PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 37 (2004) 1387–1405

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

A probabilistic spectral framework for grouping and segmentation

Antonio Robles-Kelly*, Edwin R. Hancock

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

Received 21 August 2003; received in revised form 28 October 2003; accepted 28 October 2003

Abstract

This paper presents an iterative spectral framework for pairwise clustering and perceptual grouping. Our model is expressed in terms of two sets of parameters. Firstly, there are cluster memberships which represent the affinity of objects to clusters. Secondly, there is a matrix of link weights for pairs of tokens. We adopt a model in which these two sets of variables are governed by a Bernoulli model. We show how the likelihood function resulting from this model may be maximised with respect to both the elements of link-weight matrix and the cluster membership variables. We establish the link between the maximisation of the log-likelihood function and the eigenvectors of the link-weight matrix. This leads us to an algorithm in which we iteratively update the link-weight matrix by repeatedly refining its modal structure. Each iteration of the algorithm is a three-step process. First, we compute a link-weight matrix for each cluster by taking the outer-product of the vectors of current cluster-membership indicators for that cluster. Second, we extract the leading eigenvector from each modal link-weight matrix. Third, we compute a revised link weight matrix by taking the sum of the outer products of the leading eigenvectors of the modal link-weight matrices.

Keywords: Graph-spectral methods; Maximum likelihood; Perceptual grouping; Motion segmentation

1. Introduction

Many problems in computer vision can be posed as ones of pairwise clustering. That is to say they involve grouping objects together based on their mutual similarity rather than their closeness to a cluster prototype. Such problems naturally lend themselves to a graph-theoretic treatment in which the objects to be clustered are represented using a weighted graph. Here the nodes represent the objects to be clustered and the edge-weights represent the strength of pairwise similarity relations between objects. One of the most elegant solutions to the pairwise clustering problem comes from spectral graph theory, which is a field of mathematics which aims to characterise the structural properties of graphs using the eigenvalues and eigenvectors of the Laplacian matrix. The result that is key to the grouping problem is that the eigenvalue gap (i.e. the difference between the first and second eigenvalues of the Laplacian matrix), is a measure of the degree of bijectivity of the graph (i.e. the extent to which its nodes form two distinct clusters which can be separated by a minimum cut). To exploit this property, graph spectral segmentation methods share the feature of commencing from an initial characterisation of the perceptual affinity of different image tokens in terms of a matrix of link-weights. Once this matrix is to hand then its eigenvalues and eigenvectors are located. The eigenmodes represent pairwise relational clusters which can be used to group the raw perceptual entities together.

1.1. Related literature

Roughly speaking, the problem of graph-spectral clustering is a two-step process. The first step involves the choice

* Corresponding author. Tel.: +44-1904-432774; fax: +44-1904-432767.
E-mail addresses: arobkell@cs.york.ac.uk (A. Robles-Kelly), erh@cs.york.ac.uk (E.R. Hancock).

0031-3203/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2003.10.017

of utility measure for the cluster process. There are two quantities that are commonly used to define the utility. The first of these is the association, which is a measure of total edge linkage within a cluster and is useful in defining clump structure. The second is the cut, which is a measure of linkage between different clusters and can be used to split extraneous nodes from a cluster. The second step is to show how to use eigenvectors to extract clusters using the utility measure, and this can be regarded as a post-processing step. There are several examples of the graph-spectral approach described in the literature. Some of the earliest work was conducted by Scott and Longuet-Higgins [1] who developed a method for refining the block-structure of the affinity matrix by relocating its eigenvectors. At the level of image segmentation, several authors have used algorithms based on the eigenmodes of an affinity matrix to iteratively segment image data. For instance, Sarkar and Boyer [2] have a method which uses the leading eigenvector of the affinity matrix, and this locates clusters that maximise the average association. This method is applied to locating line-segment groupings. Perona and Freeman [3] have a similar method which uses the second largest eigenvector of the affinity matrix. The method of Shi and Malik [4], on the other hand, uses the normalised cut which balances the cut and the association. Clusters are located by performing a recursive bisection using the eigenvector associated with the second smallest eigenvalue of the Laplacian (the degree matrix minus the adjacency matrix), i.e. the Fiedler vector. Focussing more on the issue of post-processing, Weiss [5] has shown how this, and other closely related methods, can be improved using a normalised affinity matrix. Recently, Shi and Meilă [6] have analysed the convergence properties of the method using Markov chains. In cognate work, Tishby and Slonim [7] have developed a graph-theoretic method which exploits the stationarity and ergodicity properties of Markov chains defined on the affinity weights to locate clusters.

Recent work has looked in more detail at the spectral grouping method. For instance, Fowlkes et al. [8] and Belongie et al. [9] have shown how it can be rendered more efficient by sub-sampling the affinity matrix and using the Nyström method to approximate the eigenvectors. Soundararajan and Sarkar [10] have investigated the role of the utility measure underpinning the graph-partitioning method. They conclude that the minimum cut and the normalised cut lead to the same average segmentations. Empirical results show that the minimum, average and normalised cuts give results that are statistically equivalent.

The problem of clustering by graph partitioning is of course one of generic utility throughout computer science. In the algorithms community, there has also been considerable effort expended at analysing the properties and behaviour of graph spectral clustering methods. For instance, Mohar [11] provides a good review of the properties of Laplace eigenvalues and eigenvectors. Recent work by Kannan et al. [12] present a new partition quality measure. The measure draws on the minimum conductance and the ratio of the inter-cluster edge weight (the cut) to the total cluster edge weight (the association). An analysis reveals that although the clustering problem is NP hard, the proposed measure leads to an approximation algorithm with poly-logarithmic guarantees.

The problem of perceptual grouping has also been extensively studied using information theoretic and probabilistic frameworks. Early work by Dickson [13] has used Bayes nets to develop an hierarchical framework for splitting and merging groups of lines. Cox et al. [14] have developed a grouping method which combines evidence from the raw edge attributes delivered by the Canny edge detector. Leite and Hancock [15] have pursued similar objectives with the aim of fitting cubic splines to the output of a bank of multi-scale derivative of Gaussian filters using the EM algorithm. Castaño and Hutchinson [16] 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 [17] has developed an evidence combining framework for extracting chains of colinear line-segments. Amir and Lindenbaum [18] have a maximum likelihood method for grouping which relies on searching for the best graph partition. The method is two step. First, grouping cues are used to construct the graph. Second, a greedy modification step is used to maximise the likelihood function. Turning our attention to information theoretic approaches, one of the best known methods is that of Hofmann and Buhmann [19], which uses mean-field theory to develop update equations for the pairwise cluster indicators. In related work, Gdalyahu et al. [20] use a stochastic sampling method. These iterative processes have some features in common with the use of iterative relaxation style operators for edge grouping. This approach was pioneered by Shashua and Ullman [21] and later refined by Guy and Medioni [22] among others. Parent and Zucker have shown how co-circularity can be used to gauge the compatibility of neighbouring edges [23].

1.2. Contribution

Although elegant by virtue of their use of matrix factorisation to solve the underlying optimisation problem, one of the criticisms which can be levelled at the graph-spectral methods is that their foundations are not statistical or information theoretic in nature. As demonstrated by Hofmann and Buhmann [19], Gdalyahu et al. [20] and Amir and Lindenbaum [18], there are significant advantages to be had from posing the problem of grouping by graph partitioning in a statistical or probabilistic setting. Since they do not do this, the post-processing strategies adopted by spectral methods are not able to characterise uncertainties in the raw affinity data or to combine evidence to overcome these uncertainties. Moreover, they lack the robustness that evidence combining

approaches offer. The aim in this paper is to overcome these shortcomings by developing a maximum likelihood framework for pairwise clustering. We parameterise the pairwise clustering problem using two sets of indicator variables. The first of these are cluster membership variables which indicate to which cluster an object belongs. The second set of variables are the elements of a link-weight matrix which convey the strength of association between pairs of nodes.

We use these two sets of parameters to develop a probabilistic model of the pairwise clustering process. We use Bernoulli trials to model the probability that pairs of nodes belong to the same pairwise cluster. The parameter of the distribution is the link-weight between pairs of nodes. The random variable associated with the Bernoulli trial is the cluster co-membership indicator that measures whether, or not, a pair of nodes belong to the same cluster. The co-membership is found by taking the product of the cluster membership indicators for pairs of nodes. We develop a log-likelihood function under the assumption that pairs of nodes associate to form clusters as the outcome of a series of independent Bernoulli trials of this sort.

The resulting log-likelihood function is in fact the total association of a logarithmic transformation of the link-weight matrix for the set of clusters. To maximise the log-likelihood function, we develop a post-processing method that is realised as an iterative clustering algorithm based on dual interleaved steps. First, the link weights are updated using the currently available cluster membership indicators. This is done by differentiating the log-likelihood function with respect to the link weights and solving the associated saddle-point equations. The update equation is particularly simple. We compute an updated link-weight matrix for each cluster by taking the outer-product of the vectors of cluster-membership indicators. We refine the structure of the updated link-weight matrix with the aim of removing noisy link-weights. To do this we decompose the updated link-weight matrix into components that originate from the different clusters. For each cluster-component of the link-weight matrices, we compute the leading eigenvector. We compute a revised link weight matrix by taking the sum of the outer products of the leading eigenvectors of the cluster link-weight matrices. This updating and refinement of the link-weight matrix is a unique feature of our method. In the pairwise clustering algorithm of Hoffman and Buhmann [19], and, the normalised cuts method of Shi and Malik [4], the link-weights remain static. The second step of the algorithm is concerned with updating the cluster membership indicators. However, since the saddle-point equations for the cluster membership indicators are not tractable in closed form, we take recourse to a naive mean field method known as soft-assign to develop update equations.

Stated in this way the dual update steps of our method is reminiscent of the EM algorithm. In fact in related work, we have developed an EM algorithm for grouping using a mixture of Bernoulli distributions [24]. However, the method proved slow to converge and resulted in overlapped clusters. By contrast, here we use the modal structure of the link-weight matrix to define the clusters. In doing so we develop on the work of Sarkar and Boyer [2]. We initialise the cluster memberships using the same sign positive eigenvectors of the initial link-weight matrix. We show how the log-likelihood function can be separated into distinct terms associated with the different modes of the link-weight matrix. This allows us to refine the cluster membership indicators using the leading eigenvector of the updated link-weight matrices for the different clusters.

It is important to stress that although there have been some attempts at using probabilistic methods for grouping elsewhere in the literature [16], our method has a number of unique features which distinguish it from these alternatives. First, although there have been successful attempts to develop probabilistic methods for grouping via graph partitioning, these do not use spectral information, and are instead based on search heuristics. Second, although our method relies on the iterative updating of cluster membership indicators it differs from the methods of Hofmann and Buhmann [19], and Shi and Malik [4] by virtue of the fact that the link-weight matrix is iteratively refined.

2. Grouping by matrix factorisation

To commence, we require some formalism. The grouping problem is characterised by the set of objects to be clustered V and a $|V| \times |V|$ matrix of link-weights A. The element $A_{i,j}$ of the link weight matrix represents the strength of association between the objects i and j. We will work with link-weights which are constructed to fall in the interval $[0, 1]$. When the link weight is close to one, then there is a strong association between the pair of nodes; when it is close to zero then there is a weak association. The aim in grouping is to partition the object-set V into disjoint subsets. If V_ω represents one of these subsets and Ω is the index-set of different partitions (i.e. the different pairwise clusters), then $V = \bigcup_{\omega \in \Omega} V_\omega$ and $V_{\omega'} \cap V_{\omega''} = \emptyset$ if $\omega' \neq \omega''$.

To represent the assignment of nodes to clusters, we introduce a cluster membership indicator $s_{i\omega}$. This quantity measures the degree of affinity of the node i to the cluster $\omega \in \Omega$ and is in the interval $[0, 1]$. When the cluster membership is close to 1 then there is a strong association of the node to the cluster; when the value is close to 0 then the association is weak.

Later on, it will be convenient to work with a matrix representation of the cluster membership indicators. Hence, we introduce a vector of indicator variables for the cluster indexed ω $\underline{s}_\omega = (s_{1\omega}, s_{2\omega}, \ldots)^{\mathrm{T}}$. The vectors are used as the columns of the $|V| \times |\Omega|$ cluster membership matrix $S = (\underline{s}_1 | \underline{s}_2 | \cdots | s_{|\Omega|})$ whose rows are indexed by the set of nodes and whose columns are indexed by the set of clusters.

In this paper, we are interested in how matrix factorisation methods can be used to partition the nodes into disjoint clusters. One way of viewing this is as the search for the

permutation matrix which re-orders the elements of A into non-ovelapping blocks. Sarkar and Boyer [2] 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 $\underline{x}^T A \underline{x}$, where A is the 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.

We confine our 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 relevant eigenmodes of the weighted adjacency matrix. The eigenvalues $\lambda_1, \lambda_2, \ldots$ of A are the solutions of the equation $|A - \lambda I| = 0$ where I is the $N \times N$ identity matrix. The corresponding eigenvectors $\underline{x}_{\lambda_1}, \underline{x}_{\lambda_2}, \ldots$ are found by solving the equation $A\underline{x}_{\lambda_i} = \lambda_i \underline{x}_{\lambda_i}$. Let the set of positive same-sign eigenvectors be represented by $\Omega = \{\omega | \lambda_\omega > 0 \wedge [(\underline{x}^*_\omega(i) > 0 \forall i) \vee \underline{x}^*_\omega(i) < 0 \forall i]\}$, where $\underline{x}^*_\omega$ indicates a same-sign eigenvector. 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 $V_\omega = \{i | \underline{x}^*_\omega(i) \neq 0\}$.

3. Maximum likelihood framework

In practice the link-weight matrix is likely to be subject to noise and error. As a result, the eigenvector clustering algorithm described above will produce poor clusters. To overcome this problem, Sarkar and Boyer allow a certain fraction of the components of the eigenvectors to flip sign. This simple method is aimed a modelling the effect of noise on the eigenvectors and is motivated by a perturbation analysis.

The aim in this paper is to develop a more sophisticated probabilistic method. We commence from a simple model of the cluster formation process based on a series of independent Bernoulli trials. The linkage of each pair of nodes within a cluster is treated as a separate Bernoulli trial. We treat the link-weight for the pair of nodes as the success probability of the trial. The random variable associated with the trial is the product of cluster indicators for the pair of nodes; this indicates whether the two nodes belong to the same cluster. Using this model we develop a joint likelihood function for the link-weights and the cluster membership indicators. This likelihood function can be used to make both a maximum likelihood re-estimate of the link-weight matrix and a maximum a posteriori probability estimate of the cluster membership indicators. In the case of re-estimating the link-weight matrix, the cluster indicators are treated as data.

It is important to stress that the dual update steps which constitute our algorithm are decoupled from the raw image data, once the initial link-weight matrix has been computed. The two steps are aimed at improving the structure of the link-weight matrix and the pairwise clusters that can be extracted from it. One way of viewing this process is that of applying a kind of relaxation process which smooths the link-weight matrix by re-enforcing adjacent elements within a block.

3.1. Joint likelihood function

Our grouping process aims to estimate the cluster membership indicators S and to obtain an improved estimate of the link-weight matrix A. We pose both problems in terms of the conditional likelihood $P(S|A)$. The problem of recovering indicator variables is one of maximum a posteriori probability estimation of S given the current link-weight matrix A. Here the link weight matrix plays the role of fixed data. The re-estimation of A is posed as maximum likelihood estimation, with the cluster membership indicators playing the role of fixed data.

To develop the two update steps, we turn our attention to the conditional likelihood function $P(S|A) = P(\underline{s}_1, \underline{s}_2, \ldots, \underline{s}_{|\Omega|}|A)$. To simplify the likelihood function, we make a number of independence assumptions. We commence by applying the chain rule of conditional probability to rewrite the likelihood function as a product of conditional probabilities

$$P(S|A) = P(\underline{s}_1|\underline{s}_2, \ldots, \underline{s}_{|\Omega|}, A) \times P(\underline{s}_2|\underline{s}_3, \ldots, \underline{s}_{|\Omega|}, A) \times \cdots \times P(\underline{s}_{|\Omega|}|A). \quad (1)$$

To simplify this product, we assume that vectors of class indicators are conditionally independent of one-another given the matrix of link-weights. Hence,

$$P(\underline{s}_i|\underline{s}_{i+1}, \ldots, \underline{s}_{|\Omega|}, A) = P(\underline{s}_i|A).$$

It is important to stress that this condition may not hold in general. It is violated when there is cluster overlap or linkage, and this effect is measured by the between cluster cut. However, when we initialise the process with cluster indicator variables computed from the eigenvectors of the adjacency matrix, then the condition is satisfied due to the fact that the eigenvectors are orthogonal. Using this simplification, then, we can write the conditional likelihood as a product over the cluster indices

$$P(S|A) = \prod_{\omega \in \Omega} P(\underline{s}_\omega|A). \quad (2)$$

An important consequence of this factorisation over clusters, is that when combined with our cluster membership model developed in Section 3.2, it leads to a likelihood function in which there is no dependence on the between cluster link structure. This is in contrast with other work on pairwise clustering (e.g. the normalised cut), where both the within and between cluster link weights play a role.

Next we apply the definition of conditional probability to re-write terms under the product in the following manner:

$$P(S|A) = \prod_{\omega\in\Omega} \frac{P(A|\underline{s}_\omega)P(\underline{s}_\omega)}{P(A)}. \tag{3}$$

To further develop the expression for the likelihood we turn our attention to the conditional probability for the link-weight matrix given the vector of cluster indicators for the cluster ω, i.e. $P(A|\underline{s}_\omega)$. Again applying the chain rule of conditional probability, we can perform the following factorisation over the non-diagonal elements of the link-weight matrix:

$$P(A|\underline{s}_\omega) = \prod_{(i,j)\in\Phi} P(A_{i,j}|A_{k,l}, k > i, l > j, \underline{s}_\omega), \tag{4}$$

where $\Phi = V \times V - \{(i,i)|i \in V\}$ is the set of non-diagonal elements of A. To simplify the factorisation, we assume that the element $A_{i,j}$ is conditionally dependant only on the cluster indicators for the nodes indexed i and j. Hence, we can write

$$P(A_{i,j}|A_{k,l}, k > i, l > j, \underline{s}_\omega) = P(A_{i,j}|s_{i\omega}, s_{j\omega}).$$

Under this simplification

$$P(A|\underline{s}_\omega) = \prod_{(i,j)\in\Phi} P(A_{i,j}|s_{i\omega}, s_{j\omega}). \tag{5}$$

Substituting this expression into that for the joint-likelihood we have that

$$P(S|A) = \prod_{\omega\in\Omega} \left\{ \frac{P(\underline{s}_\omega)}{P(A)} \prod_{(i,j)\in\Phi} \left[\frac{p(s_{i\omega}, s_{j\omega}|A_{i,j})P(A_{i,j})}{P(s_{i\omega}, s_{j\omega})} \right] \right\}. \tag{6}$$

As stated earlier, we aim to recover revised estimates of both the link-weight matrix and the cluster indicators. These estimates are realised using dual interleaved update operations. The recovery of the revised link weight matrix is posed as the maximum likelihood parameter estimation problem

$$A^* = \arg\max_A P(S|A). \tag{7}$$

The recovery of the cluster membership indicators, on the other hand, is posed as the maximum a posteriori probability estimation problem

$$S^* = \arg\max_S P(S|A). \tag{8}$$

Since we are interested in the joint dependence of the link-weight matrix A and the cluster membership indicators S, we turn our attention instead to the maximisation of the log-likelihood function for the observed pattern of link weights. Further, since we assume that the link-weights belonging to each cluster are independent of one another, we can write

$$\mathcal{L}(A,S) = \sum_{\omega\in\Omega} \sum_{(i,j)\in\Phi} \ln\, p(s_{i\omega}, s_{j\omega}|A_{i,j}). \tag{9}$$

In the next section, we describe a simple model for the conditional probability density for the indicator variables given the current estimate of the link-weight matrix elements i.e. $p(s_{i\omega}, s_{j\omega}|A_{i,j})$. The cluster membership indicators play the role of random variables, and the link-weights the role of distribution parameters. In Section 3.3, we describe how the log-likelihood function may be optimised with respect to the cluster indicator variables, given the initial estimates of the link-weights. We also describe how the estimates of link-weights may be refined once cluster membership indicators are to hand.

3.2. Bernoulli model

We now describe the generative model which underpins our pairwise clustering method. The model assumes that pairs of nodes associate to clusters as the outcome of a Bernoulli trial. The idea is that the observed link structure of the pairwise clusters arises as the outcome of a series of Bernoulli trials. The probability that a link form between a pair of nodes if simply the link-weight between the nodes. To be more formal, let us consider the pair of nodes i and j. We are concerned with whether or not this pair of nodes both belong to the cluster indexed ω. The random variable that governs the outcome of the Bernoulli trial is the product of indicator variables $\zeta_{i,j,\omega} = s_{i\omega}s_{j\omega}$. There of four combinations of the two indicator variables $s_{i\omega}$ and $s_{j\omega}$. For the single case when $s_{i\omega} = s_{j\omega} = 1$, then the two nodes have a pairwise association to the cluster indexed ω, and $\zeta_{i,j\omega} = 1$. In the three cases when either $s_{i\omega} = 0$ or $s_{j\omega} = 0$, then the pair of nodes do not both associate to the cluster ω and $\zeta_{i,j,\omega} = 0$. We model the cluster formation process as a series of independent Bernoulli trials over all pairs of nodes. According to this model, success is the event that both nodes belong to the same cluster, while failure is the event that they do not. The probability of success, i.e. the parameter of the Bernoulli trial is the link-weight $A_{i,j}$. Success is the event $\zeta_{i,j,\omega} = 1$, and there is a single combination of cluster indicators $s_{i,\omega} = s_{j\omega} = 1$ that results in this outcome. The remaining probability mass $1 - A_{i,j}$ is assigned to the three cases which result in failure, i.e. those for which $\zeta_{i,j,\omega} = 0$. This simple model is captured by the distribution rule

$$p(s_{i\omega}, s_{j\omega}|A_{i,j}) = \begin{cases} A_{i,j} & \text{if } s_{i\omega} = 1 \text{ and } s_{j\omega} = 1, \\ (1 - A_{i,j}) & \text{if } s_{i\omega} = 0 \text{ or } s_{j\omega} = 0. \end{cases} \tag{10}$$

This rule can be written in the more compact form

$$p(s_{i\omega}, s_{j\omega}|A_{i,j}) = A_{i,j}^{s_{i\omega}s_{j\omega}}(1 - A_{i,j})^{1-s_{i\omega}s_{j\omega}}. \tag{11}$$

We could clearly have adopted a more complicated model, by not distributing the probability uniformly among the three

cases for which $\zeta_{i,j,\omega}=0$. Moreover, the model is developed under the assumption that the quantities $s_{i\omega}$ and $s_{j\omega}$, and hence $\zeta_{i,j,\omega}$ are binary in nature. When we come to update the cluster indicators, then we relax this condition and the quantities no longer belong to the set $\{0,1\}$, but instead belong to the interval $[0,1]$.

After substituting this distribution into the log-likelihood function, we find that

$$\mathscr{L}(A,S)=\sum_{\omega\in\Omega}\sum_{(i,j)\in\Phi}\{s_{i\omega}s_{j\omega}\ln A_{ij} + (1-s_{i\omega}s_{j\omega})\ln(1-A_{i,j})\}. \quad (12)$$

Performing algebra to collect terms, the log-likelihood function simplifies to

$$\mathscr{L}(A,S)=\sum_{\omega\in\Omega}\sum_{(i,j)\in\Phi}\left\{s_{i\omega}s_{j\omega}\ln\frac{A_{ij}}{1-A_{ij}} + \ln(1-A_{i,j})\right\}. \quad (13)$$

The structure of the log-likelihood function deserves further comment. The first term, which depends on the cluster membership indicators, is closely related to the association measure for the configuration of clusters. Classically, the association of the cluster indexed ω is defined to be $Assoc(\omega)=\sum_{i\in V}\sum_{j\in V}s_{i\omega}s_{j\omega}A_{i,j}=\vec{s}_{\omega}^{\mathrm{T}}A\vec{s}_{\omega}$. Hence, our log-likelihood function is the sum of the individual cluster associations for the logarithmically transformed link weight matrix. Our method hence does not take into account the cut-measure between clusters. The cut between the clusters indexed ω_a and ω_b is defined to be $Cut(\omega_a,\omega_b)=\sum_{i\in\omega_a}\sum_{j\in\omega_b}s_{i\omega_a}s_{j\omega_b}A_{i,j}$. There is considerable debate in the spectral clustering literature concerning the choice of utility measure. Maximising the association is widely thought to work well with compact well separated clusters, and is at the heart of the Sarkar and Boyer method. Minimising the cut, on the other hand can remove outliers from an otherwise well-defined cluster. To strike a balance between these two behaviours has lead to the development of more sophisticated measures such as the normalised cut [4]. As evidenced by the recent paper of Kannan et al. [12], the debate on the optimal choice of utility measure continues. However, as noted by Weiss [5] the post-processing of the spectral representation can play a pivotal role in determining the quality of the clusters recoverable. This is not surprising. For instance, techniques such as relaxation labelling have proved to be very effective in improving the results of otherwise limited initial labellings. However, the aim here is to commence from a principal starting point. Our Bernoulli model, and indeed any simple probability distribution, is unlikely to lead to a measure that has a structure similar to the normalised cut or the conductance measure defined by Kannan et al. [12]. Hence, we turn our attention to the post-processing of the likelihood function using spectral analysis. As we will demonstrate experimentally, this leads to results which are comparable to and sometimes better than the normalised cut.

There are a number of additional points concerning the structure of the log-likelihood function. First, when the cluster membership indicators are initialised using the components of the same-sign eigenvectors of A (as described later), it gauges only the within-cluster structure of the link-weight matrix. There are no contributions from between cluster links. The second feature is that the structure of the log-likelihood function is reminiscent of that underpinning the expectation-maximisation algorithm. The reason for this is that the product of cluster-membership variables $s_{i\omega}s_{j\omega}$ plays a role similar to that of the a posteriori measurement probability in the EM algorithm, and weight contributions of the link-weights to the likelihood function.

In a recent paper, we have developed an EM algorithm for grouping which is based on a mixture of Bernoulli distributions. However, this method proved slow to converge and resulted in overlapped clusters. To overcome these problems, in this paper we aim to develop an iterative grouping method which focuses on refining the modal structure of the link-weight matrix.

3.3. Maximising the likelihood function

In this section, we focus on how the log-likelihood function can be maximised with respect to the link-weights and the cluster membership variables. This is a three-step process. We commence by showing how the maximum likelihood link-weight matrix can be located by taking the outer-product of the vectors of cluster membership indicators. Second, we show how to remove noise from the link-weight matrix using a process which we refer to as modal sharpening. This involves decomposing the link-weight matrix into components corresponding to the same-sign eigenvectors. For each component or cluster there is an individual link-weight matrix. For each such matrix, we compute the leading eigenvector. The modal sharpening process involves reconstructing the overall link-weight matrix by summing the outer products of the leading eigenvectors of the cluster link-weight matrices, The third, and final component of the update process is to update the link-weights. This is done by applying a naive mean field method to the likelihood function.

3.3.1. Updating the link-weight matrix

Our aim is to explore how the log-likelihood function can be maximised with respect to the link-weights and the cluster membership indicators. In this section, we turn our attention to the first of these. To do this we compute the derivatives of the expected log-likelihood function with respect to the elements of the link-weight matrix

$$\frac{\partial\mathscr{L}}{\partial A_{ij}}=\sum_{\omega\in\Omega}\left\{s_{i\omega}s_{j\omega}\frac{1}{A_{ij}(1-A_{ij})}-\frac{1}{1-A_{ij}}\right\}. \quad (14)$$

The matrix of updated link weights $\hat{A}_\omega$ may be found by setting the derivatives to zero and solving the equation $\partial\mathscr{L}/\partial A_{ij} = 0$. The derivative vanishes when

$$\hat{A}_{ij} = \frac{1}{|\Omega|}\sum_{\omega\in\Omega} s_{i\omega}s_{j\omega}. \tag{15}$$

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. We can make the structure of the updated link-weight matrix clearer if we make use of the vector of membership variables for the cluster indexed ω, i.e. $\underline{s}_\omega = (s_{1\omega}, s_{2\omega}, \ldots)^{\mathrm{T}}$. With this notation the updated link-weight matrix is $\hat{A} = (1/|\Omega|)\sum_{\omega\in\Omega}\underline{s}_\omega\underline{s}_\omega^{\mathrm{T}}$. Hence, the updated link-weight matrix is simply the average of the outer-products of the vectors of cluster membership indicators. We can make the cluster-structure of the link-weight matrix clearer if we introduce the link-weight matrix $\hat{A}_\omega = \underline{s}_\omega\underline{s}_\omega^{\mathrm{T}}$ for the cluster indexed ω. With this notation, we can write the updated link weight matrix as the sum of contributions from different clusters, i.e.

$$\hat{A} = \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\hat{A}_\omega. \tag{16}$$

Finally, we note that the updated link-weight matrix can be written in the compact form $\hat{A} = (1/|\Omega|)SS^{\mathrm{T}}$.

3.3.2. Modal sharpening of the link-weight matrix

In practice, the link-weight matrix may be noisy and hence the cluster structure may be subject to error. In an attempt to overcome this problem, in this section we turn our attention to how the updated link-weight matrix may be refined with a view to improving its block structure. The aim here is to suppress structure which is not associated with the principal modes of the matrix.

We commence by focussing in more detail on the significance of the update process described in the previous section. To do this, we return to the expression for the log-likelihood function. The component of the log-likelihood which depends on the cluster indicators is

$$\hat{\mathscr{L}}(A,S) = \sum_{\omega\in\Omega}\sum_{(i,j)\in\Phi}\left\{s_{i\omega}s_{j\omega}\ln\frac{A_{ij}}{1-A_{ij}}\right\}. \tag{17}$$

We can rewrite this component of the log-likelihood function using matrix notation as $\hat{\mathscr{L}}(A,S) = Tr[S^{\mathrm{T}}TS]$ where S is the cluster membership matrix defined earlier and T is the $|V|\times|V|$ matrix whose elements are given by

$$T_{ij} = \ln\frac{A_{ij}}{1-A_{i,j}}. \tag{18}$$

Since the trace of a matrix product is invariant under the cyclic permutation of the matrices, we have that $\hat{\mathscr{L}}(A,S) = Tr[SS^{\mathrm{T}}T]$. From the previous section of this paper, we know that the matrix SS^{T} is related to the updated link-weight matrix by the equation $SS^{\mathrm{T}} = |\Omega|\hat{A}$. Hence, we can write $\hat{\mathscr{L}}(A,S) = |\Omega|Tr[\hat{A}T]$. As shown by Scott and Longuet-Higgins [1] in a study of correspondence matching, this quantity may be maximised by performing an eigendecomposition on the matrix T and setting the columns of $\hat{A}$ equal to the eigenvectors of T. It has been shown by Dieci [25] that the eigenvectors of the matrices A and $\ln A$, have identical directions. This suggests a means by which we might refine our estimate of the link-weight matrix.

Returning to our analysis, we note that the eigenvector expansion of the matrix A is

$$A = \sum_{k=1}^{|V|}\lambda_k\underline{x}_k\underline{x}_k^{\mathrm{T}}. \tag{19}$$

This matrix may be approximated by the same-sign eigenvectors, i.e.

$$A \simeq \sum_{\omega\in\Omega}\lambda_\omega\underline{x}_\omega^*\underline{x}_\omega^{*\mathrm{T}}. \tag{20}$$

We can exploit this property to develop a means of refining the structure of the updated link-weight matrix, with the aim of improving its block structure. To commence, let the rank-one matrix $\hat{A}_\omega = \underline{s}_\omega\underline{s}_\omega^{\mathrm{T}}$. represent the component of the updated link-weight matrix which results from the cluster of nodes indexed ω. We can write

$$A = \sum_{\omega\in\Omega}\hat{A}_\omega + E, \tag{21}$$

where E is an error matrix. Since $\hat{A}_\omega$ is rank one, it has only one non-zero eigenvalue λ_ω^*. Let the eigenvector corresponding to this eigenvalue be ϕ_ω^*. With this notation, we can write

$$A = \sum_{\omega\in\Omega}\lambda_\omega^*\phi_\omega^*(\phi_\omega^*)^{\mathrm{T}} + E. \tag{22}$$

Hence, provided that the error matrix E is small, then we can approximate the updated link-weight matrix $\hat{A}$ by the matrix

$$A^* = \sum_{\omega\in\Omega}\frac{\lambda_\omega^*}{|\Omega|}\phi_\omega^*(\phi_\omega^*)^{\mathrm{T}}. \tag{23}$$

Thus, we construct a new link-weight matrix from the leading eigenvectors of the cluster link-weight matrices $\hat{A}_\omega = \underline{s}_\omega\underline{s}_\omega^{\mathrm{T}}$. The eigenvalues and eigenvectors of the new link-weight matrix are the leading eigenvalues and eigenvectors of the individual cluster adjacency matrices. We refer to this process as modal sharpening. The effect is to impose a strong block structuring on the link-weight matrix. For each cluster or mode of the link-weight matrix, we effectively partition the nodes into foreground and background. The background nodes for each cluster are then removed from further consideration. This can be viewed as a form of noise removal.

The modal decomposition of the link-weight matrix also suggests an initialisation. We assign cluster-membership

probabilities so that they are close to the eigenmodes of the raw adjacency matrix. To do this we use the same sign eigenvectors by setting

$$s_{iw} = \frac{|\underline{x}^*_\omega(i)|}{\sum_{i\in V_\omega} |\underline{x}^*_\omega(i)|}. \tag{24}$$

Since each node is associated with a unique cluster, this means that the updated affinity matrix is composed of non-overlapping blocks. Moreover, the link-weights are guaranteed to be in the interval $[0,1]$. Finally, it is important to note that the updating of the link-weights is a unique feature of our algorithm which distinguishes it from the pairwise clustering methods of Hoffman and Buhmann [19] and Shi and Malik [4].

3.3.3. Updating cluster membership variables

We can repeat the gradient-based analysis of the log-likelihood function to develop update equations for the cluster-membership variables. Recall that we have relaxed the condition $s_{j\omega} \in \{0,1\}$ so that the cluster membership indicators $s_{j\omega}$ instead belong to the interval $[0,1]$. As a result, we can compute the derivatives of the expected log-likelihood function with respect to the cluster-membership variable

$$\frac{\partial \mathscr{L}(A,S)}{\partial s_{i\omega}} = \sum_{j\in V} s_{j\omega} \ln \frac{A_{ij}}{1-A_{ij}}. \tag{25}$$

Since the associated saddle-point equations are not tractable in closed form, we use the soft-assign ansatz of Bridle [26] to update the cluster membership assignment variables. This is a form of naive mean field theory [27]. According to mean field theory the cluster memberships should be updated by replacing them with their expected values [19]. Rather than performing the detailed expectation analysis, soft-assign allows the cluster memberships to be approximated by exponentiating the partial derivatives of the expected log-likelihood function. The updated cluster memberships are given by

$$\hat{s}_{i\omega} = \frac{\exp[\partial \mathscr{L}(A,S)/\partial s_{i\omega}]}{\sum_{i\in V} \exp[\partial \mathscr{L}(A,S)/\partial s_{i\omega}]} = \frac{\exp\left[\sum_{j\in V} s_{j\omega} \ln A_{i,j}/(1-A_{ij})\right]}{\sum_{i\in V} \exp\left[\sum_{j\in V} s_{j\omega} \ln A_{i,j}/(1-A_{ij})\right]}. \tag{26}$$

After simplifying the argument of the exponential, the update formula reduces to

$$\hat{s}_{i\omega} = \frac{\prod_{j\in V} \{A_{i,j}/(1-A_{ij})\}^{s_{j\omega}}}{\sum_{i\in V} \prod_{j\in V} \{A_{ij}/(1-A_{ij})\}^{s_{j\omega}}}. \tag{27}$$

It is worth pausing to consider the structure of this update equation. First, the updated link weights are an exponential function of the current ones. Second, the exponential constant is greater than unity, i.e. there is re-enforcement of the cluster memberships, provided that $A_{i,j} > \frac{1}{4}$.

We can take the analysis of the cluster membership update one step further and establish a link with the eigenvectors of the updated adjacency matrix. To this end we make use of the matrix T defined in Section 3.3.2. We turn our attention to the argument of the exponential appearing in Eq. (26) and write

$$\sum_{j\in V} s_{j\omega} \ln \frac{A_{i,j}}{1-A_{i,j}} = (T\underline{s}_\omega)_i. \tag{28}$$

In other words, the argument of the exponential is simply the ith component of the vector obtained by the matrix multiplication $T\underline{s}_\omega$.

Next, consider the case when the vector $\underline{s}_\omega$ is an eigenvector of the matrix T. The eigenvector equation for the matrix T is $T\underline{z}_\omega = \hat{\lambda}_\omega \underline{z}_\omega$, where $\hat{\lambda}_\omega$ is the ωth eigenvalue and $\underline{z}_\omega$ is the corresponding eigenvector. Hence, when the vector of cluster memberships $\underline{s}_\omega$ is an eigenvector of T, then we can write $(T\underline{s}_\omega)_i = \underline{z}_\omega(i)$, where $\underline{z}_\omega(i)$ is the ith component of the vector $\underline{z}_\omega$. If this is the case, then we can identify the pairwise clusters with the eigenmodes of T, and the update equation becomes

$$\hat{s}_{i\omega} = \frac{\exp[\hat{\lambda}_\omega z_\omega(i)]}{\sum_{i\in V} \exp[\hat{\lambda}_\omega z_\omega(i)]} = \frac{\tilde{\lambda}_\omega^{z_\omega(i)}}{\sum_{i\in V} \tilde{\lambda}_\omega^{z_\omega(i)}}, \tag{29}$$

where $\hat{\lambda}_\omega = \ln \tilde{\lambda}_\omega$. This update process becomes particularly simple when it is applied to the adjacency matrix obtained by modal sharpening. Let the elements of matrix T^* be given by $T^*_{i,j} = \ln(A^*_{i,j}/(1-A^*_{i,j}))$. Since the logarithm of matrix T is a polynomial in T (i.e. a primary matrix function) [28] and matrix T is positive, definite and invertible, the directions of the eigenvectors of the matrices T and $\ln T$ are identical [25]. Hence, we can compute the eigenvectors of T^* by the eigenvectors of T. As a result, the updated cluster membership variables can be computed directly from the eigenvectors of the matrix T^*, i.e. the leading eigenvector $\tilde{\phi}_\omega$. Thus, we can write

$$\hat{s}_{i\omega} = \frac{\tilde{\lambda}_\omega^{\tilde{\phi}_\omega(i)}}{\sum_{i\in V} \tilde{\lambda}_\omega^{\tilde{\phi}_\omega(i)}}. \tag{30}$$

In this way, by computing the eigenmodes of the matrix T^*, we can update the individual cluster membership indicators.

3.4. Algorithm description

We use the update steps developed in Sections 3.3.1, 3.3.2 and 3.3.3 to develop an iterative grouping algorithm. The steps of the algorithm are as follows:

Step 0: The algorithm is initialised using the initial link weight matrix A. This is computed from raw image data and is domain specific. Some examples of how this is done are provided later on in Section 6 for line-segment grouping and in Section 7 for motion segmentation.

Step 1: The same-sign eigenvectors are extracted from the current link-weight matrix A. These are used to compute the cluster-membership matrix S using Eq. (30). The number of same-sign eigenvectors determines the number of clusters for the current iteration. This number may vary from iteration to iteration. In our experiments, the complexity of computing the first eigenvector using the power method was on average $5.8N^2$, where N is the order of matrix A.

Step 2: For each cluster we compute the link-weight matrix $A_\omega = \underline{s}_\omega \underline{s}_\omega^{\mathrm{T}}$. We perform an eigen-decomposition on each cluster link-weight matrix to extract the non-zero eigenvalue λ_ω^* and the corresponding eigenvector ϕ_ω^*. Since the matrix A_ω is rank one since it is defined as the product of two vectors, the computation of the first eigenvector can be regarded for computational purposes as a normalisation of the vectors $\underline{s}_\omega$. Therefore, the complexity can be reduced to approximately $3N$ for each cluster.

Step 3: We perform modal sharpening by applying Eq. (23) to the leading (i.e. sole non-zero) eigenvalues and the corresponding eigenvectors of the cluster link-weight matrices $\hat{A}_\omega$. The resulting revised link-weight matrix is A^*. The complexity of computing matrix A^* is N^2.

Step 4: An updated matrix of cluster membership variables $\hat{S}$ is computed. This is done by applying Eq. (27) to the revised link-weight matrix obtained by modal sharpening, i.e. A^*, and the current cluster membership matrix S. Making use of the fact that the denominator of Eq. (26) is the sum of the quantities in the numerator for every $\hat{s}_{i\omega}$, the complexity of this step can be reduced to approximately $5N$.

Step 5: The updated cluster membership matrix $\hat{S}$ is used to compute the updated link-weight matrix $\hat{A} = (1/|\Omega|)SS^{\mathrm{T}}$. This revised link-weight matrix is passed to Step 1. The average complexity of this step in our experiments was $2.4N^2$.

Steps 1–5 are iterated in sequence until convergence is reached. In our experiments, the algorithm converged in an average of 3 iterations, where each iteration had in average a complexity of $9.2N^2 + 8N$.

With the algorithm description at hand, we illustrate the behaviour of the method by showing the evolution of the cluster-membership variables and the link-weight matrix over the steps described above. To this end, we have generated a set of four point-patterns consisting of 190 points corresponding to two clusters. The first cluster consists of 50 points distributed normally around a fixed centre point. The second cluster is an annulus consisting of 150 normally distributed points. For both clusters, the variance of the Gaussian kernel was set to 1.5.

We assign the linking probability between the point indexed i and the point indexed j using the exponential distribution $A_{ij} = \exp(-kD_{ij}^2)$ where D_{ij}^2 is the squared Euclidean distance on the x–y plane and $k \in (0, \infty]$ is a constant.

We focus our attention on the evolution of the link-weights and the cluster-membership variables corresponding to the first cluster. In the top row of Fig. 1

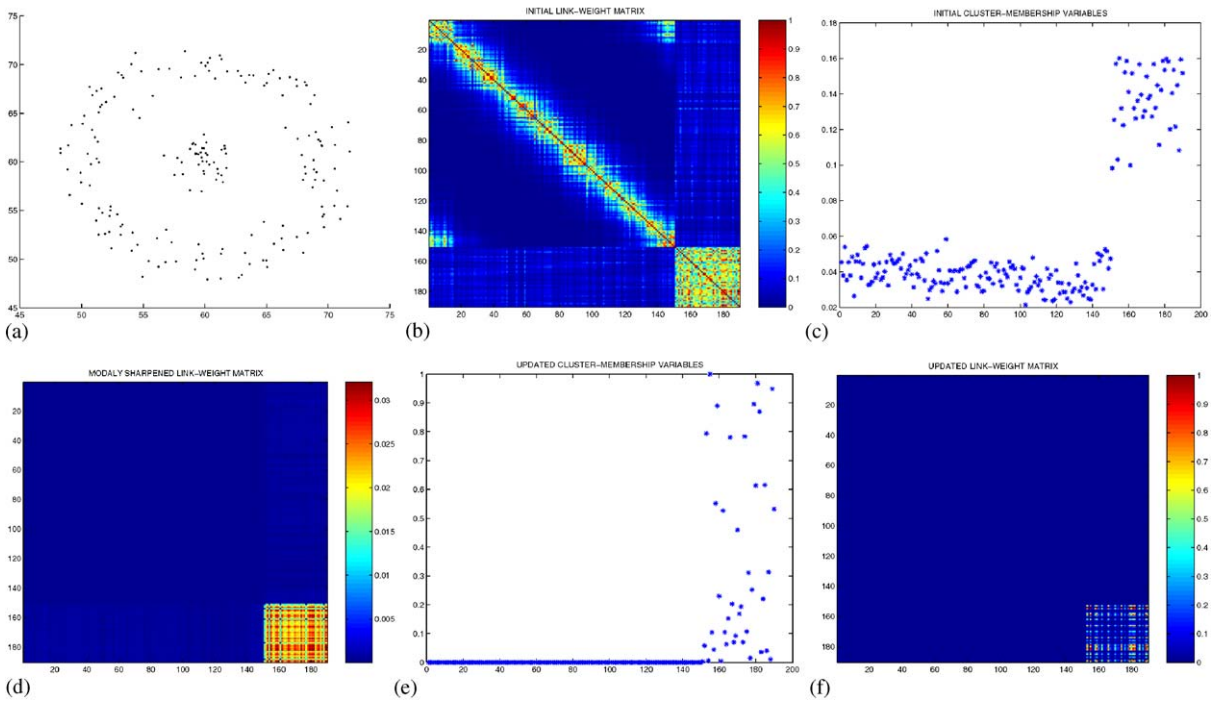

Fig. 1. (a) Example of the point-patterns under study; (b) Initial adjacency matrix ($k = 0.275$); (c) Leading eigenvector; (d) The matrix A^*; (e) Updated cluster membership variables; (f) The matrix $\hat{A}$ (see text for details).

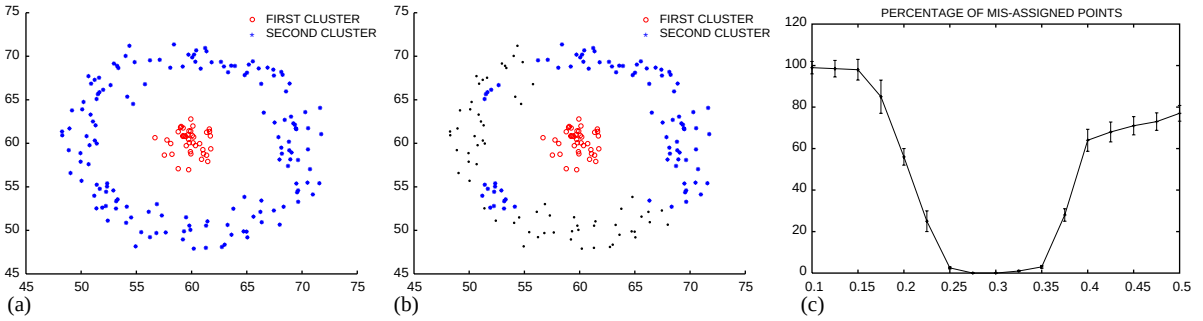

Fig. 2. Clustering results with (a) $k = 0.3$ and (b) $k = 0.4$; (c) fraction of mis-assigned points as a function of k.

we show an example of the point-patterns under study, the initial affinity matrix A and the corresponding leading eigenvector. From the affinity matrix, it is clear that there are two clusters. The strongest cluster is in the bottom right-hand corner, and corresponds to the cluster of points in the centre. The weaker cluster is in the top left-hand corner and corresponds to the annulus of surrounding points. In the bottom row of Fig. 1 from left-to-right we show the matrix A^*, the updated cluster-membership variables and the matrix $\hat{A}$. There are three important effects of the algorithm steps on the cluster variables. First, the first eigenvector of the adjacency matrix A presents two well defined groups of coefficient-values that correspond to the two clusters in the point-pattern. Second, when the matrix A^* is computed a strong block structure is imposed by the modal sharpening step. Finally, the soft-assign step nulls all the cluster-membership variables that correspond to the elements in the second cluster (top left-hand corner of A).

Next, we study the effect of varying k in the output of our clustering algorithm. To perform this study, we have computed the adjacency matrices of our four point-patterns varying k from 0.1 to 0.5. In Fig. 2 we show the clustering results with $k = 0.3, 0.4$ and the fraction of points that are mis-assigned by the clustering algorithm. From Fig. 2c, we can conclude that the output is stable and reliable over the $[0.25, 0.35]$ interval, with $k = 0.3$ as its optimum.

At this point, we pause to stress that although this iterative process clearly has features which are reminiscent of the EM algorithm, there are important differences. These mainly stem from our use of the modal sharpening process to improve the block and cluster structure of the link-weight matrix. We have recently reported the use of an EM algorithm based on mixtures of Bernoulli distributions. We will compare the method described in this paper with this EM algorithm in our experimental evaluation. The algorithm commences from the initial set of cluster memberships defined by the eigenmodes of the raw affinity matrix A. In the E or expectation step we compute a matrix of a posteriori cluster membership probabilities Q. This matrix is found by taking the expectation of the cluster-membership matrix i.e. $Q = E(S)$. In the M or maximisation step we perform two updates. First, we compute the updated link-weight matrix using the formula $\hat{A}=E(SS^{\mathrm{T}})$. Second, we compute a revised matrix of cluster-membership indicators $\hat{S}$ using a variant of the soft-assign method outlined in Section 3.3. Hence, the main differences are that the number of clusters is set at the outset of the algorithm, and that there is no modal sharpening of the link-weight matrix.

4. Convergence analysis

In this section, we provide some analysis of the convergence properties of the new clustering algorithm. We are interested in the relationship between this modal analysis and the updated cluster membership variables. Using the shorthand in Eq. (18) and substituting in Eq. (14) the update formulae for the link-weight matrix and the cluster membership indicators given in Eqs. (15) and (29), it is a straightforward matter to show that the corresponding updated log-likelihood function is given by

$$\mathcal{L}(A,S) = \sum_{\omega\in\Omega}\sum_{(i,j)\in\Phi}\left\{T_{ij}\frac{\tilde{\lambda}_\omega^{z_\omega(i)+z_\omega(j)}}{\sum_{i'\in V}\tilde{\lambda}_\omega^{z_\omega(i')}\sum_{j'\in V}\tilde{\lambda}_\omega^{z_\omega(j')}} + \ln(1-A_{ij})\right\}. \tag{31}$$

We would like to understand the conditions under which the likelihood is maximised by the update process. We hence compute the partial derivative of $\mathcal{L}(A,S)$ with respect to $\tilde{\lambda}_\omega$. After collecting terms and some algebra we find

$$\frac{\partial\mathcal{L}(A,S)}{\partial\tilde{\lambda}_\omega} = \sum_{\omega\in\Omega}\sum_{(i,j)\in\Phi}\left\{\frac{T_{ij}\tilde{\lambda}_\omega^{z_\omega(i)+z_\omega(j)}}{\tilde{\lambda}_\omega\left(\sum_{i'\in V}\tilde{\lambda}_\omega^{z_\omega(i')}\right)^2}\left(z_\omega(i)+z_\omega(j)-2\times\frac{\sum_{i'\in V}z_\omega(i')\tilde{\lambda}_\omega^{z_\omega(i')}}{\sum_{i'\in V}\tilde{\lambda}_\omega^{z_\omega(i')}}\right)\right\}. \tag{32}$$

Since the natural logarithm function is strictly increasing, the maximum of the likelihood will occur at the same point as the maximum of the log-likelihood function. Hence, we set the partial derivative to zero. This condition is satisfied when

$$z_\omega(i) + z_\omega(j) = 2\,\frac{\sum_{i'\in V} z_\omega(i')\tilde{\lambda}_\omega^{z_\omega(i')}}{\sum_{i'\in V}\tilde{\lambda}_\omega^{z_\omega(i')}}. \tag{33}$$

Unfortunately, this condition is not always guaranteed to exist. However, from Eq. (32) we can conclude that the following will always be the best approximation:

$$\tilde{\lambda}_\omega^{z_\omega(i)+z_\omega(j)} \ll \tilde{\lambda}_\omega \left(\sum_{i'\in V}\tilde{\lambda}_\omega^{z_\omega(i')}\right)^2. \tag{34}$$

If T^* is a non-negative irreducible symmetric matrix then the coefficients of the leading eigenvector $\underline{z}_*$ associated with the eigenvalue $\hat{\lambda}_\omega = \ln\tilde{\lambda}_\omega$ are each positive [29]. As a result, the quantity $\sum_{i'\in V}\tilde{\lambda}_\omega^{z_\omega(i')}$ will be maximised when $\hat{\lambda}_\omega$ is maximum. Hence, $\mathscr{L}(A,S)$ will be maximised by the first (maximum) eigenvalue of T^*.

Further, since the matrix T^* is symmetric and non-negative, the leading eigenvalue of every principal minor of matrix T^* does not exceed the value of its maximal eigenvalue [30]. From this monotonicity principle, we can conclude that the maximal eigenvalue of matrix T^* can be used for measuring the degree of convergence. To do this, it is enough to compare the maximal eigenvalue of the matrix T^* when it is passed on to Step 1 from Step 5 (see algorithm description). If the leading eigenvalue starts increasing over iteration number, then convergence has been reached.

5. 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.

5.1. 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. This problem has attracted considerable interest in the literature. For instance, Heitger and von der Heydt [31] have shown how to model the line extension field using directional filters whose shapes are motivated by studies of the visual field of monkeys. Parent and Zucker [23] use edge co-circularity compatibility. Williams et al. [32] 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.

Here we follow the former approach and to provide an initial characterisation of the matrix of link-weights using a grouping field. 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 the polar angle θ_{ij} of the line $\Gamma_{i,j}$ with respect to the base-line Λ_i and its length ρ_{ij}. 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 [31] 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 $\xi_{i,j} = k\cos^2\theta_{i,j}$.

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

$$k = \frac{\rho_{i,j}}{\hat{\rho}_{i,j}\cos^2\theta_{i,j}}. \tag{35}$$

The geometry of the lines and their relationship to the interpolating polar lemniscate is illustrated in Fig. 3a. It is important to note that the polar angle is defined over the interval $\theta_{ij} \in (-\pi/2, \pi/2]$ 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 distribution

$$A_{ij} = \exp[-\mu k], \tag{36}$$

where μ is a positive 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 end-points 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 colinear or parallel. The linking probability is small when either the relative separation of the end-points 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.

5.2. Experiments

In this section, we provide some experiments to illustrate the utility of our new perceptual grouping method when applied to line-linking. There are two aspects to this study.

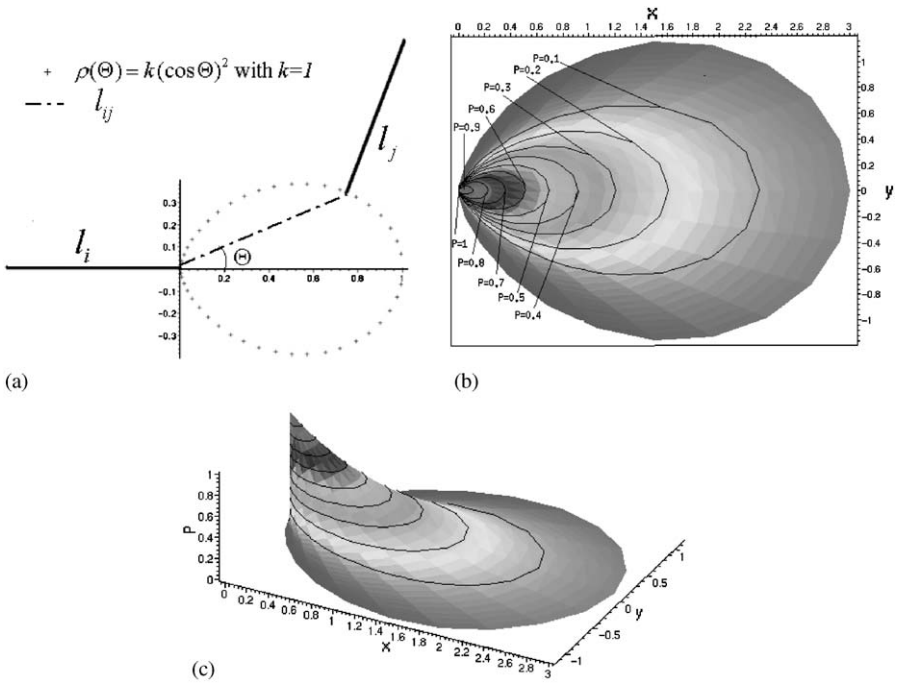

Fig. 3. (a) Geometric meaning of the parameters used to obtain P_{ij}; (b) Plot showing the level curves; (c) 3D plot showing P_{ij} on the z-axis.

We commence by providing some examples for synthetic images. Here we investigate the sensitivity of the method to clutter and compare it with an eigendecomposition method. The second aspect of our study focuses on real world images with known ground-truth.

In our experiments, we provide comparison with three different algorithms. The first of these is the EM algorithm described in Ref. [24], which uses a mixture of Bernoulli distributions. This algorithm does not however use modal decomposition of the link-weight matrix. The second method is that of Sarkar and Boyer [2], which we have outlined briefly in Section 3. Thirdly, there is the normalised cut (recursive bisection) method of Shi and Malik [4].

5.2.1. Synthetic images

Our first experiment concerns a hexagonal arrangement of lines to which increasing numbers of randomly distributed distractors have been added. The positions, orientations and lengths of the distractors have been drawn from uniform distributions. The images used in this study are shown in the first column of Fig. 4. The distractor density increases from top to bottom in the first column of the figure. From the arrangement of lines in each panel of the figure, we compute the link-weight matrix using Eq. (36).

In the second column of the figure, we show the results obtained using the Sarkar and Boyer method. The third column shows the result obtained using the standard EM algorithm. In the fourth column, we show the results obtained using the normalised cuts method. Finally, the fifth column shows the results obtained using our new method.

To display the results of our method we first label the lines according to the cluster associated with the largest membership variable. For the line indexed i, the cluster-label is $\theta_i = \arg\max_{\omega\in\Omega} \hat{s}_{i\omega}$. Next, we identify the cluster which contains the largest number of lines from the hexagonal pattern. Suppose that the index of this cluster is denoted by ω_p. The lines displayed are those belonging to the set $\Theta_{\omega_p} = \{\theta_i | \theta_i = \omega_p\}$.

There are a number of conclusions that can be drawn from these examples. First, the quality of the results obtained increases as we move from left-to-right across the figure. In the case of the Boyer and Sarkar method, little of the distractor structure is removed. In the case of the EM algorithm and the normalised cuts method, most of the background is removed, but a few distractors remain attached to the hexagonal pattern of lines.

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. Figs. 5a and b, respectively, show the fraction of nodes merged with the foreground and the fraction of nodes which leak into the background as a function

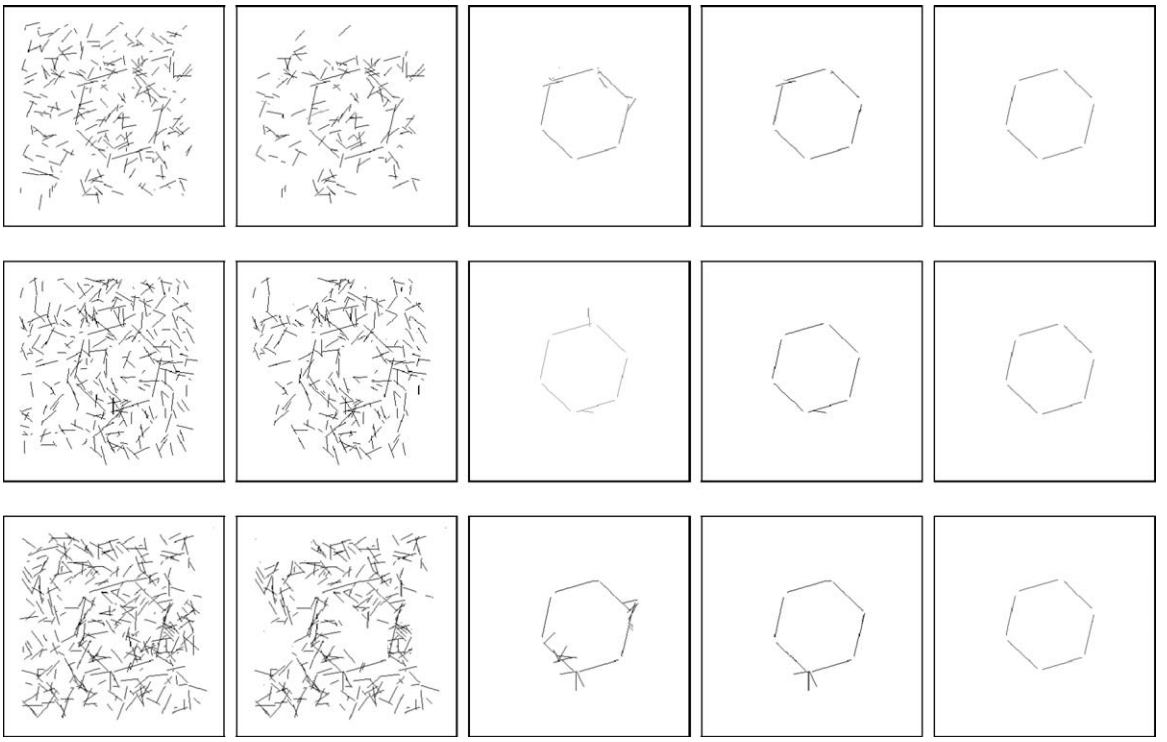

Fig. 4. Left-hand column: patterns containing 250, 300 and 350 randomly positioned background lines; each subsequent column shows the result obtained with the Sarkar and Boyer algorithm (second column), the results when a standard EM algorithm is used (third column), the cluster memberships obtained using the normalized cut (fourth column) and the cluster-memberships obtained using our new method (last column) for each of the images shown in the first column.

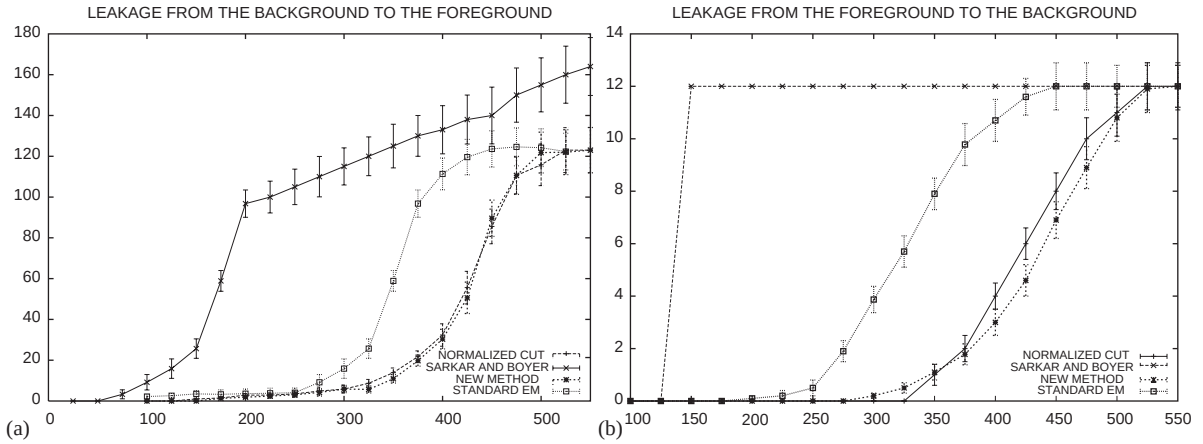

Fig. 5. Comparison between the non-iterative eigendecomposition approach and the two variants of the EM-like algorithm.

of the number of distractors. The four curves shown in each plot are for the non-iterative eigendecomposition method of Sarkar and Boyer, the EM algorithm, the Shi and Malik normalised cuts method, and for the new method described in this paper.

Next, we turn our attention to the fraction of foreground lines which leak into the background (i.e. those which are erroneously identified as distractors). From Fig. 5b a similar pattern emerges to that in Figure 5a. In other words, the worst performance is delivered by the Sarkar and Boyer

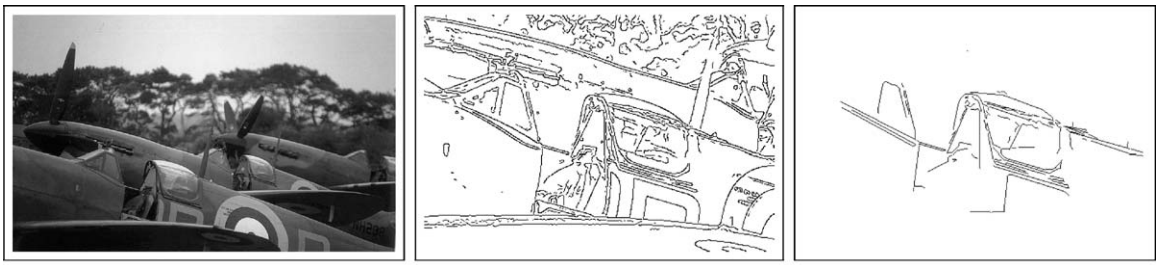

Fig. 6. Real-world images: (a) raw image, (b) results of Canny edge detection and (c) the result of applying the eigendecomposition algorithm.

method [2], and the EM algorithm gives intermediate performance. However, now the new method gives a margin of improvement over the Shi and Malik normalised cuts method.

Finally, we present results on a real-world image in Fig. 6. The edges shown in Fig. 6b have been extracted from the raw image using the Canny edge-detector. Straight-line segments have been extracted using the method of Yin [33]. The resulting groupings obtained with our new method are shown in Fig. 6c.

6. Motion segmentation

The second application of our pairwise clustering method focuses on the segmentation of independently moving objects from image sequences. The motion vectors used in our analysis have been computed using a single resolution block matching algorithm [34]. The method measures the similarity of motion blocks using spatial correlation and uses predictive search to efficiently compute block-correspondences in different frames. The block matching algorithm assumes that the translational motion from frame to frame is constant. The current frame is divided into blocks that will be compared with the next frame in order to find the displaced coordinates of the corresponding block within the search area of the reference frame. Since the computational complexity is much lower than the optical flow equation and the pel-recursive methods, block matching has been widely adopted as a standard for video coding and hence it provides a good starting point.

However, the drawback of the single resolution block-matching scheme is that while the high resolution field of motion vectors obtained with small block sizes captures fine detail, it is susceptible to noise. At low resolution, i.e. for large block sizes, the field of motion vectors is less noisy but the fine structure is lost. To strike a compromise between low-resolution noise suppression and high resolution recovery of fine detail, there have been several attempts to develop multi-resolution block matching algorithms. These methods have provided good predictive performance and also improvements in speed. However, one of the major

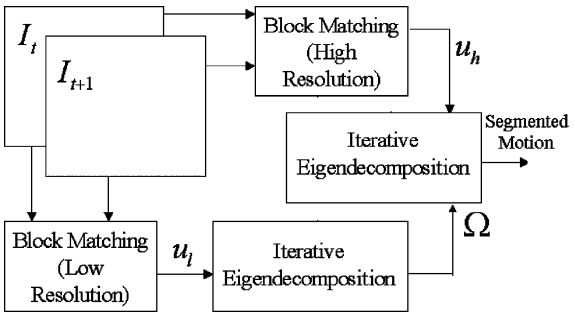

Fig. 7. Motion segmentation system.

problems with the multi-resolution block matching method is that random motions can have a significant degradational effect on the estimated motion field. For these reasons, we have used a single high-resolution block matching algorithm to estimate the raw motion field. This potentially noisy information is refined in the motion segmentation step, where we exploit hierarchical information.

We pose the problem of grouping motion blocks into coherent moving objects as that of finding pairwise clusters. The 2D velocity vectors for the extracted motion blocks are characterised using a matrix of pairwise similarity weights. Suppose that $\hat{\mathbf{n}}_i$ and $\hat{\mathbf{n}}_j$ are the unit motion vectors for the blocks indexed i and j. The elements of the initial link-weight matrix are given by

$$A_{i,j}^{(0)} = \begin{cases} \frac{1}{2}(1 + \hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j) & \text{if } i \neq j, \\ 0 & \text{otherwise.} \end{cases} \tag{37}$$

6.1. Hierarchical motion segmentation

As mentioned earlier, we use a single-level high-resolution block-matching method to estimate the motion field. The resulting field of motion vectors is therefore likely to be noisy. To control the effects of motion-vector noise, we have developed a multi-resolution extension to the clustering approach described above.

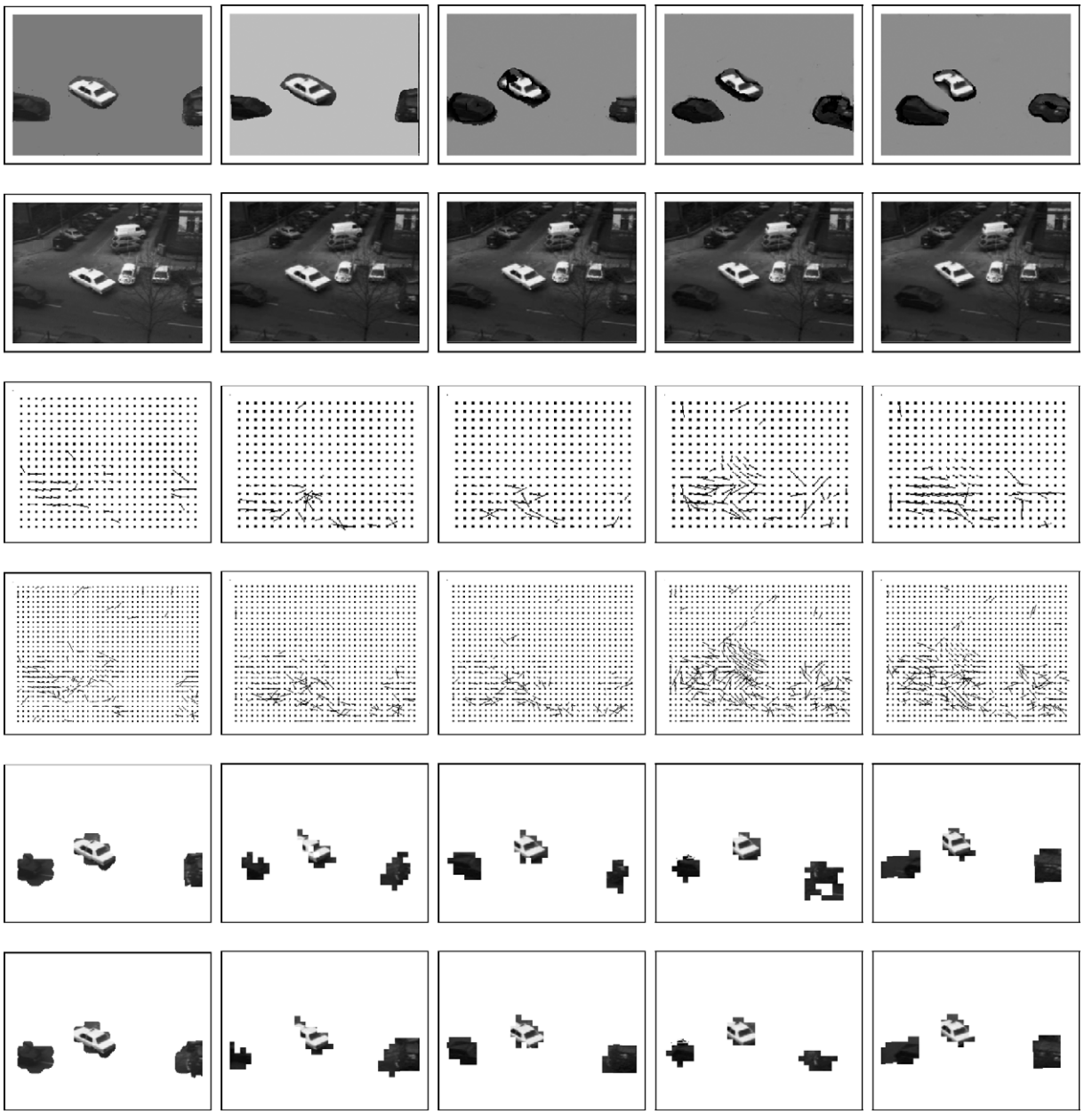

Fig. 8. Top row: ground truth for the 1st, 4th, 8th, 12th and 16th frame of the "Hamburg Taxi" sequence; second row: original frames; third and fourth rows: low and high resolution motion fields; fifth row: Final motion segmentation obtained using the normalised cut; bottom row: motion segmentation using our new method.

The adopted approach is as follows:

- We obtain the a high resolution field of motion vectors U_H using blocks of size k-pixels and a low-resolution motion field U_L using blocks of size $2k$ pixels.
- We apply our clustering algorithm to the low resolution motion field U_L. We note the number of clusters N_L detected.
- We make a second application of our clustering algorithm to the high-resolution motion field U_H. Here we select only the first N_L eigenvalues of the motion-vector similarity matrix as cluster centres.

In this way, we successively perform the motion estimation at low and high resolution. The number of clusters detected at low resolution is used to constrain the number

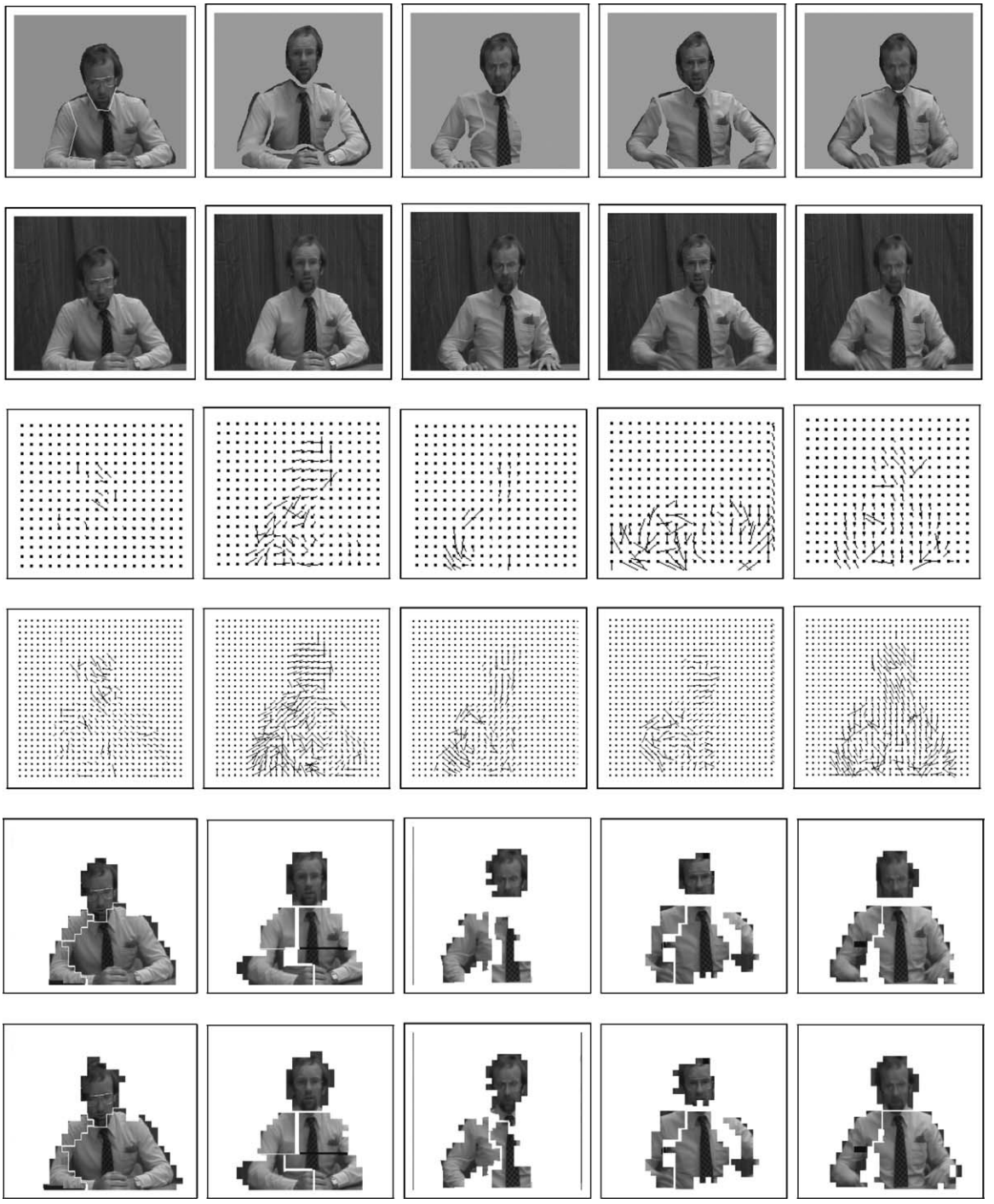

Fig. 9. Top row: ground truth for the 1st, 4th, 8th, 12th and 16th frame of the "Trevor White" sequence; second row: original frames; third and fourth rows: low and high resolution motion fields; fifth row: motion segmentation obtained using the normalised cut; bottom row: motion segmentation obtained using our new method.

of permissible high resolution clusters. This allows the high-resolution clustering process to deal with fine detail motion fields without succumbing to noise. There is scope to extend the method and develop a pyramidal segmentation strategy. The structure of the hierarchical system can be seen in Fig. 7.

6.2. Motion experiments

We have conducted experiments on motion sequences with known ground truth. In Fig. 8, we show some results obtained with five frames of the well-known "Hamburg Taxi" sequence. The top row shows the hand-labelled ground-truth segmentation for the motion sequence. The second row shows the corresponding image frames from the motion sequence. In the third and fourth rows we, respectively, show the low and high resolution block motion vectors. At low resolution we use 16×16 pixel blocks to perform motion correspondence and compute the motion vectors; for the high resolution motion field the block size is 8×8 pixels. The fifth row in the figure shows the moving objects segmented from the motion field using the normalised cut method. The sixth row shows the motion segmentation obtained using the new method described in this paper. Turning our attention to the results delivered by our new method (i.e. the sixth row of the figure) in each frame there are three clusters which match closely to the ground truth data shown. In fact, the three different clusters correspond to distinct moving vehicles in the sequence. These clusters again match closely to the ground-truth data. The results obtained using the normalised cut are good, but some of regions are slightly undersegmented.

Fig. 9 repeats these experiments for the "Trevor White" sequence. The sequence of rows is the same as in Fig. 3. Here the block sizes are, respectively, 24×24 and 12×12 pixels. There are three motion clusters which correspond to the head, the right arm, and the chest plus left arm. These clusters again match closely to the ground-truth data. The normalised cuts method again under-segments the motion regions when compared with our new method.

In Table 1 we provide a more quantitative analysis of these results. The table lists the fraction of the pixels in each region of the ground truth data which are mis-assigned by the clustering algorithm. There are different columns in the table for the normalised cuts method and our new method. Turning our attention to our new method, the best results are obtained for the chest-region, the taxi and the far-left car, where the error rate is a few percent. For the far-right car and the head of "Trevor White", the error rates are about 10%. The problems with the far-right car probably relate to the fact that it is close to the periphery of the image. The normalised cuts method consistently gives error rates which are about 2% worse than our new method.

Table 1
Error percentage for the two image sequences

Sequence	Cluster	% Error (Normalised cut) (%)	% Error (our approach) (%)
Trevor White	Right arm	7.2	8
Trevor White	Chest	7.4	6
Trevor White	Head	13.6	12
Ham. Taxi	Taxi	6	4
Ham. Taxi	Far Left Car	4	3
Ham. Taxi	Far Right Car	14	10

7. Conclusions

In this paper, we have developed a maximum likelihood framework for pairwise clustering. The method commences from a specification of the pairwise clustering problem in terms of a matrix of link-weights and a set of cluster membership indicators. The likelihood function underpinning our method is developed under the assumption that the cluster membership indicators are random variables which are generated by Bernoulli trials. The parameter of the Bernoulli trials are the link-weights. Based on this model, we develop an iterative process for updating the link-weights and the cluster membership indicators in interleaved steps, reminiscent of the EM algorithm. We show that the log-likelihood function is maximised by the leading eigenvector of the link-weight matrix. We apply the resulting pairwise clustering process to a number of image segmentation and grouping problems.

There are a number of ways in which the work presented in this paper can be extended and improved. First, we intend to investigate alternatives to the Bernoulli model of the clustering process. For instance, a different choice of distribution may provide us with a means of locating spanning trees or relational skeletons in the raw data. Second, our present method does not facilitate data-closeness between the final arrangement of clusters and the raw data. In fact it can be viewed as a type of relaxation process which applies contiguity constraints to the blacks of the link-weight matrix. Our future work will therefore focus on developing a clustering process which minimises the Kullback–Leibler divergence between the initial matrix of link-weights and the final arrangement of pairwise clusters.

References

[1] G.L. Scott, H.C. Longuet-Higgins, Feature grouping by relocalisation of eigenvectors of the proximity matrix, in: British Machine Vision Conference, 1990, pp. 103–108.

[2] S. Sarkar, K.L. Boyer, Quantitative measures of change based on feature organization: eigenvalues and eigenvectors, Comput. Vision Image Understanding 71 (1) (1998) 110–136.

[3] P. Perona, W.T. Freeman, Factorization approach to grouping, in: Proceedings of ECCV, Freiburg, Germany, 1998, pp. 655–670.
[4] J. Shi, J. Malik, Normalized cuts and image segmentations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 731–737.
[5] Y. Weiss, Segmentation using eigenvectors: a unifying view, in: IEEE International Conference on Computer Vision, 1999, pp. 975–982.
[6] M. Meilă, J. Shi, Learning segmentation by random walks, in: Advances in Neural Information Processing Systems, Vol. 13, MIT Press, Cambridge, MA, 2001, pp. 873–879.
[7] N. Tishby, N. Slonim, Data clustering by markovian relaxation and the information bottleneck method, in: T.K. Leen, T.G. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems, Vol. 13, MIT Press, Cambridge, MA, 2001, pp. 640–646.
[8] C. Fowlkes, S. Belongie, J. Malik, Efficient spatiotemporal grouping using the Nyström method, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, 2001, pp. I:231–238.
[9] S. Belongie, C. Fowlkes, F. Chung, J. Malik, Spectral partitioning with indefinite kernels using the Nyström extension, in: Proceedings of the European Conference on Computer Vision, Copenhagen, Denmark, 2002, pp. III: 531–542.
[10] P. Soundararajan, S. Sarkar, Investigation of measures for grouping by graph partitioning, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2001, pp. 239–246.
[11] B. Mohar, Some applications of Laplace eigenvalues of graphs, Graph Symmetry: Algebraic Methods and Applications, Kluwer Academic Publishers, Dordrecht, 2000, pp. 225–275.
[12] R. Kannan, S. Vampala, A. Vetta, On clusterings: good, bad and spectral, in: Proceedings of the 41st Symposium on the Foundations of Computer Science, Redondo Beach, California, USA, 2000, pp. 367–377.
[13] W. Dickson, Feature grouping in a hierarchical probabilistic network, Image Vision Comput. 9 (1) (1991) 51–57.
[14] I.J. Cox, J.M. Rehg, S. Hingorani, A bayesian multiple-hypothesis approach to edge grouping and contour segmentation, Int. J. Comput. Vision 11 (1) (1993) 5–24.
[15] J.A.F. Leite, E.R. Hancock, Iterative curve organisation with the em algorithm, Pattern Recognition Lett. 18 (1997) 143–155.
[16] R. Castaño, S. Hutchinson, A probabilistic approach to perceptual grouping, Comput. Vision Image Understanding 64 (3) (1996) 339–419.
[17] D. Crevier, A probabilistic method for extracting chains of collinear segments, Image Vision Comput. 76 (1) (1999) 36–53.
[18] A. Amir, M. Lindenbaum, A generic grouping algorithm and its quantitative analysis, Trans. Pattern Anal. Mach. Intell. 20 (2) (1998) 168–185.
[19] T. Hofmann, M. Buhmann, Pairwise data clustering by deterministic annealing, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1) (1997) 1–14.
[20] Y. Gdalyahu, D. Weinshall, M. Werman, A randomized algorithm for pairwise clustering, in: Advances in Neural Information Processing Systems, Vol. 11, MIT Press, Cambridge, MA, 1999, pp. 424–430.
[21] A. Shashua, S. Ullman, Structural saliency: the detection of globally salient structures using a locally connected network, in: Proceedings of the 2nd International Conference in Computer Vision, Tarpon Springs, Florida, USA, 1998, pp. 321–327.
[22] G. Guy, G. Medioni, Inferring global perceptual contours from local features, Int. J. Comput. Vision 20 (1/2) (1996) 113–133.
[23] P. Parent, S. Zucker, Trace inference, curvature consistency and curve detection, IEEE Trans. Pattern Anal. Mach. Intell. 11 (8) (1989) 823–839.
[24] A. Robles-Kelly, E.R. Hancock, An expectation–maximisation framework for segmentation and grouping, Image and Vision Comput. 20 (9–10) (2002) 725–738.
[25] L. Dieci, Considerations on computing real logarithms of matrices, hamiltonian logarithms and skew-symmetric logarithms, Linear Algebra its Appl. 244 (1996) 35–54.
[26] J.S. Bridle, Training stochastic model recognition algorithms can lead to maximum mutual information estimation of parameters, in: NIPS, Vol. 2, Denver, Colorado, USA: MIT, 1990, pp. 211–217.
[27] Z. Ghahramani, M. Jordan, Factorial hidden markov models, Mach. Learning 29 (2–3) (1997) 245–273.
[28] R.A. Horn, C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, Cambridge, MA, 1991.
[29] R.S. Varga, Matrix Iterative Analysis, 2nd Edition, Springer, Berlin, 2000.
[30] F.R. Gantmacher, in: Matrix Theory, Vol. 2, Chelsea, New York, 1971.
[31] F. Heitger, R. von der Heydt, A computational model of neural contour processing, in: IEEE Conference on Computer Vision and Pattern Recognition, 1993, pp. 32–40.
[32] L.R. Williams, D.W. Jacobs, Stochastic completion fields: a neural model of illusory contour shape and salience, Neural Comput. 9 (4) (1997) 837–858.
[33] Yin Peng-Yeng, Algorithms for straight line fitting using k-means, Pattern Recognition Lett. 19 (1998) 31–41.
[34] C.H. Hsieh, P.C. Lu, J.S. Shyn, E.H. Lu, Motion estimation algorithm using inter-block correlation, IEE Electron. Lett. 26 (5) (1990) 276–277.

About the Author—ANTONIO A. ROBLES-KELLY received his B. Eng. degree in Electronics and Communications from the Inst. Tecnológico y de Estudios Superiores de Monterrey with honours in 1998. In 2001, he visited the University of South Florida as part of the William Gibbs/Plessey Award to the best research proposal to visit an overseas research lab. The award is considered in consultation with GEC-Marconi Underwater Systems Ltd. He received his Ph.D. in Computer Science from the University of York in 2003. Currently, he is a Research Associate under the MathFit-EPSRC framework at York.

His research interests are in the areas of Computer Vision, Pattern Recognition and Computer Graphics. Along these lines, he has done work on segmentation and grouping, graph-matching, shape-from-X and reflectance models. He is also interested in the differential

structure of surfaces. His research has found applications in areas such as database organisation, 3D surface recovery and reflectance model approximation.

About the Author—EDWIN HANCOCK studied physics as an undergraduate at the University of Durham and graduated with honours in 1977. He remained at Durham to complete a Ph.D. in the area of high energy physics in 1981. Following this he worked for ten years as a researcher in the fields of high-energy nuclear physics and pattern recognition at the Rutherford-Appleton Laboratory (now the Central Research Laboratory of the Research Councils). During this period he also held adjunct teaching posts at the University of Surrey and the Open University. In 1991 he moved to the University of York as a lecturer in the Department of Computer Science. He was promoted to Senior Lecturer in 1997 and to Reader in 1998. In 1998 he was appointed to a Chair in Computer Vision.

Professor Hancock now leads a group of some 15 faculty, research staff and Ph.D. students working in the areas of computer vision and pattern recognition. His main research interests are in the use of optimisation and probabilistic methods for high and intermediate level vision. He is also interested in the methodology of structural and statistical pattern recognition. He is currently working on graph-matching, shape-from-X, image databases and statistical learning theory. His work has found applications in areas such as radar terrain analysis, seismic section analysis, remote sensing and medical imaging. Professor Hancock has published some 80 journal papers and 300 refereed conference publications. He was awarded the Pattern Recognition Society medal in 1991 and an outstanding paper award in 1997 by the journal Pattern Recognition. In 1998 he became a fellow of the International Association for Pattern Recognition.

Professor Hancock has been a member of the Editorial Boards of the journals IEEE Transactions on Pattern Analysis and Machine Intelligence, and, Pattern Recognition. He has also been a guest editor for special editions of the journals Image and Vision Computing and Pattern Recognition. He has been on the programme committees for numerous national and international meetings. In 1997 with Marcello Pelillo, he established a new series of international meetings on energy minimisation methods in computer vision and pattern recognition.