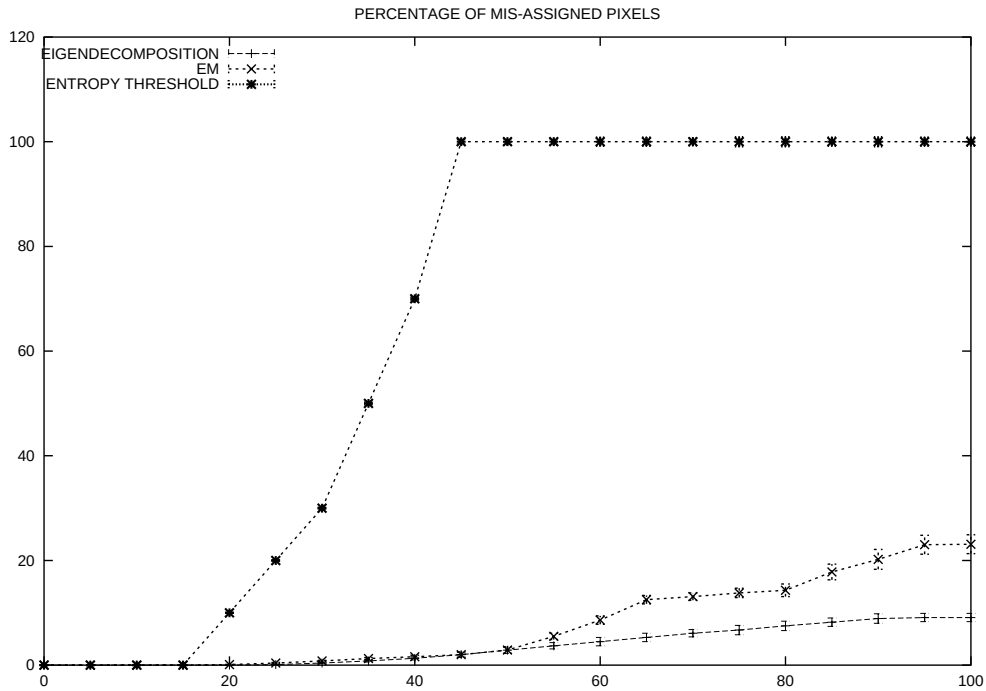

Fig. 9. Number of segmentation errors versus noise standard deviation with sign-flipping (top curve), thresholded link-weight matrix (middle curve) and EM (lower curve).

angle and gap-length ratio. Here the method again fails for large opening angles and large gap-lengths. However, the onset of algorithm failure is now more abrupt. Of particular importance is the fact that the onset of grouping errors is dependent only on the opening angle and not upon the gap-length ratio β. These results are in accord with the psychophysical results obtained by Field [9].

Fig. 6(c) and (d) repeats this analysis for the algorithm of Sarkar and Boyer [20] with sign-flipping tolerance. Now the angle at which there is onset of grouping errors is dependent on the gap-length ratio β. In particular, for small gap-lengths the onset angle for grouping errors becomes small. In other words, the method is more fragile than the EM algorithm. Finally, it is important to stress that the Sarkar and Boyer method allows line-segments to be assigned to more than one eigencluster since it tolerates sign-flips. In Fig. 6e we plot the fraction of ambiguously assigned line-segments as a function of the opening angle and the gap-length ratio β. The fraction of ambiguously assigned lines is greatest for large gap-lengths and large angles.

4.2.2. Real-world images

Finally, we present results on real-world images. We have used the airplane and turtle images shown in Fig. 7(a) and (e). The edges shown in Fig. 7(b) and (f) have been extracted from the raw images using the Canny edge-detector. Straight-line segments have been extracted using the method of Yin [16]. The resulting clusters obtained with the EM algorithm can be seen in Fig. 7(c) and (g). Here the EM algorithm typically converged in 3–4 iterations. For comparison, in Fig. 7(d) and (h) we show the results obtained with the eigendecomposition method of Sarkar and Boyer [20]. In both cases the grouping obtained by the EM algorithm is cleaner and contains less spurious clutter.

5. Grey-scale image segmentation

The second application of our new pairwise clustering method involves segmenting grey-scale images into regions. To compute the initial affinity matrix, we use the difference in grey-scale values at different pixel sites. Suppose that g_i is the grey-scale value at the pixel indexed i and g_j is the grey-scale value at the pixel indexed j. The corresponding entry in the affinity matrix is

$$A_{i,j}^{(0)} = \exp[-k_g(g_i - g_j)^2] \tag{23}$$

where k_g is a heuristically set constant. If we are segmenting an $R \times C$ image of R rows and C columns, then the affinity matrix is of dimensions $RC \times RC$. Hence, the affinity matrix is rather large (256×256). This initial characterisation of the affinity matrix is similar to that used by Shi and Malik [22] in their normalised-cut method of image segmentation. It is important to stress that this is an extremely simplistic method that does not, for instance, include information concerning the spatial proximity of the pixels.

By applying the clustering method to this initial affinity matrix we iteratively segment the image into regions. Each pairwise cluster corresponds to a distinct region.

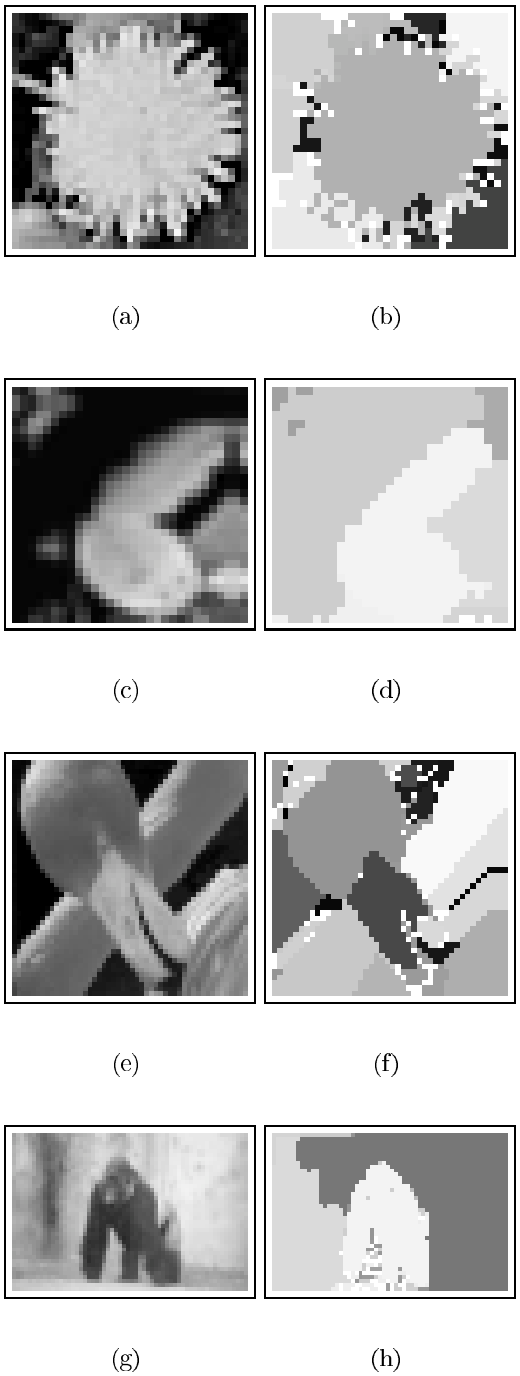

Fig. 10. Segmentations of grey-scale images.

5.1. Experiments

We have conducted experiments on both synthetic and real images. We commence with some examples on synthetic images aimed at establishing some of the properties of the resulting image segmentation method. In Fig. 8(a) and (b) we investigate the effect of contrast variations. The left-hand panel shows an image which is divided into two rectangular regions. Within each region there is a variation in intensity whose distribution is generated using a Lambertian sphere. In the right-hand panel we show the resulting segmentation. There are two modes, i.e. detected regions. These correspond to the two rectangular regions in the original image. There is no fragmentation due to the spherical intensity variation.

Next we consider the effect of added random noise. In Fig. 8(c),(e),(g), and (i) we show a sequence of images in which we have added Gaussian noise of zero mean and known standard deviation to the grey-scale values in an image containing three rectangular regions. In the sequence, each pair of images represents the original noise corrupted image (on the left) and the resulting segmentation (on the right) as the standard deviation of the noise in increased. For the different image pairs, standard deviation of the added Gaussian noise is 35, 50, 65 and 95% of the grey-scale difference between the regions. The final segmentations are obtained with an average of 2.3 iterations. The final segmentations contain 3, 3, 4 and 4 clusters, respectively. The method begins to fail once the noise exceeds 60%. It is also worth noting that the region boundaries and corners are well reconstructed.

Fig. 9 offers a more quantitative evaluation of the segmentation capabilities of the method. Here we compute the fraction of mislabelled pixels in the segmented images as a function of the standard deviation of the added Gaussian noise. The plot shows three performance curves obtained with (a) the eigendecomposition algorithm method of Sarkar and Boyer with sign-flipping (leftmost curve), (b) the Sarkar and Boyer method applied to the thresholded link-weight matrix (middle curve), and (c) the EM method (rightmost curve). The two variants of the Sarkar and Boyer method fail abruptly at low noise-levels. However, thresholding of the link-weight matrix is slightly more robust than using sign-flip tolerance. The EM algorithm performs much better.

To conclude this section, in Fig. 10 we provide some example segmentations on real-world images. In the left-hand column we show the original image, while the right-hand column shows the final segmentation. In Table 1 we list the value of the parameter k_g, the number of clusters and the average number of iterations taken to obtain each segmentation. On the whole the results are quite promising. The segmentations capture the main region structure of the images. Moreover, they are not unduly disturbed by brightness variations or texture. It should be stressed that these results are presented to illustrate the scope offered by our new clustering algorithm and not to make any claims concerning its utility as a tool for image segmentation. To do so would require comparison and sensitivity analysis well beyond the scope of this paper.

6. Conclusions

In this paper, we have presented a new perceptual clustering algorithm which uses the EM algorithm to

estimate link-weights and cluster membership probabilities. The method is based on a mixture model over pairwise clusters. The cluster membership probabilities are modelled using a Bernoulli distribution for the link-weights. We apply the method to the problems of region segmentation and of line-segment grouping. In the case of line-segment grouping, the method appears robust to severe levels of background clutter. Although more preliminary, the results obtained for region segmentation are promising and underline the flexibility of the new method.

There are a number of ways in which the method proposed in the paper can be improved. Presently, we use a soft-assign method to update the cluster membership variables. We are currently investigating whether this step can be rendered more efficient using matrix factorisation method along the lines suggested by Perona and Freeman [17].

Acknowledgement

This work was supported by CONACYT, under grant No. 146475/151752.

References

[1] A. Dempster, N. Laird, D. Rubin, Maximum-likehood from incomplete data via the EM algorithm, Journal of Royal Statistical Society, Series B (Methodological) 39 (1977) 1–38.
[2] J.S. Bridle, Training stochastic model recognition algorithms can lead to maximum mutual information estimation of parameters, NIPS 2 (1990) 211–217.
[3] D. Crevier, A probabilistic method for extracting chains of collinear segments, Image and Vision Computing 76 (1) (1999) 36–53.
[4] W. Dickson, Feature grouping in a hierarchical probabilistic network, Image and Vision Computing 9 (1) (1991) 51–57.
[5] G. Guy, G. Medioni, Inferring global perceptual contours from local features, International Journal of Computer Vision 20 (1/2) (1996) 113–133.
[6] F. Heitger, R. von der Heydt, A computational model of neural contour processing, IEEE CVPR (1993) 32–40.
[7] T. Hofmann, M. Buhmann, Pairwise data clustering by deterministic annealing, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1) (1997) 1–14.
[8] J.M. Rehg, I.J. Cox, S. Hingorani, A bayesian multiple-hypothesis approach to edge grouping and contour segmentation, International Journal of Computing and Vision 11 (1) (1993) 5–24.
[9] D.J. Field, Contour integration by the human visual system: evidence for a local association field, Vision Research 33 (2) (1993) 173–193.
[10] J. Matas, J. Kittler, Junction detection using probabilistic relaxation, Image and Vision Computing 11 (4) (1993) 197–202.
[11] A. Robles Kelly, E.R. Hancock, Grouping-line segments using eigenclustering, Proceedings of the British Machine Vision Conference (2000).
[12] J.A.F. Leite, E.R. Hancock, Iterative curve organisation with the EM algorithm, Pattern Recognition Letters 18 (1997) 143–155.
[13] J.M. Leitao, M. Figueiredo, A.K. Jain, On fitting mixture models, Proceedings of the Second International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science, Springer-Verlag, York, UK, vol. 1654, 1999, pp. 54–67.
[14] Z. Ghahramani, N. Ueda, R. Nakano, G.E. Hinton, Smem algorithm for mixture models, Neural Computation 12 (9) (2000) 2109–2128.
[15] P. Parent, S. Zucker, Trace inference, curvature consistency and curve detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (8) (1989) 823–839.
[16] P.-Y. Yin, Algorithms for straight line fitting using k-means, Pattern Recognition Letters 19 (1998) 31–41.
[17] P. Perona, W.T. Freeman, Factorization approach to grouping, Proceedings of ECCV (1998) 655–670.
[18] J. Risanen, Stochastic Complexity in Statistical Enquiry, World Scientific, Singapore, 1989.
[19] R. Castaño, S. Hutchinson, A probabilistic approach to perceptual grouping, Computer Vision and Image Understanding 64 (3) (1996) 339–419.
[20] S. Sarkar, K.L. Boyer, Quantitative measures of change based on feature organization: eigenvalues and eigenvectors, Computer Vision and Image Understanding 71 (1) (1998) 110–136.
[21] A. Shashua, S. Ullman, Structural saliency: the detection of globally salient structures using a locally connected network, Proceedings of Second International Conference in Computer Vision (1988) 321–327.
[22] J. Shi, J. Malik, Normalized cuts and image segmentations, Proceedings of IEEE CVPR (1997) 731–737.
[23] N. Vlassis, A. Likas, A kurtosis-based dynamic approach to gaussian mixture modeling, IEEE Transactions in Systems, Man and Cybernetics 29 (4) (1999) 393–399.
[24] Y. Weiss, Segmentation using eigenvectors: a unifying view, IEEE International Conference on Computer Vision (1999) 975–982.
[25] L.R. Williams, D.W. Jacobs, Local parallel computation of stochastic completion fields, Neural Computation 9 (4) (1997) 859–882.
[26] L.R. Williams, D.W. Jacobs, Stochastic completion fields: a neural model of illusory contour shape and salience, Neural Computation 9 (4) (1997) 837–858.
[27] Y. Sakamoto, M. Ishiguro, G. Kitagawa, Akaike Information Criterion Statistics, KTK Scientific Publishers, 1988.