Graph Matching using Adjacency Matrix Markov Chains

Antonio Robles-Kelly* Edwin R. Hancock
Department of Computer Science
The University of York, York, Y01 5DD, UK
email: arobkell,erh@cs.york.ac.uk

Abstract

This paper describes a spectral method for graph-matching. We adopt a graphical models viewpoint in which the graph adjacency matrix is taken to represent the transition probability matrix of a Markov chain. The node-order of the steady state random walk associated with this Markov chain is determined by the co-efficent order of the leading eigenvector of the adjacency matrix. We match nodes in different graphs by aligning their sequence order in the steady-state walk. The method proceeds from the nodes with the largest leading eigenvector co-efficient. We develop a brushfire search method to assign correspondences between nodes using the rank-order of the eigenvector co-efficients in first-order n eighbourhoods of the graphs. We demonstrate the utility of the new graph-matching method on both synthetic and real graphs.

1 Introduction

Recently, graph-spectral methods have provided promising results for intermediate level vision tasks including segmentation and grouping. This is a term applied to a family of techniques that aim to characterise the global structural properties of graphs using the eigenvalues and eigenvectors of the adjacency matrix [3]. The techniques furnished by spectral graph theory have provided powerful solutions to a number of problems in computer science including routing and information retrieval. In the computer vision literature there have been a number of attempts to use spectral properties for graph-matching, object recognition and image segmentation. Umeyama has an eigendecomposition method that matches graphs of the same size [14]. Borrowing ideas from structural chemistry, Scott and Longuet-Higgins were among the first to use spectral methods for correspondence analysis [10]. 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, in which point sets are matched by comparing the eigenvectors of the point proximity matrix. Here the proximity matrix is constructed by computing the Gaussian weighted distance between points. The eigen-vectors of the proximity matrices can be viewed as the basis vectors of an orthogonal transformation on the original point identities. In other words, the components of the eigenvectors represent mixing angles for the transformed

*Supported by CONACYT, under grant No. 146475/151752.

points. Matching between different point-sets is effected by comparing the pattern of eigenvectors in different images. Shapiro and Brady's method can be viewed as operating in the attribute domain rather than the structural domain. Horaud and Sossa[5] 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. In another application involving indexing into large data-bases, Sengupta and Boyer[11] have used property matrix spectra to characterise line-patterns. Various attribute representations are suggested and compared. Shokoufandeh, Dickinson and Siddiqi [1] have shown how graphs can be encoded using local topological spectra for shape recognition from large data-bases.

Although formally elegant, the main limitation of these graph-spectral 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. However, Luo and Hancock [8] have shown how some of these difficulties can be overcome by using the EM algorithm in conjunction with the singular value decomposition used by Scott and Longuet-Higgins. However, this is an iterative method which is time consuming.

In this paper we aim to investigate whether graph-spectral ideas can be used in conjunction with a simple non-iterative search heuristic. We turn to the literature on Markov chains and random walks on graphs [7, 6]. Suppose that the transition probability matrix for a Markov chain is represented as a weighted graph. The steady state random walk on the graph associated with the Markov chain is the sequence of nodes defined by the co-efficient order of the leading eigenvector of the transition probability matrix [15]. This property has been exploited widely in the literature on randomised algorithms [9]. For instance Tishby and Slonim [13] have used to for pairwise clustering. The aim in this paper is to exploit the property to develop a graph-matching algorithm. We match graphs by aligning the steady-state random walks associated associated with their adjacency matrices.

The alignment is performed usin a brushfire search procedure which uses the rank order of the coefficients of the leading eigenvector. This procedure has features in common with the method of Shapiro and Brady which finds correspondences between weighted graphs so as to minimise the Euclidean distance between the leading eigenvector [12]. This method is notoriously susceptible to differences in the sizes of the graphs being matched. Our method, on the other hand, searches for correspondences by combining the rank-order of the nodes and edge consistency constraints. As we shall demonstrate, this gives us improved robustness to size difference.

2 Eigenvectors of the Adjacency Matrix

We are concerned with the problem of matching a data graph $G_D = (V_D, E_M)$ to a model graph $G_M = (V_M, E_M)$. Here V_D and V_M are the node-sets of the two graphs, and, $E_D \subset V_D \times V_D$ and $E_M \subset V_M \times V_M$ are their edge-sets. Our aim is to find the correspondences between the nodes in the data graph V_D and their counterparts in the model-graph V_M using constraints provided by the edge-set of the model-graph. We denote the resulting set of correspondence matches by the function $f : V_D \rightarrow V_M$ from the set of data-graph nodes to the set of model-graph nodes. Here we adopt a graph-

spectral approach to the problem. That is to say, we aim to use characterise the structure of the two graphs using the eigenvalues and eigenvectors of the node adjacency matrix. The elements of this matrix are unity if a pair of nodes are connected to one-another by an edge and are zero otherwise. The diagonal entries of the matrix are set equal to unity. As a result, the adjacency matrix for the data-graph is 1 if $i = j$ or $(i, j) \in E_D$ and 0 otherwise. The adjacency matrix for the model-graph is constructed in the same way and is denoted by A_M. We are interested in the eigen-system associated with the adjacency matrix. The eigenvalues of A_D are found by solving the polynomial equation $|A_D - \lambda I| = 0$. The eigenvectors ϕ_i associated with the eigenvalue λ_i are found by solving the system of linear equations $A_D\phi_i = \lambda_i\phi_i$.

Our matching algorithm makes use of the leading eigenvectors, i.e. those associated with the largest positive eigenvalues, of the adjacency matrices A_D and A_M. The reason for this is as follows. Let us consider a a random walk on the graph $G = (V_M, E_M)$. The walk commences at the node j_1 and proceeds via the sequence of edge-connected nodes $S = j_1, j_2, j_3, \ldots$ where $(j_i, j_{i-1}) \in E$. Suppose that the transition probability associated with the move between the nodes j_l and j_m is $P_{l,m}$. If the random walk can be represented by a Markov chain, then the probability of visiting the nodes in the sequence above is $P_S = P(j_1)\prod_{i=1}^{|V_M|} P_{i+1,i}$. This Markov chain can be represented using the transition probability matrix P whose element with row l and column m is $P_{l,m}$. The leading eigenvector of the transition probability matrix determines the steady state random walk for the Markov chain [15]. Specifically, the steady state random walk is determined by the magnitude order of the co-efficients of the leading eigenvector. This condition holds provided that adjacency matrix is real, symmetric and non-negative. The resulting Markov chain is ergodic and has a single stationary state. Hence, if the adjacency matrices of the model and data graphs are taken to represent transition probability matrices, then we can attempt to find correspondences between the two graphs by aligning the random walks associated with their steady-state Markov chains.

To proceed, suppose that the leading eigenvector for the data-graph adjacency matrix is denoted by $\phi_D^* = (\phi_D^*(1), ..., \phi_D^*(|V_D|))^T$ while that for the model graph is denoted by $\phi_M^* = (\phi_M^*(1), ...\phi_M^*(|V_M|))^T$. The associated eigenvalues are λ_D^* and λ_M^*. The designation of the two graphs as "data" and "model" is a matter of convention. Here we take the data graph to be the graph which possesses the largest leading eigenvalue, i.e. $\lambda_D^* > \lambda_M^*$.

Our aim is to use the sequence of nodes defined by the rank order of the magnitudes of the components of the leading eigenvector as a means of locating correspondences. The rank order of the nodes in the data graph is given by the list of sorted node-indices $O_D = (j_1, j_2, j_3, ..., j_{|V_D|})$ where $\phi_D^*(j_1) > \phi_D^*(j_2) > \phi_D^*(j_3) > ... > \phi_D^*(j_{|V_D|})$. The subscript n of the node-index $j_n \in V_D$ is hence the rank-order of the eigenvector component $\phi_D^*(j_n)$. The rank-ordered list of model-graph nodes is $O_M = (i_1, i_2, i_3, ..., i_{|V_D|})$ where $\phi_M^*(i_1) > \phi_M^*(i_2) > \phi_M^*(i_3) > ... > \phi_M^*(i_{|V_M|})$. To develop our matching algorithm we will need to determine the rank-order of specified nodes. Accordingly, we define the operator $R(j_n, S, \phi_D^*)$ which returns the rank-order n of the node index j_n in the set $S \subseteq V_D$ using the coefficients of the eigenvector ϕ_D^*.

3 Correspondence Matching

The idea underpinning our graph-matching algorithm is to use the rank order provided by the components of the leading eigenvector to locate correspondence matches. We pose this as a brushfire search which is driven from the rank order of the nodes in the data-graph. In a nutshell, the idea is to traverse the rank-ordered list of data-graph nodes, commencing with the node of largest co-efficient and terminating with the node of smallest co-efficient. The search is initialised by placing the model-graph node of largest co-efficient in correspondence with the data-graph node of largest co-efficient. We then proceed by assigning correspondences to the first-neighbours of each data graph-node using the rank-order of model-graph nodes which satisfy edge consistency constraints.

3.1 Seeding the Algorithm

We commence by placing the first ranked node from the data-graph in correspondence with the first ranked node of the model graph, i.e. $f(j_1) = i_1$. We proceed with our brushfire search, by considering the first-neighbours of the data-graph node j_1. The candidate matches which may be assigned to these nodes and which satisfy the edge-connectivity constraints provided by the model graph are the first-neighbours of the model-graph node i_1. For the data graph, the set of first-neighbour nodes is $N^D_{j_1} = \{j|(j, j_1) \in E_D\}$ and the set of candidate correspondences from the model-graph is $N^M_{f(j_1)} = \{i|(i, f(j_1)) \in E_M\}$. We rank the nodes in the two sets according to the coefficients of the leading eigenvectors of the associated adjacency matrix. We place nodes of the same rank order in correspondence with one-another. The assignment rule is

$$\forall j \in N^D_{j_1}[f(j) = i \Leftrightarrow R(j, N^D_{j_1}, \phi^*_D) = R(i, N^M_{i_1}, \phi^*_M)] \quad (1)$$

We propagate this procedure by visiting each node in the data-graph in the order specified by the ranked-list O_D. This is an iterative process which spreads like a brush-fire from the seed node j_1.

3.2 Propagating the Brushfire Search

Suppose that we have reached the n^{th} ranked node, i.e. j_n, in the data-graph. To keep track of the nodes visited and the correspondences assigned we maintain two lists. The first of these is the set of data-graph nodes $L^D(j_n)$ to which correspondences have yet to be assigned. The second is the list of available model graph nodes $L^M(i_n)$ which have yet to be placed in correspondence with the data-graph nodes. The algorithm proceeds as follows. First, we find the set of first-neighbours of the data-graph node j_n which remain without correspondences. This set is given by $C^D_{j_n} = L^D_{j_n} \cap N^D_{j_n}$. Since we are following a chain of edge-connected nodes, the data-graph node j_n will already have been assigned a correspondence match since it is one of the first-neighbours of the node j_{n-1} which was visited in the previous iteration of the algorithm. We would like to preserve edge-connectivity constraints while assigning correspondences to the unvisited first neighbours of j_n. Hence, we find the set of nodes in the model-graph which are connected to the assigned correspondence of the node j_n. This set is given by $N^D_{f(j_n)} = \{i|(f(j_n), i) \in E_D\}$. The set of nodes which preserve the edge consistency constraints provided by the

model-graph and which are available for assignment to the nodes of the data-graph is $C^M_{f(j_n)} = L^M(j_n) \cap N^D_{f(j_n)}$.

We assign correspondences from the set of model-graph nodes $C^M_{f(j_n)}$ to the set of date-graph nodes $C^D_{j_n}$ on the basis of the rank-order of the coefficients of the leading eigenvector of the adjacency matrix. However, the two sets may be of different cardinality. If the set $C^D_{j_n}$ is of smaller size than the set $C^M_{f(j_n)}$, then the nodes with low rank leading eigenvector coefficients may be discarded. If, on the other hand, the set $C^M_{f(j_n)}$ is null (i.e. empty) or of smaller size than the set $C^M_{j_n}$, then we must find an alternative way of assigning correspondences. To do this we introduce null or dummy correspondences. We therefore pad the set $C^M_{f(j_n)}$ with $P_{j_n} = |C^D_{j_n}| - |C^M_{f(j_n)}|$ dummy nodes denoted by Φ. The ranks of these nodes are $|C^M_{f(j_n)}| + 1, |C^M_{f(j_n)}| + 2$,....., $|C^M_{f(j_n)}| + P_{j_n}$. The resulting set of padded model-graph nodes is

$$\hat{C}^M_{f(j_n)} = C^M_{f(j_n)} \cup \Phi^{|C^D_{j_n}| - |C^M_{f(j_n)}|} \tag{2}$$

The correspondences for the nodes belonging to the set $C^D_{j_n}$ are assigned as follows

$$\forall j \in C^D_{j_n} \left[f(j) = i \Leftrightarrow R(j, C^D_{j_n}, \phi^*_D) = R(i, \hat{C}^M_{f(j_n)}, \phi^*_M) \right] \tag{3}$$

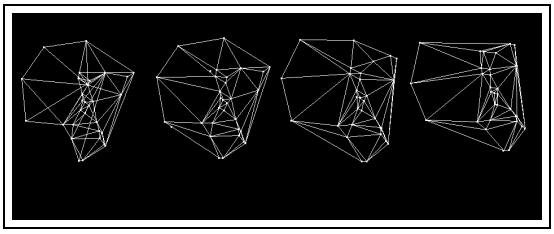
Once the correspondences have been assigned, then we update the two lists of nodes available for correspondence. The updated set of data-graph nodes which have yet to be assigned correspondences is $L^D(j_{n+1}) = L^D(j_n) - C^D_{j_n}$ while the set of nodes available for assignment is $L^M(j_{n+1}) = L^M(j_n) - C^M_{j_n}$.

This process is repeated untill all of the nodes in the data-graph have been assigned correspondences, i.e. $L^D(j_n) = \emptyset$.

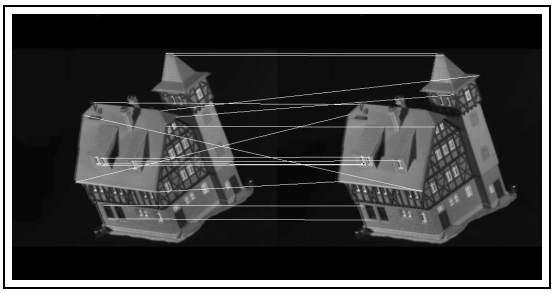
4 Experiments

We have conducted some experiments with the CMU house sequence. This sequence consists of a series of images of a model house which have been captured from different viewpoints. To construct graphs for the purposes of matching, we have first extracted corners from the images using the corner detector of Luo, Cross and Hancock [2]. The graphs used in our experiments are the Delaunay triangulations of these points. The Delaunay triangulations of the example images are shown in Figure 1a. We have matched pairs of graphs representing increasingly different views of the model house. To do this, we have matched the first image in the sequence, with each of the subsequent images. In Figure 1 b, c and d we show the sequence of correspondence matches. In each case the left-hand graph contains 34 nodes, while the right-hand graphs contain 30, 32 and 34 nodes. ¿From the Delaunay graphs it is clear that there are significant structural differences in the graphs. The numbers of correctly matched nodes in the sequence are respectively 24, 22 and 21 nodes. By comparison, the more complicated iterative EM algorithm of Luo and Hancock [8] gives 29, 23 and 11 correct correspondences. As the difference in viewing direction increases, the fraction of correct correspondences decreases from 80% for the closest pair of images to 60% for the most distant pair of images.

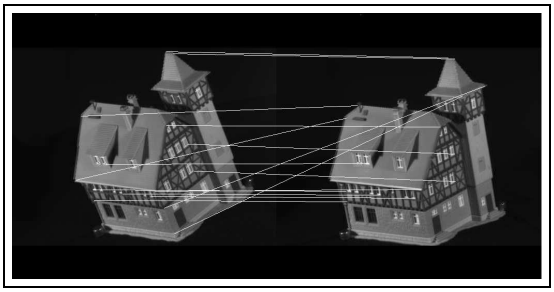
We have conducted some comparison with a number of alternative algorithms. The first of these share with our method the feature of using matrix factorisation to locate



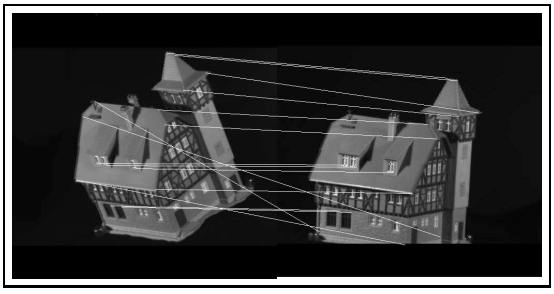
(a)



(b)



(c)



(d)

Figure 1: Delaunay triangulations and sequence of correspondences

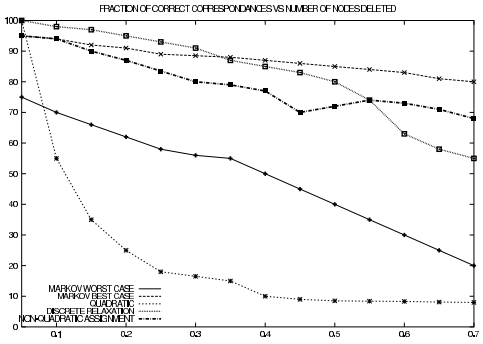


Figure 2: Sensitivity study results.

correspondences and have been reported by Umeyama [14] and Shapiro and Brady [12]. Since these two algorithms can not operate with graphs of different size, we have taken pairs of graphs with identical numbers of nodes from the CMU sequence; these are the second and fourth images which both contain 32 nodes. Here the Umeyama method and the Shapiro and Brady method both give 6 correct correspondences, while both the Luo and Hancock [8] method and our own give 22 correct correspondences.

Finally, we have conducted some experiments with synthetic data to measure the sensitivity of our matching method to structural differences in the graphs and to provide comparison with alternatives. Here we have generated random point-sets and have constructed their Delaunay graphs. We have simulated the effects of structural errors by randomly deleting nodes and re-triangulating the remaining point-set. In Figure 2 we show the fraction of correct correspondences as a function of the fraction of nodes. We plot two performance curves for our new method. The curve labeled "Markov worst case" shows the result if any of the nodes are deleted from the graph. The curve labeled "Markov best case" shows the result obtained if the seed node is protected. This latter result is much better than the former. In the case of random deletion, the fraction of correct matches falls to 80% when the fraction of deleted nodes is 10%. By contrast, the Shapiro and Brady method gives on a few percentage of correct correspondences at this level of corruption. Also shown on the plot are the performance curves for the Wilson and Hancock [16] and Gold and Rangarajan [4] methods. In the case of random node deletion, our method gives performance that is better than the Gold and Rangarajan method but worse than the Wilson and Hancock method. If the seed node is protected, then the method is comparable with the Wilson and Hancock method. The main conclusion to be drawn from this study is that our method may form the basis of a robust algorithm if a more sophisticated search procedure is used.

5 Conclusions

In this paper we have described a spectral method for correspondence matching. The method makes use of a brushfire search procedure to find correspondences using the rank-

order of the co-efficients of the leading eigenvector of the adjacency matrix. The search procedure commences from the node of largest co-efficient and proceeds via first-order neighbourhoods to assign correspondences on the basis of local rank order. The method proves to be robust to structural differences in the graphs being matching and outperforms the method of Shapiro and Brady.

There are clearly a number of ways in which the method can be improved. Currently, we use the rank-order of the eigenvector co-efficients to establish correspondences. The robustness of this procedure is clearly critically dependent on the stability of the seed node. One future possibility is to treat the missing correspondences as the states of a hidden Markov model which can be used to match the eigenvectors.

References

[1] K. Siddiqi A. Shokoufandeh, S. J. Dickinson and S. W. Zucker. Indexing using a spectral encoding of topological structure. In *Proceedings of the Computer Vision and Pattern Recognition*, 1998.

[2] Luo Bin and E. R. Hancock. Procrustes alignment with the em algorithm. In *8th International Conference on Computer Analysis of Images and Image Patterns*, pages 623–631, 1999.

[3] Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[4] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE T-PAMI*, 18(4), 1996.

[5] R. Horaud and H. Sossa. Polyhedral object recognition by indexing. *Pattern Recognition*, 1995.

[6] M. Jerum and A. Sinclair. Approximating the permament. *SIAM J. Comput.*, 18:1149–1178, 1989.

[7] L. Lovász. Random walks on graphs: A survey. *Royal Society Mathematical Studies*, 2, 1993.

[8] Bin Luo and E. R. Hancock. Structural graph matching using the EM algorithm and singular value decomposition. *To appear in IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2001.

[9] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge Univeristy Press, 1995.

[10] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two images. In *Proceedings of the Royal Society of London*, number 244 in B, 1991.

[11] K. Sengupta and K. L. Boyer. Modelbase partitioning using property matris spectra. *Computer Vision and Image Understanding*, 70(2), 1998.

[12] L. S. Shapiro and J. M. Brady. A modal approach to feature-based correspondence. In *British Machine Vision Conference*, 1991.

[13] Naftali Tishby and Noam Slonim. Data clustering by markovian relaxation and the information bottleneck method. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 640–646. MIT Press, 2001.

[14] S. Umeyama. An Eigen Decomposition Approach to Weighted Graph Matching Problems. *IEEE PAMI*, 10:695–703, 1988.

[15] R. S. Varga. *Matrix Iterative Analysis*. Springer, second edition, 2000.

[16] R.C. Wilson, A.N. Evans, and E.R. Hancock. Relational matching by discrete relaxation. *IVC*, 13(5):411–421, June 1995.