Graph Matching using Spectral Embedding and Semidefinite Programming

Xiao Bai, Hang Yu, Edwin R. Hancock
Computer Science Department
University of York

Abstract

This paper describes how graph-spectral methods can be used to transform the node correspondence problem into one of point-set alignment. We commence by using the ISOMAP algorithm to embed the nodes of a graph in a low-dimensional Euclidean space. With the nodes in the graph transformed to points in a metric space, we can recast the problem of graph-matching into that of aligning the points. Here we use semidefinite programming to develop a variant of the Scott and Longuet-Higgins algorithm to find point correspondences. We experiment with the resulting algorithm on a number of real-world problems.

1 Introduction

Graphs are general and powerful data structures that can be used to represent both physical objects and abstract concepts. The graph matching problem is a generic one in computer vision, arising whenever a match must be found between the features of two related patterns. There is a considerable literature on the problem of graph matching. Broadly speaking, the work can be divided into three areas. The first of these is concerned with defining a measure of relational similarity [18] [9] [6]. The second issue is that of how to develop more principled statistical measures of similarity [20]. The third one is that of optimization [14] [16]. However, the main challenge in graph matching is how to deal with differences in node and edge structure. There have been successful attempts to use spectral methods for both structure graph matching [2] [16], and for point pattern matching [4] [10]. For instance Umeyama [16] has developed a least-squares approach and showed how an eigendecomposition method can be used to recover the permutation matrix by using a singular value decomposition of the adjacency matrices. Scott and Longuet-Higgins [4], on the other hand, align point-sets by performing singular value decomposition on a point association weight matrix. Shapiro and Brady [10] have reported a correspondence method which relies on measuring the similarity the eigenvector of a Gaussian point-proximity matrix.

Although Umeyama's algorithm [16] is elegant in its formulation and can deal with both weighted or unweighted graphs, it can not be applied to graphs which contain different numbers of nodes and for weighted graphs the method is susceptible weight errors. One way to overcome these problems is to cast the problem of recovering correspondences in a statistical setting using the EM algorithm [11]. However, the resulting algorithm is time consuming because of its iterative character. In this paper we take an alternative view

BMVC 2004 doi:10.5244/C.18.32

of spectral matching algorithms to develop an efficient method that is robust to differences in structure. Spectral methods can be viewed as embedding the nodes of a graph in a space spanned by the eigenvectors of the adjacency matrix. In the case of the Umeyama algorithm [16], matching is effected by finding the transformation matrix that best aligns the embedded points. The Shapiro and Brady [10] algorithm finds correspondences by seeking the closest embedded points. Kosinov and Caelli [15] have improved this method by allowing for scaling in the eigenspace.

Our aim in this paper is to seek an embedding of the nodes of a graph which allows matching to be effected using simple point-pattern matching methods. In the mathematics literature, there is a considerable body of work aimed at understanding how graphs can be embedded on a manifold so as to minimize a measure of distortion. Broadly speaking there are three ways in which the problem can be addressed. First, the graph can be interpolated by a surface whose genus is determined by the number of nodes, edges and faces of the graph. Second, the graph can be interpolated by a hyperbolic surface which has the same pattern of geodesic (internode) distances as the graph [7]. Third, a manifold can be constructed whose triangulation is the simplicial complex of the graph [1]. A review of methods for efficiently computing distance via embedding is presented in the recent paper of Hjaltason and Samet [5]. Recently, there has been considerable interest in the pattern analysis community in how to embed complex relational data in a low dimensional manifold. Collectively [8, 13, 3], these methods are known as manifold learning theory. Their collective aim is to develop variants of the classical methods of PCA and MDS, that can be used to better capture localised variations on the structure of the data.

In this paper we investigate whether methods from manifold learning theory can be combined with spectral graph theory to develop effective tools for graph structure matching. The idea is to use manifold learning methods to embed the graphs in a low dimensional coordinate space, and to use point-pattern matching techniques to find correspondences between nodes. We proceed as follows. We commence by using a spectral method to compute geodesic distances between nodes. This analysis relies on the heat kernel of the graph. We then use a strategy similar to ISOMAP [8] to embed the graphs in a Euclidean pattern space. This is done by applying MDS (Multidimensional Scaling) [17] to the matrix of geodesic distances between nodes. Once embedded in this space, we can use point-alignment methods to match the nodes of the graphs. To do this we develop a variant of the Scott and Longuet-Higgins [4] algorithm. There are two problems with the existing method. First, the correspondences are obtained from the matrix formed by taking the outer-product of the singular vectors of the point association matrix. This matrix does not have a clear interpretation, and the resulting search for correspondences using the maximum row and column entries is ad-hoc in nature. Second, the method breaks down when the point-sets being matched are of very different sizes. Here rather than performing singular value decomposition on the inter-point-set proximity matrix, we use semidefinite programming to locate correspondences. By using SDP we overcome the first of these problems, since it leads to a correspondence matrix which is doubly stochastic, and hence has clear meaning. Second, the resulting matching method is more robust to size difference than that of Scott and Longuet-Higgns.

2 Metric Embedding of Graphs

We are interested in the abstract problem of embedding the nodes of a graph into a Euclidean space. Here we use Isomap [8] as a way to solve the low-distortion graph embedding problem. The idea behind Isomap is to apply classical MDS [17] to the matrix of geodesic distances between data-points. In this way the data is mapped from a high-dimensional input space to the low-dimensional space of a nonlinear manifold. Although the method was originally devised for dimensionality reduction, we can use it here for the low-distortion graph embedding problem. Here we use a spectral method to approximate the geodesic distances between nodes.

2.1 Geodesic Distances from Graph Spectra

In this section, we develop method for approximating the geodesic distance between nodes by exploiting the spectral properties of the Laplacian matrix. To commence, suppose that the graph under study is denoted by $G = (V,E)$ where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. Since we wish to adopt a graph-spectral approach we introduce the adjacency matrix A for the graph where

$$A(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We also construct the diagonal degree matrix Deg, whose elements are given by $Deg(u,u) = \sum_{v \in V} A(u,v)$. From the degree matrix and the adjacency matrix we construct the Laplacian matrix $L = Deg - A$, i.e. the degree matrix minus the adjacency matrix. The spectral decomposition of the Laplacian matrix is $L = \Phi \Lambda \Phi^T$ where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the diagonal matrix with the ordered eigenvalues as elements and $\Phi = (\phi_1 | \phi_2 | .... | \phi_{|V|})$ is the matrix with the ordered eigenvectors as columns.

We are interested in the heat equation associated with the Laplacian, i.e. $\frac{\partial h_t}{\partial t} = -Lh_t$ where h_t is the heat kernel and t is time. The solution is found by exponentiating the Laplacian eigenspectrum, i.e. $h_t = \Phi \exp[-t\Lambda]\Phi^T$. The heat kernel is a $|V| \times |V|$ matrix, and for the nodes u and v of the graph G the resulting component is

$$h_t(u,v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i(u)\phi_i(v) \tag{2}$$

When t tends to zero, then $h_t \simeq I - Lt$, i.e. the kernel depends on the local connectivity structure or topology of the graph. If, on the other hand, t is large, then $h_t \simeq \exp[-t\lambda_m]\phi_m\phi_m^T$, where λ_m is the smallest non-zero eigenvalue and ϕ_m is the associated eigenvector, i.e. the Fiedler vector. Hence, the large time behavior is governed by the global structure of the graph.

It is interesting to note that the heat kernel is also related to the path length distribution on the graph. If $D_k(u,v)$ is the number of paths of length k between nodes u and v then

$$h_t(u,v) = \exp[-t] \sum_{k=1}^{|V|^2} D_k(u,v) \frac{t^k}{k!} \tag{3}$$

The path-length distribution is itself related to the eigenspectrum of the Laplacian. By equating the derivatives of the spectral and path-length forms of the heat kernel it is straightforward to show that

$$D_k(u,v) = \sum_{i=1}^{|V|} (1-\lambda_i)^k \phi_i(u)\phi_i(v) \tag{4}$$

We can use this result to compute the distances between nodes in the graph. The distance $d(u,v)$ is found by searching for the minimum value of k that results in a non-zero number of paths between nodes.

2.2 Metric Embedding using Isomap

Our goal is to find a low-distortion or distortion-free embedding from the graph metric space into a normed space. Here we use Isomap [8] as a way to solve the low-distortion graph embedding problem. The idea behind Isomap is to apply classical MDS [17] to map data points from their high-dimensional input space to low-dimensional coordinates of a nonlinear manifold. The key contribution is hence to apply MDS to the pairwise distances not in the input Euclidean space, but in the geodesic space of the manifold.

Although the method was originally devised for dimensionality reduction, we can use it here for the low-distortion graph embedding problem. Viewed as an isometric feature mapping, Isomap is a mapping $f : X \rightarrow Y$ from the observation space X to a Euclidean feature space Y that preserves as closely as possible the intrinsic metric structure of the observations, i.e. the distances between observations as measured along geodesic(shortest) paths of X [8]. The distortion in this embedding is nearly 1.

For graphs, the embedding procedure is straightforward. We first construct the shortest path distance matrix S for each graph. Each element $d_{u,v}$ in S is the shortest path distance between the pair of nodes u and v of the graph. We embed each graph in a Euclidean space by performing MDS on the matrix S.

The pairwise geodesic distances between nodes $d(u,v)$ are used as the elements of an $N \times N$ dissimilarity matrix S, whose elements are defined as follows

$$S(u,v) = \begin{cases} d(u,v) & \text{if } u \neq v \\ 0 & \text{if } u = v \end{cases} \tag{5}$$

The first step of MDS is to calculate a matrix T whose element with row r and column c is given by $T(r,c) = -\frac{1}{2}[d^2(r,c) - \hat{d}^2(r,.) - \hat{d}^2(.,c) + \hat{d}^2(.,.)]$, where $\hat{d}(r,.) = \frac{1}{N}\sum_{c=1}^{N} d(r,c)$ is the average dissimilarity value over the rth row, $\hat{d}(.,c)$ is the similarly defined average value over the cth column and $\hat{d}(.,.) = \frac{1}{N^2}\sum_{r=1}^{N}\sum_{c=1}^{N} d(r,c)$ is the average similarity value over all rows and columns of the similarity matrix T.

We subject the matrix T to an eigenvector analysis to obtain a matrix of embedding co-ordinates X. If the rank of T is $k, k \leq N$, then we will have k non-zero eigenvalues. We arrange these k non-zero eigenvalues in descending order, i.e. $l_1 \geq l_2 \geq \ldots \geq l_k > 0$. The corresponding ordered eigenvectors are denoted by $\vec{u}_i$ where l_i is the ith eigenvalue. The embedding co-ordinate system for the graphs obtained from different views is $X = [\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_s]$, where $\vec{f}_i = \sqrt{l_i}\vec{u}_i$ are the scaled eigenvectors. For the graph-node indexed u, the embedded vector of co-ordinates is $\vec{x}_i = (X_{u,1}, X_{u,2}, ..., X_{u,s})^T$.

3 Semidefinite Programming for Graph Matching

By applying Isomap to the two graphs to be matched, we obtain two point sets I and J, containing m and n features respectively. We now follow a way similar to Scott and Longuet-Higgins' method. We regard the points in I and J as lying in the same plane. We then represent the '*proximities*' between the features in I and the features in J. We use the Gaussian form

$$G_{ij} = exp(-m_{ij}^2/2\delta^2) \tag{6}$$

to compute the matrix of proximity weights. In the equation, m_{ij} is the Mahalanobis distance between two nodes, which is

$$m_{ij}^2(M) = (x_i - x_j)'\Sigma^{-1}(x_i - x_j) \tag{7}$$

and Σ is the point set covariance matrix. The use of the Mahalanobis metric instead of the Euclidean distance has several advantages. First, it automatically accounts for the scaling of the coordinate axes. Second, it corrects for correlation between the different features. Third, it can provide curved as well as linear decision boundaries.

With an inter-graph node distance matrix to hand, then one way to find correspondences is to use the Scott and Longuet-Higgins algorithm [4]. This involves performing the singular value decomposition (SVD) $G = TDU$. The matrices of T and U are orthogonal. The matrix D contains the singular values along its diagonal in descending numerical order. The final step is to compute the correlation between T's rows and U's columns, giving an association matrix $P = TEU$, where E is obtained by replacing each diagonal element in D by a 1. The element P_{ij} indicates the strength of attraction between feature $i \in I$ and $j \in J$. The rows of P, index the features in the first graph, and its columns those in the second graph. If P_{ij} is both the largest element in row i and column j then we regard these features as being in one-to-one correspondence with one-another. If P_{ij} is the greatest element in row i but not the greatest in column j, then we may regard $i \in I$ competing unsuccessfully for partnership with $j \in J$. Similar remarks apply if P_{ij} is the greatest element in its column but not in its row [4].

However, the Scott and Longuet-Higgins method can prove to be sensitive to instabilities in the singular vectors. For this reason we turn to semidefinite programming as an alternative. The semidefinite programming problem(SDP) is essentially an ordinary linear program where the nonnegativity constraint is replaced by a semidefinite constraint on matrix variables. It is interesting to note that SDP is a special instance of a more general problem class called *conic linear programs*, where one seeks to minimize a linear objective function subject to linear constraints and a cone constraint [19]. The process has many applications, ranging from control theory to structural design. In particular, many hard optimization problems can be relaxed to a problem with convex quadratic constraints which, in turn, can be formulated as an SDP [12]. The handbook [21] has described the application of SDP on combinatorial optimization, on nonconvex quadratic programming, on eigenvalue and nonconvex optimization, etc.

We seek the matrix P that best correlates with G in the sense of maximizing the inner product:

$$P : G = \sum_i \sum_j P_{ij} G_{ij} = trace(P^T G)$$

By using the *lifting process*, we define the variable matrix X

$$X = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix}$$

where x is obtained by vectorizing the matrix P and the problem can be transformed into the following

$$\begin{array}{lll} \min & C \bullet X & \\ s.t. & F_i \bullet X = a_i & i = (1...mn) \\ & ROW_j \bullet X = b_j & j = (1....n) \\ & COL_k \bullet X = c_k & k = (1....m) \\ & X \succeq 0 & \end{array} \tag{8}$$

here we introduce the constraint matrices

$$F_i = \begin{bmatrix} 0 & \cdots \frac{1}{2} & \cdots & 0 \\ 0 & \cdots 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \\ \frac{1}{2} & \cdots -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots 0 & \cdots & 0 \\ 0 & \cdots 0 & \cdots & 0 \end{bmatrix} i = (1...mn)$$

and

$$ROW_i = \begin{bmatrix} 0 & \cdots \frac{1}{2} & \cdots \frac{1}{2} & 0 \\ 0 & \cdots 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \\ \frac{1}{2} & \cdots 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{2} & \cdots 0 & \cdots & 0 \\ 0 & \cdots 0 & \cdots & 0 \\ 0 & \cdots 0 & \cdots & 0 \end{bmatrix} (i = 1...m)$$

the analogous matrices can be defined for column constraints.

In this case, we will finally obtain the association matrix P that maximizes trace $P^T G$ from X, and the solution matrix will be limited to be doubly stochastic.

4 Experiments

In this section, we provide some experimental evaluation of the new graph-matching method. There are two aspects to this study. First, we compare our method with some alternative methods by using synthetic data. Second, we evaluate our method on real-world data.

We commence with some synthetic data experiments. The aim is to evaluate how the new method works under controlled structural corruption and to compare it with some alternative methods. These alternatives are Shapiro and Brady [10] and Scott and Longuet-Higgins' [4] feature set matching methods. These two methods use coordinate information for the feature points, and do not incorporate the graph structure information. We also investigated Umeyama's [16] method. In our method we are concerned with matching the Delaunay triangulation of corner-features. Besides the coordinates of the feature points the Delaunay graph will incorporate important structural information.

Our first experiment is based on synthetic data. We have generated random point-sets containing 30 2D points. We use the position of the points to generate a Delaunay graph. We have 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 Figure 4, we show the fraction of correct correspondences as a function of the noise standard deviation for our method, Shapiro and Brady's [10] method, Umeyama's method and Scott and Longuet-Higgins' method [4]. To take this study one step further in Figure 3, we investigate the effect of structural noise. Here we have added a controlled fraction of additional nodes at random positions and have recomputed the Deluanay triangulations. We plot the fraction of correct correspondences as a function of the fraction of added nodes. The plot compares the result of applying our method to the data, and the results obtained using Scott and Longuet-Higgins' method, and Shapiro and Brady's method. Since Umeyama's algorithm can not handle graphs of different size, we have not compared with this method. The main feature to note is that our method outperforms the two alternatives. This means our method can solve the structure matching problem when the graphs are of different size.

To take this study one step further, we perform some real-world data experiments. We apply our matching method to two image sequences (MOVI and Desk). There are rotation, scaling, and perspective distortion present. Example images from these sequences are shown in Fig 1 and correspond to different camera viewing directions. The detected feature points and their Delaunay triangulations are overlayed on the images. The first four images are from the MOVI sequence and each contain about 140 nodes. The second four images are from Desk sequence and each contain about 400 nodes.

In Fig 2 ,we test our method on some pairs of images. In Table 1 we summarize the matching results for the MOVI houses. Here we list the number of nodes in the Delaunay graphs, the number of correct correspondence, the number of correspondence errors, and the number of points without correspondence. We also selected a pair of images which contain the same number of corner points (image 1 and image 4 from MOVI sequence 140 nodes). Although the number of corners is the same, there are differences in the both identities of the detected points and their structural arrangement. We compared these images' matching results by using our algorithm, the Umeyama's algorithm, Scott and Longuet-Higgins' algorithm and Shapiro and Brady's method. The compare results are summarized in Table 2. From these results, it is clear that our new method is better than these ones.

Images	Points	Correct correspondence	False correspondence	No correspondence
house1	140	-	-	-
house2	134	112	8	14
house3	130	109	6	15
house5	140	110	8	22

Table 1: Experiments Results for MOVI House Sequence Images

Methods	Correct correspondence	False correspondence	No correspondence
Our Method	110	8	22
Umeyama	84	30	26
Scott and Longuett-Higgins	97	17	26
Shapiro and Brady	83	17	40

Table 2: Summary of Comparison of the Four Matching Algorithms

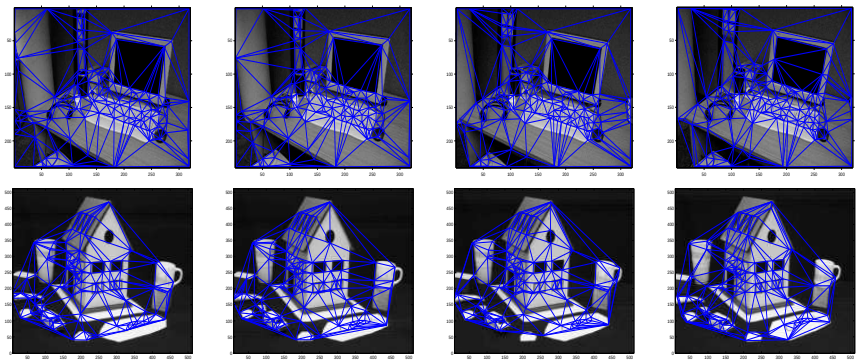

Figure 1: Delaunay graphs overlayed on the house images

5 Conclusion and Future Work

This paper has presented an efficient approach to graph structure matching. The approach is to first use the Isomap algorithm to embed the graphs in a Euclidean space by using the geodesic distance between nodes. Each node is hence transformed to a point in co-ordinate space. We match points corresponding to nodes in different graphs by using Semidefinite Programming .

In our experiments we have demonstrated the feasibility of the proposed method and have applied it to real-world data. Our algorithm can also be applied to other types of data. We plan to extend our work in several directions these are: first apply this algorithm to trees and try to solve the correspondence problem for trees, second to exploit the feasibility of using the embedded vector representations for indexing purposes, third we plan to use alternative methods for low distortion metric embedding and compare their effects.

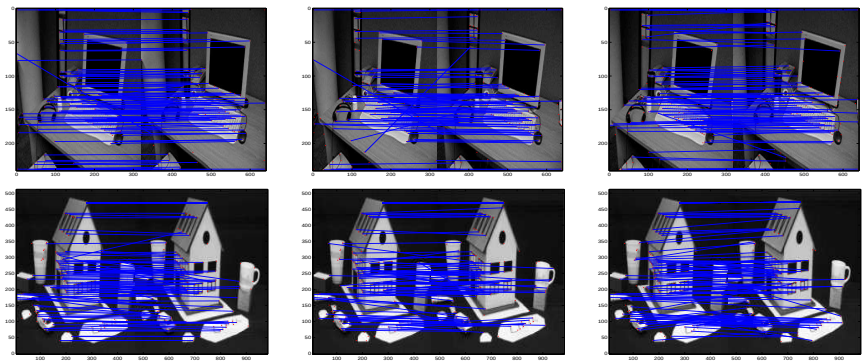

Figure 2: Our algorithm for CMU and MOVI house sequences

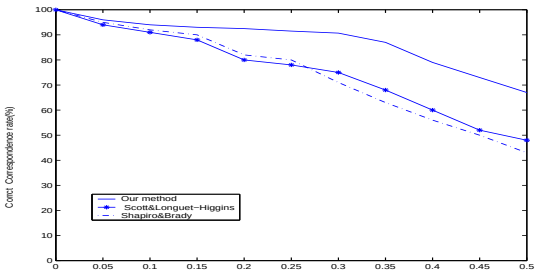

Figure 3: Correspondences for three methods for graph matching with differing numbers of nodes

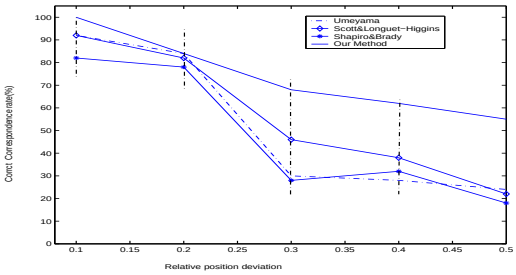

Figure 4: Comparison of four methods for graphs with same number of nodes

References

[1] A.Ranicki. Algebraic l-theory and topological manifolds. *Cambridge University Press*, 1992.

[2] A.Shokoufandeh, S.Dickinson, K.Siddiqi, and S.Zucker. Indexing using a spectral encoding of topological structure. *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 491–497, 1999.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Neural Information Processing Systems*, 14:634–640, 2002.

[4] G.L.Scott and H.C.Longuett-Higgins. An algorithm for associating the features of two images. *Proceedings of the Royal Society of London B*, 244:21–26, 1991.

[5] G.R.Hjaltason and H.Samet. Properties of embedding methods for similarity searching in metric spaces. *PAMI*, 25:530–549, 2003.

[6] H.Bunke and K.Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19:255–259, 1998.

[7] H.Busemann. The geometry of geodesics. *Academic Press*, 1955.

[8] J.B.Tenenbaum, V.D.Silva, and J.C.Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:586–591, 2000.

[9] L.G.Shapiro and R.M.Haralick. A metric for comparing relational descriptions. *IEEE PAMI*, 7:90–94, 1985.

[10] L.S.Shapiro and J.M.Brady. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, 10:283–288, 1992.

[11] B. Luo and E.R. Hancock. Structural graph matching using the em algorithm and singular value decomposition. *IEEE PAMI*, 23:1120–1136, 2001.

[12] M.Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[13] N.Linial, E.London, and Y.Rabinovich. The geometry of graphs and some its algorithmic application. *Combinatorica*, 15:215–245, 1995.

[14] S.Gold and A.Rangarajan. A graduated assignment algorithm for graph matching. *IEEE PAMI*, 18, 1996.

[15] S.Kosinov and T.Caelli. Inexact multisubgraph matching using graph eigenspace and clustering models. *9th International Workshop on Structural and Syntactic Pattern*, LNCS 2396:133–142, 2002.

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

[17] Cox T. and Cox M. Multidimensional scaling. *Chapman and Hall*, 1994.

[18] J. Ullman. An algorithm for subgraph isomorphism. *Jounal of ACM*, 23:31–42, 1976.

[19] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[20] W.Christmas, J.Kittler, and M.Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE PAMI*, 17, 1995.

[21] H. Wolkowicz, R. Saigal, and L. Vandenberghe. Handbook of semidefinite programming. *Kluwer Academic Publishers*, 2000.