Structural Graph Matching using the EM Algorithm and Singular Value Decomposition

Bin Luo
University of York,York, UK
and Anhui university, PR China.
luo@cs.york.ac.uk

Edwin Hancock
University of York,York, UK.
erh@cs.york.ac.uk

Abstract

This paper describes an efficient algorithm for inexact graph-matching. The method is purely structural, that is to say it uses only the edge or connectivity structure of the graph and does not draw on node or edge attributes. We make two contributions. Commencing from a probability distribution for matching errors, we show how the problem of graph-matching can be posed as maximum likelihood estimation using the apparatus of the EM algorithm. Our second contribution is to cast the recovery of correspondence matches between the graph-nodes in a matrix framework. This allows us to efficiently recover correspondence matches using singular value decomposition. We experiment with the method on both real-world and synthetic data. Here we demonstrate that the method offers comparable performance to more computationally demanding methods.

1 Introduction

Graph-matching is a task of pivotal importance in high-level vision since it provides a means by which abstract pictorial descriptions can be matched to one-another. Since the process of eliciting graph structures from raw image data is a task of some fragility due to noise and the limited effectiveness of the available segmentation algorithms, graph-matching is invariably approached by inexact means [10] An importnnt idea here has been to use edit-distance to compare graphs [10] and it has recently been shown that the edit distance is related to the size of the maximum common subgraph [2]. Another powerful way to deal with inexactness is to model the structural errors present in the graph-matching problem in a probabilistic way. Wong and You [15] made one of the first contributions here by defining an entropy measure for structural graph-matching. Boyer and Kak [1] also adopted an information theoretic approach, but worked instead with attribute relations. Using a probabilistic relaxation framework Christmas, Kittler and Petrou [3] have developed a statistical model for pairwise attribute relations. Working in the purely structural domain, Wilson and Hancock [14] have derived probability distributions for the relational errors that occur when there is significant graph corruption. Once a measure of graph similarity is to hand then the search for the set of correspondence matches may be posed as an optmisation or energy minimisation problem.

Another important method which draws ideas from the field of mathematics known as spectral graph theory [4], is to cast the graph-matching problem in a matrix setting and to use the eigenvalues and eigenvectors of the adjacency matrix as a representation of relational structure. For instance, Umeyama has an eigendecomposition method that matches graphs of the same size [13]. Borrowing ideas from structural chemistry, Scott and Longuet-Higgins were among the first to use spectral methods for correspondence analysis [11]. They showed how to recover correspondences via singular value decomposition on the point association matrix between different images. In keeping more closely with the spirit of spectral graph theory, yet seemingly unaware of the related literature, Shapiro and Brady [12] developed an extension of the Scott and Longuet-Higgins method which performs multidimensional scaling on the point-set proximity matrices to extract a feature-vector for matching. Horaud and Sossa[8] have adopted a purely structural approach to the recognition of line-drawings. Their representation is based on the immanental polynomials for the Laplacian matrix of the line-connectivity graph. By comparing the coefficients of the polynomials, they are able to index into a large data-base of line-drawings.

Although formally elegant, the main limitation of these matrix methods is their inability to cope with graphs of different sizes. This means that they can not be used when significant levels of structural corruption are present. Moreover, there has been little attempt to render them robust using probabilistic or statistical methods. Based on these observations our aim in this paper is to cast the statistical matching of graphs into a matrix representation and to exploit singular value methods to efficiently recover correspondences. We commence by developing a likelihood function for the graph-matching problem. This treats the graph to be matched (the data-graph) as observed data and the set of correspondences with the available model (the model-graph) as hidden variables. Accordingly, we construct a mixture model over the set of correspondences between the nodes of the data-graph and those of the model-graph. We adopt a Bernoulli model for the probability distribution of the correspondence errors encountered in matching the data-graph to the model-graph. The existence or otherwise of correspondence errors is gauged using the edge-consistency of the pattern of matches.

2 Likelihood Function

Our overall goal in this paper is to develop a maximum likelihood framework for structural graph matching. In this section we develop the likelihood function underpinning our study. To commence we must define some notation. We use the notation $G = (V, E)$ to denote the graphs under match, where V is the set of nodes and E is the set of edges. Our aim in matching is to associate nodes $V_D = \{x_1, x_2, ...x_{|V_M|\}}$ in a graph $G_D = (V_D, E_D)$ representing data to be matched against those from the set $V_M = \{y_1, y_2, ..., y_{|V_M|}\}$ in a graph $G_M = (V_M, E_M)$ representing an available model. Formally, the matching is represented by a function from the nodes in the data graph G_D to those in the model graph G_M. Suppose that the state of match between the two graphs is represented by the function $f : V_D \to V_M$ from the nodes of the data-graph to those of the model-graph. We will use latin letters to denote nodes from the data-graph and greek letters to denote nodes from the model-graph. Hence, the statement $f^{(n)}(a) = \alpha$ means that the node $a \in V_D$ is assigned the label or symbol $\alpha \in V_M$.

One of the goals in this paper is to show how the two graphs can be matched using matrix factorisation methods. We therefore introduce some matrix notation to represent the graphs. To this end we define a $|V_D| \times |V_M|$ matching matrix $S^{(n)}$ whose elements are assignment variables which convey the following meaning

$$s_{a\alpha} = \begin{cases} 1 & \text{if } f(a) = \alpha \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We represent the structure of the two graphs using a $|V_D| \times |V_D|$ adjacency matrix D for the data graph and a $|V_M| \times |V_M|$ adjacency matrix M for the model graph. The elements of the adjacency matrix for the data graph are defined as follows

$$D_{ab} = \begin{cases} 1 & \text{if } (a,b) \in E_D \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

while those for the model graph are defined to be

$$M_{\alpha\beta} = \begin{cases} 1 & \text{if } (\alpha,\beta) \in E_M \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Since we are working with undirected graphs, the two adjacency matrices are symmetric, i.e. $D = D^T$ and $M = M^T$.

Having introduced the necessary formalism, we now proceed to develop our maximum likelihood framework for graph-matching. We seek the matrix of assignment variables that maximises the conditional likelihood of the observed data-graph given the available model graph. Hence, we seek the matrix of assignment variables which satisfies the condition

$$S = \arg\max_{\hat{S}} P(G_D | G_M, \hat{S}) \tag{4}$$

Next we construct a mixture model over the set of possible correspondences. We follow the standard approach to constructing the likelihood function for a mixture distribution. This involves factorising the likelihood function over the observed data (i.e. the nodes of the data-graph) and summing over the hidden or unobserved variables (i.e. the corresponding nodes in the model-graph). As a result we write

$$P(G_D, G_M | S) = \prod_{a \in V_D} \sum_{\alpha \in V_M} p(x_a | y_\alpha, S) \tag{5}$$

where $p(x_a|y_\alpha, S)$ is the probability that data-graph node a is in correspondence with the model-graph node α under the matrix of assignment variables S.

In order to proceed, we require a model for the observation density $p(x_a|y_\alpha, S)$. We commence from the assumption that the observation density is factorial over the parameters of the mixture model, i.e. the set of assignment variables. If this is the case, then we can write

$$P(x_a | y_\alpha, S) = \prod_{b \in V_D} \prod_{\beta \in V_M} P(x_a | y_\alpha, s_{b\beta}) \tag{6}$$

Next we develop a model for the probability distribution for the observed set of correspondences between the nodes of the data and the model graphs given the current set of assignment parameters, i.e $P(x_a|y_\alpha, s_{b\beta})$. Our model draws on the recent work of Wilson and Hancock [14] and assumes that the observed data-graph nodes are derived from the model-graph nodes through a Bernoulli distribution. The parameter of this distribution is the probability of correspondence error P_e. The idea behind this model is that the modal-graph node x_a can emit a symbol y_α drawn from the set of model-graph nodes. The probability that this symbol is the correct correspondence is $1 - P_e$ while the probability that it is in error is P_e. To gauge the correctness of the emitted symbol, we check whether the nodes a and b of the data-graph are matched to a valid edge $(\alpha, \beta) \in E_m$ of the model-graph. To test for edge-consistency, we make use of the quantity

$$D_{a,b} M_{\alpha\beta} s_{b\beta} = \begin{cases} 1 & \text{if } (a,b) \in E_D \text{ and } (\alpha,\beta) \in E_M \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Using this switching property, the Bernoulli distribution becomes

$$P(x_a | y_\alpha, s_{b\beta}) = (1 - P_e)^{D_{ab} M_{\alpha\beta} s_{b\beta}} P_e^{1 - D_{ab} M_{\alpha\beta} s_{b\beta}} \tag{8}$$

With the factorial assumption and the distribution rule to hand, the observation density becomes

$$P(x_a | y_\alpha, S) = \prod_{b \in V_D} \prod_{\beta \in V_M} (1 - P_e)^{D_{ab} M_{\alpha\beta} s_{b\beta}} P_e^{1 - D_{ab} M_{\alpha\beta} s_{b\beta}} \tag{9}$$

This expression is exponential in character. It can be rewritten in as a natural exponential function

$$P(x_a | y_\alpha, S) = K \exp\left[\mu \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab} M_{\alpha\beta} s_{b\beta}\right] \tag{10}$$

where $\mu = \ln \frac{1 - P_e}{P_e}$ and $K = P_e^{|V_D| \times |V_M|}$. Finally, the corresponding log-likelihood function for the assignment matrix is

$$\mathcal{L}(S) = \tag{11}$$

$$\sum_{a \in V_D} \log\left\{ \sum_{\alpha \in V_M} K \exp\left[\mu \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab} M_{\alpha\beta} s_{b\beta}\right]\right\}$$

Unfortunately, because of the mixture structure the direct estimation of the matrix of assignment variables S from the log-likelihood function is not tractable in closed form. For this reason, in the next section we explain how the expectation-maximisation algorithm may be used instead.

3 Expectation-Maximisation

Having developed our computational model which poses the graph-matching problem in a maximum-likelihood framework, in this section we provide a concrete algorithm for recovering the parameters of the underlying mixture-model. We choose to use the EM algorithm originally introduced by Dempster, Laird and Rubin [5]. The utility measure underpinning the algorithm is the expected log-likelihood function. The basic idea underlying the algorithm is to iterate between the interleaved expectation and maximisation 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 maximisation phase, the model parameters are recomputed to maximise the expected value of the incomplete data likelihood.

3.1 Expected log-likelihood function

For our graph-matching problem, maximisation of the expectation of the conditional likelihood is equivalent to maximising the weighted log-likelihood function

$$\Lambda(S^{(n+1)}|S^{(n)}) = \sum_{a\in V_D}\sum_{\alpha\in V_M} P(y_\alpha|x_a, S^{(n)}) \ln P(x_a|y_\alpha, S^{(n+1)}) \tag{12}$$

where $S^{(n)}$ indicates the matrix of assignment variables taken at iteration n of the EM algorithm. Hence, the *a posteriori* correspondence matching probabilities computed at iteration n, i.e. $P(y_\alpha|x_a, S^{(n)})$ are used to weight the iteration $n+1$ contributions to the log-likelihood function.

With the expected log-likelihood function to hand, the maximum-likelihood matrix of assignment variables is the one which satisfies the condition

$$S^{(n+1)} = \arg\max_{\hat{S}} \Lambda(\hat{S}|S^{(n)}) \tag{13}$$

One way to realise the update process is by parallel iterative local gradient ascent. In the next section we show how the expected log-likelihood function can be recast in a matrix framework. This allows us to realise the update procedure more efficiently using singular value decomposition.

3.2 Matrix Representation

To commence, we note that when the distribution function for the assignment variables is substituted from Equation (10) the expected log-likelihood function becomes

$$\Lambda(S^{(n+1)}|S^{(n)}) = \sum_{a,b;\alpha,\beta} Q^{(n)}_{a\alpha}\left\{\ln K + \mu D_{ab} M_{\alpha\beta} s^{(n+1)}_{b\beta}\right\} \tag{14}$$

where we have introduced the $|V_D| \times |V_M|$ matrix $Q^{(n)}$ whose elements $Q^{(n)}_{a\alpha} = P(y_\alpha|x_a, S^{(n)})$ are set equal to the *a posteriori* probability of correspondence match between the data-graph node a and the model-graph node α at iteration n of the EM algorithm.

The critical quantity in determining the update direction for maximum likelihood matches is

$$\hat{\Lambda}(S^{(n+1)}|S^{(n)}) = \sum_{a,b;\alpha,\beta} Q^{(n)}_{a\alpha} D_{ab} M_{\alpha\beta} s^{(n+1)}_{b\beta} \tag{15}$$

In matrix form the expected log-likelihood is

$$\hat{\Lambda}(S^{(n+1)}|S^{(n)}) = Tr[D^T Q^{(n)} M (S^{(n+1)})^T] \tag{16}$$

3.3 Maximisation

The maximisation step of the EM algorithm can be stated as that of recovering the set of correspondence indicators $S^{(n+1)}$ which satisfies the condition

$$S^{(n+1)} = \arg\max_{S} Tr[D^T Q^{(n)} M S^T] \tag{17}$$

In other words, the utility measure gauges the degree of correlation between the edge-sets of the two graphs under the weighted permutation structure induced by the correspondence probabilities.

To locate the updated set of correspondence indicators we use the extremum principal reported by Scott and Longuet-Higgins [11]. Their result is as follows. Suppose that G is a positive definite $|V_D| \times |V_M|$ matrix. They have shown how the $|V_D| \times |V_M|$ orthogonal matrix R that maximises the quantity $Tr[GR^T]$ may be found by performing singular value decomposition. To do this they perform the matrix factorisation $G = V\Delta U^T$, where V is a $|V_D| \times |V_D|$ orthogonal matrix, U is a $|V_M| \times |V_M|$ orthogonal matrix and Δ is a $|V_D| \times |V_M|$ matrix whose diagonal elements $\Delta_{i,j} = 0$ if $i \neq j$ and whose "diagonal" elements $\Delta_{i,i}$ are non-zero. Suppose that E is the matrix obtained from Δ by making the diagonal elements $\Delta_{i,i}$ unity. The matrix R which maximises $Tr[GR^T]$ is $R = VEU^T$. This extremum principle may be applied to our graph matching problem if we make the substitution $G = D^T Q^{(n)} M$ and perform the singular value decomposition $D^T Q^{(n)} M = V\Delta U^T$ to obtain R. This matrix satisfies the condition

$$R = \arg\max_{\hat{R}} Tr[D^T Q^{(n)} M \hat{R}^T] \tag{18}$$

Provided that the matrix $D^T Q^{(n)} M$ is positive-definite, then the elements of R are real.

Although this extremum principle is useful, it is not entirely suited to our needs. The reasons for this are that the elements of R can not be interpreted as probabilities since they are neither guaranteed to be positive, nor are they normalised. Furthermore, they can not be interpreted as assignment indicators since they are not binary in nature. To overcome these difficulties, we follow Scott and Longuet-Higgins by testing the elements of R to obtain a matrix of binary correspondence indicators $S^{(n+1)}$. If the element $R_{a,\alpha}$ is the maximum value for both the row and column that contains it, then the assignment indicator $s^{(n+1)}_{a,\alpha}$ is set to unity. Otherwise it is set to zero. As a result the updated set of correspondence indicators is

$$s^{(n+1)}_{a\alpha} = \begin{cases} 1 & \text{if } R_{a\alpha} = \arg\max_{b\beta} R_{b\beta} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

3.4 Expectation

In the expectation step of the EM algorithm, the *a posteriori* probabilities of the hidden data are computed from the component densities appearing in the mixture-distribution. This is done by applying the Bayes theorem. At iteration $n+1$ we have

$$P(y_\alpha|x_a, S^{(n+1)}) = \frac{p(x_a|y_\alpha, S^{(n)})\pi^{(n)}_\alpha}{\sum_{\alpha\in V_M} p(x_a|y_\alpha, S^{(n)})\pi^{(n)}_\alpha} \tag{20}$$

where

$$\pi^{(n)}_\alpha = \frac{1}{|V_D|}\sum_{a\in V_D} P(y_\alpha|x_a, S^{(n)}) \tag{21}$$

4 Experiments

In this section of the paper, we provide some experimental evaluation of the new graph-matching technique. There are two aspects to this study. We commence with a sensitivity study using synthetic data. The aim here is to evaluate how the new method performs under controlled structural corruption and to compare it with some alternatives reported elsewhere in the literature. The second part of the study evaluates the method on real-world data.

4.1 Sensitivity Study

Our sensitivity study is divided into two parts. We compare our method with some alternative methods for inexact graph-matching which rely on matrix factorisation techniques. These methods do not work when the graphs are of different size. Here we keep the graphs of fixed equal size and investigate the effect of corrupting the pattern of edges.

4.1.1 Inexact Graph Matching

We commence by studying the effect of controlled structural error on the graphs being matched. The graphs used in our study are the Delaunay triangulations of randomly generated point-sets. The effects of structural error are simulated by deleting a predefined fraction of randomly selected nodes and re-triangulating the remaining points.

We compare the performance of our new matching method with three alternatives. These are the dictionary-based relaxation scheme of Wilson and Hancock [14], the quadratic assignment method of Gold and Rangarajan [7] and the non-quadratic graduated assignment method of Finch, Wilson and Hancock [6]. Figure 1 compares the four algorithms. Here we show the fraction of correct correspondences as a function of the fraction of nodes deleted from the graphs. The main feature to note is that the new graph matching method delivers performance that is intermediate between the discrete relaxation method and the non-linear graduated assignment method. This is an interesting observation when we compare the computational overheads associated with the three methods.

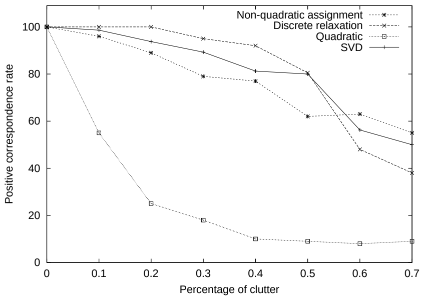

Figure 1. Sensitivity study for graphs of different size.

4.1.2 Factorisation Methods

In this subsection we provide comparison with two methods for weighted graph-matching which share with our own method the feature of relying on matrix factorisation. The methods selected for this comparison are

- Umeyama's weighted graph-matching method which seeks the permutation matrix $\mathcal{P}$ that minimises quantity $J(\mathcal{P}) = ||\mathcal{P}M - D||$ [13]. The method performs the singular value decompositions $M = U_M \Delta_M U_M^T$ and $D = U_D \Delta_D U_D^T$, where the U's are orthogonal matrices and the Δ's are diagonal matrices. Once these factorisations have been performed, the required permutation matrix is $\mathcal{P} = U_D U_M^T$.
- Shapiro and Brady's [12] weighted graph-matching method which uses the modal structure of the two weighted adjacency matrices D and M. The modal structure of the two adjacency graphs is found by solving the eigenvalue equation $D\phi_l^D = \lambda_l \phi_l^D$. where λ_l is the l^{th} eigenvalue of the adjacency matrix D and ϕ_l^D is the corresponding eigenvector. The eigenvectors are ordered according to the size of the associated eigenvalues and are used as the columns of the modal matrix $\Phi_D = \left(\phi_1^D, \phi_2^D, \phi_3^D, ...\right)$. This procedure is repeated to construct a second modal matrix Φ_M for the model-graph adjacency matrix M. The column index of these two modal matrices refers to the order of the eigenvalues while the row-index is the index of the nodes in the graphs. Shapiro and Brady find correspondences by locating pairs of rows which have minimum distance, i.e.

$$s_{a,\alpha} = \begin{cases} 1 & \text{if } \alpha = \arg\min_{\alpha'} \sum_{l=1}^{N} ||\Phi_D(a,l) - \Phi_M(\alpha',l)||^2 \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

These two methods rely on weighted adjacency matrices rather than the binary ones defined earlier. To conduct our experiments, we have generated random 2D point-sets. We use the positions of these points to generate the weights of the adjacency matrix. Suppose that $\vec{x}_\alpha^M$ and $\vec{x}_\beta^M$ represent the co-ordinate vectors associated with the nodes indexed a and b. The weight associated with the edge connecting the nodes is

$$M_{\alpha\beta} = \exp\left[-k||\vec{x}_\alpha^M - \vec{x}_\beta^M||^2\right] \quad (23)$$

These two methods are not effective when the graphs under study contain different numbers of nodes. To compare with our method we have therefore kept the number of points fixed and have added Gaussian errors to the point positions. The parameter of the noise process is the standard deviation of the positional jitter. In our experiments, we express this parameter as a fraction of the average minimum distance between points (the relative standard deviation). It is important to stress that the methods compared here use different representations of the arrangement of the points. The Shapiro and Brady, and Umeyama methods use the weighted adjacency matrix. Our method, on the other hand, uses a binary adjacency matrix to represent the Delaunay triangulation of the points.

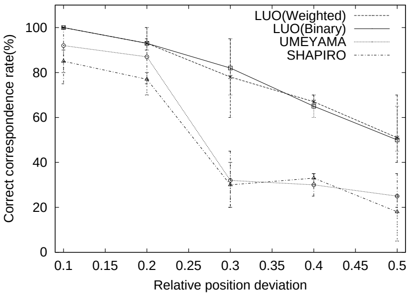

Figure 2. Comparison of the four eigendecomposition methods for graphs with the same number of nodes.

In Figure 2 we show the fraction of correct correspondences as a function of the relative standard deviation for our new method (bold curve), Umeyama's [13] method (solid curve) and the method of Shapiro and Brady [12] (dotted curve). The main feature to note is that our method outperforms the two alternatives. There is little to distinguish the performance of the Shapiro and Brady [12], and Umeyama [13] methods. Both fail abruptly once the relative standard deviation exceeds 0.2, i.e. the noise standard deviation is greater than 20% of the average closest point distance. Our method, on the other hand, degrades almost linearly with the noise standard deviation. However, it must be stressed that the results are not completely comparable. In the case of Shapiro and Brady, and Umeyama [13], we are measuring the sensitivity of the method to noise on the entries of the weighted adjacency matrices. In the case of our method, we are measuring the sensitivity of the method to errors in the edge-sets of the graphs used for matching.

Finally, we illustrate the results obtained when we apply our method to the weighted adjacency matrix rather than the binary adjacency matrix. The dot-dashed curve in Figure 1 shows the fraction of correct correspondences as a function of the relative standard deviation of the point-position jitter. The method performs considerably better than the Shapiro and Brady, and Umeyama methods. However, there is little to distinguish its performance from that obtained with the binary adjacency matrix.

4.2 Real-world data

We commence our real-world evaluation of the graph-matching method on images of indoor scenes. Here we are concerned with matching the Delaunay triangulations of corner-features. We use the corner detector recently reported by [9] to extract point features. Figure 3 shows two examples of the indoor images used in our study. Superimposed on the images are the detected corners and their associated Delaunay triangulations. The two images are taken from different viewpoints. There is rotation, scaling and perspective distortion present. Moreover, several of the objects in the scene are at different depths and move relative to one-another. As a result there are significant structural differences in the two Delaunay graphs. Figure 4 shows the correspondences between the corners as lines between the two images. After checking by hand, the fraction of correct correspondences is 77%.

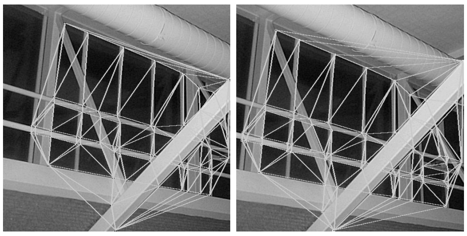

Figure 3. Test images overlayed with Delaunay graphs.

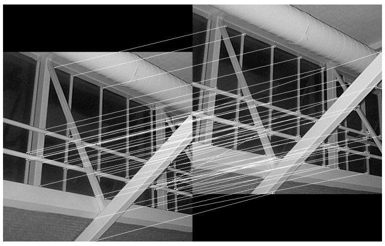

Figure 4. Correspondences.

We have performed our experiments using images taken from the CMU/VASC model-house sequence. The images used in our study are shown in Figure 5 and correspond to different camera viewing directions. The detected corner features and their Delaunay triangulations are overlayed on the images. There are clearly significant structural differences in the graphs. Figure 6 shows the results obtained when pairs of images in the sequence are matched. The results are summarised in Table 1. Here we list the number of detected corners in the images being matched, the number of corners that are in correct correspondence, the number of corners that are in error, and the number of corners for which there are no correspondences (i.e. there is no row and column maximum). The method breaks down after the 4th image in the sequence.

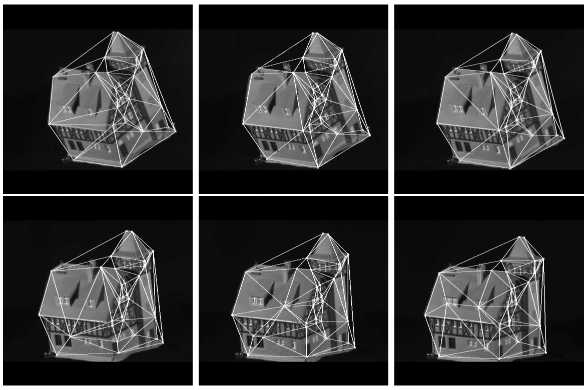

Figure 5. Delaunay graphs overlayed on the toy house images.

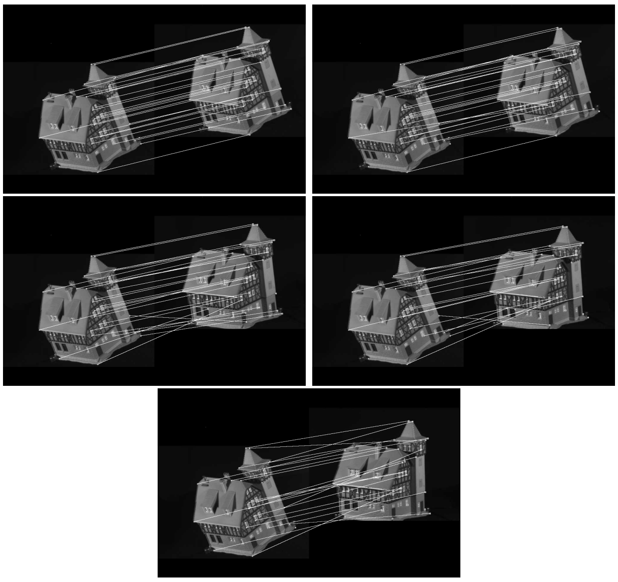

Figure 6. Correspondences for pairs of images with increasing difference in viewing angle.

To provide some comparison, we have selected a pair of images which contain the same number of corner points (image 2 and image 4). Although the number of corners is the same, there are differences in the both identities of the detected points and their structural arrangement. For these images we compare the matches returned by the unweighted and weighted versions of our algorithm (referred to as Luo), the method of Umeyama and the method of Shapiro and Brady. The results are shown in Figures 7 and 8 and the numbers of correct matches are summarised in

Table 2. From these results it is clear that the new method returns considerably better matches.

Images	Corners	Correct	False	Unmatched
house 1	30	–	–	–
house 2	32	29	0	1
house 3	32	28	1	1
house 4	32	23	5	2
house 5	34	11	10	9
house 6	33	5	16	9

Table 1. Summary of experimental results for the house sequence images.

Methods	Correct	False	Unmatched
Luo(Weighted)	22	6	4
Luo(Unweighted)	22	6	4
Umeyama	6	11	15
Shapiro	6	11	15

Table 2. Summary of the comparison of the three matching algorithms.

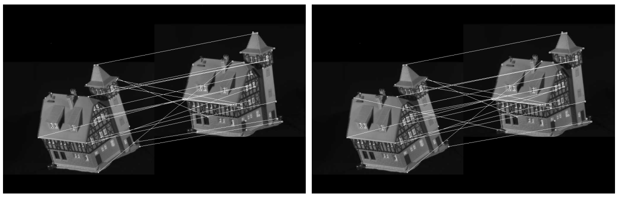

Figure 7. Correspondences from the Umeyama (left) and Shapiro (right) algorithms.

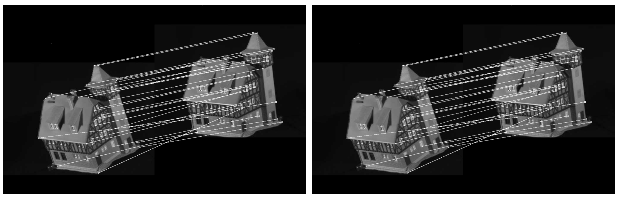

Figure 8. Correspondences from the unweighted (left) and weighted (right) variants of our algorithm.

5 Conclusions

Our main contributions in this paper are twofold. First, we have cast the problem of graph-matching into a maximum likelihood framework by constructing a mixture model over the set of hidden correspondences and adopting a Bernoulli model for the distribution of edge-matching errors. Second, we have used the apparatus of the EM algorithm to show how the problem of estimating the correspondence indicators may be cast into a compact matrix setting. This allows us to use singular value decomposition to estimate the correspondence indicators in the M-step. The result is an efficient algorithm that can be used to accurately match inexact graphs under considerable levels of structural corruption.

When viewed from the perspective of recent work on matrix-based graph-matching, the important contribution of this paper is to show how point-sets of different sizes can be matched using singular value decomposition.

References

[1] K. Boyer and A. Kak. Structural Stereopsis for 3D Vision. *IEEE PAMI*, 10:144–166, 1988.

[2] H. Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE PAMI*, 21:917–922, 1999.

[3] W.J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE PAMI*, 17(8):749–764, 1995.

[4] F.R.K. Chung. *Spectral Graph Theory*. American Mathmatical Society Ed., CBMS series 92, 1997.

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Soc. Ser. B (methodological)*, 39:1–38, 1977.

[6] A.M. Finch, R.C. Wilson, and E.R. Hancock. An energy function and continuous edit process for graph matching. *Neural Computation*, 10(7):1873–1894, 1998.

[7] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE PAMI*, 18(4):377–388, 1996.

[8] R. Horaud and H. Sossa. Polyhedral object recognition by indexing. *Pattern Recognition*, 28(12):1855–1870, 1995.

[9] B. Luo, A.D.J. Cross, and E.R. Hancock. Corner detection via topographic analysis of vector potential. *Pattern Recognition Letters*, 20:635–650, 1999.

[10] A. Sanfeliu and K.S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. Systems, Man and Cybernetics*, 13(3):353–362, May 1983.

[11] G.L. Scott and H.C. Longuet-Higgins. An Algorithm for Associating the Features of 2 Images. *Proceedings of the Royal Society of London Series B-Biological*, 244(1309):21–26, 1991.

[12] L.S. Shapiro and J. M. Brady. Feature-based Correspondence - An Eigenvector Approach. *Image and Vision Computing*, 10:283–288, 1992.

[13] S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *IEEE PAMI*, 10:695–703, 1988.

[14] R.C. Wilson and E.R. Hancock. Structural matching by discrete relaxation. *IEEE T-PAMI*, 19(6):634–648, June 1997.

[15] A.K.C Wong and M You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE PAMI*, 7:509–609, 1985.