Article:
Cross, A.D.J. and Hancock, E.R. orcid.org/0000-0003-4496-2028 (1998) Graph matching with a dual-step EM algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence. pp. 1236-1253. ISSN 0162-8828

https://doi.org/10.1109/34.730557

Graph Matching With a Dual-Step EM Algorithm

Andrew D.J. Cross and Edwin R. Hancock

Abstract—This paper describes a new approach to matching geometric structure in 2D point-sets. The novel feature is to unify the tasks of estimating transformation geometry and identifying point-correspondence matches. Unification is realized by constructing a mixture model over the bipartite graph representing the correspondence match and by affecting optimization using the EM algorithm. According to our EM framework, the probabilities of structural correspondence gate contributions to the expected likelihood function used to estimate maximum likelihood transformation parameters. These gating probabilities measure the consistency of the matched neighborhoods in the graphs. The recovery of transformational geometry and hard correspondence matches are interleaved and are realized by applying coupled update operations to the expected log-likelihood function. In this way, the two processes bootstrap one another. This provides a means of rejecting structural outliers. We evaluate the technique on two real-world problems. The first involves the matching of different perspective views of 3.5-inch floppy discs. The second example is furnished by the matching of a digital map against aerial images that are subject to severe barrel distortion due to a line-scan sampling process. We complement these experiments with a sensitivity study based on synthetic data.

Index Terms—EM Algorithm, graph-matching, affine geometry, perspective geometry, relational constraints, Delaunay graph, discrete relaxation.

1 Introduction

The estimation of transformational geometry from point-sets is key to many problems of computer vision and robotics [29], [31]. Broadly speaking, the aim is to recover a matrix representation of the transformation between image and model coordinate systems. Estimating the matrix requires a set of correspondence matches between features in the two coordinate systems [37]. In other words, the feature points must be labeled. Posed in this way, there is a basic chicken-and-egg problem. Before good correspondences can be estimated, there needs to be reasonable bounds on the transformational geometry. Yet, this geometry is, after all, the ultimate goal of computation. This problem is usually overcome by invoking constraints to bootstrap the estimation of feasible correspondence matches [17], [27]. One of the most popular ideas is to use the epipolar constraint to prune the space of potential correspondences [17]. If reliable correspondences are not available, then a robust fitting method must be employed [36], [35]. This involves removing rogue correspondences through outlier rejection. An example is furnished by the recent work of Torr and Murray [37].

In this paper, we adopt a somewhat different approach to the problem of recovering transformational geometry. We take the view that the available correspondences are, at best, uncertain and may contain a substantial proportion of errors. However, rather than rejecting those correspondences which give rise to a large registration error, we attempt to iteratively correct them. In a nutshell, our idea is to bootstrap by alternating between estimating transformational parameters and refining correspondence matches. The framework for this study is furnished by a variant of the EM algorithm. Specifically, we use a structural gating process inspired by Jordan and Jacob's [23] hierarchical mixture of experts architecture to control contributions to the log-likelihood function for the transformation parameters. The structural gating is based on the consistency of the correspondence matches and draws on adjacency constraints for the point sets under consideration.

1.1 Related Literature

The problem of point pattern matching has attracted sustained interest in both the vision and statistics communities for several decades. For instance, Kendall [24] has generalized the process to projective manifolds using the concept of Procrustes distance. In the vision literature, the problem has attracted increased recent interest because of the pivotal role of rigidity constraints in recovering structure from motion sequences. As a concrete example, McReynolds and Lowe show how rigidity constraints can be used in perspective matching [27]. Historically, it was Ullman [39] who was one of the first to recognize the importance of exploiting rigidity constraints in the correspondence matching of point-sets. Recently, several authors have drawn inspiration from Ullman's ideas in developing general purpose correspondence matching algorithms using the Gaussian weighted proximity matrix.

There are two contrasting uses of the proximity-matrix which deserve special mention. Scott and Longuet-Higgins [33] locate correspondences by finding a singular value decomposition of the interimage proximity matrix. Shapiro and Brady [36], [35], on the other hand, match by

- *The authors are with the Department of Computer Science, University of York, York, Y01 5DD, UK. E-mail: {erh, adjc}@minster.york.ac.uk.*

Manuscript received 7 Mar. 1998; revised 30 Sept. 1998. Recommended for acceptance by Y.-F. Wang.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107484.

comparing the modal eigenstructure of the intraimage proximity matrix. In fact, these techniques provide the basic ground-work on which the deformable shape models of Cootes et al. [8] and Sclaroff and Pentland [34] build.

This work on the coordinate proximity matrix is closely akin to that of Umeyama [40] who shows how point-sets abstracted in a structural manner using weighted adjacency graphs can be matched using an eigendecomposition method. These ideas have been extended to accommodate parametererized transformations [41], which can be applied to the matching of articulated objects [42]. More recently, there have been several attempts at modeling the structural deformation of point-sets. For instance, Amit and Kong [5] have used a graph-based representation (graphical templates) to model deforming two-dimensional shapes in medical images. Lades et al. [26] have used a dynamic mesh to model intensity-based appearance in images.

However, these contributions fall well short of formally integrating structural constraints into the recovery of transformational geometry. The basic idea behind this paper is to address this deficiency. We take the view that, although the importance of structural constraints in the recovery of correspondence matches has been clearly identified, the adopted statistical framework leaves considerable scope for improvement. In particular, there is little attempt to explicitly characterize the representation of point-structure or to quantify in a statistical way acceptable geometric deformations.

1.2 Paper Overview

The aim in this paper is to develop a synergistic framework for matching. Specifically, we aim to facilitate feedback between the two problems of estimating transformational geometry and locating correspondence matches. The key idea is to use a bipartite graph to represent the current configuration of correspondence match. This graphical structure provides an architecture that can be used to gate contributions to the likelihood function for the geometric parameters using structural constraints. Correspondence matches and transformation parameters are estimated by applying the EM algorithm to the gated likelihood function. In this way, we arrive at dual maximization steps. When a Gaussian measurement process is assumed, then maximum likelihood parameters are found by minimizing the structurally gated squared error residuals between features in the two images being matched. Correspondence matches are updated so as to maximize the a posteriori probability of the observed structural configuration on the bipartite association graph.

It is important to stress that the idea of using a graphical model to provide structural constraints on parameter estimation is a task of generic importance. Although the EM algorithm has been used to extract affine and Euclidean parameters from point-sets [43], [15] or line-sets [28], there has been no attempt to impose structural constraints on the correspondence matches. Viewed from the perspective of graphical template matching [5], [26], our EM algorithm allows an explicit deformational model to be imposed on a set of feature points. Since the method delivers statistical estimates for both the transformation parameters and their associated covariance matrix, it offers significant advantages in terms of its adaptive capabilities. When viewed in this way, our method has some conceptual similarity with Pollefeys and Van Gool's [32] stratified self-calibration. Here, the calibration process is bootstrapped by interleaving the estimation of affine geometry and refining correspondences by imposing rigidity constraints.

The outline of this paper is as follows. Section 2 concerns the geometry of point-sets. Here, we review the affine and perspective transformations needed to register point-sets in our matching experiments. In Section 3, we provide a Bayesian framework which can be used to assess the relational consistency of correspondence matches using the neighborhood structure of the point-sets. Section 4 unifies the notions of geometric registration and correspondence matching introduced in Sections 2 and 3. The framework adopted here is a variant of the EM algorithm. Experimental evaluation of the method is presented in Section 5. Finally, Section 6 offers some conclusions and suggests directions for future investigation.

2 Point Registration

Point registration revolves around transforming the coordinates of the point-sets under a predefined geometry. The process is a critical ingredient in intermediate level vision [36], [35], [33]. Concrete applications include camera calibration [6], object recognition [17], motion analysis [25], and image mosaicing [21]. Depending on the imaging geometry, the transformation may be Euclidean, affine, or perspective. The simplest case is the Euclidean similarity transformation, which involves only translation, rotation and isotropic scaling. In the affine case, there is an additional anisotropic scaling process. Most complicated of all is the case of perspective geometry, which involves foreshortening in the vanishing point direction.

As with any estimation process, the basic problem is how to operate robustly when noise or poor image segmentation are limiting factors. More importantly, an essential prerequisite to transformation parameter estimation is the availability of a candidate set of correspondence matches. It is for this reason that robust estimation strategies [36], [35], [37] are favored over simpler least-squares methods [27]. Concrete examples include the recent work of Torr and Murray [37], where the aim is to exclude both noise-points and false-matches from the estimation process. However, although robust-estimation methods aim to exclude statistical outliers, they base their rejection criterion purely upon the distance of points from current estimates of the model-location. In other words, although there are extensive examples of utilizing geometric constraints, there is rarely any attempt to check the structural consistency of the correspondence matches. Here, we take the view that existing schemes are limited by the chicken-and-egg relationship between transformation matrix estimation and correspondence matching. The novel contribution of this paper is to develop a synergistic matching process in which the two processes iteratively bootstrap one another via information exchange. The framework for this study is provided by the EM algorithm.

2.1 Point Sets

Our goal is to recover the parameters of a geometric transformation $\mathbf{\Phi}^{(n)}$ that best maps a set of image feature points **w** onto their counterparts in a model **z**. In order to do this, we represent each point in the image data set by an augmented position vector $\vec{w}_i = (x_i, y_i, 1)^T$, where i is the point index. This augmented vector represents the two-dimensional point position in a homogeneous coordinate system. We will assume that all these points lie on a single plane in the image. In the interests of brevity, we will denote the entire set of image points by $\mathbf{w} = \{\vec{w}_i, \forall i \in \mathcal{D}\}$, where $\mathcal{D}$ is the point index-set. The corresponding fiducial points constituting the model are similarly represented by $\mathbf{z} = \{\vec{z}_j, \forall j \in \mathcal{M}\}$, where $\mathcal{M}$ denotes the index-set for the model feature-points $\vec{z}_j$.

Our aim in this paper is to investigate the matching of point-sets under two specific geometries. The first and simplest of these is affine geometry. The more complex case is that of plane perspective geometry.

2.2 Affine Geometry

In the case of the affine transformation, there are six free parameters. These model the two components of translation of the origin on the image plane, the overall rotation of the coordinate system, and the global scale, together with the two parameters of shear. These parameters can be combined succinctly into an augmented matrix that takes the form

$$\mathbf{\Phi}^{(n)} = \begin{pmatrix} \phi_{1,1}^{(n)} & \phi_{1,2}^{(n)} & \phi_{1,3}^{(n)} \\ \phi_{2,1}^{(n)} & \phi_{2,2}^{(n)} & \phi_{2,3}^{(n)} \\ 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

With this representation, the affine transformation of coordinates is computed using the following matrix multiplication

$$\vec{z}_j^{(n)} = \mathbf{\Phi}^{(n)} \vec{z}_j. \tag{2}$$

Clearly, the result of this multiplication gives us a vector of the form $\vec{z}_j^{(n)} = (x, y, 1)^T$. The superscript n indicates that the parameters are taken from the nth iteration of our algorithm. Our goal is to recover the elements $\phi_{i,j}^{(n)}$ of the parameter matrix $\mathbf{\Phi}^{(n)}$, which describes a coordinate system transformation that the best bring the set of image points **w** into registration with the model set **z** at iteration n.

The recovery of the parameters requires a minimum of three points that are known to be in correspondence. If more than three correspondences are known, then the parameter recovery process is overconstrained and can be solved using least-squares estimation. Since the affine transformation can be represented in a linear fashion, the least-squares estimate is easily recovered by matrix inversion.

2.3 Perspective Geometry

Perspective geometry is distinguished from the simpler Euclidean (translation, rotation, and scaling) and affine (the addition of shear) cases by the presence of significant foreshortening. We represent the perspective transformation by the parameter matrix

$$\Phi^{(n)} = \begin{pmatrix} \phi_{1,1}^{(n)} & \phi_{1,2}^{(n)} & \phi_{1,3}^{(n)} \\ \phi_{2,1}^{(n)} & \phi_{2,2}^{(n)} & \phi_{2,3}^{(n)} \\ \phi_{3,1}^{(n)} & \phi_{3,2}^{(n)} & \phi_{3,3}^{(n)} \end{pmatrix}. \tag{3}$$

Using homogeneous coordinates, the transformation between model and data is computed in the following way

$$\vec{z}_j^{(n)} = \frac{1}{\vec{z}_j^T . \Xi^{(n)}} \Phi^{(n)} \vec{z}_j, \tag{4}$$

where $\Xi^{(n)} = \left(\phi_{3,1}^{(n)}, \phi_{3,2}^{(n)}, 1\right)^T$ is a column-vector formed from the elements in the bottom row of the transformation matrix.

Because the transformation equations are nonlinear, the recovery of perspective geometry [4], [16], [11], [29], [20] is more difficult than the affine case. The main problems stem from the numerical instabilities associated with the denominator of the transformation equations. Haralick et al. [16] review the origins of the three-point pose estimation problem in the geometry and photogrammetry literature, providing an analysis of numerical sensitivity. One way to circumvent some of the numerical problems is to use weak-perspective or paraperspective geometry. For instance, DeMenthon and Davis [11] have an iterative algorithm that recovers linear weak-perspective pose if the correspondences between 3D features and 2D image points are known. Horaud et al. [20] have extended these ideas to develop an iterative algorithm for recovering paraperspective pose. Finally, Jacobs [22] has an efficient voting algorithm which uses a hashing technique based on image triangles to recover the perspective pose of planar objects in 3D scenes. Notwithstanding these important contributions to the estimation of perspective pose, it must be stressed that, in this paper, the problem serves as an exemplar of our new matching architecture. As a result, our primary interest is not with issues of efficiency or numerical stability. Moreover, we satisfy ourselves with a simple demonstration on the matching of planar rather than 3D objects.

3 Relational Graph Matching

The gating layer of our matching architecture represents the state of correspondence match between the point-sets. Rather than using epipolar constraints to establish putative correspondences [37], we use constraints provided by the spatial adjacency of the points. These constraints are elicited by separately triangulating the data and model points. We use the neighborhood consistency of the correspondences in the triangulations to weight the contributions to the log-likelihood function. In the remainder of this section, we describe how the relational consistency of the correspondence match can be modeled in a probabilistic manner.

3.1 Point Correspondences

One of our goals in this paper is to exploit structural constraints to improve the recovery of transformation parameters from sets of feature points. We abstract the representation of correspondences using a bipartite graph. Because

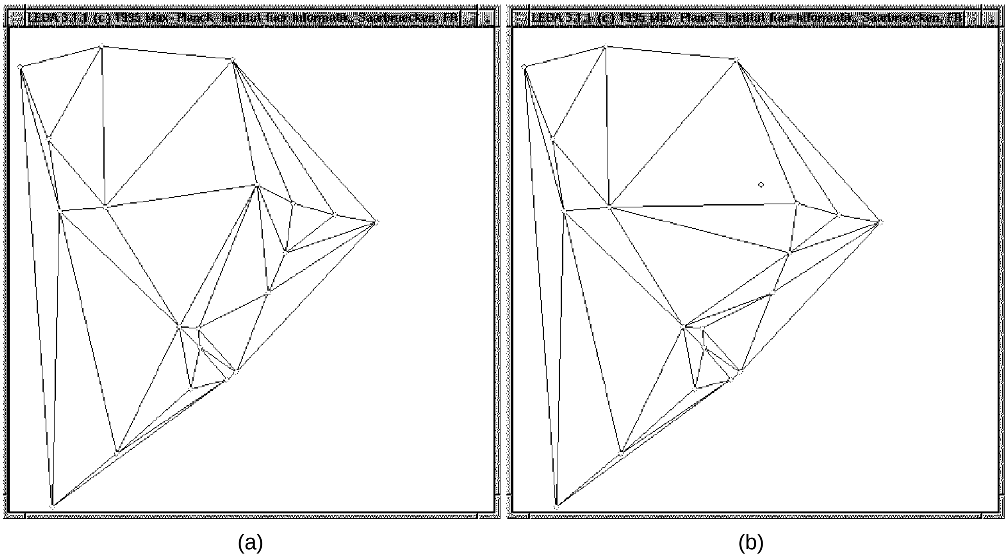


Fig. 1. An example of Delaunay graph editing by node deletion. (a) The original graph. (b) The edited graph resulting from a node deletion.

of its well documented robustness to noise and change of viewpoint, we adopt the Delaunay triangulation as our basic representation of image structure [38], [13]. We establish Delaunay triangulations on the data and the model by seeding Voronoi tessellations from the feature-points [1], [2], [3].

The process of Delaunay triangulation generates relational graphs from the two sets of point-features. An example is shown in Fig. 1. More formally, the point-sets are the nodes of a data graph $G_D = \{\mathcal{D}, E_D\}$ and a model graph $G_M = \{\mathcal{M}, E_M\}$. Here, $E_D \subseteq \mathcal{D} \times \mathcal{D}$ and $E_M \subseteq \mathcal{M} \times \mathcal{M}$ are the edge-sets of the data and model graphs. Key to our matching process is the idea of using the edge-structure of Delaunay graphs to constrain the correspondence matches between the two point-sets. This correspondence matching is denoted by the function $f : \mathcal{D} \rightarrow \mathcal{M}$ from the nodes of the data-graph to those of the model graph. According to this notation, the statement $f^{(n)}(i) = j$ indicates that there is a match between the node $i \in \mathcal{D}$ of the model-graph to the node $j \in \mathcal{M}$ of the data-graph at iteration n of the algorithm. We use the binary indicator or assignment variable

$$s_{i,j}^{(n)} = \begin{cases} 1 & \text{if } f^{(n)}(i) = j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

to represent the configuration of correspondence matches.

3.2 Relational Constraints

In performing the matches of the nodes in the data graph G_D, we will be interested in exploiting structural constraints provided by the edges of the model graph G_M. These constraints are purely symbolic in nature and are represented by configurations of nodes in the model graph. We use representational units or subgraphs that consist of neighborhoods of nodes that are connected to a center node by arcs to impose consistency constraints. For convenience, we refer to these structural subunits or N-ary relations as supercliques. The superclique centered on the node indexed i in the data graph G_D with arc-set E_D is denoted by the set of nodes $C_i^D = i \cup \{k; (i,k) \in E_D\}$. The matched realization of this superclique is denoted by the relation

$$\Gamma_i = \left(f(u_1), f(u_2), \ldots, f\left(u_{|C_i^D|} \right) \right).$$

Key to our matching scheme is the idea of computing the probability of the match of the data-graph node i to the model-graph node j. We realize this goal by comparing the configuration of matches residing on the data-graph superclique C_i^D with the configuration of nodes that constitute the model-graph superclique centered on the node j, i.e., $C_j^M = j \bigcup \{l; (j,l) \in E_M\}$.

To realize this comparison we require a dictionary of possible mappings between the nodes of the data-graph clique C_i^D and those of the model-graph superclique C_j^M so as to bind matches for the purposes of comparison. If the two supercliques are of the same size, then these so-called structure-preserving mappings are found by permuting the noncenter nodes. However, when the supercliques are of different size, then we must pad the smaller unit with dummy nodes to raise it to the same size as the larger unit. This increases the complexity of the task of dictionary compilation. First, we must insert one or more dummy edges into the smaller superclique between each pair of the existing edges. Second, we perform cyclic permutation of each of the resulting padded configurations. This process is illustrated in Fig. 2. The process effectively models the disruption of the adjacency structure of the data graph caused by the addition of clutter elements or the loss of elements due to segmental drop-out. Since it is intrinsically symbolic in nature, the resulting dictionary is invariant to scene translations, scalings, or rotations.

To be more formal, the set of feasible mappings, or dictionary, for the model-graph superclique C_j^M is denoted by $\Theta_j = \{S\}$. The individual structure-preserving mappings are sets of Cartesian pairs which associate individual data-graph

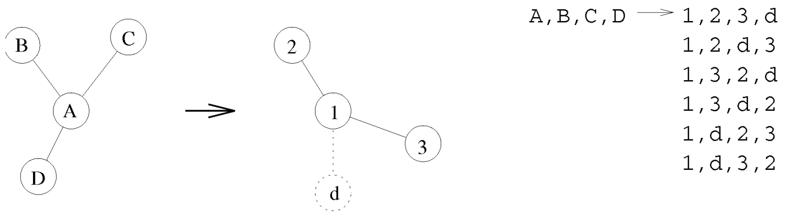


Fig. 2. Example superclique mapping: The diagram shows the cyclic permutation with dummy insertion for matching a data-graph superclique with four nodes being matched onto a model-graph superclique containing only three nodes.

nodes from the superclique C_i^D with counterparts in the model-graph superclique C_j^M. The center nodes are always paired with one another. The neighborhood nodes are paired either with other neighborhood nodes or with dummy nodes. Each dictionary item is a structure-preserving mapping of the form

$$S = (i,j) \cup \Big\{(k,l);(k,l) \in \Big[\big(C_i^D - \{i\}\big) \cup dummy\Big] \times \Big[\big(C_j^M - \{j\}\big) \cup dummy\Big]\Big\}. \tag{6}$$

It is the size of the dictionary which poses the main computational bottleneck in the application of our matching scheme. For instance, if we are considering the matching of supercliques of the same size, i.e., no padding is required, then there are $\left|C_j\right|$ cyclic dictionary items of the superclique C_j. If, on the other hand, the cyclicity constraint is lifted, then there are $\left|C_j\right|!$ items. When padding is introduced, then the complexity is increased. If the model-graph relation S is being compared with the match residing on the data-graph clique C_j, then there are

$$\frac{\left(|S|-1\right)!}{\left(\left|C_j\right|-1\right)!\left(|S|-\left|C_j\right|\right)!}$$

cyclic dictionary items and

$$\frac{\left(|S|-1\right)!\left|C_j\right|}{\left(|S|-\left|C_j\right|\right)!}$$

noncyclic dictionary items.

3.3 Structural Matching Probabilities

In this section, we present a simple model which can be used to assign probabilities to putative correspondence matches based on their consistency with the assigned matches on neighboring nodes of the graph. This model draws on a refinement of the relational consistency measure originally reported by Wilson and Hancock [45]. Our goal is to compute the probability of assigning the correspondence match $f^{(n+1)}(i) = j$ to the center node i of the data-graph clique C_i^D at iteration $n + 1$. In order to draw on contextual information concerning the consistency of this putative correspondence match, we condition the probability on the matches assigned to the neighboring nodes of the clique at iteration n of the algorithm, The relevant set of neighborhood matches is denoted by the configuration

$$\hat{\Gamma}_i^{(n)} = \Big\{f^{(n)}(l); l \in C_i^D - \{i\}\Big\}.$$

In other words, we aim to compute the correspondence matching probability

$$P\Big(f^{(n+1)}(i) = j \Big| \hat{\Gamma}_i^{(n)}\Big) = \frac{P\Big(f^{(n+1)}(i) = j, \hat{\Gamma}_i^{(n)}\Big)}{P\Big(\hat{\Gamma}_i^{(n)}\Big)}. \tag{7}$$

Since the probability of the assigned neighborhood configuration $P\Big(\hat{\Gamma}_i^{(n)}\Big)$ appearing in the denominator is a fixed property of the data-graph clique C_i^D at iteration n, we can confine our attention to developing the joint configurational probability $P\Big(f^{(n+1)}(i) = j, \hat{\Gamma}_i^{(n)}\Big)$ appearing in the numerator. According to the Bayes formula,

$$P\Big(f^{(n+1)}(i) = j \Big| \hat{\Gamma}_i^{(n)}\Big) = \frac{P\Big(f^{(n+1)}(i) = j, \hat{\Gamma}_i^{(n)}\Big)}{\sum_{j \in \mathcal{M}} P\Big(f^{(n+1)}(i) = j, \hat{\Gamma}_i^{(n)}\Big)}. \tag{8}$$

To simplify the development, we use the notation $\Gamma_{i,j} = \Big\{f^{(n+1)}(i) = j, f^{(n)}(l), \forall l \in C_i^D - \{i\}\Big\}$ to represent the configuration of matched nodes on the superclique C_i^D with the putative update $f^{(n+1)}(i) = j$ at the center node.

As we noted in Section 3.2, the consistent labelings available for gauging the quality of the putative match $f^{(n)}(i) = j$ are represented by the set of relational mappings from the data-graph clique C_i^D onto the model graph clique centered on the node j, i.e., C_j^M is encapsulated by the dictionary Θ_j. To evaluate the consistency of the putative configuration of matches $\Gamma_{i,j}$, we use the Bayes rule to expand the probability $P(\Gamma_{i,j})$ over the structure preserving mappings between the supercliques C_i^D and C_j^M belonging to the dictionary Θ_j. In other words, we write

$$P\Big(\Gamma_{i,j}\Big) = \sum_{S \in \Theta_j} P\Big(\Gamma_{i,j} \Big| S\Big) . P(S). \tag{9}$$

The development of a useful graph-mapping measure from this expression requires models of the processes at play in producing matching errors. These models are represented in terms of the joint conditional matching probabilities $P\Big(\Gamma_{i,j} \Big| S\Big)$ and of the joint priors $P(S)$ for the consistent relations in the dictionary. In developing the required models, we will limit our assumptions to the case of matching errors which are memoryless and occur with a uniform probability distribution.

To commence our modeling of the conditional probabilities, we assume that the various types of matching error for nodes belonging to the same superclique are memoryless. In direct consequence of this assumption, we may factorize the required probability distribution over the symbolic constituents of the relational mapping under consideration. As a result, the conditional probabilities $P\left(\Gamma_{i,j}|S\right)$ may be expressed in terms of a product over label confusion probabilities

$$P\left(\Gamma_{i,j}|S\right) = \prod_{(k,l)\in S} P\left(f(k)\,|\,l\right). \tag{10}$$

Our next step is to propose a two component model of the processes which give rise to erroneous matches. The first of these processes is initialization error, which we aim to rectify by iterative label updates. We assume that initialization errors occur with a uniform and memoryless probability P_e. The second source of error is structural disturbance of the relational graphs caused by noise, clutter or segmentation error. We assume that structural errors can also be modeled by a uniform distribution which occurs with probability P_ϕ. This probability governs the insertion of dummy nodes necessary to make the comparison of differently sized cliques feasible. Dummy nodes are inserted into the smaller clique to raise it to the same size as the larger clique. Under these dual assumptions concerning the nature of matching errors, the confusion probabilities appearing under the product of (10) may be assigned according to the following distribution rule

$$P\left(f(k)\,|\,l\right) = \begin{cases} \left(1 - P_\phi\right)\left(1 - P_e\right) & \text{if } f(k) = l \\ \left(1 - P_\phi\right)P_e & \text{if } f(k) \neq l \text{ and } l \neq \text{dummy} \\ P_\phi & \text{if } k = \text{dummy or} \\ & \quad l = \text{dummy} \end{cases} \tag{11}$$

The three cases under this distribution rule require further explanation. The first case corresponds to the situation in which there is agreement between the current match and that demanded by the dictionary item S. The second case corresponds to matching disagreements which do not involve dummy nodes. The third case arises when the supercliques under consideration are of different size. If either the data-graph node k or model-graph node l is a dummy node inserted for the purposes of padding, then the null-match probability is assigned.

As a natural consequence of this distribution rule, the joint conditional probability is a function of three physically meaningful variables. The first of these is the Hamming distance $H(\Gamma_{i,j}, S)$ between the assigned matching and the feasible relational mapping S. This quantity counts the number of conflicts between the putative configuration of matches $\Gamma_{i,j}$ assigned to the data-graph superclique C_i^D and those assignments demanded by the relational mapping S onto the model-graph superclique C_j^M. With the binary indicator variables used to represent the matching process, the Hamming distance is given by

$$H\left(\Gamma_{i,j}, S\right) = \sum_{(k,l)\in S} \left(1 - s_{k,l}^{(n)}\right). \tag{12}$$

The second variable is the sum of the number of dummy nodes required for padding. This second quantity is equal to the size difference between data-graph clique C_i^D and the model-graph clique C_j^M and is denoted by

$$\Psi\left(\Gamma_{i,j}\right) = \left\| C_i^D \right| - \left| C_j^M \right\|. \tag{13}$$

The final variable is the size of the larger superclique, i.e.,

$$R_{i,j} = \max\left[\left|C_i^D\right|, \left|C_j^M\right|\right]. \tag{14}$$

With these ingredients, the resulting expression for the joint conditional probability acquires an exponential character

$$\begin{aligned} P\left(\Gamma_{i,j}|S\right) &= \left[\left(1 - P_\phi\right)\left(1 - P_e\right)\right]^{R_{i,j} - H\left(\Gamma_{i,j},S\right) - \Psi\left(\Gamma_{i,j}\right)} \\ &\times \left[\left(1 - P_\phi\right)P_e\right]^{H\left(\Gamma_{i,j},S\right)} \\ &\times \left[P_\phi\right]^{\Psi\left(\Gamma_{i,j}\right)}. \end{aligned} \tag{15}$$

Finally, in order to compute the superclique matching probability $P(\Gamma_{i,j})$, we require a model of the joint-priors for the dictionary items. Here, we assume that the unit probability mass is uniformly distributed over the relevant items, i.e.,

$$P(S) = \frac{1}{\left|\Theta_j\right|}. \tag{16}$$

Collecting together terms in the expression for $P(\Gamma_i\,|\,S)$ and substituting for the joint priors for the dictionary items, we obtain the following expression for the superclique matching probability

$$P\left(\Gamma_{i,j}\right) = \frac{K_{i,j}}{\left|\Theta_j\right|} \sum_{S\in\Theta_j} \exp\left[-\left(k_e H\left(\Gamma_{i,j}, S\right) + k_\phi \Psi\left(\Gamma_{i,j}\right)\right)\right], \tag{17}$$

where $K_{i,j} = \left[\left(1 - P_e\right)\left(1 - P_\phi\right)\right]^{R_{i,j}}$. The two exponential constants appearing in the above expression are related to the matching-error probability and the null match probability, i.e.,

$$k_e = \ln \frac{\left(1 - P_e\right)}{P_e}$$

and

$$k_\phi = \ln \frac{\left(1 - P_e\right)\left(1 - P_\phi\right)}{P_\phi}.$$

In the work reported here, we set the correspondence error probability and the structural error probability according to the size differences between the graphs, i.e.,

$$P_e = P_\phi = \frac{2\left\| |\mathcal{M}| - |\mathcal{D}| \right\|}{\left\| |\mathcal{M}| + |\mathcal{D}| \right\|}. \tag{18}$$

The probability distribution given in (17) may be regarded as providing a natural way of softening the hard relational constraints operating in the model graph. The most striking and critical feature of the expression for $P(\Gamma_{i,j})$ is that the consistency of match is gauged by a series of exponentials that are compounded over the dictionary of consistently mapped relations.

The key idea underlying our dual-step EM algorithm for recovering transformational geometry is to gate contributions to the expected log-likelihood function using relational constraints. In order to realize this process, we require the expected value of the assignment variables, i.e., $E\left[s_{i,j}^{(n)}\right]$. With the ingredients outlined in this section, the expected assignment variable is equal to

$$E\left[s_{i,j}^{(n)}\right] = \zeta_{i,j}^{(n)} = \frac{\sum_{S\in\Theta_j} P\left(\Gamma_{i,j} \middle| S\right) P(S)}{\sum_{j\in\mathcal{M}} \sum_{S\in\Theta_j} P\left(\Gamma_{i,j} \middle| S\right) P(S)}. \tag{19}$$

Substituting from (17), we make the role of Hamming distance and the number of padding or dummy nodes more explicit by writing

$$\zeta_{i,j}^{(n)} = \frac{\frac{K_{i,j}}{\left|\Theta_j\right|} \sum_{S\in\Theta_j} \exp\left[-\left(k_e H\left(\Gamma_{i,j}, S\right) + k_\phi \Psi\left(\Gamma_{i,j}\right)\right)\right]}{\sum_{j\in\mathcal{M}} \frac{K_{i,j}}{\left|\Theta_j\right|} \sum_{S\in\Theta_j} \exp\left[-\left(k_e H\left(\Gamma_{i,j}, S\right) + k_\phi \Psi\left(\Gamma_{i,j}\right)\right)\right]}. \tag{20}$$

4 The Unified Matching Algorithm

Our aim is to extract geometric transformation parameters and correspondence matches from the two point-sets using the EM algorithm. When couched probabilistically, the goal can be succinctly stated as that of jointly maximizing the data-likelihood $p(\mathbf{w} \mid \mathbf{z}, f, \Phi)$ over the space of correspondence matches f and the matrix of transformation parameters Φ. We realize this process using a dual-step or hierarchical version of the EM algorithm. The utility measure underpinning the algorithm is the expected log-likelihood function which allows parameters to be estimated when confronted with incomplete data. The basic idea underlying the algorithm is to iterate between the expectation and maximization steps until convergence is reached. Expectation involves updating the a posteriori probabilities of the missing data using the most recently available parameter estimates. In the maximization phase, the model parameters are recomputed to maximize the expected value of the incomplete data likelihood.

According to the original work of Dempster et al. [12] the expected likelihood function is computed by weighting the current log-probability density by the a posteriori measurement probabilities estimated from the preceding maximum likelihood parameters. Here, we wish to exploit Jordan and Jacobs [23] idea of augmenting the maximum likelihood process with a graphical model. From an architectural standpoint, the graphical model can be regarded as a supervisor network which effectively gates contributions to the expected log-likelihood function. The novelty of the work reported here is to develop a variant of this idea in which it is the bipartite graph, i.e., f, which gates the likelihood function for the transformation parameters Φ. This graph represent the current state of correspondence match between the two point-sets.

We extract both maximum likelihood geometric transformation parameters and maximum a posteriori matching probabilities by applying coupled update operations to the gated likelihood function. In this way, the consistency of the structural matching process can guide the pose recovery process. Likewise, error probabilities derived from the position residuals are used to guide the correspondence matching process. When the joint likelihood function is maximized in this way, then the correspondence matches play the role of missing data.

4.1 Mixture Model

Our basic aim is to jointly maximize the data-likelihood $p(\mathbf{w} \mid \mathbf{z}, f, \Phi)$ over the space of correspondence matches f and the matrix of geometric transformation parameters Φ. We can regard the set of data-graph measurement vectors, i.e., $\mathbf{w}$, as the input to our process. The model-graph measurements, i.e., $\mathbf{z}$, on the other hand, are the outputs which are to be stochastically recovered from the transformation parameters, i.e., Φ and the structural state of the bipartite correspondence matching graph, i.e., f. For this reason, rather than commencing our discussion from the complete likelihood function $p(\mathbf{w} \mid \mathbf{z}, f, \Phi)$, we turn to the incomplete data likelihood, i.e., $p(\mathbf{w} \mid f, \Phi)$.

Our first model assumption is that the incomplete data-likelihood can be factorized over the set of data-graph measurement vectors. In other words, we assume that the measurement process is conditionally independent given the transformation parameters and the model-graph measurements. As a result, we write

$$p\left(\mathbf{w} \middle| f, \Phi\right) = \prod_{i\in\mathcal{D}} p\left(\vec{w}_i \middle| f, \Phi\right). \tag{21}$$

The next step is to account for the missing variables by developing a mixture model over the set of model-graph measurements. Using the Bayes rule, we expand the incomplete data-likelihood over the set of model-graph measurements

$$p\left(\vec{w}_i \middle| f, \Phi\right) = \sum_{j\in\mathcal{M}} p\left(\vec{w}_i, \vec{z}_j \middle| f, \Phi\right). \tag{22}$$

Our key modeling ingredient is to exploit the binary assignment variables as exponential indicators in developing a measurement density for the correspondence matches. We take the view that if data graph node i matches to model-graph node j, then it is the measurement density $p\left(\vec{w}_i, \vec{z}_j \middle| \Phi\right)$ that is appropriate in gauging the coordinate similarity of the points $\vec{w}_i$ and $\vec{z}_j$. If this is not the case, then we assign a uniform measurement density ρ which is independent of coordinates and models the outlier process. With this modeling ingredient, we may write

$$p\left(\vec{w}_i, \vec{z}_j \middle| f, \Phi\right) = p\left(\vec{w}_i, \vec{z}_j \middle| \Phi\right)^{s_{i,j}} \rho^{1-s_{i,j}}. \tag{23}$$

We are now in a position to assemble the expression for the incomplete data-likelihood by substituting (22) and (23) into (21). The result is

$$p\left(\mathbf{w} \middle| f, \Phi\right) = \prod_{i\in\mathcal{D}} \sum_{j\in\mathcal{M}} p\left(\vec{w}_i, \vec{z}_j \middle| \Phi\right)^{s_{i,j}} \rho^{1-s_{i,j}}. \tag{24}$$

Our route to maximizing the incomplete data-likelihood is to apply the EM algorithm [12], [23] to the expected log-likelihood function. The idea behind the EM algorithm is to compute the conditional log-likelihood for a new parameter set given the preceding parameter estimates. In the case of our incomplete data-likelihood, the conditional log-likelihood is

$$Q\left(\Phi^{(n+1)} \middle| \Phi^{(n)}\right) = \sum_{i\in\mathcal{D}} \sum_{j\in\mathcal{M}} P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(n)}\right) \left[s_{i,j}^{(n)} \ln p\left(\vec{w}_i, \vec{z}_j \middle| \Phi^{(n+1)}\right) + \left(1 - s_{i,j}^{(n)}\right) \ln \rho\right]. \tag{25}$$

To simplify matters, we make a mean-field approximation and replace $s_{i,j}^{(n)}$ by its average value, i.e., we make use of the fact that $E\left(s_{i,j}^{(n)}\right) = \zeta_{i,j}^{(n)}$. In this way, the structural matching probabilities gate contributions to the expected likelihood function, i.e.,

$$Q\left(\Phi^{(n+1)} \middle| \Phi^{(n)}\right) = \sum_{i\in\mathcal{D}} \sum_{j\in\mathcal{M}} P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(n)}\right) \left[\zeta_{i,j}^{(n)}\left(\ln p\left(\vec{w}_i, \vec{z}_j \middle| \Phi^{(n+1)}\right) - \ln \rho\right) + \ln \rho\right]. \tag{26}$$

This is an important statement of our matching framework. The first term inside the round-braces is appropriate to parameter estimation, while the second term is appropriate to the computation of correspondence matches. In the case of parameter estimation and as demonstrated by Dempster, et al. [12], maximizing the incomplete expected likelihood function is equivalent to maximizing the following conditional likelihood function

$$\hat{Q}\left(\Phi^{(n+1)} \middle| \Phi^{(n)}\right) = \sum_{i\in\mathcal{D}} \sum_{j\in\mathcal{M}} P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(n)}\right) \zeta_{i,j}^{(n)}\left(\ln p\left(\vec{w}_i, \vec{z}_j \middle| \Phi^{(n+1)}\right)\right). \tag{27}$$

The structure of this component of the expected log-likelihood function requires further comment. The measurement densities $p\left(\vec{w}_i, \vec{z}_j \middle| \Phi^{(n+1)}\right)$ model the distribution of error-residuals between the observed data-point position $\vec{w}_i$ and the predicted position of the model point $\vec{z}_j$ under the current set of transformation parameters $\Phi^{(n+1)}$. The log-likelihood contributions at iteration $n + 1$ are weighted by the a posteriori measurement probabilities $P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(n)}\right)$ computed at the previous iteration n of the algorithm. The individual contributions to the expected log-likelihood function are gated by the structural matching probabilities $\zeta_{i,j}^{(n)}$. This chimes with the hierarchical mixture of experts algorithm of Jordan and Jacobs [23], where an expert layer is responsible for gating. However, whereas Jordan and Jacobs gating layer is parametric, ours is structural.

We now turn our attention to the recovery of correspondence matches. As we mentioned earlier, the component of the expected log-likelihood appropriate to this task is the second term under the round braces of (26). Since the uniform density ρ is a constant whose value is less than unity, the goal of correspondence matching is to maximize the quantity

$$Z = \sum_{i\in\mathcal{D}} \sum_{j\in\mathcal{M}} P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(n)}\right) \zeta_{i,j}^{(n)}. \tag{28}$$

It is interesting to note that this is just the MAP criterion used by Wilson and Hancock [45] in their work on graph-matching by discrete relaxation.

4.2 Expectation

In the expectation step of the EM algorithm, the a posteriori probabilities of the missing data (i.e., the model-graph measurement vectors, $\vec{z}_j$) are updated by substituting the revised parameter vector into the conditional measurement distribution. Using the Bayes rule, we can rewrite the a posteriori measurement probabilities in terms of the components of the corresponding conditional measurement densities

$$P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(n+1)}\right) = \frac{\alpha_{i,j}^{(n)} p\left(\vec{w}_i, \vec{z}_j \middle| \Phi^{(n)}\right)}{\sum_{j'\in\mathcal{M}} \alpha_{j'}^{(n)} p\left(\vec{w}_i, \vec{z}_{j'} \middle| \Phi^{(n)}\right)}. \tag{29}$$

The mixing proportions are computed by averaging the a posteriori probabilities over the set of data-points, i.e.,

$$\alpha_{i,j}^{(n+1)} = \frac{1}{|\mathcal{D}|} \sum_{i\in\mathcal{D}} P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(n)}\right). \tag{30}$$

In order to proceed with the development of a point registration process, we require a model for the conditional measurement densities, i.e., $p\left(\vec{w}_i, \vec{z}_j \middle| \Phi^{(n)}\right)$. Here, we assume that the required model can be specified in terms of a multivariate Gaussian distribution. The random variables appearing in these distributions are the error residuals for the position predictions of the jth model point delivered by the current estimated transformation parameters. Accordingly, we write

$$p\left(\vec{w}_i, \vec{z}_j \middle| \Phi^{(n)}\right) = \frac{1}{(2\pi)^{\frac{3}{2}}\sqrt{|\Sigma|}} \exp\left[-\frac{1}{2}\epsilon_{i,j}\left(\Phi^{(n)}\right)^T \Sigma^{-1} \epsilon_{i,j}\left(\Phi^{(n)}\right)\right]. \tag{31}$$

In the above expression, Σ is the variance-covariance matrix for the vector of error-residuals $\epsilon_{i,j}\left(\Phi^{(n)}\right) = \vec{w}_i - \vec{z}_j^{(n)}$ between the components of the predicted measurement vectors $\vec{z}_j^{(n)}$

and their counterparts in the data, i.e., $\vec{w}_i$. Formally, the matrix is related to the expectation of the outer-product of the error-residuals i.e., $\Sigma = E\left[\epsilon_{i,j}\left(\Phi^{(n)}\right)\epsilon_{i,j}\left(\Phi^{(n)}\right)^T\right]$. Accordingly, we compute the following estimate of Σ

$$\tilde{\Sigma} = \frac{\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{M}} P\left(\vec{z}_j\middle|\vec{w}_i,\Phi^{(n)}\right)\zeta_{i,j}^{(n)}\left(\vec{w}_i-\vec{z}_j^{(n)}\right)\left(\vec{w}_i-\vec{z}_j^{(n)}\right)^T}{\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{M}} P\left(\vec{z}_j\middle|\vec{w}_i,\Phi^{(n)}\right)\zeta_{i,j}^{(n)}}. \quad (32)$$

4.3 Maximization

As pointed out earlier, the maximization step of our unified or synergistic matching algorithm is based on dual coupled update processes. The first of these aims to locate maximum a posteriori probability correspondence matches. The second update operation is concerned with locating maximum likelihood transformation parameters. We effect the coupling by allowing information flow between the two processes. In the remainder of this section, we detail the maximization steps used to realize this coupling.

4.3.1 Maximum a posteriori Probability Matches

Point correspondences are sought so as to maximize the a posteriori probability of structural match. The updated configuration of correspondence matches is located by maximizing the second component of the expected log-likelihood function defined in (26). The resulting update formula is

$$f^{(n+1)}(i) = \arg\max_{j\in\mathcal{M}} P\left(\vec{z}_j\middle|\vec{w}_i,\Phi^{(n)}\right)\zeta_{i,j}^{(n+1)}. \quad (33)$$

Once this update equation has been applied, the unmatched model-graph nodes are identified for removal from the triangulation. At this point, the edited set of model feature-points is retriangulated to correct potential structural errors. We provide more details of this graph-editing process in the next subsection. The structural matching probabilities $\zeta_{i,j}^{(n+1)}$ are also updated using (20) as outlined in Section 3.

4.3.2 Updating the Triangulation

In order to overcome this source of potential structural corruption, at the end of each iteration we retriangulate the graph in order to accurately reflect the structure of the points under the current estimate of the transformation parameters. We have recently shown how this process of active graph reconfiguration or editing can be realized as a MAP estimation process [44]. However, rather than confining itself purely to node relabeling, the algorithm now encompasses the possibility of node insertions or deletions together with the implied modification of the Delaunay edge-set. This process is illustrated in Fig. 1. Details of the derivation are outside the scope of this paper. The basic idea is to gauge the net effect of deleting a node by examining those contributions to the consistency measure that arise from modification of the supercliques containing the node in question. This set is constructed by identifying those nodes that form a superclique with node i in graph G_D, i.e., $C_i^D - \{i\}$, and determining the new superclique set for these nodes in the reconfigured graph G_D'. We let χ_i^+ denote the superclique set of object i in graph G_D and χ_i^- denote the corresponding superclique set in the reconfigured graph G_M'. With this notation, the change in the MAP criterion of (28) caused by the deletion of the node i is proportional to

$$\Delta_i^- = P_\phi \sum_{l\in\chi_i^-}\frac{K_{i,j}}{\left|\Theta_j\right|}\sum_{S\in\Theta_j}\exp\left[-k_e H\left(\Gamma_{l,j},S\right)\right]. \quad (34)$$

By contrast, when considering the change in the MAP criterion caused by reinsertion of the node i, it is the superclique set χ_i^+ to which we turn our attention. The corresponding change to the MAP criterion is proportional to

$$\Delta_i^+ = P\left(\vec{z}_{f(i)}\middle|\vec{w}_i,\Phi^{(n)}\right)\sum_{l\in\chi_i^+}\frac{K_{i,j}}{\left|\Theta_j\right|}\sum_{S\in\Theta_j}\exp\left[-k_e H\left(\Gamma_{l,j},S\right)\right]. \quad (35)$$

The decision criteria for node deletion or reinsertion are as follows. We delete node i provided $\Delta_i^+ < \Delta_i^-$ and reinstate it provided $\Delta_i^+ > \Delta_i^-$. In addition to this structural modification, we can improve the robustness of parameter estimation by removing points in the model-set which have no correspondence in the data-set when computing the expected log-likelihood function in the expectation step of the EM algorithm. Once these points are removed, we must once again retriangulate the point set in order to reflect the change in structure. At each iteration of the maximization stage, we also try reintroducing any deleted points back into the data set.

4.3.3 Maximum Likelihood Parameters

With the Gaussian measurement process described in Section 4.2, the maximization step of the EM algorithm simply reduces to computing the weighted squared error criterion

$$Q'\left(\Phi^{(n+1)}\middle|\Phi^{(n)}\right) = -\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{M}} P\left(\vec{z}_j\middle|\vec{w}_i,\Phi^{(n)}\right)\zeta_{i,j}^{(n)}\epsilon_{i,j}\left(\Phi^{(n+1)}\right)^T\Sigma^{-1}\epsilon_{i,j}\left(\Phi^{(n+1)}\right). \quad (36)$$

In other words, the a posteriori probabilities $P\left(\vec{z}_j\middle|\vec{w}_i,\Phi^{(n)}\right)$ and the structural matching probabilities $\zeta_{i,j}^{(n)}$ effectively regulate the contributions to a weighted squared-error criterion. In the remainder of this subsection, we provide details of how maximization is realized for the two different geometries described in Section 2.

- *Affine Geometry:* In the case of affine geometry, the transformation is linear in the parameters. This allows us to locate the maximum-likelihood parameters directly by solving the following system of saddle-point equations for the independent affine parameters $\phi_{k,l}^{(n+1)}$ running over the indices $k = 1, 2$ and $l = 1, 2, 3$

$$\frac{\partial Q'\left(\Phi^{(n+1)}\middle|\Phi^{(n)}\right)}{\partial \phi_{k,l}^{(n+1)}} = 0. \tag{37}$$

For the affine transformation the set of saddle-point equations is linear and are, hence, easily solved by using matrix inversion. It is straightforward to show that the updated matrix of affine parameters must satisfy the following implied system of linear equations

$$\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{M}} P\left(\vec{z}_j\middle|\vec{w}_i,\Phi^{(n)}\right)\zeta_{i,j}^{(n)}\times$$
$$\left[\left(\vec{w}_i - \Phi^{(n+1)}\vec{z}_j\right)^T \Sigma^{-1}\right]\vec{z}_j\, U = 0, \tag{38}$$

where the elements of the matrix U are the partial derivatives of the affine transformation matrix with respect to the individual parameters, i.e.,

$$U = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}. \tag{39}$$

As a result, the updated solution matrix is given by

$$\Phi^{(n+1)} = \left[\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{M}} P\left(\vec{z}_j\middle|\vec{w}_i,\Phi^{(n)}\right)\zeta_{i,j}^{(n)}\vec{z}_j\, U^T \vec{z}_j^T\, \Sigma^{-1}\right]^{-1}$$
$$\times\left[\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{M}} P\left(\vec{z}_j\middle|\vec{w}_i,\Phi^{(n)}\right)\zeta_{i,j}^{(n)}\vec{w}_i\, U^T \vec{z}_j^T\, \Sigma^{-1}\right]. \tag{40}$$

This allows us to recover a set of improved transformation parameters at iteration $n + 1$. Once these are computed, the a posteriori measurement probabilities may be updated by applying the Bayes formula to the measurement density function. The update procedure involves substituting the parameter matrix of (1) into the Gaussian density of (31) and applying the Bayes theorem.

- *Perspective Geometry:* In the case of perspective geometry where we have used homogeneous coordinates, the saddle-point equations are no longer soluble in a closed-form linear fashion. The maximum likelihood transformation parameters satisfy the condition

$$\Phi^{(n+1)} = \arg\min_{\Phi}\sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{M}} P\left(\vec{z}_j\middle|\vec{w}_i,\Phi^{(n)}\right)$$
$$\zeta_{i,j}^{(n)}\left(\vec{w}_i - \vec{z}_j^{(n+1)}\right)^T \Sigma^{-1}\left(\vec{w}_i - \vec{z}_j^{(n+1)}\right). \tag{41}$$

Following Horaud et al. [20], we solve this maximization problem using the Levenberg-Marquardt technique. This is a nonlinear optimization technique that offers a compromise between the steepest gradient and inverse Hessian methods. The former is used when close to the optimum while the latter is used far from it. In other words, when close to the optimum, parameter updating takes place with step-size proportional to the gradient $\nabla_\Phi Q'\left(\Phi\middle|\Phi^{(n)}\right)$. When far from the optimum, the optimization procedure uses second-order information residing in the Hessian, H, of $Q'\left(\Phi\middle|\Phi^{(n)}\right)$; the corresponding step-size for the parameter matrix Φ is $H^{-1}\nabla_\Phi Q\left(\Phi\middle|\Phi^{(n)}\right)$. Central to the Levenberg-Marquardt method is the idea of exerting control over these two update modes using a positive parameter λ. This parameter defines the elements of the matrix Λ

$$\Lambda_{k,l} = \begin{cases} 1+\lambda & \text{if } k = l \\ 1 & \text{otherwise} \end{cases} \tag{42}$$

According to the Levenberg-Marquardt method, the step-size $\delta\phi_l$ for the parameter ϕ_l is found by solving the following set of linear equations

$$\sum_{l=1}^{3}\Lambda_{k,l}\frac{\partial^2 Q'\left(\Phi\middle|\Phi^{(n)}\right)}{\partial\phi_k\partial\phi_l}\delta\phi_l = \frac{\partial Q'\left(\Phi\middle|\Phi^{(n)}\right)}{\partial\phi_k}. \tag{43}$$

The parameter λ is chosen to be large if the log-likelihood increases when an optimization step is taken; in this case, the optimization process operates in steepest gradient mode. If, on the other hand, the expected likelihood decreases, then λ is reduced towards zero; in this case, the optimization process operates in inverse Hessian mode. When controlled effectively, this method is less prone to local convergence than the standard steepest gradient descent method, while offering efficiency gains over the inverse Hessian method. In fact, we find the Levenberg-Marquardt optimization converges in 5-10 iterations. Steepest-gradient takes up to 10 times longer to converge. Although our research-code is not optimized for efficient execution, typical registration experiments involving up to 100 points take tens of seconds on an SGI RS5000 workstation.

Before proceeding, it is important to offer a word of caution and to stress that degeneracies may be encountered in the estimation process if we confront affine data with a perspective model. We have not investigated this pathological case in our experiments.

Our final comment on the implementation concerns the execution speed of the algorithms. The main computational bottleneck is the evaluation of the structural matching probabilities. As outlined in Section 3.2, here, the main source of complexity is the size of the dictionary. Compared with this, the computation of transformation parameters represents a relatively small overhead. In the case of affine geometry, the recovery process can be realized rapidly using efficient matrix inversion. In the perspective case, parameter recovery is slower since iterative numerical optimization is needed. However, it must be stressed that we have not investigated the use of efficient search algorithms here because of the bottleneck associated with the dictionary prior. As a concrete example, when run on a SGI RS5000 processor, our registration method takes tens of minutes.

5 Experimental Results

In this section, we will provide experimental evaluation of our new coupled matching process. This investigation has two distinct strands. First, we will provide an algorithm sensitivity analysis. Here, we confine ourselves to the affine variant of the matching process. The aim is to experimentally compare our dual-step algorithm with the performance of each of its components taken individually. In other words, we provide comparison with *maximum a posteriori probability* correspondence matching and least squares parameter recovery. These two comparisons serve to demonstrate that the combined modeling of both point correspondences and transformation geometry yields significant advantages in terms of accuracy of convergence over their individual use.

In the second strand of our experimental investigation, we will furnish examples demonstrating the use of the dual-step matching scheme on real world imagery. Here, we will use two different data sets. The first of these involves perspective views of 3.5-inch floppy discs. The second example involves matching distorted aerial image data against a digital map.

5.1 Algorithm Comparison

The aim of this section is to demonstrate how the dual-step matching algorithm performs in comparison with the repeated iteration of its individual component parts. In this part of our study, we confine our attention to the matching of synthetic point-sets under affine geometry. The two algorithms used for comparison are

- **Maximum a posteriori probability correspondence matching:** In this set of experiments, we aim to demonstrate how our dual-step EM method performs in comparison with MAP correspondence matching using structural constraints [45]. The update process is realized by keeping the parameter matrix static at the value $\Phi^{(0)}$ and iteratively updating the discrete correspondence assignments. The update rule for the correspondence matches is

$$f^{(n+1)}(i) = \arg\max_{j \in \mathcal{M}} P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(0)}\right) \zeta_{i,j}^{(n)}. \tag{44}$$

  This aspect of our study aims to investigate the role played by the explicit modeling of the transformational geometry.

- **Maximum Likelihood Registration:** Here, we focus on the iterative estimation of affine registration parameters by applying the standard EM algorithm to the ungated log-likelihood function. This corresponds to clamping the structural gating probabilities with a uniform and static distribution over the complete space of possible correspondences during the iteration of the parameter estimation process. The update equation for the affine parameter matrix is

$$\Phi^{(n+1)} = \left[\sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(n)}\right) \vec{z}_j \, U^T \vec{z}_j^{\,T} \, \Sigma^{-1} \right]^{-1}$$

$$\times \left[\sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} P\left(\vec{z}_j \middle| \vec{w}_i, \Phi^{(n)}\right) \vec{w}_i \, U^T \vec{z}_j^{\,T} \, \Sigma^{-1} \right]. \tag{45}$$

  The aim here is to investigate the effect of structural gating on the estimation of transformation parameters.

The results of the comparative study have been obtained using random point sets. This allows us to compare algorithm sensitivity in a controlled manner under varying noise conditions. This experimental methodology also allows the results to be averaged over a large number of random experiments and meaningful error bars to be derived from the population statistics. Each of our reported datapoints is averaged over 100 random experiments. The errorbars are the standard errors over the set of trials.

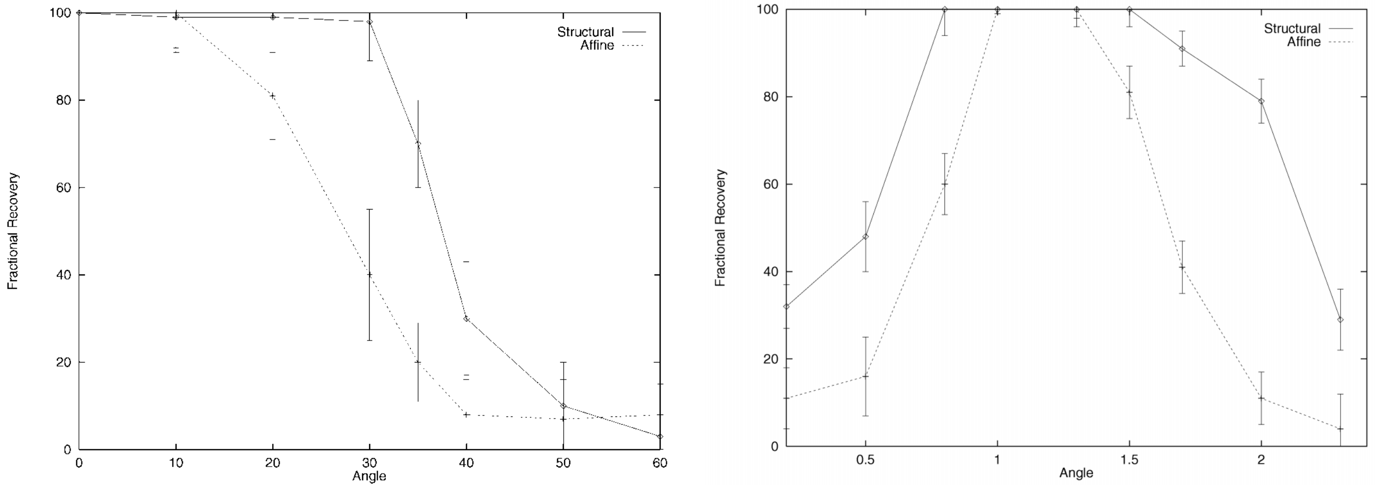
5.1.1 Comparison With Standard Structural Matching

In order to demonstrate the relative stability of the dual-step matching scheme under initial parameter choice, we will compare it with a structural graph matching scheme. The algorithm used in this comparison is essentially the discrete relaxation process of Wilson and Hancock [45]. This structural matching technique results solely from the iteration of the MAP update process defined in (34), leaving the parameter estimates static. The aim of our study is to demonstrate the sensitivity of our method to initial differences in isotropic image scale and overall point-set orientation. The study is based on random dot patterns in which the ground-truth transformational geometry is known.

In Figs. 3a and 3b, we respectively show the final fraction of points in correct correspondence as a function of the initial difference in orientation and scale. The dotted lines show the sensitivity of the standard structural matching scheme, while the solid lines are for the dual-step matching method. It is clear that our dual-step expectation-maximization approach consistently outperforms the standard relational matching scheme. In particular, the range of both rotation and scale over which the dual-step EM scheme successfully recovers meaningful results is significantly greater than that for the purely structural scheme. For instance, our dual-step method copes well with angle differences of up to 35 degrees, whereas the structural method must be initialized to within 10 degrees. In the case of the scale difference, the dual-step method copes with differences in the range 0.7 to 1.6, whereas the MAP scheme only functions effectively over the range 0.9 to 1.1. However, it must be stressed that the structural method could be rendered considerably more robust if affine invariant measures are used to compute the initial a posteriori matching probabilities.

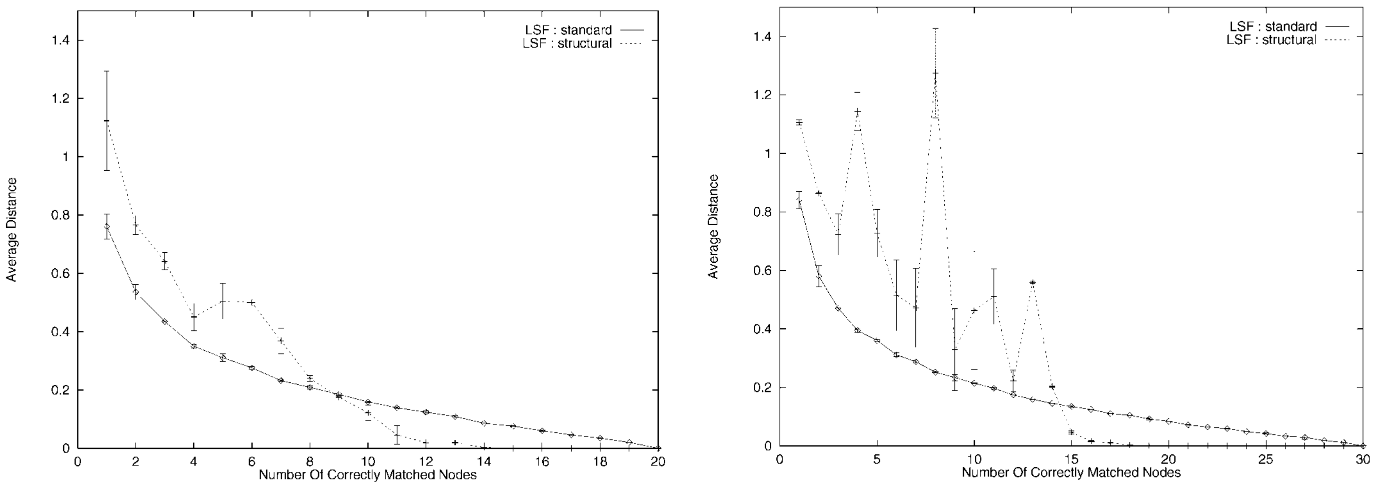
5.1.2 Comparison With Least Squares Fitting Algorithms

Our aim in this experiment is to demonstrate the effect of the structural component upon the recovery of affine transformation parameters. We commence our simulation study by generating random point sets. By applying Voronoi tessellation and Delaunay triangulation to the point-sets, we have constructed model graphs. The corresponding data-graphs are generated by deleting at random a controlled number of points from the original point-set and randomly re-inserting new points. The fraction of modified points is taken as a measure of structural corruption. Our measure of registration accuracy is the average magnitude of the, unweighted, distance between the points that are known to be in correspondence. Suppose that $T \subset \mathcal{M} \times \mathcal{D}$ is the set of ground-truth correspondences between the uncorrupted



(a) (b)

Fig. 3. Sensitivity study: The two plots show how the dual-step EM algorithm (solid curve) outperforms MAP-matching (dashed curve) in terms of the fraction of correct correspondence matches. (a) Affine rotation. (b) Affine scaling.



(a) (b)

Fig. 4. Comparing the dual-step matching scheme with a least squares approach: The plots compare the performance of the dual-step EM algorithm (dotted curve) with standard least-squares estimation (solid curve). (a) Graph size 20 nodes. (b) Graph size 30 nodes.

portion of the data-graph and the model. If n_∞ is the final iteration number for the matching algorithm, then the measure of registration accuracy is

$$A = \frac{1}{|T|}\sum_{(i,j)\in T}\left|\vec{w}_i - \vec{z}_j^{(n_\infty)}\right|. \tag{46}$$

It is important to stress that this figure of merit includes only unmodified points from the original model graphs. In other words, we exclude nodes deleted from the graphs in the re-triangulation step.

Fig. 4a shows a comparison of the new matching scheme (dashed curve) and the ungated maximum likelihood method (solid curve) of affine parameter estimation. Here, the size of original point-set is 20. Fig. 4b repeats this experiment when the size of the point-set is 30. The plots show the average registered point-distance as a function of the fraction of correct correspondence matches.

The main feature to note from these plots is that, provided sufficient correspondences are available, then the dual-step method outperforms the least-squares method in terms of the registration error. Moreover, provided that the degree of structural error is less than 50 percent, then the average registration error for the dual-step method is considerably smaller than for the conventional EM registration method. In fact, when this is the case, then an average point error of less than 0.01 is achievable.

This limiting degree of structural corruption deserves further comment. In Fig. 5, we show a plot of the fraction of supercliques containing a structural error as a function of the fraction of added noise points for the Delaunay graph. From this plot, it is clear that when the fraction of corrupt nodes is 50 percent, then 70 percent of the supercliques contain at least one corrupt edge. In other words, our dual-step EM approach outperforms its conventional counterpart provided that more than 30 percent of the structure of the Delaunay graph remains is intact. Although our matching method is dependent on structural information, it can tolerate a significant degree of structural corruption.

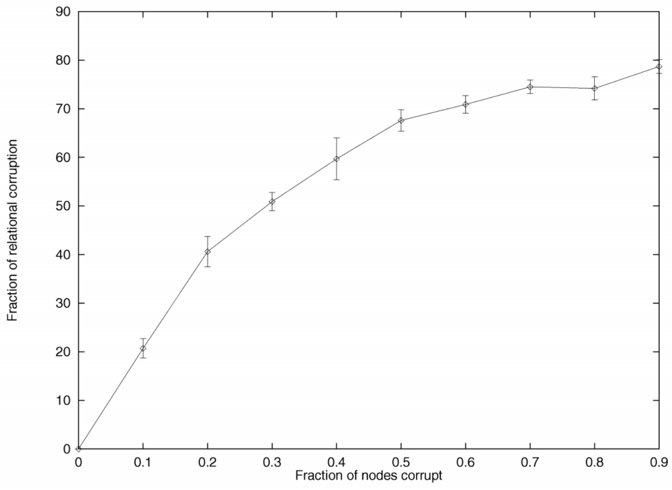


Fig. 5. Effect of adding controlled fractions of relational clutter to Delaunay graphs.

5.2 Real World Imagery

In order to demonstrate the effectiveness of the new matching process on real world imagery, we will consider the following two data-sets:

- **Disk Set:** This data set consists of a set of digital photographs of 3.5-inch floppy disks. This data-set was chosen since it allows for controlled shifts in viewpoint to be made. The images under investigation include both small viewpoint shifts that are nearly affine, and very large shifts where the controlled introduction of strong perspective foreshortening will be investigated.
- **Road Network:** In this experiment, we are concerned with the registration of aerial infrared images against a digital map. The images were taken at nighttime. Under these imaging conditions, the most prominent features are those that radiate absorbed heat. In the urban scenes under study, these features are the tarmac roads. We therefore chose the road networks as the basis for our graph structures. The nodes in our Delaunay graphs are junctions detected in the road network. It is important to note that these images are distorted due to the geometry of the line-scan process. The images are captured using horizontal line-scan as the aircraft moves in the vertical direction. The line-scan process is controlled by the rotation of a mirror. For this reason, the images are subject to barrel distortion in the horizontal-direction. In the vertical-direction, there are also sampling irregularities due to the aircraft changing heading due to banking or turbulence.

We will first consider the task of recognizing planar objects in different 3D poses, which is posed by the set of images of floppy disks. The object used in this study is placed on a desktop. The different object viewing angles are contrived so as to introduce increasing degrees of perspective foreshortening. The feature points used to triangulate the object are corners which are extracted by hand. Fig. 6 shows a sequence of object-views with the triangulations of the hand segmented feature-points superimposed. The first oblique view in the sequence is taken as the object-model; the remaining object-poses are used to test the matching process.

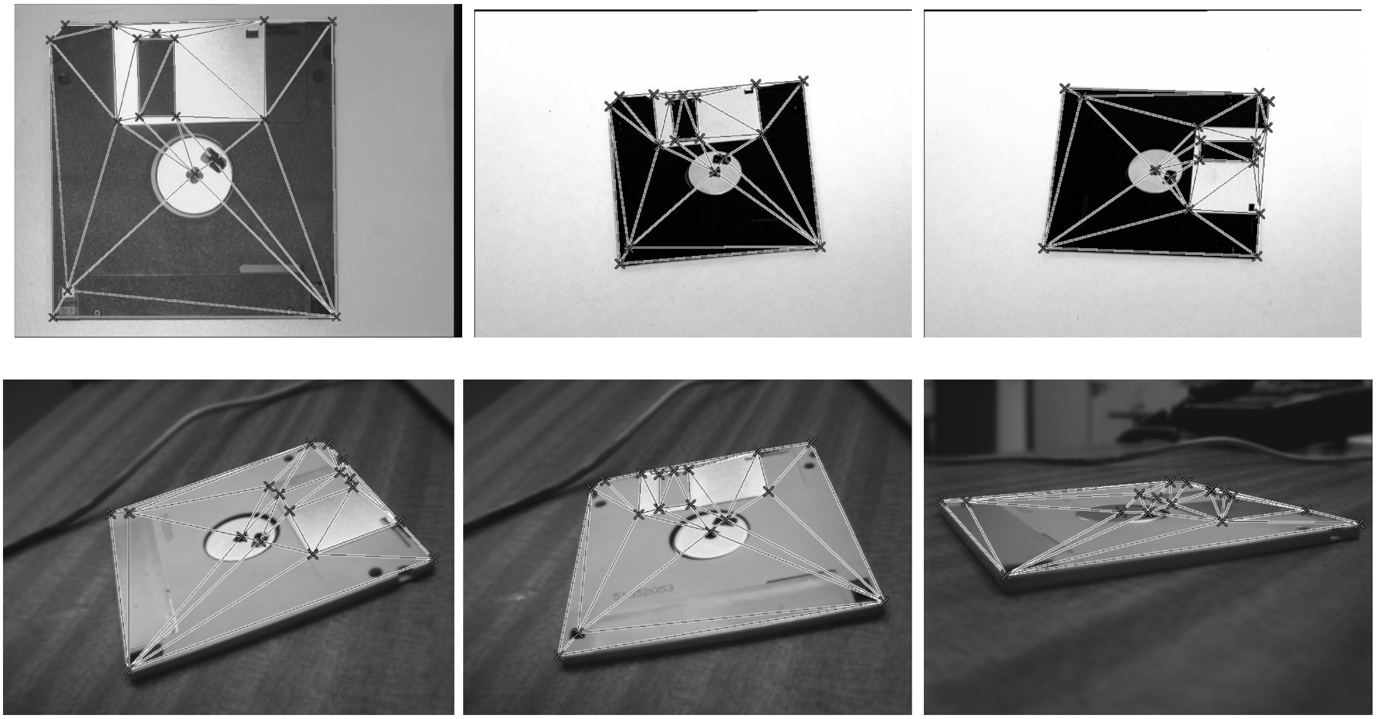


Fig. 6. The six views of used in the matching experiments.

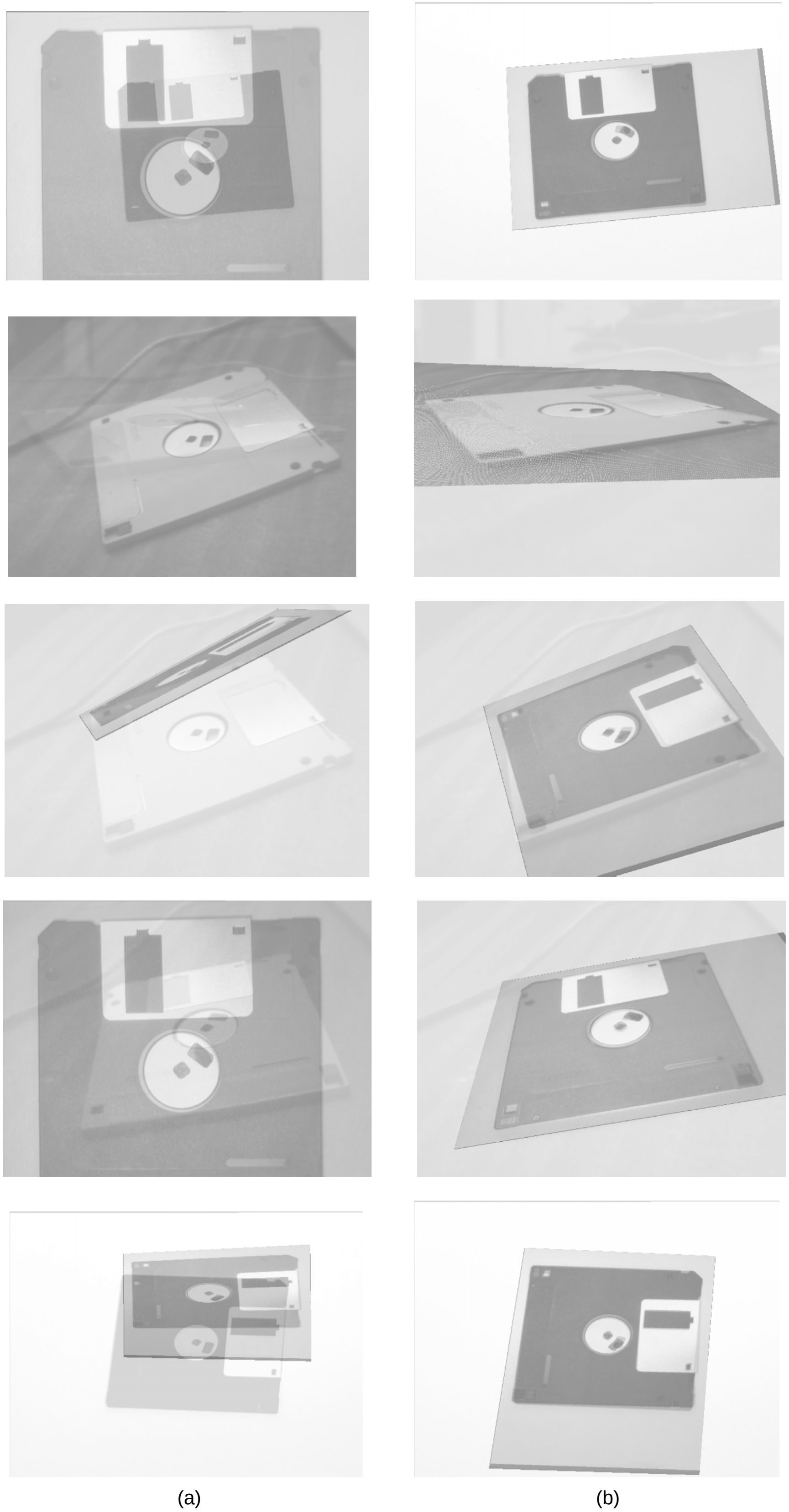


Fig. 7. Some initial (a) and final (b) registration configurations.

Fig. 7 shows the initial and final poses for the registration of various images in the dataset. In each case, the fraction of correct initial correspondences was approximately 50 percent. From the superimposed images, it is clear that the recovered final poses are accurate.

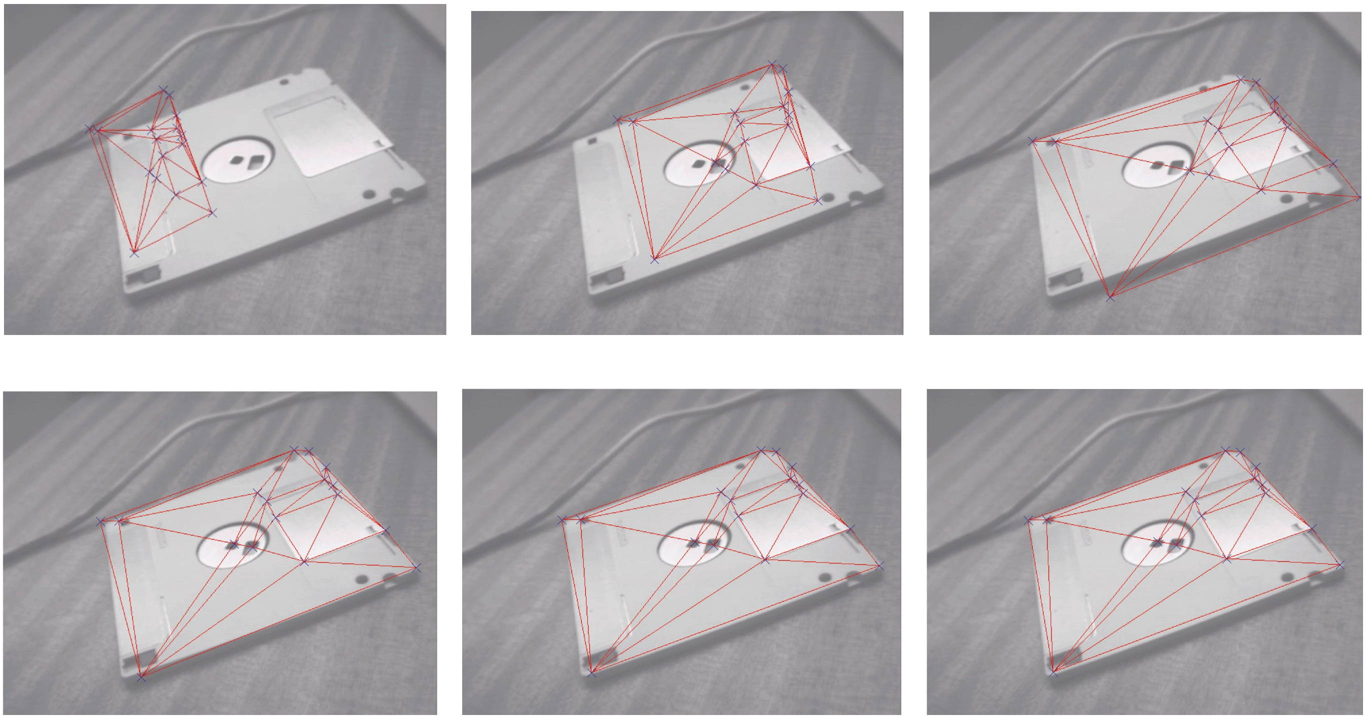


Fig. 8. The iterative registration of the model: The different iterations of the algorithm are ordered from left-to-right and top-to-bottom.

Next, we provide an illustration of the iterative properties of our matching algorithm. The sequence in Fig. 8 shows the iterative recovery of the full perspective geometry. Here, we illustrate the iterative registration of the fourth (taken to be the data-graph) and fifth (taken to be the model-graph) images in the test sequence. In this case, the direction of perspective foreshortening is rotated through 90 degrees with respect to the disc. Each state shows the model-graph of the fifth image mapped onto the fourth image using the current set of recovered perspective parameters. The initial parameters are selected at random. The registration process converges on a good solution after three iterations.

To illustrate that the matching process is not sensitive to structural errors, we now provide some examples when the data-graph used in the previous experiment is subsumed in clutter. Figs. 9 and 10 show the initial and final correspondence matches. In each case, the model-graph is on the left and the data-graph is on the right. The blue lines between the two halves of the figures represent correspondence matches between the graphs. Initially, there are many matching errors. Once the algorithm has converged, the consistency of the pattern of matches is significantly improved. Moreover, there are no erroneous matches to the extraneous clutter nodes. Finally, the model-pose is in good agreement with the corresponding points in the data.

To illustrate that our method is not sensitive to the accuracy of the assumed transformational geometry, we now provide some experiments which involve matching perspectively distorted images using only affine geometry. Here, we use the first floppy disc image as a model, while the fourth floppy disc image is again the data. Clearly, in this case, the affine transformation is insufficient to represent the image deformation. Fig. 11 shows the iterative registration for this experiment. The registration converges upon a pose that is a good approximation to the full perspective transformation in about six iterations. In Fig. 12, we show the Delaunay triangulation iterating in synchronization with the image registration of Fig. 11.

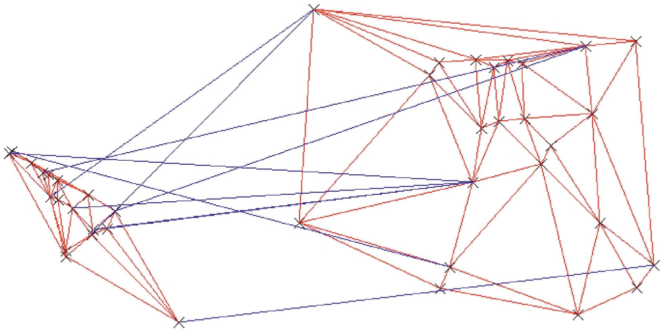


Fig. 9. Initial correspondence match for a cluttered scene.

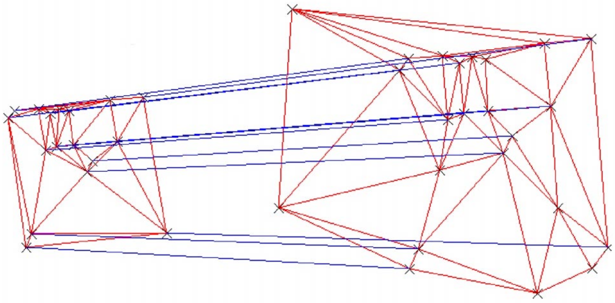


Fig. 10. Final correspondence match for a cluttered scene.

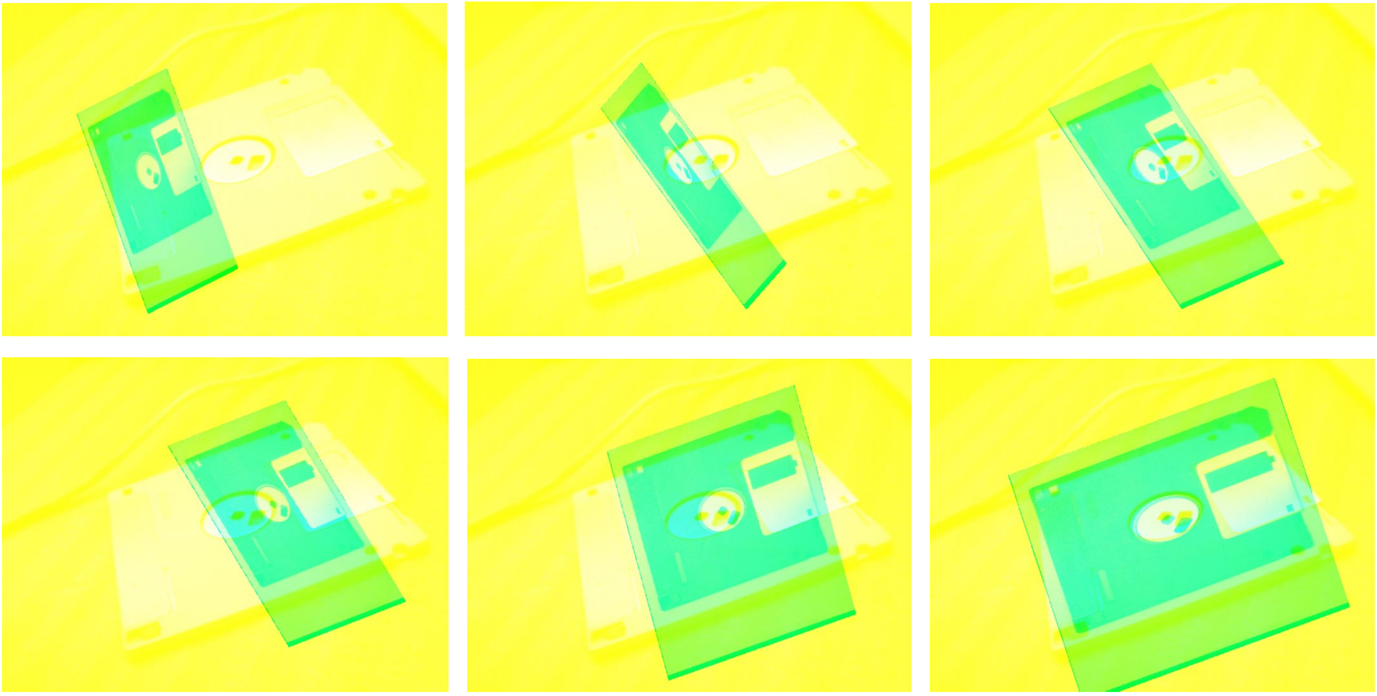


Fig. 11. Iterative convergence using an affine transformation: The different iterations of the algorithm are ordered from left-to-right and top-to-bottom.

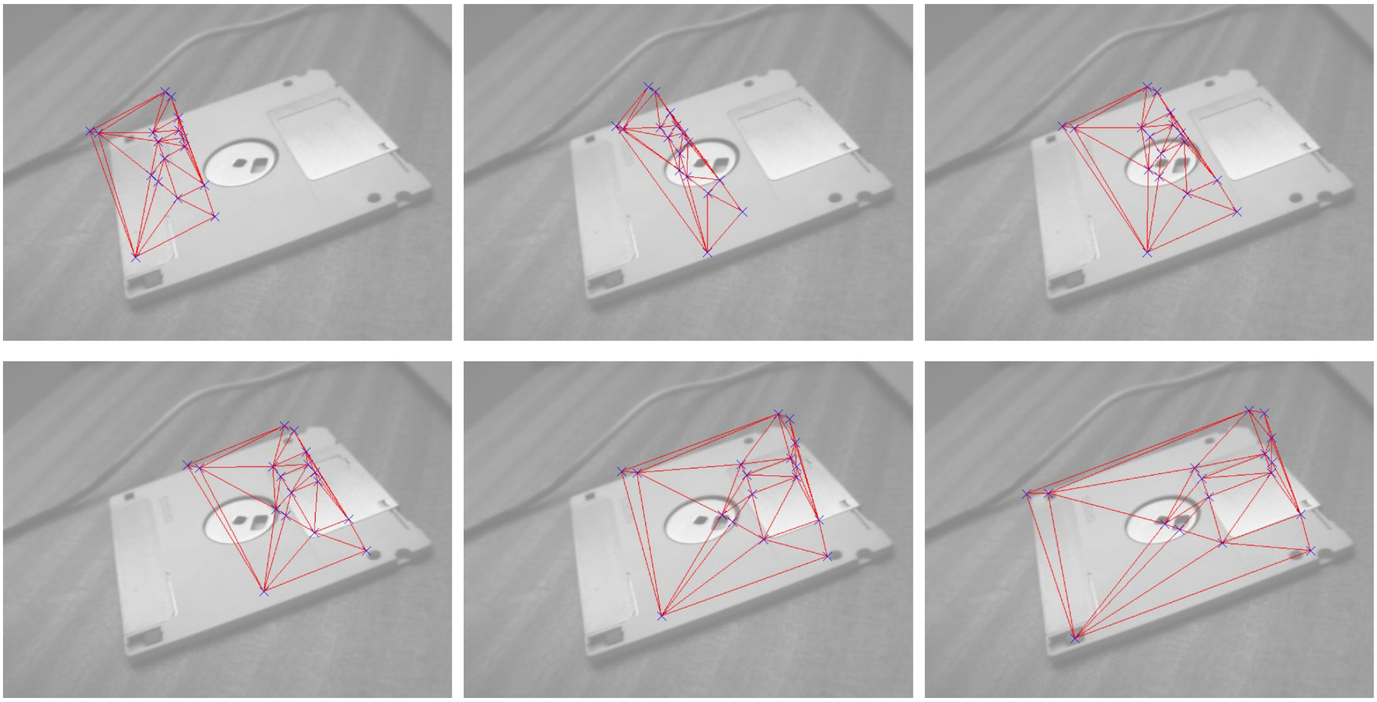


Fig. 12. The graphs iterating in synchronization with the registration process: The different iterations of the algorithm are ordered from left-to-right and top-to-bottom.

The final piece of experimentation involves the registration of a digital map against a set of aerial infrared images. Fig. 13 shows the map data together with the raw images used in this example. The salient structure in this imagery is a road network. The feature points used in our matching experiments are junctions in the road network. These points are used to seed the Delaunay triangulation. There are three factors which complicate the matching process. First, there are cartographic errors. As a result, there are features for which no correspondence exists even when the map is brought into exact registration with the images. Second, there is a significant amount of barrel distortion in these images. This process is not faithfully captured by our affine transformation model. Finally, the extracted Delaunay triangulations exhibit a significant degree of structural corruption. Fig. 13 shows the final affine transformations of the

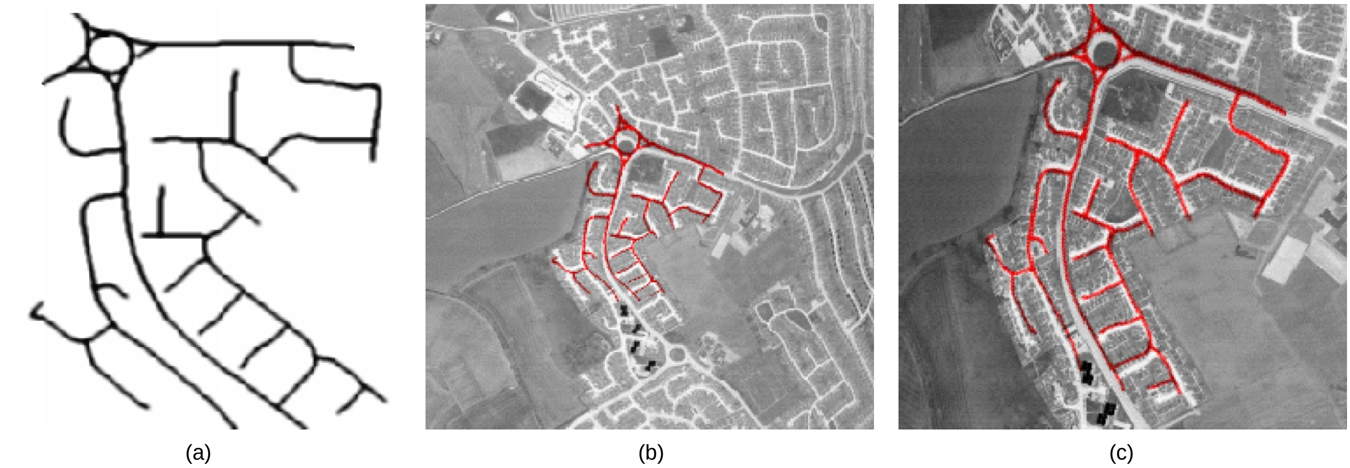


Fig. 13. Aerial image registration. (a) The digital map. (b) The registration with the high altitude image. (c) The registration with the low altitude image.

map superimposed on the different aerial images. The matching process commences from a random initial estimate of the affine transformation matrix. It is clear that the recovered transformations are reasonably accurate given the poor geometric model.

6 Conclusions

Our main contribution in this paper has been to develop a new synergistic matching algorithm. This two-step iterative process involves coupled operations to locate point-correspondences and estimate geometric transformation parameters. Point correspondences are located by maximum a posteriori graph-matching. Maximum likelihood parameters are recovered using the expectation-maximization algorithm. These coupled iterative processes communicate by exchanging separate pieces of matching information. The point-correspondences passed by the matching process improve the robustness of maximum likelihood parameter estimation. In their turn, the maximum likelihood parameters are used to estimate a posteriori measurement probabilities which improve the accuracy of the point-correspondences.

We illustrate the effectiveness of the resulting matching process under affine and perspective geometries. These transformations are of generic importance in computer vision with applications in image mosaicing, pose recovery and camera calibration. Here, the coupled matching process is shown to outperform structural matching. Moreover, the use of point-correspondences is shown to offer significant advantages in the control of added image noise.

In other words, we have presented a flexible matching method which unifies relational graph matching and pose-recovery. The framework is Bayesian and relies on some fairly nonrestrictive assumptions concerning the Gaussian origin of measurement errors and observational independence. Our future plans revolve around the use of improved optimization methods and more ambitious point-deformation models. In fact, we have recently taken some steps in this direction by demonstrating how the optimization of our relational consistency measure can be realized using soft-assign [14]. Finally, although we have demonstrated our matching process for the problem of image registration, the problem of calibration may well provide a more topical vehicle where our new technique could be compared with a number of promising alternatives [37], [32].

References

[1] N. Ahuja, “Dot Pattern Processing Using Voronoi Neighborhoods,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no. 3, pp. 336-343, 1982.

[2] N. Ahuja, B. An, and B. Schachter, “Image Representation Using Voronoi Tessellation,” *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 286-295, 1985.

[3] N. Ahuja and M. Tuceryan, “Extraction of Early Perceptual Structure in Dot Patterns: Integrating Region, Boundary and Component Gestalt,” *Computer Vision, Graphics, and Image Processing*, vol. 48, pp. 304-356, 1989.

[4] T.D. Alter, “3D Pose From Three Points Using Weak Persective,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 802-808, 1994.

[5] Y. Amit and A. Kong, “Graphical Templates for Model Registration,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 225-236, 1996.

[6] P.A. Beardsley, A. Zisserman, and D.W. Murray, “Navigation Using Affine Structure and Motion” *Proc. Third European Conf. Computer Vision*, pp. 85-96, 1994.

[7] J-D. Boissonnat, “Geometric Structures for Three-Dimensional Shape Representation,” *ACM Trans. Graphics*, vol. 3, pp. 266-286, 1984.

[8] T.F Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, “Active Shape Models—Their Training and Application,” *Computer Vision, Graphics and Image Understanding*, vol. 61, pp. 38-59, 1995.

[9] T.F Cootes and C.J. Taylor, “Combining Point Distribution Models With Shape Models Based on Finite-Element Analysis,” *Image and Vision Computing*, vol. 13, pp. 403-409, 1995.

[10] A.D.J. Cross and E.R. Hancock, “Perspective Pose Recovery With a Dual Step EM Algorithm,” *Advances in Neural Information Processing Systems*, M. Jordan, M. Kearns, and S. Solla, eds., vol. 10, pp. 780-786. Cambridge, Mass.: MIT Press, 1998.

[11] D.F. DeMenthon and L.S. Davis, “Exact and Approximate Solutions of the Perspective Three-Point Problem,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no, 11, pp. 1,100-1,105, 1992.

[12] A.P. Dempster, N.M. Laird, and D.B. Rubin, “Maximum-Likelihood From Incomplete Data via the EM Algorithm,” *J. Royal Statistical Soc. Series B* (*methodological*), vol. 39, pp. 1-38, 1977.

[13] O.D. Faugeras, E. Le Bras-Mehlman, and J-D. Boissonnat, "Representing Stereo Data With the Delaunay Triangulation," *Artificial Intelligence*, vol. 44, pp. 41-87, 1990.

[14] A.M. Finch, R.C. Wilson, and E.R. Hancock, "An Energy Function and Continuous Edit Process for Graph Matching," *Neural Computation*, vol. 10, pp. 1,873-1,894, 1998.

[15] S. Gold, A. Rangarajan, and E. Mjolsness, "Learning With Pre-Knowledge: Clustering With Point and Graph-Matching Distance measures," *Neural Computation*, vol. 8, pp. 787-804, 1996.

[16] R.M Haralick, C.N. Lee, K. Ottenberg, and M. Nolle, "Review and Analysis of Solutions of the Three-Point Perspective Pose Estimation Problem," *Int'l J. Computer Vision*, vol. 13, pp. 331-356, 1994.

[17] R.I. Hartley, "Projective Reconstruction and Invariants From Multiple Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, pp. 1,036-1,041, 1994.

[18] R.I. Hartley, "In Defence of the Eight-Point Algorithm," *Proc. Fifth Int'l Conf. Computer Vision*, pp. 1,064-1,070, 1995.

[19] F. Hempel, E. Ronchetti, R. Rousseeuw, and W. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. John Wiley, 1986.

[20] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy, "Object Pose: The Link Between Weak Perspective, Para-Perspective and Full Perspective," *Int'l J. Computer Vision*, vol. 22, pp. 173-189, 1997.

[21] M. Irani, P.Anandan, and S. Hsu, "Mosaic-Based Representations of Video Sequences and Their Applications," *Proc. Fifth Int'l Conf. Computer Vision*, pp. 605-611, 1995.

[22] D.W. Jacobs, "Optimal Matching of Planar Models in 3D Scenes," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 269-274, 1991.

[23] M.I. Jordan and R.A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation*, vol. 6, pp. 181-214, 1994.

[24] D.G. Kendall, "Shape Manifolds: Procrustean Metrics and Complex Projective Spaces," *Bulletin London Math. Soc.*, vol. 16, pp. 81-121, 1984.

[25] J.J. Koenderink and A.J. Van Doorn, "Affine Structure From Motion" *J. Optical Soc. Am.—Series A*, vol. 8, pp. 377-385, 1992.

[26] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. von der Maalsburg, R.P. Wurtz, and W.Konen, "Distortion-Invariant Object-Recognition in a Dynamic Link Architecture," *IEEE Trans. Computers*, vol. 42, no. 3, pp. 300-311, 1993

[27] D.P. McReynolds and D.G. Lowe, "Rigidity Checking of 3D Point Correspondences Under Perspective Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, pp. 1,174-1,185, no. 12, 1996.

[28] S. Moss and E.R. Hancock, "Registering Incomplete Radar Images With the EM Algorithm," *Image and Vision Computing*, vol. 15, pp. 637-648, 1997.

[29] D. Oberkampf, D.F. DeMenthon, and L.S. Davis, "Iterative Pose Estimation Using Coplanar Feature Points," *Computer Vision and Image Understanding*, vol. 63, pp. 495-511, 1996.

[30] H. Ogawa, "Labeled Point Pattern Matching by Delaunay Triangulation and Maximal Cliques," *Pattern Recognition*, vol. 19, pp. 35-40, 1986.

[31] C.J. Poelman and T. Kanade, "A Para-Perspective Factorization Method for Shape and Motion Recovery," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 206-218, 1997.

[32] M. Polleyfeys and L. Van Gool, "A Stratified Approach to Metric Self Calibration," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition Conference*, pp. 407-412, no. 3, 1997.

[33] G.L. Scott and H.C. Longuet-Higgins, "An Algorithm for Associating the Features of Two Images," *Proc. Royal Soc. London Series B-Biological*, vol. 244, pp. 21-26, 1991.

[34] S. Sclaroff and A.P. Pentland, "Modal Matching for Correspondence and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 545-661, 1995.

[35] L.S. Shapiro and J.M. Brady, "Feature-Based Correspondence—An Eigenvector Approach," *Image and Vision Computing*, vol. 10, pp. 283-288, 1992.

[36] L.S. Shapiro and J.M. Brady, "Rejecting Outliers and Estimating Errors in an Orthogonal-Regression Framework," *Phil. Trans. Royal Soc. A*, vol. 350, pp. 403-439, 1995.

[37] P. Torr and D.W. Murray, "The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix," *Int'l J. Computer Vision*, vol. 24, pp. 271-300, 1997.

[38] M. Tuceryan and T Chorzempa, "Relative Sensitivity of a Family of Closest Point Graphs in Computer Vision Applications," *Pattern Recognition*, vol. 25, pp. 361-373, 1991.

[39] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, Mass.: MIT Press, 1979.

[40] S. Umeyama, "An Eigen Decomposition Approach to Weighted Graph Matching Problems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 695-703, 1988.

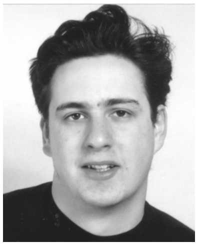
[41] S. Umeyama, "Least Squares Estimation of Transformation Parameters Between Point Sets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376-380, 1991.

[42] S. Umeyama, "Parameterized Point Pattern Matching and its Application to Recognition of Object Families," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 2, pp. 136-144, 1993.

[43] J. Utans, "Mixture Models and the EM Algorithms for Object Recognition Within Compositional Hierarchies," ICSI Berkeley Technical Report TR-93-004, 1993.

[44] R.C. Wilson, A.D.J. Cross and E.R. Hancock, "Structural Matching with Active Triangulations," *Computer Vision and Image Understanding*, vol. 72, pp. 21-38, 1998.

[45] R.C. Wilson and E.R. Hancock, "Structural Matching by Discrete Relaxation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 634-648, 1997.



Andrew D.J. Cross gained his BSc in computational physics with first-class honors from the University of Manchester Institute of Science and Technology in 1994. Between 1994 and 1998, he undertook research in the area of optimization methods for computer vision at the University of York. He was awarded the DPhil degree for this work in July 1998. Following a period of postdoctoral research at York, Dr. Cross took up an appointment with NewTek in San Antonio, Texas. His interests are in computer vision, graphics, and image processing.

Edwin R. Hancock gained his BSc in physics in 1977 and his PhD in high energy nuclear physics in 1981, both from the University of Durham, UK. After a period of postdoctoral research working on charm-photo-production experiments at the Stanford Linear Accelerator Center, he moved into the fields of computer vision and pattern recognition in 1985. Between 1981 and 1991, he held posts at the Rutherford-Appleton Laboratory, the Open University, and the University of Surrey. Dr. Hancock is currently a reader in the Department of Computer Science at the University of York. He leads a group of some 15 researchers in the areas of computer vision and pattern recognition. He has published more than 180 refereed papers in the fields of high-energy nuclear physics, computer vision, image processing, and pattern recognition. He was awarded the 1990 Pattern Recognition Society Medal and received an honorable mention in 1997. Dr Hancock serves as an associate editor of the journal *Pattern Recognition* and has been a guest editor for the *Image and Vision Computing* journal. He is currently guest-editing a special edition of the *Pattern Recognition* journal devoted to energy minimization methods in computer vision and pattern recognition. He chaired the 1994 British Machine Vision Conference and has been a program committee member for several national and international conferences.