PERGAMON

Pattern Recognition 36 (2003) 2213–2230

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Spectral embedding of graphs

Bin Luo[a,b], Richard C. Wilson[a], Edwin R. Hancock[a,*]

[a] Department of Computer Science, University of York, York YO10 5DD, UK
[b] Anhui University, People's Republic of China

Received 10 June 2002; accepted 3 January 2003

Abstract

In this paper we explore how to embed symbolic relational graphs with unweighted edges in a pattern-space. We adopt a graph-spectral approach. We use the leading eigenvectors of the graph adjacency matrix to define eigenmodes of the adjacency matrix. For each eigenmode, we compute vectors of spectral properties. These include the eigenmode perimeter, eigenmode volume, Cheeger number, inter-mode adjacency matrices and intermode edge-distance. We embed these vectors in a pattern-space using two contrasting approaches. The first of these involves performing principal or independent 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 illustrate the utility of the embedding methods on neighbourhood graphs representing the arrangement of corner features in 2D images of 3D polyhedral objects. Two problems are investigated. The first of these is the clustering of graphs representing distinct objects viewed from different directions. The second is the identification of characteristic views of single objects. These two studies reveal that both embedding methods result in well-structured view spaces for graph-data extracted from 2D views of 3D objects.

Keywords: Graph spectra; Graph clustering; View based recognition; Principal component analysis; Independent component analysis; Multidimensional scaling

1. Introduction

Relational graphs have proved alluring as structural representations for both 2D and 3D shape in computational vision. Barrow and Burstall [1] and Fischler and Enschlager [2] were among the first to demonstrate the potential of relational graphs as abstractions for pictorial information. Since then graph-based representations have been exploited widely for the purposes of shape representation, segmentation, matching and recognition. However, one of the problems that hinders the manipulation of large sets of graphs is that of measuring their similarity. This problem arises in a number of situations where graphs must be matched or clustered together. The large-scale matching problem arises in tasks involving recognition from image databases [3]. The graph-clustering task arises when the unsupervised learning of the class-structure of sets of graphs is attempted. Concrete examples here include the organisation of large structural data-bases [4] or the discovery of the view-structure of objects [5].

One of the problems that hinders this endeavour is that graphs are neither vectorial in nature nor easily transformed into vectors. The reasons for this are twofold. First, there is no canonical ordering of the nodes or edges of a graph. Hence, there is no natural way to map the nodes or edges to the components of a vector. Second, most graph-matching or graph manipulation problems are inexact in nature. That is to say that the graphs are noisy in nature and hence contain different numbers of nodes or edges. Hence, even if an ordering can be established then there needs to be a means of dealing with pattern-vectors of different length. Since they are not easily vectorised, it is not straightforward to characterise the mean and variance of a set of graphs. Hence, standard pattern recognition methods cannot be used to analyse

[*] Corresponding author. Tel.: +44-1904-433374; fax: +44-1904-432767.
E-mail address: erh@cs.york.ac.uk (E.R. Hancock).

0031-3203/03/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/S0031-3203(03)00084-0

or cluster sets of graphs. One way around this problem is to adopt a pairwise clustering approach [6]. This involves measuring the pairwise similarity of the graphs and clustering them by searching for sets of graphs which exhibit a strong mutual affinity to one-another.

There are a number of ways in which the similarity of graphs may be measured. One of the classical methods is to use the concept of graph edit distance. This is an extension of the classical string edit distance, or Levenshtein distance [7], to graphs. The idea of using graph edit distance was first explored by Fu and his co-workers [8,9]. Here edit distances are computed using separate costs for the relabeling, the insertion and the removal of both nodes and edges. Recently, Bunke [10,11] has shown that the graph edit distance and the size of the maximum common subgraph are related under certain restrictions on the edge and node edit costs. Torsello and Hancock [12] have exploited this observation to efficiently compute tree-edit distance. By using the Motzkin-Strauss theorem they show how to compute an approximation to the edit distance using relaxation labelling. Another approach to computing graph similarity is to adopt a probabilistic framework. Here there are two contributions worth mentioning. First, Christmas et al. [13] have developed an evidence combining framework for graph-matching which uses probability distribution functions to model the pairwise attribute relations defined on graph-edges. Second, Wilson and Hancock [14] show how to measure graph-similarity using a probability distribution which models the number of relabeling and graph-edit operations when structural errors are present [14,15].

Although graph-matching allows the pairwise comparison of shock-graphs, it does not allow space of structural variations to be explored in detail. Graph-matching may provide a fine measure of distance between structures, and this in turn may be used to cluster similar graphs. However, it does not result in an ordering of the graphs that has metrical significance under structural variations due to graded shape-changes. Hence, we aim to address the problem of how to organise graphs into a pattern-space in which similar structures are close to one-another, and dissimilar structures are far apart. In particular, we aim to embed graphs in a vector-space where the dimensions correspond to principal modes in structural variation. There are a number of ways in which this can be achieved. The first is to compute the distance between graphs and to use multidimensional scaling (MDS) to embed the individual graphs in a low-dimensional space [12]. However, this approach does not necessarily result in a pattern-space where the dimensions reflect the modes of structural variation of the different graphs under study. The second approach is to extract feature vectors from the graphs. A pattern-space can be constructed from such vectors by performing modal analysis on their covariance matrix. Unfortunately, and for the reasons noted above, the process of embedding graphs in a vector-space is not a straightforward one.

To overcome the problem of how to map the structure of a graph onto a vector of fixed length, we turn to graph-spectral decomposition methods. Spectral graph theory is a branch in 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 [16]. There are a number of well-known results. For instance, the degree of bijectivity of a graph is measured by the eigenvalue gap, the distribution of cycle length can be computed using a moments expansion of the eigenvalues, and the steady-state random walk on a graph is given by the leading eigenvector of the adjacency matrix. Although conceptually alluring, the main problem with spectral properties is that they are notoriously sensitive to small changes in the structure of the adjacency matrix.

In this paper we work with the spectral decomposition (or eigendecomposition) of the adjacency matrix. Associated with each eigenmode is an adjacency matrix. We aim to investigate whether graphs can be represented in a stable way using vectors of spectral attributes for the leading eigenmodes of the adjacency matrix. The attributes studied include the perimeter length, the volume and the Cheeger constants for the mode adjacency matrices. We adopt the following procedure for vectorising the graphs. Each component of the vector is taken to represent a different spectral mode of the original graph adjacency matrix. The order of the components of the vector is the magnitude order of the eigenvalues of the adjacency matrix. For each spectral mode, we use the components of the associated eigenvectors to compute spectral attributes. In this way we solve the problem of finding correspondences between nodes and vector-components.

Once the feature-vectors for the eigenmodes of the adjacency matrices 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 MDS. Here we compute a matrix of pairwise similarities between pairs of graphs using the L2 distance norm.

2. Graph spectra

In this paper we are concerned with the set of graphs $G_1, G_2, \ldots, G_k, \ldots, G_N$. The kth 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 index i and column index j is

$$A_k(i,j) = \begin{cases} 1 & \text{if } (i,j) \in E_k, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

From the adjacency matrices $A_k, k = 1,\dots,N$ at hand, we can calculate the eigenvalues λ_k by solving the equation $|A_k - \lambda_k I| = 0$ and the associated eigenvectors ϕ_k^ω by solving the system of equations $A_k\phi_k^\omega = \lambda_k^\omega \phi_k^\omega$, where ω is the eigenmode index. We order the eigenvectors according to the decreasing magnitude of the eigenvalues, i.e. $|\lambda_k^1| > |\lambda_k^2| > \cdots > |\lambda_k^{|V_k|}|$. The eigenvectors are stacked in order to construct the modal matrix $\Phi_k = (\phi_k^1|\phi_k^2|\dots|\phi_k^{|V_k|})$.

With the eigenvalues and eigenvectors of the adjacency matrix to hand, the spectral decomposition for the adjacency matrix of the graph indexed k is

$$A_k = \sum_{\omega=1}^{|V_k|} \lambda_k^\omega \phi_k^\omega (\phi_k^\omega)^{\mathrm{T}}. \tag{2}$$

If $\Lambda_k = diag(\lambda_k^1, \dots, \lambda_k^{|V_k|})$ is the diagonal matrix with the eigenvalues of A_k as diagonal elements, then the spectral decomposition of the adjacency matrix can be written as

$$A_k = \Phi_k \Lambda_k \Phi_k^{\mathrm{T}}. \tag{3}$$

Associated with the eigenmode with index ω is the mode adjacency matrix:

$$S_k^\omega = \phi_k^\omega (\phi_k^\omega)^{\mathrm{T}}. \tag{4}$$

The aim in this paper is to explore whether the properties of these matrices can be used to construct feature vectors for the graphs under study. We explore two different approaches. The first of these involves computing features for individual mode adjacency matrices. The second involves the use of relational features which describe the arrangement of mode adjacency matrices.

For each graph, we use only the first n eigenmodes of the adjacency matrix. The truncated modal matrix is

$$\Phi_k = (\phi_k^1|\phi_k^2|\dots|\phi_k^n). \tag{5}$$

3. Spectral features

Our aim is to use spectral features computed from the eigenmodes of the adjacency matrices for graphs under study to construct feature-vectors. To overcome the correspondence problem, we use the order of the eigenvalues to establish the order of the components of the feature-vectors. We study a number of features suggested by spectral graph theory.

3.1. Unary features

We commence by considering unary features for the eigenmodes of the adjacency matrix. The features studied are listed below.

3.1.1. 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

$$B_k = (\lambda_k^1, \lambda_k^2, \dots, \lambda_k^n)^{\mathrm{T}}. \tag{6}$$

This vector represents the spectrum of the graph G_k.

3.1.2. Eigenmode volume

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

$$Vol_k(S) = \sum_{i \in S} D_k(i), \tag{7}$$

where

$$D_k(i) = \sum_{j \in V_k} E_{i,j}$$

is the degree of node i in the graph G_k. If D_k is the degree vector for the graph G_k, then the vector of volumes for the eigenmodes is found using the projection

$$Vol_k = \Phi_k^{\mathrm{T}} D_k \tag{8}$$

In other words, the volume associated with the eigenmode indexed ω in the graph-indexed k is

$$Vol_k(\omega) = \sum_{i \in V_k} \Phi_k(i,\omega) D_k(i). \tag{9}$$

The eigenmodes volume feature-vector for the graph-indexed k is $B_k = (Vol_k(1), Vol_k(2)), \dots, Vol_k(n))^{\mathrm{T}}$.

3.1.3. Eigenmode perimeter

For a subgraph S the set of perimeter nodes is $\Delta(S) = \{(u,v)\,|\,(u,v) \in E \wedge u \in S \wedge v \notin S\}$. 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 perimeter length of the adjacency matrix for the eigenmode indexed ω is

$$\Gamma_k(\omega) = \sum_{v \neq \omega} \sum_{i \in V_k} \sum_{j \in V_k} \Phi_k(i,\omega)\Phi_k(j,v)A_k(i,j). \tag{10}$$

The perimeter values are ordered according to the modal index to form the graph feature vector $B_k = (\Gamma_k^1, \Gamma_k^2, \dots, \Gamma_k^n)^{\mathrm{T}}$.

3.1.4. Cheeger constant

The Cheeger constant for the subgraph S is defined as

$$H(S) = \frac{|\Delta(S)|}{\min[Vol(S), Vol(\hat{S})]}. \tag{11}$$

The analogue of the Cheeger constants for the eigenmodes of the adjacency matrix is

$$H_k(\omega) = \frac{\Gamma_k(\omega)}{\min[Vol_k(\omega), Vol_k(\hat{\omega})]}, \tag{12}$$

where

$$Vol_k(\hat{\omega}) = \sum_{\omega=1}^{n} \sum_{i \in V_k} \Phi_k(i,\omega) D_k(i) - Vol_k(\omega) \tag{13}$$

is the volume of the complement of the eigenmode indexed ω. Again, the eigenmode Cheeger numbers are ordered to form a spectral feature-vector $B_k = (H_k(1), H_k(2), \ldots, H_k(n))^{\mathrm{T}}$.

3.2. Binary features

In addition to the unary features, we have studied pairwise attributes for the eigenmodes.

3.2.1. Inter-mode adjacency matrix

Our first pairwise representation is found by projecting the adjacency matrix onto the basis spanned by the eigenvectors. The projection or inter-mode adjacency matrix is given by

$$U_k = \Phi_k^{\mathrm{T}} A_k \Phi_k. \tag{14}$$

The element of the matrix with row index u and column index v is

$$U_k(u,v) = \sum_{i\in V_k}\sum_{j\in V_k} \Phi_k(i,u)\Phi_k(j,v)A_k(i,j). \tag{15}$$

These matrices are converted into long vectors. This is done by stacking the columns of the matrix U_k in eigenvalue order. The resulting vector is $B_k = (U_k(1,1), U_k(1,2), \ldots, U_k(1,n), U_k(2,1) \ldots, U_k(2,n,), \ldots, U_k(n,n))^{\mathrm{T}}$. Each entry in the long-vector corresponds to a different pair of spectral eigenmodes.

3.2.2. Inter-mode distances

The between mode distance is defined as the path length, i.e. the minimum number of edges, between the most significant nodes associated with each eigenmode of the adjacency matrix. The most significant node associated a particular eigenmode of the adjacency matrix is the one having the largest co-efficient in the associated eigenvector. For the eigenmode indexed u in the graph indexed k, the most significant node is

$$i_u^k = \arg\max_i \Phi_k(i,u). \tag{16}$$

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 eigenmode indexed u and v is

$$d_{u,v} = \arg\min_l (A_k)^l(i_u^k, i_v^k). \tag{17}$$

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

4. Embedding the spectral vectors in a pattern space

In this section we describe three methods for embedding graphs in eigenspaces. The first of these involves performing principal components analysis on the covariance matrices for the spectral pattern-vectors. The second involves independent component analysis (ICA). The third method involves performing MDS on a set of pairwise distance between vectors.

4.1. Pattern space embedding by PCA

Our first method makes use principal components analysis and follows the parametric eigenspace idea of Murase and Nayar [17]. The graphs extracted from each image are vectorised in the way outlined in Section 3. The N different image vectors are arranged in view order as the columns of the matrix:

$$S = [B_1|B_2|\ldots|B_k|\ldots|B_N]. \tag{18}$$

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:

$$C = SS^{\mathrm{T}}. \tag{19}$$

We extract the principal components directions for the relational data by performing an eigendecomposition on the covariance matrix C. The eigenvalues λ_i are found by solving the eigenvalue equation

$$|C - \lambda I| = 0 \tag{20}$$

and the corresponding eigenvalues $\vec{e}_i$ are found by solving the eigenvector equation

$$C\vec{e}_i = \lambda_i \vec{e}_i. \tag{21}$$

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 $\vec{e} = (\vec{e}_1, \vec{e}_2, \vec{e}_3)$. The individual graphs represented by the long vectors B_k, $k = 1, 2, \ldots, N$ can be projected onto this eigenspace using the formula

$$\vec{x}_k = \vec{e}^{\mathrm{T}} B_k. \tag{22}$$

Hence each graph G_k is represented by a three-component vector $\vec{x}_k$ in the eigenspace.

4.2. Pattern space embedding by ICA

Our second approach uses ICA to embed the graphs in a pattern space. We explore how to decompose a set of graphs into significantly different independent components. These can then be used for graph clustering by projecting the original graphs into the pattern space spanned by the independent components.

The ICA algorithm used in this paper is Cardoso and Soulourniac's JADE algorithm [3]. JADE is a statistically based algorithm. The main features of the algorithm are as follows. As with other ICA algorithms, the first step is data whitening or sphering. The aim is to eliminate correlations from the data. This can be achieved by removing the mean of the data and using PCA on the data covariance matrix. As a result, the whitened vector set is $Z = WB$, where W is the estimated whitening matrix. The second step of JADE is estimate the 4th-order cumulants Q_z. In the noiseless case, Q_z can be calculated as follows:

$$Q_Z(I_n) = E\{|Z|^2 ZZ^{\mathrm{T}}\} - (n+1)I_n, \tag{23}$$

where I_n is the n-order identity matrix and $E(\cdot)$ is the expectation operator. Next, a joint diagonalisation is performed to find a matrix $\hat{V}$ to minimise the non-diagonal entries of the cumulants matrices,

$$\hat{V} = \arg\min \Sigma_i Off(V^{\mathrm{T}} Q_Z V). \tag{24}$$

Again we use the first 3 most significant independent components to represent the graphs extracted from the images. The co-ordinate system of the pattern-space is spanned by the three independent components by $\hat{e} = (\hat{V}_l, \hat{V}_2, \hat{V}_3)$. The individual graphs represented by the long vectors Z_k, $k = 1, 2, \ldots, N$ can be projected onto this pattern space using the formula $\vec{x}_k = \vec{e}^{\mathrm{T}} Z_k$. Hence each graph G_k is represented by a 3-component vector $\vec{x}_k$ in the pattern space.

4.3. Multidimensional scaling

MDS [18] is a procedure which allows data specified in terms of a matrix of pairwise distances to be embedded in a Euclidean space. The classical MDS method was proposed by Torgenson [19] and Gower [20]. Shepard and Kruskal developed a different scaling technique called ordinal scaling [21]. 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 $i1$ and $i2$, the distance is

$$d_{i1,i2} = \sum_{\alpha=1}^{K} [B_{i1}(\alpha) - B_{i2}(\alpha)]^2. \tag{25}$$

The pairwise similarities $d_{i1,i2}$ are used as the elements of an $N \times N$ dissimilarity matrix D, whose elements are defined as follows:

$$D_{i1,i2} = \begin{cases} d_{i1,i2} & \text{if } i_1 \neq i2, \\ 0 & \text{if } i1 = i2. \end{cases} \tag{26}$$

In this paper, we use the classical MDS to embed 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

$$T_{rc} = -\frac{1}{2}[d_{rc}^2 - \hat{d}_{r.}^2 - \hat{d}_{.c}^2 + \hat{d}_{..}^2], \tag{27}$$

where

$$\hat{d}_{r.} = \frac{1}{N}\sum_{c=1}^{N} d_{rc} \tag{28}$$

is the average dissimilarity value over the rth row, $\hat{d}_{.c}$ is the similarly defined average value over the cth column and

$$\hat{d}_{..} = \frac{1}{N^2}\sum_{r=1}^{N}\sum_{c=1}^{N} d_{r,c} \tag{29}$$

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 $k, k \leqslant N$, then we will have k non-zero eigenvalues. We arrange these k non-zero eigenvalues in descending order, i.e. $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_k > 0$. The corresponding ordered eigenvectors are denoted by $\vec{e}_i$ where λ_i is the ith eigenvalue. The embedding co-ordinate system for the graphs obtained from different views is

$$X = [\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_k], \tag{30}$$

where $\vec{f}_i = \sqrt{\lambda_i}\vec{e}_i$ are the scaled eigenvectors. For the graph indexed i, the embedded vector of co-ordinates is

$$\vec{x}_i = (X_{i,1}, X_{i,2}, X_{i,3})^{\mathrm{T}}. \tag{31}$$

5. View-based object recognition

To provide an experimental vehicle for our eigen-space representation of graphs, we focus on the problem of view based object recognition. This topic has been studied in the computer vision literature for over three decades [22–24]. 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 [17] or information abstraction [24]. This is a process of learning [25,26] that may involve either feature extraction, principal components analysis or the abstraction of the main structures using a relational description [27]. Once a condensed image representation is to hand, then the aim is to embed the different images in a low-dimensional representation which can be traversed with viewing direction. Recognition and pose recovery may be effected by finding the closest representative view. In other words, the aim is to embed high-dimensional view based image data in a low dimensional structure which is suitable for view indexing.

Broadly speaking there are two different approaches to this problem. The first of these is to construct an eigenspace [17]. This approach was first introduced by Murase and Nayar [17], and has since been refined in a number of different ways [28,29]. The idea is to perform principal components analysis on the images collected as the viewing direction and illumination direction [30] are varied. This is achieved by first storing each image as a long-vector. Next the covariance matrix for the long-vectors is found. The eigenvectors of the covariance matrix define the directions of principal components in the space spanned by the long-vectors. Dimensionality reduction is achieved by projecting the original images onto the principal component directions and selecting the components corresponding to the leading eigenvectors. The method has mainly been applied to pixel based image representations.

The second approach to the problem is older and involves constructing a relational abstraction of the features present in the raw images [23,22,31]. The aim here is to extract surfaces or boundary groupings from 2.5D range data or 2D image data. From this data the view occurrence of the different image structures is noted. Hence a group of images which all yield the same feature configuration are deemed to belong to a common view [32]. View indexing can be achieved by matching a relational arrangement of image structures to the set of corresponding representative view graphs. This approach to the problem has its origins in the work of Freeman on characteristic views. It has also stimulated the study of aspect graphs [22–24,33–37]. The topic draws heavily on work from psychology [38,39] and differential topology [40,31].

6. Experiments

Our experimental vehicle is provided by 2D views of 3D objects. We have collected sequences of views for a number of objects. For the different objects the image sequences are obtained under slowly varying changes in viewer angle. From each image in each view sequence, we extract corner features. We use the extracted corner points to construct Delaunay graphs. In our experiments we use three different sequences. Each sequence contains images with equally spaced viewing directions. For each sequence we show a sample of 10 images. In Fig. 1 we show the CMU/VASC model house image sequence and the associated graphs. In Fig. 2 we show the first ten samples of the raw images and extracted graphs for the INRIA MOVI toy house sequence. Finally, Fig. 3 shows the first ten samples of a third house sequence for a model of Swiss chalet. In Table 1, we list the number of feature points for each view for the three different image sequences. There are a number of "events" in the sequences. For instance in the MOVI sequence, the right-hand gable wall disappears after the 12th frame, and the left-hand gable wall appears after the 17th frame. Several of the background objects also disappear and reappear. In the Swiss chalet sequence, the front face of the house disappears after the 15th frame.

We perform two different sets of experiments with this data. The first is to determine whether the spectral feature vectors can be used to cluster the different views of the same object together, and to differentiate between different objects. Here we amalgamate 10 uniformly spaced images from each of the three different view sequences into a single data-base. The second set of experiments focuses on whether, the spectral feature vectors can be used for the finer task of determining the view structure of the individual objects. Here we treat each object sequence separately.

6.1. Graph clustering

In our first set of experiments, our aim is to investigate which combination of spectral feature-vector and embedding strategy gives the best set of graph-clusters. In other words, we aim to explore which method gives the best definition of clusters for the different objects. The data-base used in our study contains 10 images from each of the CMU, MOVI and chalet sequences.

In Fig. 5 we compare the results obtained with the different spectral feature vectors. In the figure we show the matrix of pairwise Euclidean distances between the feature-vectors for the different graphs (this is best viewed in colour). The matrix has 30 rows and columns (i.e. one for each of the images in the three sequences with the three sequences concatenated), and the images are ordered according to the position in the sequence. From left-to-right and top-to-bottom, the different panels show the results obtained when the feature-vectors are constructed using the eigenvalues of the adjacency matrix, the volumes, the perimeters, the Cheeger constants, the inter-mode adjacency matrix and the inter-mode distance. From the pattern of pairwise distances, it is clear that the eigenvalues and the inter-mode adjacency matrix give the best block structure in the matrix. Hence these two attributes may be expected to result in the best clusters.

To test this assertion, in Fig. 4 we show the leading eigenvectors of the embedding spaces for the spectral feature-vectors. The left-hand column shows the results obtained with PCA. The middle column shows the results obtained with independent component analysis. The right-hand column shows the results obtained with MDS. The different rows are for the different attributes. From the plots, it is clear that the best clusters are obtained when MDS is applied to the vectors of eigenvalues and the inter-mode adjacency matrix. PCA, on the other hand, does not give a space in which there is a clear cluster-structure. Fig. 5 shows the matrices of L2 distances between each pair of graphs. Clear block structures are present in the distance for the vector of leading eigenvalues and the inter-mode adjacency matrix. In other words, both of the vector of

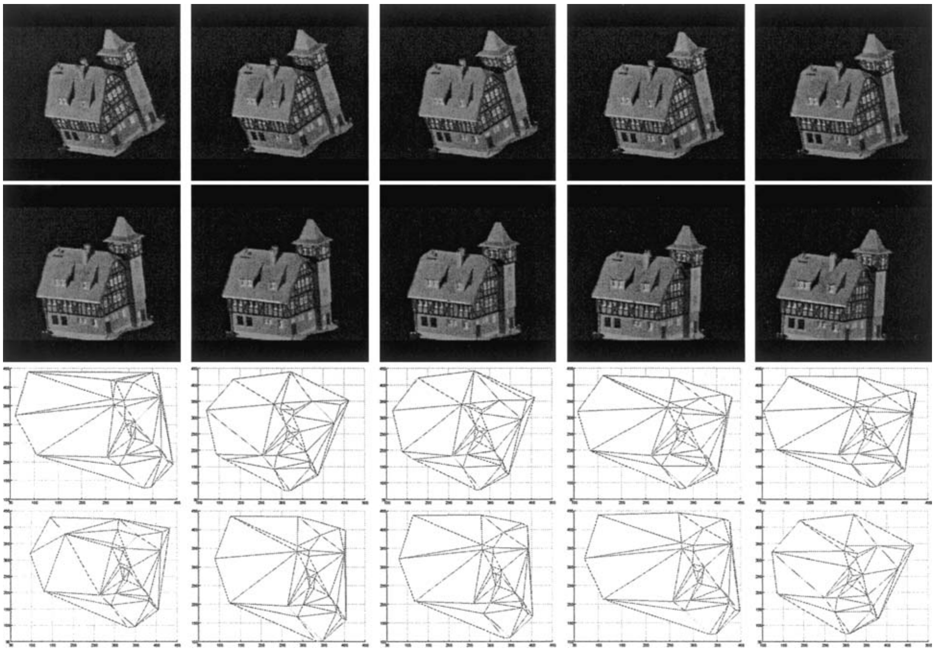

Fig. 1. CMU sequence and corresponding graphs.

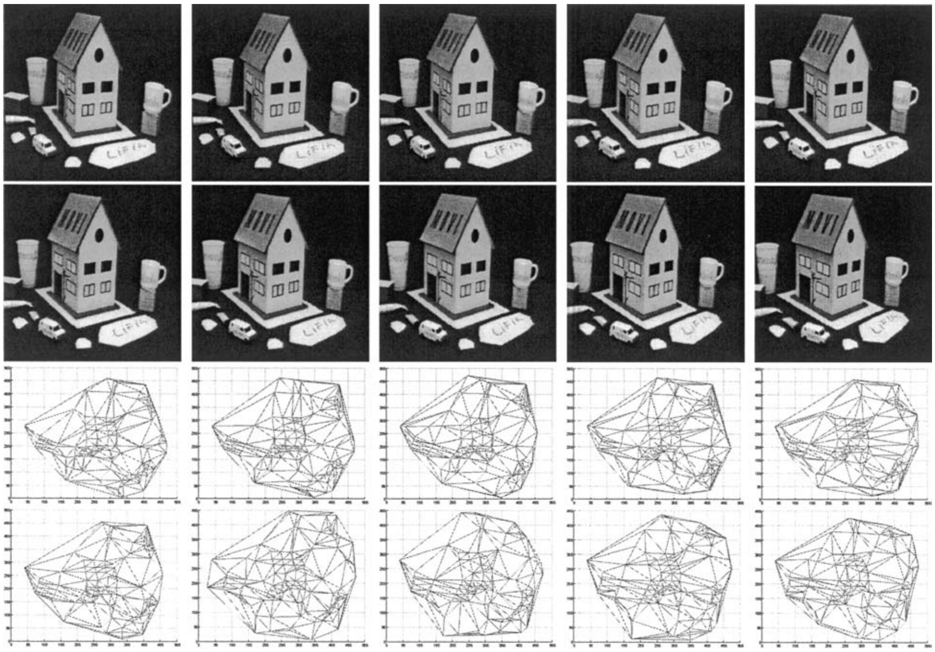

Fig. 2. MOVI sequence and corresponding graphs.

leading eigenvalues and the inter-mode adjacency matrix result in good clusters.

Next, we show the clustering results for the two well behaved features. i.e. the vector of leading eigenvalues and the inter-mode adjacency matrix. In Figs. 6–11, we show the clustering results obtained with vectors of different spectral attributes. For clarity, the results are displayed by positioning image "thumbnails" at the position associated with the

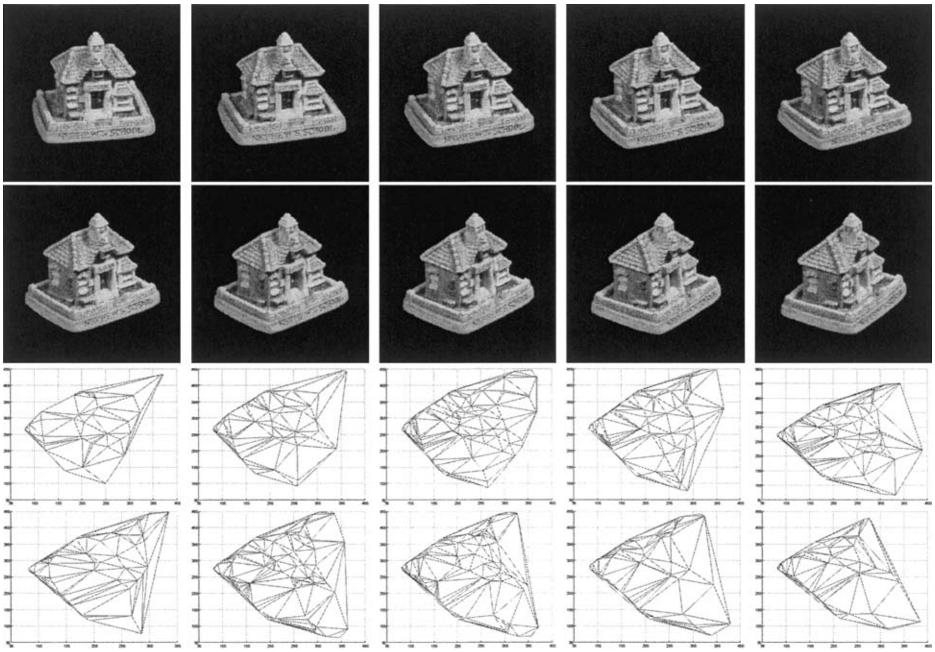

Fig. 3. The chalet sequence and corresponding graphs.

Table 1
Number of feature points extracted from the three image sequences

Image number	1	2	3	4	5	6	7	8	9	10
CMU	30	32	32	30	30	32	30	30	30	31
MOVI	140	134	130	136	137	131	139	141	133	136
Chalet	40	57	92	78	90	64	113	100	67	59

corresponding spectral feature vector in relevant eigenspace. In each case we visualise the thumbnails in the space spanned by the leading two eigenvectors generated by the embedding method (PCA, ICA or MDS). First, we investigate the clusters obtained from the vector of leading eigenvalues of the adjacency matrix. Fig. 6 shows the result obtained using PCA, Fig. 7 that obtained using ICA and Fig. 8 that obtained using MDS. There are a number of conclusions that can be drawn from these plots. First, the cluster associated with the Swiss chalet is always the least compact. Second, the best clusters are produces using MDS. That is to say the clusters are most compact and best separated. This is perhaps surprising, since this method uses only the set of pairwise distances between graphs, and overlooks the finer information residing in the vectors of spectral features. Although the clusters delivered by PCA are poorer, the results can be improved by using ICA. However, in none of the three cases is there any overlap between clusters. More importantly, they can be separated by straight-lines, i.e. they are linearly separable.

In Figs. 9–11, we repeat this sequence of experiments for the inter-mode adjacency matrix. This is a pairwise relational attribute. In each case the clusters are slightly improved. This is most marked when PCA is used to perform the embedding.

Finally, we compare the performance of the graph embedding methods using a measure of their classification accuracy. Each of the six graph spectral features mentioned above are used. We have assigned the graphs to classes using the K-means classifier. The classifier has been applied to the raw Euclidean distances, and to the distances in the reduced dimension feature-spaces obtained using PCA, ICA and MDS. In Table 2, we list the number of correctly classified graphs. From the table, it is clear that the eigenvalues and the inter-mode adjacency matrix are the best features since they return higher correct classification rates. Inter-mode distance is the worst feature for clustering graphs. We note also that classification in the feature-space produced by PCA is better than in the original feature vector spaces. However, the best results come from the MDS embedded class spaces.

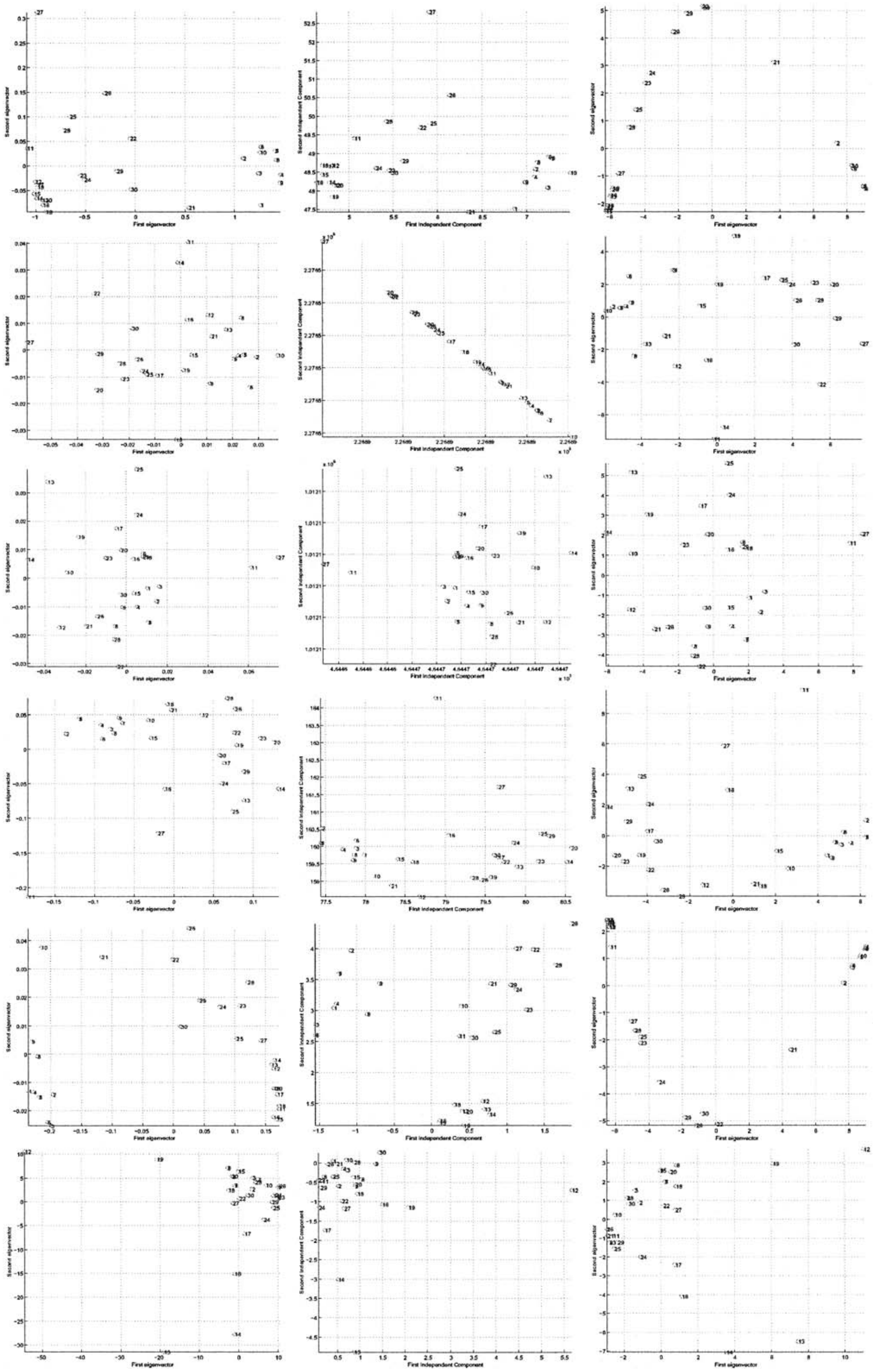

Fig. 4. PCA, ICA and MDS embedding using the spectral features of binary adjacency graph spectra, volumes, perimeters, Cheeger constants, inter-mode adjacency matrix and inter-mode distances.

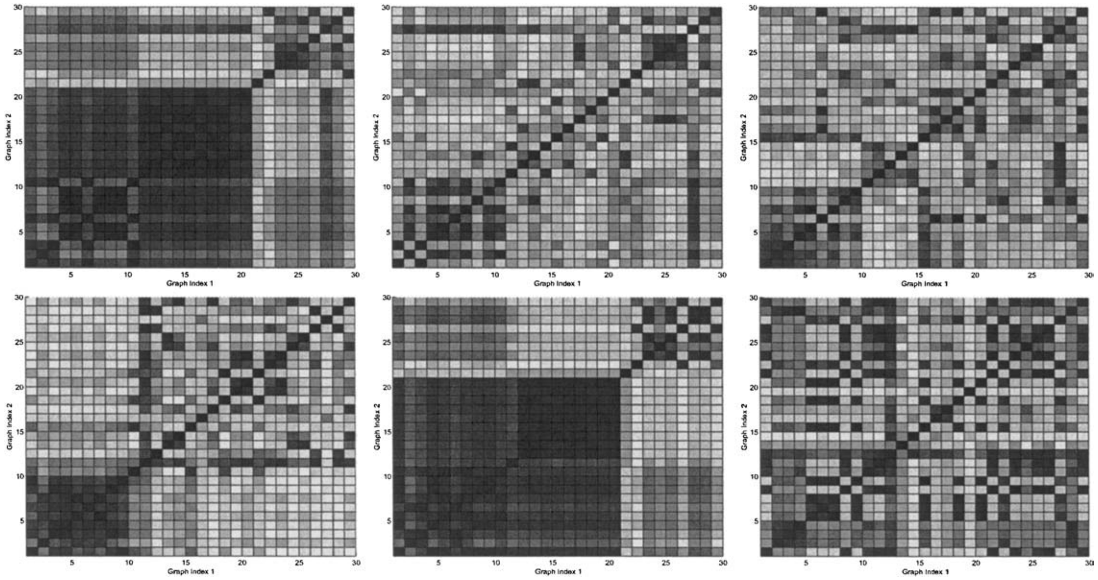

Fig. 5. Distance maps using the spectral features of binary adjacency graph spectra, volumes, perimeters, Cheeger constants, inter-mode adjacency matrix and inter-mode distances.

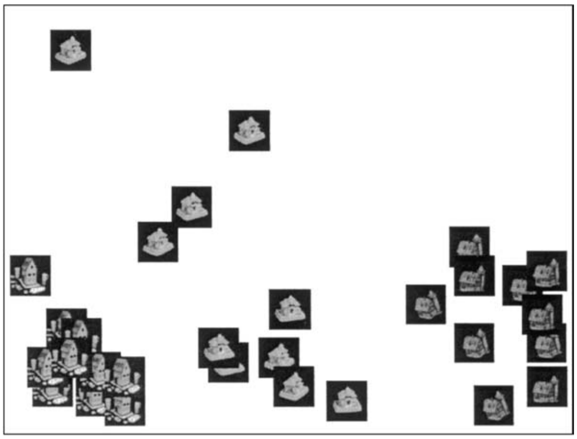

Fig. 6. PCA space clustering using the eigenvalues.

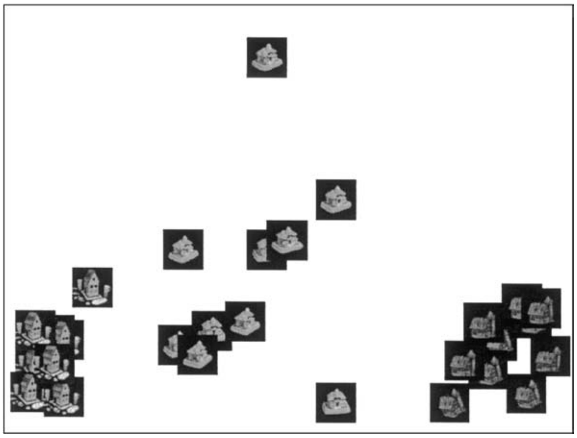

Fig. 7. ICA space clustering using the eigenvalues.

6.2. View structure

In the second set of experiments, we investigate the different image sequences separately. Here the aim is to determine whether the methods explored in the previous section can identify the finer view structure of the different objects. At this point it is important to note that we have used Delaunay graphs to abstract the corner features. Our view-based analysis explores how the edge-structure of the graphs changes with varying object pose. Since the Delaunay graph is the neighbourhood graph of the Voronoi tessellation, i.e. the locus of the median line between adjacent points, it may be expected to reflect changes in the shape of the arrangement of corner features. We investigate which combination of spectral feature-vector and embedding strategy gives the best view manifold for the different objects. In particular, we aim to determine whether the different views are organised into a well structured view trajectory, in which subsequent views are adjacent to one-another.

Figs. 12–17 show the results obtained with vectors of different spectral attributes. In each case, the top row shows the results obtained for the CMU sequence, the second row those for the MOVI house and the bottom row shows those for the chalet sequence. In the left-hand column of each

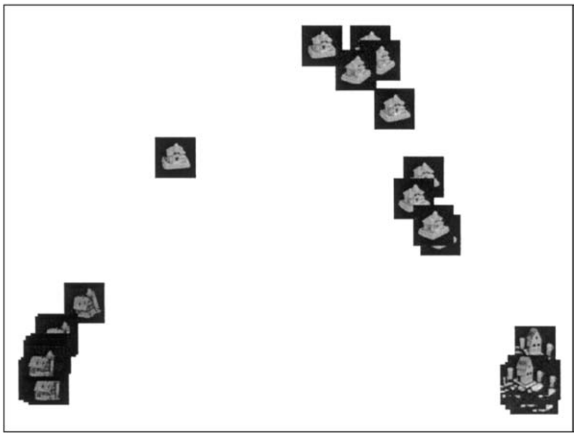

Fig. 8. MDS space clustering using the eigenvalues.

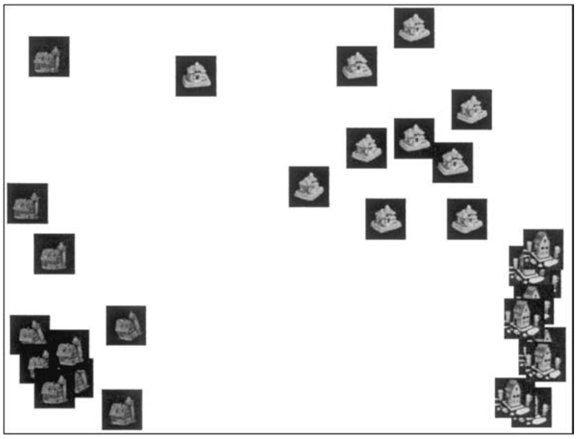

Fig. 9. PCA space clustering using the inter-mode adjacency matrix.

figure, we show the matrix of pairwise distances between the relevant spectral feature vectors. In the second column, we show the eigenspace extracted by applying PCA to the covariance matrix for the spectral feature vectors. The third column shows the eigenspace generated by applying ICA to the covariance matrix. Finally, the right-hand column shows the result of applying MDS to the matrix of distances.

Figs. 12–15 show the results obtained when the unary attributes are used. These are respectively vectors of ordered eigenvalues, volumes, perimeters and Cheeger constants. 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.

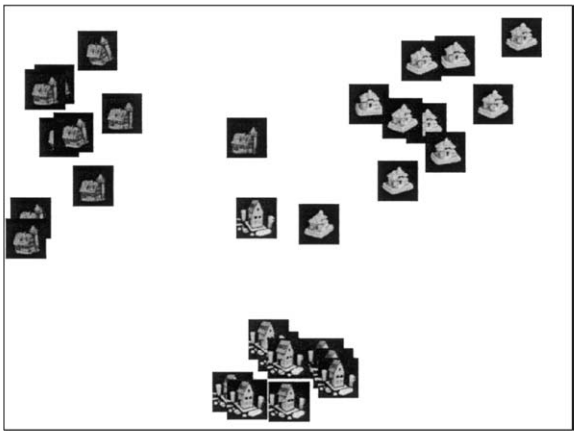

Fig. 10. ICA space clustering using the inter-mode adjacency matrix.

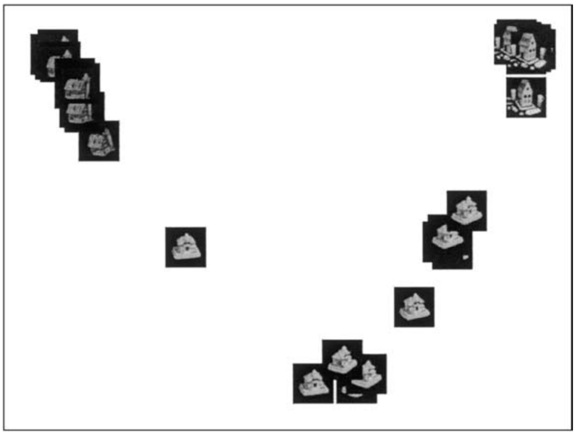

Fig. 11. MDS space clustering using the inter-mode adjacency matrix.

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. The volume and perimeter give the most marked clusters in the distribution of pairwise distances, where the view event-structure for the MOVI sequence and the Swiss chalet sequence is clearly seen as a block-structure in the plots. However, in the case of the PCA and MDS plots the similarities are less marked. In the case of the volume attribute, in the PCA plots there is a clear cluster-structure, while in the MDS plots there is a clear break in the trajectories. In the corresponding plots for the perimeter attribute there is neither clear cluster structure in the PCA plot, nor trajectory structure in the MDS plot. Finally, the Cheeger constant does not result in a clear block structure in the distance

Table 2
Correct classifications

Features	Eigenvalues	Volumes	Perimeters	Cheeger	Adjacency	Distances
Raw vector	29	26	26	13	25	12
PCA	29	27	26	17	25	12
ICA	30	24	26	17	18	12
MDS	29	28	27	16	29	17

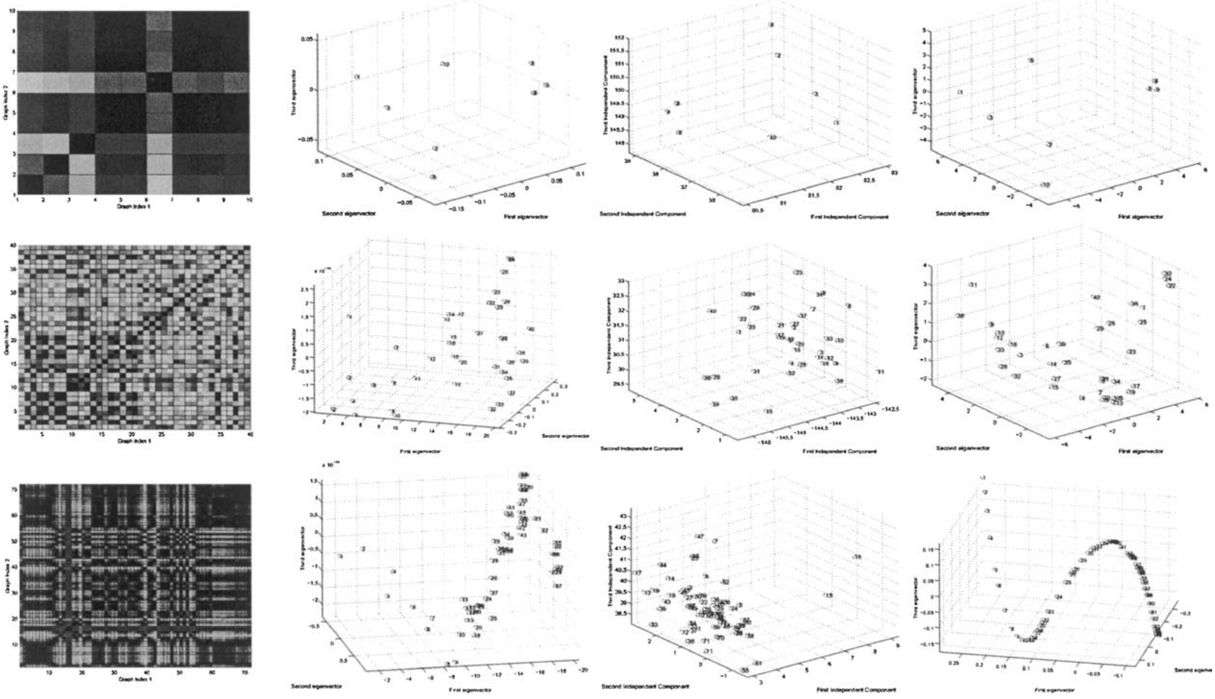

Fig. 12. Binary adjacency graph spectra.

plot. There is a good trajectory structure in the MDS plot for the Swiss chalet, but little structure of practical use in the remaining plots.

To conclude this subsection, we turn our attention to the pairwise attributes. First we consider the inter-mode adjacency matrix. This is a relational quantity and reflects the arrangement of eigenmodes of the original adjacency matrix. From Fig. 16 we can see that this gives good clusters in the PCA plots and good trajectories in the MDS plots for all three data-sets. Our last experiment uses the cluster distance measure. Fig. 17 shows that in all the three cases the block structures are obvious. The trajectories for the MOVI sequence and the chalet sequence are particularly clear.

We now provide some comparison of the different spectral representations. To do this we plot the normalised squared eigenvalues against the eigenvalue magnitude order i:

$$\hat{\lambda}_i^2 = \frac{\lambda_i^2}{\sum_{i=1}^{n} \lambda_i^2}.$$

In the case of the parametric eigenspace, these represent the fraction of the total data variance residing in the direction of the relevant eigenvector. In the case of MDS, the normalised squared eigenvalues represent the variance of the inter-graph distances in the directions of the eigenvectors of the similarity matrix.

In Fig. 18, we show the eigenvalues of the spectral-vector covariance matrix. The plots in the left-hand column are for the CMU sequence, the plots in the middle column are for the MOVI sequence, and the plots in the right-hand column are for the chalet sequence. The top row is for the unary features while the bottom row is for the binary

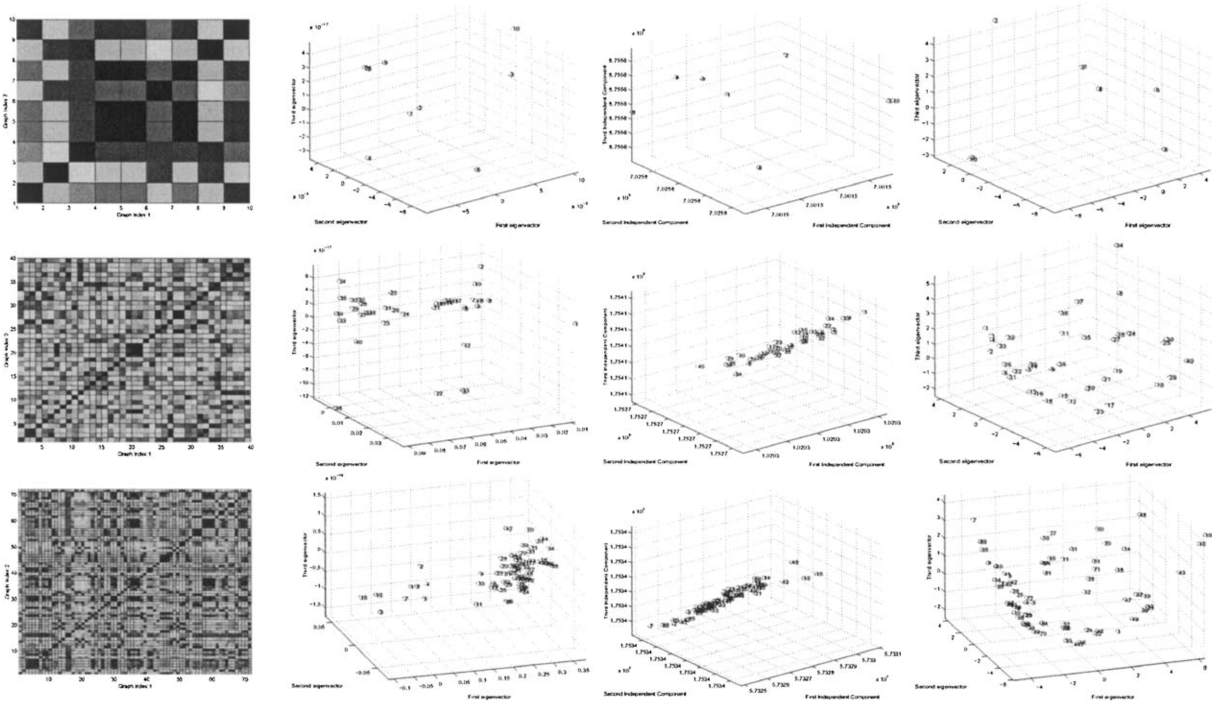

Fig. 13. Volumes.

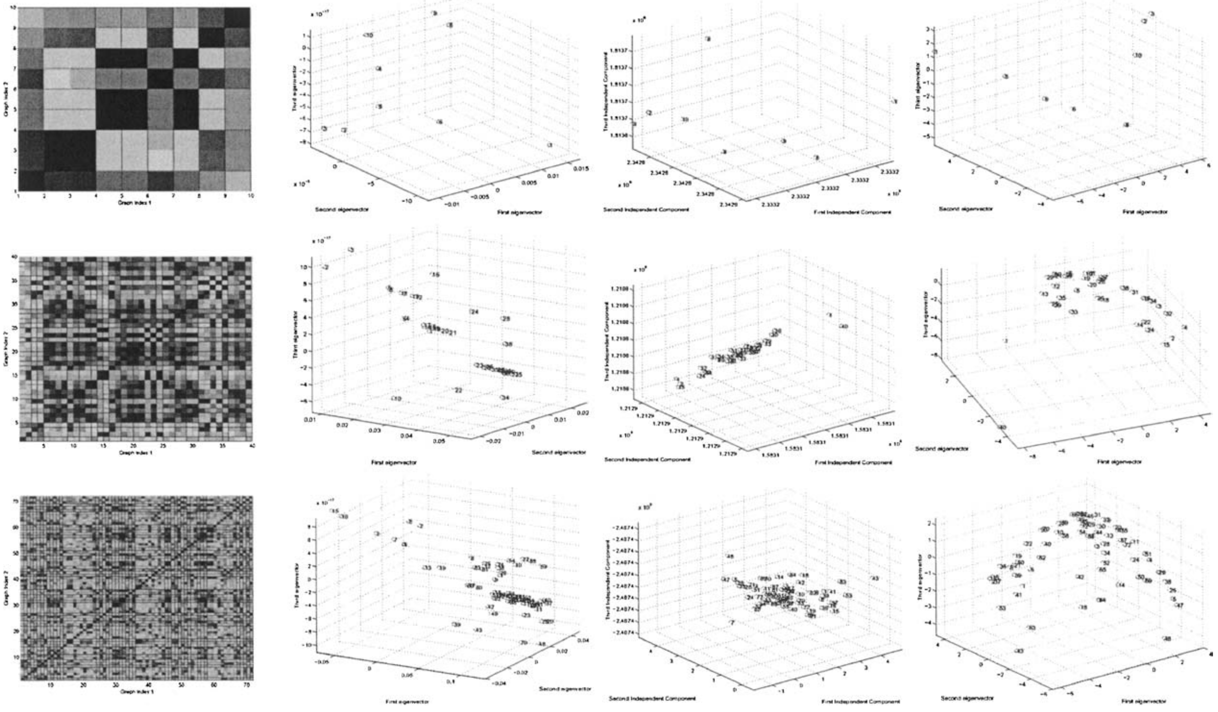

Fig. 14. Perimeters.

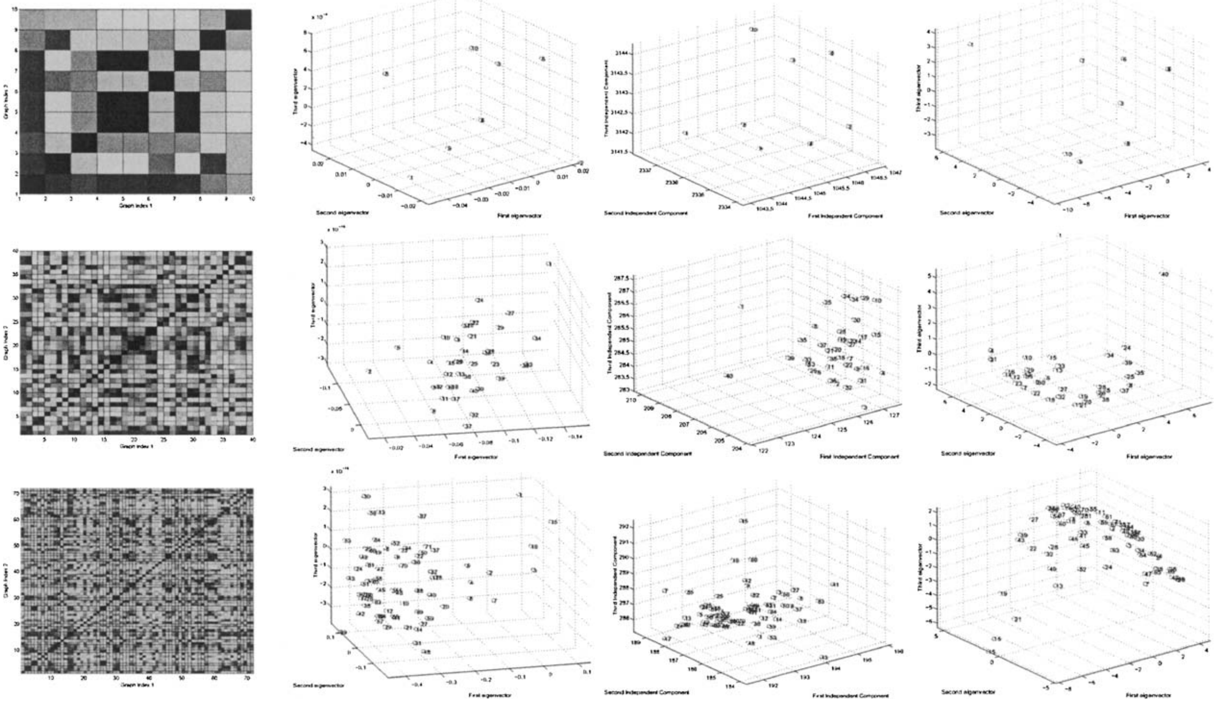

Fig. 15. Cheeger constants.

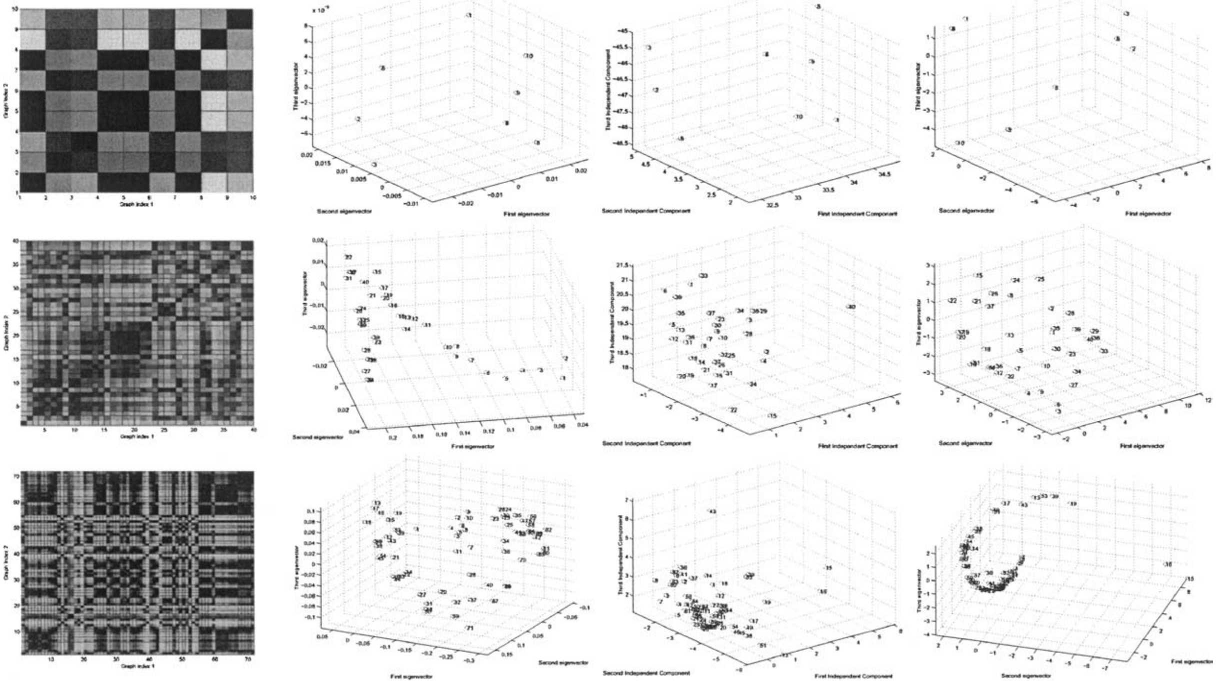

Fig. 16. Inter-mode adjacency matrix.

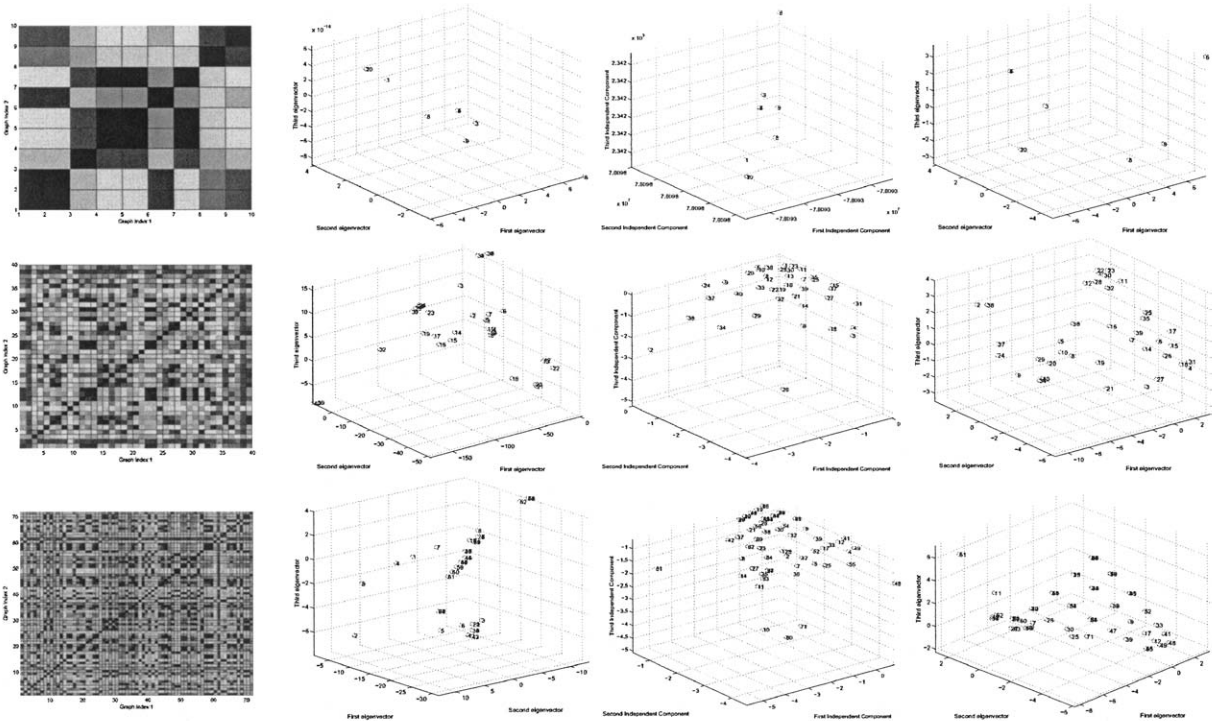

Fig. 17. Inter-mode distances.

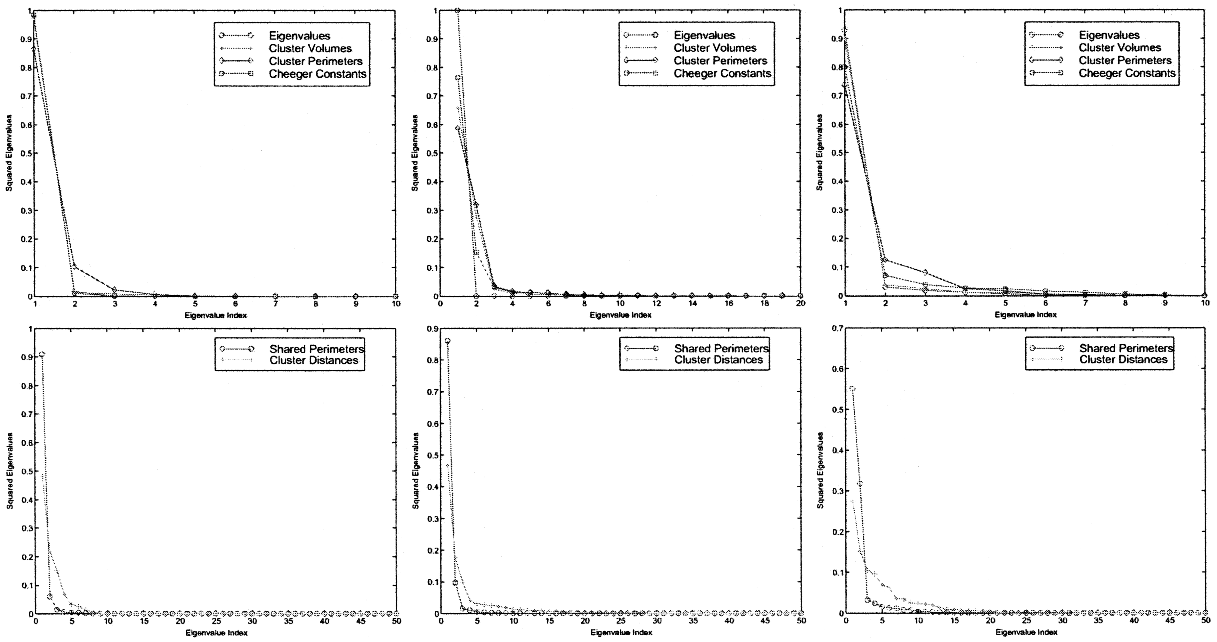

Fig. 18. Comparison of graph spectral features in eigenspaces.

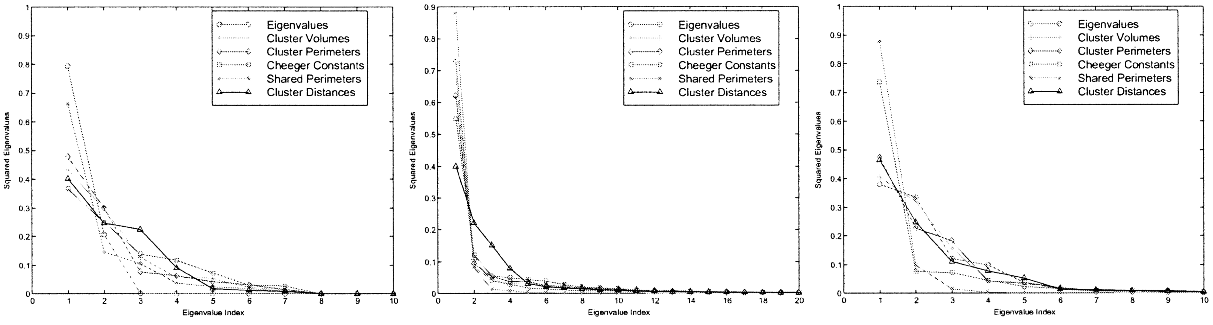

Fig. 19. Comparison of graph spectral features in MDS spaces.

features. The main feature to note is that the vector of adjacency matrix eigenvalues has the fastest rate of decay, i.e. the eigenspace has a lower latent dimensionality, while the vector of Cheeger constants has the slowest rate of decay, i.e. the eigenspace has greater dimensionality. In the case of the binary attributes, the inter-mode adjacency matrix results in the eigenspace of lower dimensionality. In Fig. 19 we repeat the sequence of plots for the three house data-sets, but merge the curves for the unary and binary attributes into a single plot by using MDS space embedding. Again the vector of adjacency matrix eigenvalues gives the space of lower dimensionality, while the vector of inter-mode distances gives the space of greatest dimensionality.

7. 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, Cheeger numbers, inter-mode adjacency matrices and inter-mode edge-distance for the eigenmodes of the adjacency matrix. The embedding strategies are PCA, ICA and MDS. Our empirical study is based on corner adjacency graphs extracted from 2D views of 3D objects. We investigate two problems. The first of these is that of clustering the different views of the three objects together. The second problem is that of organising different views of the same object into well structured trajectories, in which subsequent views are adjacent to one another and there are distinct clusters associated with different views. In both cases the best results are obtained when we apply MDS to the vectors of leading eigenvalues.

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 studying in more detail the structure of the pattern-spaces resulting from our spectral features. Here we intend to study how support vector machines and the EM algorithm can be used to learn the structure of the pattern spaces. Finally, we intend to investigate whether the spectral attributes studied here can be used for the purposes of organising large image data-bases.

References

[1] H.G. Barrow, R.M. Burstall, Subgraph isomorphism, matching relational structures and maximal cliques, Inform. Process. Lett. 4 (1976) 83–84.

[2] M. Fischler, R. Elschlager, The representation and matching of pictorical structures, IEEE Trans. Comput. 22 (1) (1973) 67–92.

[3] B. Huet, E.R. Hancock, Shape recognition from large image libraries by inexact graph matching, Pattern Recognition Lett. 20 (11–13) (1999) 1259–1269.

[4] K. Sengupta, K.L. Boyer, Organizing large structural modelbases, Pattern Anal. Mach. Intell. 17 (4) (1995) 321–332.

[5] C.M. Cyr, B.B. Kimia, 3D Object recognition using shape similarity-based aspect graph. In: ICCV'01, Vol. I, Vancouver, Canada, 2001, pp. 254–261.

[6] T. Hofman, J.M. Buhmann, Pairwise data clustering by deterministic annealing, Pattern Anal. Mach. Intell. 19 (2) (1997) 192–192.

[7] V. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Sov. Phys.-Dokl. 10 (1966) 707–710.

[8] M.A. Eshera, K.S. Fu, An image understanding system using attributed symbolic representation and inexact graph-matching, J. Assoc. Comput. Mach. 8 (5) (1986) 604–618.

[9] A. Sanfeliu, K.S. Fu, A distance measure between attributed relational graphs for pattern recognition, IEEE Trans. Syst. Man Cybern. 13 (3) (1983) 353–362.

[10] H. Bunke, K. Shearer, A graph distance metric based on the maximal common subgraph, Pattern Recognition Lett. 19 (1998) 255–259.

[11] H. Bunke, Error correcting graph matching: on the influence of the underlying cost function, IEEE Trans. Pattern Anal. Mach. Intell. 21 (1999) 917–922.

[12] A. Torsello, E.R. Hancock, Efficiently computing weighted tree edit distance using relaxation labeling, Lect. Notes Comput. Sci. 2134 (2001) 438–453.

[13] W.J. Christmas, J. Kittler, M. Petrou, Structural matching in computer vision using probabilistic relaxation, IEEE Trans. Pattern Anal. Mach. Intell. 17 (8) (1995) 749–764.

[14] R.C. Wilson, E.R. Hancock, Structural matching by discrete relaxation, IEEE Trans. Pattern Anal. Mach. Intell. 19 (6) (1997) 634–648.

[15] R. Myers, R.C. Wilson, E.R. Hancock, Bayesian graph edit distance, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 628–635.

[16] F.R.K. Chung, Spectral Graph Theory, CBMS series 92, American Mathematical Society, Providence, RI, 1997.

[17] H. Murase, S.K. Nayar, Illumination planning for object recognition using parametric eigenspaces, IEEE Trans. Pattern Anal. Mach. Intell. 16 (12) (1994) 1219–1227.

[18] C. Chatfield, A.J. Collins, Introduction to Multivariate Analysis, Chapman & Hall, London, 1980.

[19] W.S. Torgerson, Multidimensional scaling. i. theory and methods, Psychometrika 17 (1952) 401–419.

[20] J.C. Gower, Some distance properties of latent root and vector methods used in multivariate analysis, Biometrika 53 (1966) 325–328.

[21] J.B. Kruskal, Nonmetric multidimensional scaling: a numerical method, Psychometrika 29 (1964) 115–129.

[22] R. Wang, H. Freeman, Object recognition based on characteristic view classes, Proceedings of the ICPR, Vol. I, Oxford, UK, 1990, pp. 8–12.

[23] J.H. Stewman, K.W. Bowyer, Aspect graphs for convex planar-face objects, Proceedings of the IEEE Workshop on Computer Vision, Miami, FL, 1987, pp. 123–130.

[24] Z. Gigus, J. Malik, Computing the aspect graph for line drawings of polyhedral objects, IEEE Trans. Pattern Anal. Mach. Intell. 12 (2) (1990) 113–122.

[25] A.R. Pope, D.G. Lowe, Probabilistic models of appearance for 3-D object recognition, Int. J. Comput. Vision 40 (2) (2000) 149–167.

[26] J. Jia, K. Abe, Recognizing 3D objects by using models learned automatically from 2D training images, IEEE Trans. Pattern Anal. Mach. Intell. 14 (3) (2000) 315–338.

[27] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, IEEE Trans. Pattern Anal. Mach. Intell. 23 (4) (2001) 349–361.

[28] A. Leonardis, H. Bischof, Robust recognition using eigenimages, Comput. Vision Image Understanding 78 (1) (2000) 99–118.

[29] A.M. Martinez, A.C. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228–233.

[30] P.N. Belhumeur, D.J. Kriegman, What is the set of images of an object under all possible illumination conditions, Int. J. Comput. Vision 28 (3) (1998) 245–260.

[31] D.J. Kriegman, J. Ponce, Computing exact aspect graphs of curved objects: solids of revolution, Int. J. Comput. Vision 5 (2) (1990) 119–135.

[32] M.S. Costa, L.G. Shapiro, 3D Object Recognition and Pose with Relational Indexing, Comput. Vision and Image Understanding 79 (3) (2000) 364–407.

[33] D.J. Kriegman, Computing stable poses of piecewise smooth objects, Comput. Vision Graph. Image Process. 55 (2) (1992) 109–118.

[34] R. Malik, T. Whangbo, Angle densities and recognition of 3D objects, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1) (1997) 52–57.

[35] S. Petitjean, The enumerative geometry of projective algebraic-surfaces and the complexity of aspect graphs, Int. J. Comput. Vision 19 (3) (1996) 261–287.

[36] J. Rieger, The geometry of view space of opaque objects bounded by smooth surfaces, Artif. Intell. 44 (1990) 1–40.

[37] M. Seibert, A.M. Waxman, Adaptive 3-d object recognition from multiple views, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 107–124.

[38] I. Biederman, Recognition by components: a theory of human image understanding, PsychR 94 (2) (1987) 115–147.

[39] I. Biederman, Geon based object recognition, in: BMVC'93, Guildford, UK, 1993.

[40] S. Petitjean, J. Ponce, D.J. Kriegman, Computing exact aspect graphs of curved objects: algebraic surfaces, Int. J. Comput. Vision 9 (3) (1992) 231–255.

About the Author—BIN LUO received his BEng. degree in electronics and MEng. degree in computer science from Anhui university of China in 1984 and 1991, respectively. From 1996 to 1997, he was working as a British Council visiting scholar at the University of York under the Sino-British Friendship Scholarship Scheme(SBFSS). In 2002, he was awarded the Ph.D. degree in Computer Science from the University of York, the United Kingdom. He is at present a professor at Anhui University of China.

He has published some 60 papers in journals, edited books and refereed conferences. His current research interests include graph spectral analysis, large image database retrieval, image and graph matching, statistical pattern recognition and image feature extraction.

About the Author—RICHARD WILSON read physics at St John's College, University of Oxford, graduating with first class honours in 1992. Between 1992 and 1995, he undertook research at the University of York on the topic of relational graph matching for which he was awarded the D.Phil. degree. After a period of postdoctoral research, he was awarded an EPSRC Advanced Research Fellowship in 1998, which he currently holds in the Department of Computer Science at the University of York.

He has published some 70 papers in journals, edited books and refereed conferences. He received an honourable mention in the 1997 Pattern Recognition Society awards and is currently an Associate Editor of the journal Pattern Recognition. His research interests are in statistical and structural pattern recognition, high-level vision and scene understanding.

About the Author—EDWIN HANCOCK studied physics as an undergraduate at the University of Durham and graduated with honours in 1977. He remained at Durham to complete a Ph.D. in the area of high energy physics in 1981. Following this he worked for ten years as a researcher in the fields of high-energy nuclear physics and pattern recognition at the Rutherford-Appleton Laboratory (now the Central Research Laboratory of the Research Councils). During this period he also held adjunct teaching posts at the University of Surrey and the Open University.

In 1991 he moved to the University of York as a lecturer in the Department of Computer Science. He was promoted to Senior Lecturer in 1997 and to Reader in 1998. In 1998, he was appointed to a Chair in Computer Vision.

Professor Hancock now leads a group of some 15 faculty, research staff and Ph.D. students working in the areas of computer vision and pattern recognition. His main research interests are in the use of optimisation and probabilistic methods for high and intermediate level vision. He is also interested in the methodology of structural and statistical pattern recognition. He is currently working on graph-matching, shape-from-X, image data-bases and statistical learning theory. His work has found applications in areas such as radar terrain analysis, seismic section analysis, remote sensing and medical imaging. Professor Hancock has published some 60 journal papers and 200 refereed conference publications. He was awarded the Pattern Recognition Society medal in 1991 for the best paper to be published in the journal Pattern Recognition. The journal also awarded him an outstanding paper award in 1997.

Professor Hancock has been a member of the Editorial Boards of the journals IEEE Transactions on Pattern Analysis and Machine Intelligence, and, Pattern Recognition. He has also been a guest editor for special editions of the journals Image and Vision Computing and Pattern Recognition, and he is currently a guest editor of a special edition of IEEE Transactions on Pattern Analysis and Machine Intelligence devoted to energy minimisation methods in computer vision. He has been on the programme committees for numerous national and international meetings. In 1997, he established a new series of international meetings on energy minimisation methods in computer vision and pattern recognition.