Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Manifold embedding for shape analysis

Bai Xiao^{a,*}, Edwin Hancock^b, Hang Yu^c

^a School of Computer Science and Engineering, Beihang University, Beijing 100191, China

^b Department of Computer Science, University of York, York Y010 5DD, UK

^c Department of Computer Science, University of Florida, Gainesville, FL 32611-6120, USA

ARTICLE INFO

ABSTRACT

Available online 15 March 2010 Keywords: Shape analysis Manifold learning Graph embedding Spectral alignment Shape analysis played important role in computer vision based tasks. The importance of shape information relies that it usually contains perceptual information, and thus can be used for high level visual information analysis. Currently, there are many ways that shapes can be represented as a structural manner using graphs. Hence shapes can be analyzed by using graph methods. This paper describes how graph-spectral methods can be used to transform the node correspondence problem into one of point-sets alignment. We commence by using the ISOMAP algorithm to embed the nodes of a graph in a low-dimensional Euclidean space. With the nodes in the graph transformed to points in a metric space, we can recast the problem of graph-matching into that of aligning the point-sets. Here we use semidefinite programming to develop a robust point-sets correspondences algorithm. Variations in graph structure using the covariance matrix for corresponding embedded point-positions is captured. We construct a statistical point distribution model for the embedded node positions using the eigenvalues and eigenvectors of the covariance matrix. We show how to use this model to project individual graph, i.e. shape into the eigenspace of the point position covariance matrix. We illustrate the utility of the resulting method for shape analysis and recognition on COIL and MPEG-7 databases.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Shape analysis played important role in computer vision based tasks. The importance of shape information relies that it usually contains high level visual information. There has already many methods for shape analysis. The first part methods can be described as statistical modeling [4,22,15,17]. Here a well established route to construct a pattern space for the data-shapes is to use principal components analysis (PCA) [12]. This commences by encoding the image data or shape landmarks as a fixed length long vector. The data are then projected into a low-dimensional space by projecting the long vectors onto the leading eigenvectors of the sample covariance matrix. This approach has been proved to be particularly effective, especially for face data and medical images, and has lead to the development of more sophisticated analysis methods capable of dealing with quite complex pattern spaces.

However, shape can also be abstracted in a structural manner using a graph or a tree representation [26,30]. Examples include the representation of the skeletal structure of silhouettes using shock-trees [26], and the use of point proximity or region adjacency graphs to describe object arrangement [19]. The observation underpinning this paper is that although there has

* Corresponding author. E-mail address: little_point_baixiao@hotmail.com (B. Xiao). been considerable effort aimed at learning the variation of shape in the landmark and parameter domains, little effort has been devoted to learning the statistical variations in structural representations of shape. This is a big omission since structural representations can capture variations in shape in a scale invariant manner. In addition, structural representations are intrinsically of higher level, and are closer to the semantic content of a scene.

In this paper, we introduce a graph based framework for shape analysis. We combine the ideas from manifold learning and statistical modeling. After we have a set of graphs each representing a shape, the first step is to embed the nodes of each graph in a vector space. We introduce the metric graph embedding method. In the mathematics literature, there is a considerable body of work aimed at understanding how graphs can be embedded on a manifold so as to minimize a measure of distortion. Recently, there has been considerable interest in the pattern analysis community in how to embed complex relational data in a low dimensional manifold. Collectively [13,24,1], these methods are known as manifold learning theory. Their collective aim is to develop variants of the classical methods of PCA [12] and MDS [6], that can be used to better capture localised variations on the structure of the data. In this paper, we investigate whether methods from manifold learning theory can be combined with spectral graph theory to develop effective tools for graph structure matching. The idea is to use manifold learning methods



^{0925-2312/\$ -} see front matter \circledcirc 2010 Elsevier B.V. All rights reserved. doi:10.1016/j.neucom.2009.10.023

to embed the graphs in a low dimensional coordinate space, and to use point-pattern matching techniques to find correspondences between nodes. We proceed as follows. We start by performing a strategy similar to ISOMAP [13] to embed the graphs in a Euclidean pattern space. This is done by applying MDS (multidimensional scaling) [6] to the matrix of geodesic distances between nodes. Once embedded in this space, we can use pointalignment methods to match the nodes of the graphs. To do this we develop a variant of the Scott and Longuet-Higgins [10] algorithm. There are two problems with the existing method. First, the correspondences are obtained from the matrix formed by taking the outer-product of the singular vectors of the point association matrix. This matrix does not have a clear interpretation, and the resulting search for correspondences using the maximum row and column entries is ad hoc in nature. Second, the method breaks down when the point-sets being matched are of very different sizes [3].

Our first contribution is to extend manifold learning algorithms on graph space for graph embedding. By doing so, we transform graphs to point-sets in vector space by using tradition manifold learning algorithms, i.e. ISOMAP [13]. An important feature for graph embedding is that we can perform a number of graph manipulation tasks by applying simple point pattern analysis algorithms to the embedded point position vectors.

Following the first contribution, after embed each graph in point-sets, our second contribution in this paper is rather than performing singular value decomposition on the inter-point-set proximity matrix, we use semidefinite programming (SDP) [9] to locate correspondences. By using SDP we overcome the first of these problems, since it leads to a correspondence matrix which is doubly stochastic, and hence has clear meaning. Second, the resulting matching method is more robust to size difference than that of Scott and Longuet-Higgins. Once we found the correspondence between the nodes of the two graphs, we can apply our methods on image matching. We have showed our algorithms for matching on both synthetic and real-world database.

Our third contribution is to construct statistical model to learn the structure variation in graphs. Once we have solved the matching correspondence problem, we draw on ideas from linear deformable models to construct a simple and explicit statistical model for graph structure. One of the problems that limits the use of the structural clustering methods [2] is that they suffer from exponential complexity and are therefore not easily sampled from. To overcome the problem of exponential complexity we turn to the shape analysis literature, where principal components analysis has been proved to be a powerful way of capturing the variations in sets of landmark points for 2D and 3D objects [5].

Hence, we use the graph embedding to construct a statistical model for graph structure. Our aim is to construct a statistical model that can account for the distribution of embedded point positions for corresponding nodes in a sample of graphs. A reference graph is selected, and the correspondences between the nodes of each sample graph and the reference graph are established using the spatial alignment method. We capture variations in graph structure using the covariance matrix of the corresponding embedded point positions. We construct a point distribution model for the embedded node positions using the eigenvalues and eigenvectors of the covariance matrix. We show how to use this model to project individual graphs into the eigenspace of the point position covariance matrix. We also illustrate the utility of the resulting method for shape analysis. Experiments on the COIL and MPEG-7 databases are performed.

The outline of the paper is like this. In Section 2, we introduce the methods on how to embed the graph in coordinate spaces so transfer each graph to a point-set. Then the graph nodes correspondence problems can be solved by using a point-sets alignment algorithm which will be introduced in Section 3. With the correspondence at hand, in Section 4, we continue to construct a graph statistical model which can be used to compute the similarities between graphs. Finally, in Section 5, we evaluate our methods on experiments.

2. Metric embedding of graphs

We are interested in the abstract problem of embedding the nodes of a graph into an Euclidean space. Here we use Isomap [13] as a way to solve the low-distortion graph embedding problem. The idea behind Isomap is to apply classical MDS [29] to the matrix of geodesic distances between data-points. In this way the data are mapped from a high-dimensional input space to the low-dimensional space of a nonlinear manifold. Although the method was originally devised for dimensionality reduction, we can use it here for the low-distortion graph embedding problem.

2.1. Metric embedding using Isomap

To commence, suppose that the graph under study is denoted by G=(V,E) where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. Since we wish to adopt a graph-spectral approach we introduce the adjacency matrix A for the graph where

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

The graph can also be represented by geodesic distance matrix *S*. The pairwise geodesic distances between nodes d(u,v) are used as the elements of an $N \times N$ dissimilarity matrix *S*, whose elements are defined as follows:

$$S(u,v) = \begin{cases} d(u,v) & \text{if } u \neq v \\ 0 & \text{if } u = v \end{cases}$$
(2)

There are many algorithms to compute the geodesic distance matrix *S*, here for the simplicity we use the canonical Dijkstra algorithm [7].

Our goal is to find a low-distortion or distortion-free embedding from the graph metric space into a normed space. Here we use Isomap [13] as a way to solve the low-distortion graph embedding problem. The idea behind Isomap is to apply classical MDS [29] to map data points from their high-dimensional input space to low-dimensional coordinates of a nonlinear manifold. The key contribution is hence to apply MDS to the pairwise distances not in the input Euclidean space, but in the geodesic space of the manifold.

Although the method was originally devised for dimensionality reduction, we can use it here for the low-distortion graph embedding problem. Viewed as an isometric feature mapping, Isomap is a mapping $f : X \rightarrow Y$ from the observation space X to a Euclidean feature space Y that preserves as closely as possible the intrinsic metric structure of the observations, i.e. the distances between observations as measured along geodesic(shortest) paths of X [13]. The distortion in this embedding is nearly 1.

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

The first step of MDS is to calculate a matrix *K* whose element with row *r* and column *c* is given by $K(r,c) = -\frac{1}{2}[d^2(r,c) - \hat{d}^2(r,\cdot) - \hat{d}^2(\cdot,c) + \hat{d}^2(\cdot,\cdot)]$, where $\hat{d}(r,\cdot) = (1/N) \sum_{c=1}^{N} d(r,c)$ is the average dissimilarity value over the *r* th row, $\hat{d}(\cdot,c)$ is the similarly defined

average value over the *c* th column and $\hat{d}(\cdot,\cdot) = (1/N^2) \sum_{r=1}^{N} \times \sum_{c=1}^{N} d(r,c)$ is the average similarity value over all rows and columns of the similarity matrix *K*.

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

3. Semidefinite programming for graph matching

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

$$G_{ii} = \exp(-m_{ii}^2/2\delta^2) \tag{3}$$

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

$$m_{ii}^2(M) = (x_i - x_j)' \Sigma^{-1} (x_i - x_j)$$
(4)

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

With an inter-graph node distance matrix to hand, then one way to find correspondences is to use the Scott and Longuet-Higgins algorithm [10]. This involves performing the singular value decomposition (SVD) G=TDU. The matrices of T and U are orthogonal. The matrix D contains the singular values along its diagonal in descending numerical order. The final step is to compute the correlation between T's rows and U's columns, giving an association matrix P=TEU, where E is obtained by replacing each diagonal element in D by a 1. The element P_{ij} indicates the strength of attraction between feature $i \in I$ and $j \in J$. The rows of *P*, index the features in the first graph, and its columns those in the second graph. If P_{ii} is both the largest element in row *i* and column j then we regard these features as being in one-to-one correspondence with one-another. If P_{ij} is the greatest element in row *i* but not the greatest in column *j*, then we may regard $i \in I$ competing unsuccessfully for partnership with $j \in J$. Similar remarks apply if P_{ij} is the greatest element in its column but not in its row [10].

However, the Scott and Longuet-Higgins method can prove to be sensitive to instabilities in the singular vectors [3]. For this reason we turn to semidefinite programming as an alternative. The semidefinite programming problem (SDP) is essentially an ordinary linear program where the nonnegativity constraint is replaced by a semidefinite constraint on matrix variables. It is interesting to note that SDP is a special instance of a more general problem class called conic linear programs, where one seeks to minimize a linear objective function subject to linear constraints and a cone constraint [31]. The process has many applications, ranging from control theory to structural design. In particular, many hard optimization problems can be relaxed to a problem with convex quadratic constraints which, in turn, can be formulated as an SDP [21]. The handbook [34] has described the application of SDP on combinatorial optimization, on nonconvex quadratic programming, on eigenvalue and nonconvex optimization, etc.

The standard form for the primal problem is

min trace CY

s.t. trace $F_i Y = b_i (i = 1, 2, ..., m)$, $Y \ge 0$

where *C*, F_i and *Y* are real symmetric $n \times n$ matrices and b_i is a scalar. The constraint $Y \ge 0$ means that the variable matrix must lie on the closed convex cone of positive semidefinite solutions. Back to the problem of Longuet-Higgins' algorithm who try to find a solution matrix *P* which can best correlates with *G* in the sense of maximizing the inner product

$$P: G = \sum_{i} \sum_{j} P_{ij} G_{ij} = \operatorname{trace}(P^T G)$$

At the paper [10], the authors try to use spectral methods singular value decomposition (SVD) to find the solution. Although elegant and convenient, spectral methods are only guaranteed to find a locally optimal solution to the problem. For this reason in this paper we turn to the more general method of semidefinite programming to locate an optimal solution which utilizes the convexity properties of the matrix representation. To use SDP, we notice that neither *P* nor *G* is symmetric matrix. By solving this problem, we introduce the matrix *W* which is $(m+n) \times (m+n)$ size and the definition is given below:

$$W:\begin{bmatrix} O & P^T \\ P & O \end{bmatrix}$$

The same for matrix *G* we extend it into matrix *V* which is defined below:

$$V:\begin{bmatrix} O & G^T \\ G & O \end{bmatrix}$$

Matrix *V* is also $(m+n) \times (m+n)$ size.

Then maximize trace(P^TG) is to maximize trace(WV)+ trace(W^TV^T) which equals maximize trace(WV). Hence, W corresponds to C while V corresponds to Y in Eq. (5). We next to introduce the constraint matrix which in the symbol F_i which is $(m+n) \times (m+n)$ size and the definition is given below:

$$F_{1} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1/2 & \cdots & 1/2 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & & & & \\ 1/2 & 0 & \cdots & 0 & & \\ 1/2 & 0 & \cdots & 0 & & \\ \vdots & & & & & \\ 1/2 & 0 & \cdots & 0 & & \\ 1/2 & 0 & \cdots & 0 & & \\ \vdots & & & & & \\ 0 & 1/2 & \cdots & 0 & & \\ \vdots & & & & & \\ 0 & 1/2 & \cdots & 0 & & \\ \vdots & & & & & \\ 0 & 1/2 & \cdots & 0 & & \\ \vdots & & & & & \\ 0 & 1/2 & \cdots & 0 & & \\ \end{bmatrix}$$

And the cost vector b_i is a m+n length with each element equals to 1. The first m constraint matrices keeps the columns of variable matrix sum to 1 and the following n constraint matrices keeps the row variable matrix sum to 1. The reasons to set F_i matrices is like this. For example, F_1 V=1 equals: $\sum_j (1/2 \times V(1,j) + 1/2 \times V(j,1)) = 1$, $j \ge m$ while P(1,j) and P(j,1) both correspond to the same element in the matrix P, and sum of P(1,j) is the sum of the 1st column of matrix *P*. So we can use SDP methods to solve the *P* matrix. In this paper, we use the SDP solver developed by Fujisawa et al. [14]. In this case, we will finally obtain the association matrix *P* that maximizes trace (P^TG) from *Y*, and the solution matrix will be limited to be doubly stochastic.

4. Graph similarity

From previous sections, we have solved the correspondence between pairs of graphs by using the association matrix P. In this section, we continue the work from previous section, to introduce how we compute the similarity among a set of graphs. Part of this work has been published in [35]. The idea is to first find the correspondence between the embedded graph point-sets by using the methods from previous section. With the correspondence results at hand, we construct a statistical model for the embedded point-sets. We can then use some statistical properties, i.e. mean. covariance to characterize the embedded point-sets hence the corresponding graphs. From these statistical properties, we can extract a similarity measure for the graph. When a new graph is given, we can still use the constructed graph statistical model to evaluate the similarity between the new given graph and the mean of the training graph set. Below we will describe the details of the methods.

Let the sample be $T = \{G_1, G_2, ..., G_k, ..., G_K\}$ where the *k* th graph $G_k = (V_k, E_k)$ has node-set V_k and edge-set E_k . The result of performing metric embedding of the nodes of the *k* th graph is a matrix of co-ordinates Y_k . Our aim in this section is to construct a statistical model that can be used to describe the distribution of embedded node co-ordinates for the sample of graphs. Since the graphs contain different numbers of nodes, we truncate the co-ordinate matrices to remove the spatial dimensions corresponding to insignificant eigen-modes. Hence, we retain just the first *N* rows of each co-ordinate matrix. For the graph G_k the truncated node co-ordinate matrix is denoted by \hat{Y}_k .

To construct the statistical model for the sample of graph, we require correspondences between the nodes of each sample graph and the nodes of a reference structure. Here we take the reference graph to be the graph in the sample with the largest number of nodes. This graph has index $k^* = \operatorname{argmax}_{G_k \in T} |V_k|$. To locate the correspondences between the nodes of each sample graph and those of the reference graph, we use methods described in Section 3. We compute the association matrix *P* between each sample graph and the reference graph.

Once we have correspondences to hand, then we can construct the statistical model for the set of graphs. To do this we model variations in the positions of the embedded points using a point distribution model. Point distribution model has been used for shape analysis to characterize the variation within the shape space which is normally represented by feature points [5]. We commence by computing the mean point positions. The matrix of mean-position co-ordinates and the associated covariance matrix are

$$\hat{X} = \frac{1}{T} \sum_{k \in T} P_{k,k^*}^T \hat{Y}_k$$
$$\Sigma = \frac{1}{T} \sum_{k \in T} (P_{k,k^*}^T \hat{Y}_k - \hat{X}) (P_{k,k^*}^T \hat{Y}_k - \hat{X})^T$$

To construct the point-distribution model, we perform the eigendecomposition $\Sigma = \Psi \Gamma \Psi^T$ where $\Gamma = diag(\gamma_1, \gamma_2, \dots, \gamma_K)$ is the diagonal matrix of ordered eigenvectors and $\Psi = (\psi_1 | \dots | \psi_K)$ is the matrix with the correspondingly ordered eigenvectors as columns. We deform the mean-embedded node positions in the directions of the leading eigenvectors of the point-position covariance matrix Σ . Let $\tilde{\Psi}$ be the result of truncating the matrix Ψ after S columns and let b be a parameter-vector of length S. We convert the mean point position matrix with a long vector form. Let $Col_i(\hat{X})$ be the *i* th column of the mean-point position matrix \hat{X} . The long vector is given by $\hat{Z} = (Col_1^T(\hat{X}), Col_2^T(\hat{X}), \ldots)$. The long vector corresponding to deformed point set position is $\tilde{Z} = \hat{Z} + \tilde{\Psi}b$. The matrix with deformed point position as column is \hat{X} . An observed configuration of embedded nodes \tilde{Y} maybe fitted to the model. To do this the best fit parameters estimated using the least squares procedure

$$b^* = \operatorname*{argmin}_{b} (\tilde{Y} - \hat{X} - \tilde{\Psi}b)^T (\tilde{Y} - \hat{X} - \tilde{\Psi}b)$$

The best-fit parameter vector is $b^* = \tilde{\Psi}^T (\tilde{Y} - \hat{X})$.



Fig. 1. Delaunay graphs overlayed on the house images.



Fig. 2. Our algorithm for CMU and MOVI house sequences.

Finally, the similarity of a pair of graphs can be measured using the difference in their best-fit parameter vectors. Since the parameter-vector is just the projection of the corresponding graph into the eigenspace of the model, the difference is parameter vectors is related to the distance between graphs in the eigenspace. Suppose that the graphs G_{k_1} and G_{k_2} have best fit parameter vectors $b_{k_1}^*$ and $b_{k_2}^*$, respectively. The Euclidean distance between the parameter vectors is

$$d^{2}(k_{1},k_{2}) = (b_{k_{1}}^{*} - b_{k_{2}}^{*})^{T} (b_{k_{1}}^{*} - b_{k_{2}}^{*}) = (\hat{Y}_{k_{1}} - \hat{Y}_{k_{2}})^{T} \tilde{\Psi} \tilde{\Psi}^{T} (\hat{Y}_{k_{1}} - \hat{Y}_{k_{2}}).$$

5. Experiments

In this section, we provide some experiments for our methods. We divide this section into two parts. In the first part, we test our methods on image matching. We use COIL and MOVI data to find the corresponding between pairs of images. In the second part, we extend our experiment by using the matching results to perform shape image analysis. Here we use COIL data-set [23] together with MPEG-7 data-set [27].

5.1. Image matching

In this section, we provide some experimental evaluation of the new matching method.

Our experiments are based on some real-world data experiments. We also compare our methods with some alternatives. These alternatives are Shapiro and Brady [18] and Scott and Longuet-Higgins' [10] feature set matching methods. These two methods use coordinate information for the feature points, and do not incorporate the graph structure information. We also investigated Umeyama's [28] method. In our method we are concerned with matching the Delaunay triangulation of cornerfeatures. From each image in each view sequence, we extract corner features [11]. We use the detected feature points as nodes of a graph. The edge relations are obtained by computing the Voronoi tessellations of the feature points, and constructing the region adjacency graph, of the Voronoi regions. We apply our matching method to two image sequences (MOVI and Desk).

Table 1

Experiments results for MOVI house sequence images.

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

Table 2

Summary of comparison of the four matching algorithms.

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

There are rotation, scaling, and perspective distortion present. Example images from these sequences are shown in Fig. 1 and correspond to different camera viewing directions. The detected feature points and their Delaunay triangulations are overlayed on the images. The first four images are from the MOVI sequence and each contains about 140 nodes. The second four images are from the Desk sequence and each contains about 400 nodes.

In Fig. 2, we test our method on some pairs of images. In Table 1 we summarize the matching results for the MOVI houses. Here we list the number of nodes in the Delaunay graphs, the number of correct correspondence, the number of correspondence errors, and the number of points without correspondence. We also selected a pair of images which contain the same number of corner points (images 1 and 4 from MOVI sequence 140 nodes). Although the number of corners is the same, there are differences in the both identities of the detected points and their structural arrangement. We compared these images' matching results by

using our algorithm, Umeyama's algorithm, Scott and Longuet-Higgins' algorithm and Shapiro and Brady's method. The compared results are summarized in Table 2. From these results, it is clear that our new method is better than these ones.

5.2. Shape indexing

In this part we provide some experimental evaluation on our graph model for shape analysis. We use two databases for the evaluation. The first one is COIL 100 database [23], which consists of a series of 2D views of 3D objects collected at 72 equally spaced viewing directions on a great circle of the object view-sphere. For the COIL 100 database, we still extract the Delaunay graphs for each image. The second database is MPEG-7 shape silhouette database [27]. For the second database, the shock tree is a tree structure based representation of the differential structure of the boundary of a 2D shape. It is obtained by locating the shape skeleton, and examining the differential behavior of the radius of the bitangent circle from the skeleton to the object boundary, as the skeleton is traversed [16]. The shock tree extraction process has been further improved by Torsello and Hancock [30] recently. In this paper we use the methods in [30] to extract tree structure representations from the silhouettes. Example images from the data sets are shown in Figs. 3 and 4.

In the first experiment, we investigate whether the methods explored in the paper can identify finer view structure of the different objects. Here we use COIL database. In Fig. 5 we show the result of projecting the embedded node vectors for the graphs extracted from the three image sequences in the COIL database onto the eigenvectors of the embedded node position covariance matrix Σ . Since $b = \Psi^T (vec[\hat{Y}] - \hat{X})$, the projection is given by the components of the parameter vector *b*. To visualize the



Fig. 3. Sample views of the COIL 100 database objects.



Fig. 4. Sample views of the MPEG 7 database objects

distribution, we have placed a thumbnail image at the location specified by the first three components of the parameter vector *b*. The line connecting the thumbnails corresponds to the sequence order of the original images. The main feature to note is that neighboring images in the sequence are close together in the eigenspace and the generative model produce good trajectory view structures of different objects.

We have also experimented the graph statistical model as a method of clustering shapes. In this experiment, we use both the COIL and MPEG-7 databases. For the COIL database, we select 20 image from each class as training data to construct the statistical model. The rest images as testing for query. The similarity value is computed as described in Section 4. We measure the performance by using the so-called 'bullseye test' in which each image is used as a query and one counts the number of corrected images in the top 40 matches. We obtain a retrieval rate of 84.5%.

We can also use the best fit parameters b^* as feature vector for recognition. We compare the results obtained by using our algorithms with two alternative spectral methods. The first one is called Laplacian spectrum method [20] which is just use the leading eigenvalues from Laplacian matrix L of the graph as feature vector. In our experiment, we choose the first 10 leading eigenvalues to construct the feature vector for each graph. The second spectral methods has recently been introduced by Wilson et al. [33] which has reported a family of invariants that can be computed by applying symmetric polynomials to the elements from the graph spectral matrix. Here we use supervised learning methods. For each kind of object, we choose 30 graphs randomly from each class as training sets. We have used several canonical supervised learning algorithms, generative mixture models (GMM) [8], support vector machines (SVM) [32] and linear discriminant analysis (LDA) [25]. In Table 3, we give the classification accuracy with different methods i.e. ours, Laplacian spectrum [20] and symmetric polynomials [33], by using a specific learning algorithm. In this experiment, the feature vector b^* with the GMM classification methods gives the best classification rate.

In the second experiment, we use the silhouettes from MPEG-7 to extract graph representations using the method from [30]. We follow the procedure outlined in previous sections. We first embed the shock graph in co-ordinate sets by using graph embedding. Then for sets of graphs we construct the statistical

Table 3A classification performance for COIL 100 database.

	Our methods (%)	Laplacian spectrum (%)	Symmetric polynomials (%)
LDA	97	84	90
GMM	96	83	91
SVM	97	85	91



Fig. 5. Eigenprojection of graphs from 15 images in duck sequence.



Fig. 6. Shape silhouettes classification performance as a function of object numbers.

model and compute the similarity between pairs of graphs Section 4. To identify different objects from similarity matrix, we use the pairwise clustering algorithm proposed by Torsello et al. in [30] which use EM (expectation maximization) algorithm to iteratively segment the similarity matrix into disjoint subsets. In Fig. 6, we plot the proportion of shapes from the database correctly classified as the number of shape classes is increased. In this experiment, we also compare our results with the graph edit distance method computed by Torsello et al. in [30]. From Fig. 6, the classification curvature from our method is clearly higher than other methods.

6. Conclusion and future work

This paper has presented an efficient approach for shape analysis by using graph models. The approach is to first use the Isomap algorithm to embed the graph based shape representation in a Euclidean space by using the geodesic distance between graph nodes. Each node is hence transformed to a point in coordinate space. The graph nodes correspondence problems can alleviated by using point-sets alignment. With the correspondence results between graphs we continue to construct the statistical model for the graph structure. The idea underpinning the generative model is to construct a point distribution model for the embedded point position vectors. The mean and covariance matrix of the aligned embedded point position vectors are used to capture the intrinsic variation within the correspondence graphs. We demonstrated the utility of the methods for shape analysis in both COIL and shock graph databases which certainly can be used for visual information retrieval and indexing.

Our future plans involve the use of a mixture model to describe the positions of the embedded nodes, and to assess uncertainty in the computation of correspondence. We also plan to apply the algorithm to the graphs which contain high-level vision information from the images, i.e. Gestalt graphs, relational graphs. We also plans to expand our work in video based analysis.

Acknowledgement

This paper is supported by 973 Program (Project No. 2010CB327902). Prof. Edwin Hancock was supported by the EU

FET project SIMBAD (213250) and a Royal Society Wolfson Research Merit Award.

References

- M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Neural Information Processing Systems 14 (2002) 634–640.
- [2] H. Bunke, Error correcting graph matching: on the influence of the underlying cost function, IEEE Transaction on Pattern Analysis and Machine Intelligence 21 (9) (1999) 917–922.
- [3] M. Carcassoni, E.R. Hancock, Correspondence matching with modal clusters, IEEE Transaction on Pattern Analysis and Machine Intelligence 25 (12) (2003) 1609–1615.
- [4] T.F. Cootes, G.J. Edwards, C.J. Taylor, Active appearance models, European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 1407, Springer-Verlag, Freiburg, Germany, 1998, p. 484.
- [5] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models—their training and application, Computer Vision and Image Understanding 61 (1) (1995) 38–59.
- [6] T. Cox, M. Cox, Multidimensional Scaling, Chapman and Hall, 1994.
- [7] E.W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959) 269–271.
- [8] M.A.F. Figueiredo, A.K. Jain, Unsupervised learning of finite mixture models, IEEE PAMI 24 (3) (2002) 381–396.
- [9] K. Fujisawa, M. Kojima, K. Nakata, Exploiting sparsity in primal-dual interiorpoint methods for semidefinite programming, Mathematical Programming 79 (1997) 235–253.
- [10] G.L. Scott, H.C. Longuett-Higgins, An algorithm for associating the features of two images, Proceedings of the Royal Society of London B 244 (1991) 21–26.
- [11] C. Harris, M. Stephens, A combined corner and edge detector, in: Proceedings of the 4th Alvey Vision Conference, Manchester, UK, 1998, pp. 189–192.
- [12] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, 1986.
- [13] J.B. Tenenbaum, V.D. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 586–591.
- [14] K. Fujisawa, Y. Futakata, M. Kojima, K. Nakata, M. Yamashita, Sdpa-m user's manual < http://sdpa.is.titech.ac.jp/SDPA-M >, 2002.
- [15] X. Klassen, A. Srivastava, W. Mio, S.H. Joshi, Analysis of planar shapes using geodesic paths on shape spaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 372–383.
- [16] K. Siddiqi, A. Shokoufandeh, S. Dickinson, S. Zucker, Shock graphs and shape matching, International Journal of Computer Vision 30 (1999) 1–24.
- [17] C.G. Lee, C.G. Small, Multidimensional scaling of simplex shapes, Pattern Recognition 32 (1999) 1601–1613.
- [18] L.S. Shapiro, J.M. Brady, Feature-based correspondence: an eigenvector approach, Image and Vision Computing 10 (1992) 283–288.
- [19] B. Luo, E.R. Hancock, Structural graph matching using the em algorithm and singular value decomposition, IEEE Transaction on Pattern Analysis and Machine Intelligence 23 (10) (2001) 1120–1136.
- [20] B. Luo, R. Wilson, E. Hancock, Eigenspaces for graphs, International Journal for Image Graphics 2 (2002) 247–268.
- [21] M. Todd, Semidefinite optimization, Acta Numerica 10 (2001) 515-560.

- [22] H. Murase, S.K. Nayar, Illumination planning for object recognition using parametric eigenspaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 1219–1227.
- [23] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (coil 100), Columbia University, 1996.
- [24] N. Linial, E. London, Y. Rabinovichm, The geometry of graphs and some its algorithmic application, Combinatorica 15 (1995) 215–245.
- [25] R.O. Duda, P.E. Hart, D.H. Stork, Pattern Classification, Wiley Interscience, 2000.
- [26] K. Siddiqi, A. Shokoufandeh, S.J. Dickinson, S.W. Zucker, Shock graphs and shape matching, International Journal of Computer Vision 35 (1) (1999) 13–32.
- [27] S. Jeannin, M. Bober, Description of core experiments for mpeg-7 motion/ shape, Technical Report ISO/IEC JTC 1/SC 29/WG 11 MPEG99, 1999.
- [28] S. Umeyamam, An eigen decomposition approach to weighted graph matching problems, IEEE PAMI 10 (1988) 695-703.
- [29] T. Cox, M. Cox, Multidimensional Scaling, Chapman and Hall, 1994.
- [30] A. Torsello, A. Robles-Kelly, E.R. Hancock, Discovering shape classes using tree edit-distance and pairwise clustering, International Journal of Computer Vision 72 (3) (2007) 259–285.
- [31] L. Vandenberghe, S. Boyd, Semidefinite programming, SIAM Review 38 (1996) 49–95.
- [32] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
- [33] R.C. Wilson, B. Luo, E.R. Hancock, Pattern vectors from algebraic graph theory, TPAMI 27 (2003) 2220–2237.
- [34] H. Wolkowicz, R. Saigal, L. Vandenberghe, Handbook of Semidefinite Programming, Kluwer Academic Publishers, 2000.
- [35] B. Xiao, E.R. Hancock, A spectral generative model for graph structure, in: SSPR/SPR, 2006, pp. 173–181.



Bai Xiao received the B.Eng. degree in Computer Science from Beihang University of China in 2001. From 2002 to 2006, he was a Ph.D. student at Computer Science Department, University of York, UK under the supervision of Professor Edwin R. Hancock. From September 2006 to December 2008, he was a Research Officer (Fellow, Scientist) at Computer Science Department, University of Bath. He is now an Associate Professor at School of Computer Science and Engineering, Beihang University (BUAA). He has published around 30 papers in journals and referred conferences. His current research interests include graph spectral analysis, image segmentation and

recognition, statistical pattern recognition, and feature extraction.



Edwin Hancock studied physics as an undergraduate at the University of Durham and graduated with honors in 1977. He remained at Durham to complete the Ph.D. degree in the area of high-energy physics in 1981. Following this, he worked for 10 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). 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 optimization 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 databases, and statistical learning theory. He has published more than 90 journal papers and 350 refereed conference publications. He was awarded the Pattern Recognition Society medal in 1991 and an outstanding paper award in 1997 by the journal Pattern Recognition. In 1998, he became a fellow of the International Association for Pattern Recognition. He has also been a guest editor for special editions of the journals Image and Vision Computing and Pattern Recognition.



Hang Yu received the B.Eng. degree in Computer Science from Zhe Jiang University of China in 2003. He is currently a Master student at University of Florida.