Graph Spectral Approach for Learning View Structure

Bin Luo\textsuperscript{1,2} Richard C. Wilson\textsuperscript{1} Edwin R. Hancock \textsuperscript{1}
\textsuperscript{1}Department of Computer Science, University of York, York YO1 5DD, UK.
\textsuperscript{2}Anhui University, P.R. China
Tel. +44-1904-433374 E-mail: erh@cs.york.ac.uk

Abstract

In this paper we explore how to represent object view-structure by embedding the neighbourhood graphs of feature points in a pattern-space. We adopt a graph-spectral approach. We use the leading eigenvectors of the graph adjacency matrix to define clusters of nodes. For each cluster, we compute vectors of cluster properties. We embed these vectors in a pattern-space using two contrasting approaches. The first of these involves performing principal components analysis on the covariance matrix for the spectral pattern vectors. The second approach involves performing multidimensional scaling on the L2 norm for pairs of pattern vectors. We demonstrate the both methods result in well-structured view spaces for graph-data extracted from 2D views of 3D objects.

1 Introduction

View-based object recognition has been studied in the computer vision literature for over three decades \cite{10, 9, 2}. Stated simply, the idea is to compile a series of images of an object as the set of possible viewing directions is spanned. The images are then subjected to some form of dimensionality reduction \cite{7} or information abstraction \cite{2}. This is a process of learning \cite{8, 5}. Broadly speaking there are two different approaches to this problem. The first of these is to construct an eigenspace from appearance attributes \cite{7}. The second approach to the problem is older and involves constructing a relational abstraction of the features present in the raw images \cite{9, 10, 6}. Generally, speaking while the former approach has met with more success, the latter one, while conceptually alluring has proved elusive. The reason for this is that it while it is a straightforward task to map pixel-based attributes to a pattern-vector, in the case of relational graphs this is not the case. The reasons for this are two-fold. First, correspondences are required so that nodes and edges can be mapped to the relevant component of the pattern-vector. Second, there needs to be a means of accommodating graphs which contain different numbers of nodes and edges.

In this paper, to overcome these two problems, we provide a graph-spectral approach to the embedding problem. Spectral graph theory is a branch is mathematics which aims to characterise the properties of unweighted graphs using the eigenvalues and eigenvectors of the adjacency matrix or the closely related Laplacian matrix. Our aim is to use the pairwise clustering property of the eigenmodes of the adjacency matrix for the purposes of constructing pattern-vectors for graphs. The idea is as follows. Each of the leading eigenmodes of the adjacency matrix is taken to represent a pairwise cluster of nodes. The significance of the cluster is determined by the magnitude of the eigenvector. The degree of cluster membership of graph-nodes to clusters is gauged by the co-efficients of the associated eigenmodes. We perform our embedding of the graphs using vectors of graph-theoretic attributes for the clusters. We solve the correspondence problem by ordering the clusters using the magnitude order of the eigenvalues. For each cluster we calculate weighted cluster-attributes using the eigenvectors to weight contributions from the different nodes in the original graph. Once the cluster feature-vectors are to hand, then we investigate two alternative routes to embedding them in a pattern-space. The first of these involves principal components analysis. Here we construct the covariance matrix for the spectral pattern vectors of the graphs. We project the pattern-vectors onto the leading eigenvectors of the covariance matrix to give a graph pattern-space. The second approach is based on multidimensional scaling. Here we compute a matrix of pairwise similarities between pairs of graphs using the L2 distance norm.

2 Spectral Pattern Vectors

In this paper we are concerned with the set of graphs $G_1, G_2, \ldots, G_k, \ldots, G_N$. The $k$th graph is denoted by $G_k = (V_k, E_k)$, where $V_k$ is the set of nodes and $E_k \subseteq V_k \times V_k$ is the edge-set. Our approach in this paper is a graph-spectral one. For each graph $G_k$ we compute the adjacency matrix $A_k$. This is a $|V_k| \times |V_k|$ matrix whose element with row
vectors are stacked in order to construct the modal matrix $A_k$. To overcome the correspondence problem, we use the order of the graphs under study to construct feature-vectors. We study a number of features suggested by spectral graph theory. These are listed below:

**Leading eigenvalues:** Our first vector of spectral features is constructed from the ordered eigenvalues of the adjacency matrix. For the graph indexed $k$, the vector is

$$B_k = (\lambda_1^k, \lambda_2^k, ..., \lambda_n^k)^T.$$  

**Cluster volumes:** The volume $Vol(S)$ of a subgraph $S$ of a graph $G$ is defined to be the sum of the degrees of the nodes belonging to the subgraph, i.e.

$$Vol(S) = \sum_{i \in S} \deg(i).$$  

where $\deg(i)$ is the degree of node $i$. By analogy, for the modal clusters, we define the normalised volume of the cluster indexed $\omega$ in the graph-indexed $k$ to be

$$Vol^\omega_k = \frac{\sum_{i \in V_k} s^k_{i,\omega} \deg(i)}{\sum_{\omega=1}^m \sum_{i \in V_k} s^k_{i,\omega} \deg(i)}.$$  

The feature-vector for the graph-indexed $k$ is $B_k = (Vol^1_k, Vol^2_k, ..., Vol^n_k)^T$.

**Cluster perimeters:** For a subgraph $S$ the set of perimeter nodes is $\Delta(S) = \{(u,v)| (u,v) \in E \land u \in S \land v \not\in S\}$. The perimeter length of the subgraph is defined to be the number of edges in the perimeter set, i.e. $\Gamma(S) = |\Delta(S)|$. Again, by analogy, the normalised perimeter length of the modal cluster indexed $\omega$ is

$$\Gamma^\omega_k = \frac{\sum_{i \in V_k} s^\omega_{i,\omega} (1 - s^\omega_{i,\omega}) A_k(i,j)}{\sum_{\omega=1}^m \sum_{i \in V_k} s^\omega_{i,\omega} (1 - s^\omega_{i,\omega}) A_k(i,j)}.$$  

The perimeter values are ordered according to the modal index of the relevant cluster to form the graph feature vector $B_k = (\Gamma^1_k, \Gamma^2_k, ..., \Gamma^n_k)^T$.

**Shared perimeters:** The first pairwise cluster attribute studied is the shared perimeter of each pair of clusters. For the pair subgraphs $S$ and $T$ the perimeter is the set of nodes belong to the set $P(S,T) = \{ (u,v) | u \in S \land v \in T \}$. Hence, our cluster-based measure of shared perimeter for the clusters is

$$U_k(u,v) = \frac{\sum_{(i,j) \in E_k} s^k_{i,u} s^k_{j,v} A_k(i,j)}{\sum_{(i,j) \in E_k} s^k_{i,u} s^k_{j,v}}.$$  

Each graph is represented by a shared perimeter matrix $U_k$. We convert these matrices into long vectors. This is obtained by stacking the columns of the matrix $U_k$ in eigenvalue order. The resulting vector is $B_k = (U_k(1,1), U_k(1,2), ..., U_k(1,n), U_k(2,1), ..., U_k(q, n), ..., U_k(n,n))^T$. Each entry in the long-vector corresponds to a different pair of spectral clusters.

**Cluster distances:** The second pairwise attribute is the between cluster distance. This is defined as the path length, i.e. the minimum number of edges, between the most significant nodes in a pair of clusters. The most significant node in a cluster is the one having the largest co-efficient in the eigenvector associated with the cluster. For the cluster indexed $u$ in the graph indexed $k$, the most significant node is

$$i^k_u = \arg \max_i s^k_{i,u}.$$  

To compute the distance, we note that if we multiply the adjacency matrix $A_k$ by itself $l$ times, then the matrix $(A_k)^l$ represents the distribution of paths of length $l$ in the graph $G_k$. In particular, the element $(A_k)^l(i,j)$ is the number of paths of length $l$ edges between the nodes $i$ and $j$. Hence the minimum distance between the most significant nodes of the clusters $u$ and $v$ is

$$d_{u,v} = \arg \min_l (A_k)^l(i^k_u, i^k_v).$$  

If we only use the first $n$ leading eigenvectors to describe the graphs, the between cluster distances for each graph can be written as a $n$ by $n$ matrix which can be converted to a $n \times n$ long-vector $B_k = (d_{1,1}, d_{1,2}, ..., d_{1,n}, d_{2,1}, ..., d_{n,n})^T$.

### 2.1 Eigendecomposition of the image representation matrices

Our first method makes use principal components analysis and follows the parametric eigenspace idea of Murase.
and Nayar [7] the relational data for each image is vectorised in the way outlined in Section 3. The \( N \) different image vectors are arranged in view order as the columns of the matrix \( S = [B_1 | B_2 | \cdots | B_i | \cdots | B_N] \).

Next, we compute the covariance matrix for the elements in the different rows of the matrix \( S \). This is found by taking the matrix product \( C = SS^T \). We extract the principal components directions for the relational data by performing an eigendecomposition on the covariance matrix \( C \). The eigenvalues \( \lambda_i \) are found by solving the eigenvalue equation \( [C - \lambda I]v = 0 \) and the corresponding eigenvectors \( e_i \) are found by solving the eigenvector equation \( Ce_i = \lambda_i e_i \).

We use the first \( 3 \) leading eigenvectors to represent the graphs extracted from the images. The co-ordinate system of the eigenspace is spanned by the three orthogonal vectors by \( E = (e_1, e_2, e_3) \). The individual graphs represented by the long vectors \( B_i, i = 1, 2, \cdots, N \) can be projected onto this eigenspace using the formula \( \vec{x}_i = e^T B_i \). Hence each graph \( G_i \) is represented by a 3-component vector \( \vec{x}_i \) in the eigenspace.

### 2.2 Multidimensional Scaling

Multidimensional scaling (MDS) [1] is a procedure which allows data specified in terms of a matrix of pairwise distances to be embedded in a Euclidean space. The classical multidimensional scaling method was proposed by Torgerson [11] and Gower [4]. Shepard and Kruskal developed a different scaling technique called ordinal scaling [3]. Here we intend to use the method to embed the graphs extracted from different viewpoints in a low-dimensional space.

To commence we require pairwise distances between graphs. We do this by computing the L2 norms between the spectral pattern vectors for the graphs. For the graphs indexed \( i_1 \) and \( i_2 \), the distance is

\[
D_{i_1, i_2} = \frac{1}{N^2} \sum_{j=1}^{N} \left| B_{i_1}(j) - B_{i_2}(j) \right|^2.
\]

The pairwise similarities \( d_{i_1, i_2} \) are used as the elements of an \( N \times N \) dissimilarity matrix \( D \), whose elements are defined as follows

\[
D_{i_1, i_2} = \begin{cases} 
\frac{1}{N^2} \sum_{j=1}^{N} d_{r,c} & \text{if } i_1 \neq i_2 \\
0 & \text{if } i_1 = i_2 
\end{cases}
\]

In this paper, we use the classical multidimensional scaling method to embed our the view-graphs in a Euclidean space using the matrix of pairwise dissimilarities \( D \). The first step of MDS is to calculate a matrix \( T \) whose element with row \( r \) and column \( c \) is given by \( T_{rc} = -\frac{1}{2}(d_{r,c}^2 - d_r^2 - d_c^2 + \bar{d}_c^2) \), where \( \bar{d}_c = \frac{1}{N} \sum_{c=1}^{N} d_{r,c} \) is the average dissimilarity value over the \( r \)th row, \( d_r^2 \) is the similarly defined average value over the \( r \)th column and

\[
\bar{d}_c = \frac{1}{N^2} \sum_{r=1}^{N} \sum_{c=1}^{N} d_{r,c}
\]

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. \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0 \). The corresponding ordered eigenvectors are denoted by \( e_i \) where \( \lambda_i \) is the \( i \)th eigenvalue. The embedding co-ordinate system for the graphs obtained from different views is \( X = [f_1, f_2, \cdots, f_k] \), where \( f_j = \sqrt{\lambda_j} e_j \) are the scaled eigenvectors. For the graph indexed \( i \), the embedded vector of co-ordinates is \( \vec{x}_i = (X_{i,1}, X_{i,2}, X_{i,3})^T \).

### 3 Experiments

To provide an experimental vehicle for our new eigenspace representation of graphs, we focus on the problem of view based object recognition. In this section we report experiments on 2D image sequences of 3D objects under slowly varying changes in viewer angle. From each object in the view sequence, we extract corner features. From the extracted corner points we construct Delaunay graphs. We experiment with a Swiss chalet sequence. Figure 1 shows the raw images and there corresponding Delaunay triangulations.

![Figure 1. Raw images and graphs of the chalet sequence](image-url)

In Figure 2 we show the results obtained with vectors of different spectral attributes. In the left-hand column of
the figure, we show the eigenspace extracted by applying PCA to the covariance matrix for the spectral feature vectors. The middle column shows the matrix of pairwise vector distances used as input to MDS. Finally, the right-hand column shows the result of applying MDS to the matrix of distances. From the top row to the bottom row, we show the results obtained with vectors of ordered eigenvalues, cluster volumes, cluster perimeters, shared perimeters and the cluster distances. First, we compare the structure of the view-spaces obtained using PCA and MDS. Generally speaking, they are rather different. In the case of PCA, a cluster structure emerges. By contrast, in MDS the different views execute smooth trajectories. Hence, the output of PCA would appear to be best for locating clusters of similar views, while MDS provides information which might be more useful in constructing parametric eigenspaces.

We now turn to the different spectral attributes in more detail. The vector of leading eigenvalues of the adjacency matrix give the smoothest and most uniformly distributed trajectories when used in conjunction with MDS. They also result in the least clustered distribution when PCA is applied.

We now turn our attention to the shared perimeter attribute. This is a relational quantity and reflects the arrangement of modal clusters. From the row 3 of Figure 2 we can see that this gives good clusters in the PCA plots and good trajectories in the MDS plots. Finally, we consider the inter-cluster distance shown in the final row of Figure 2. Here there is the most obvious block-structure in the centre-panel, and this reflects the view structure of the object mentioned above.

4 Conclusions

In this paper we have investigated how vectors of graph-spectral attributes can be used for the purposes of embedding graphs in eigenspaces. The attributes studied are the leading eigenvalues, the volumes, perimeters, shared perimeters and cluster distances for modal clusters. The best view trajectories result when we apply MDS to the vectors of leading eigenvalues. The best clusters result when we use cluster volume or shared perimeter.

Hence, we have shown how to cluster purely symbolic graphs using simple spectral attributes. The graphs studied in our analysis are of different size, and we do not need to locate correspondences. Our future plans involve investigating whether the spectral attributes studied here can be used for the purposes of organising large image data-bases.

References


