Heat Kernel Analysis On Graphs

Xiao Bai

Submitted for the degree of Doctor of Philosophy

Department of Computer Science

THE UNIVERSITY of York

June 7, 2007

Abstract

In this thesis we aim to develop a framework for graph characterization by combining the methods from spectral graph theory and manifold learning theory. The algorithms are applied to graph clustering, graph matching and object recognition.

Spectral graph theory has been widely applied in areas such as image recognition, image segmentation, motion tracking, image matching and etc. The heat kernel is an important component of spectral graph theory since it can be viewed as describing the flow of information across the edges of the graph with time. Our first contribution is to investigate how to extract useful and stable invariants from the graph heat kernel as a means of clustering graphs. The best set of invariants are the heat kernel trace, the zeta function and its derivative at the origin. We also study heat content invariants. The polynomial co-efficients can be computed from the Laplacian eigensystem. Graph clustering is performed by applying principal components analysis to vectors constructed from the invariants or simply based on the unitary features extracted from the graph heat kernel. We experiment with the algorithms on the COIL and Oxford-Caltech databases.

We further investigate the heat kernel as a means of graph embedding. The second contribution of the thesis is the introduction of two graph embedding methods. The first of these uses the Euclidean distance between graph nodes. To do this we equate the spectral and parametric forms of the heat kernel to compute an approximate Euclidean distance between nodes. We use the resulting pattern of distances to embed the nodes of the graph on a manifold using a procedure similar to ISOMAP. The distribution of embedded points can be used to characterize the graphs, and can be used for the purpose of graph clustering as well. Experiments demonstrate that the algorithms can offer a useful margin of advantages over existing alternatives.

The second graph embedding method uses the Young-Householder decomposition of the heat kernel to map the nodes of the graphs into a vector space. This is similar to performing kernel PCA on the heat kernel. The co-ordinates of the nodes are determined by the eigenvalues and eigenvectors of the Laplacian matrix, together with a time parameter which can be used to scale the mapping. Node correspondences are located by applying a spectral alignment algorithm to the embedded nodes. Here the third contribution of the thesis is to use the heat kernel graph embedding to transform the graph matching problem into one of point-set alignment problem.

The fourth contribution of the thesis is to use the correspondence matches to construct a generative model which can be used to capture variations in graph structure using the covariance matrix for corresponding embedded point positions. This is done by using the eigenvalues and eigenvectors of the covariance matrix for the embedded node positions of sample of graphs. We show how to use this model to both project individual graph into the eigenspace of the point position covariance matrix and how to fit the model to potentially noisy graphs to reconstruct the Laplacian matrix. We illustrate the utility of the resulting method for shape analysis using data from the COIL and Caltech-Oxford databases.

Contents

List of Figures

List of Tables

Glossary of Symbols

$G = (V, E)$	Graph
V	Sets of nodes
E	Sets of edges
A	Adjacency matrix
D	Degree matrix
I	Identity matrix
L	Un-normalized Laplacian matrix
$\hat{L}$	Normalized Laplacian matrix
Λ	Eigenvalue matrix of normalized Laplacian matrix $\hat{L}$
Φ	Eigenvector matrix of normalized Laplacian matrix $\hat{L}$
λ	Eigenvalue of normalized Laplacian matrix $\hat{L}$
ϕ	Eigenvector of normalized Laplacian matrix $\hat{L}$
h_t	Heat kernel
t	Time
T_r	Matrix trace
ζ	Zeta function
q_m	Heat content co-efficient
S_r	Elementary symmetric polynomial
P_r	Power-sum symmetric polynomial
R_I	Rand index value
$\vec{B}$	Feature vector
d_G	Geodesic distance
d_E	Euclidean distance
k_s	Sectional curvature
μ_{pq}	Raw moment
W	Weight matrix
$\hat{X}$	Mean position
Σ	Covariance matrix
C_{IJ}	Correspondence matrix

Acknowledgements

I would like to express my sincere appreciation and gratitude to my supervisor, Prof. Edwin Hancock, for his support and advice on my research during more than three years of postgraduate work at York. I also thank my assessor, Dr. Richard Wilson for his impartial assessment and constructive comments on my work. Without their insightful guidance, it would not have been possible for me to complete this thesis. My sincere thanks also go to the good friends I made in York, with whom I shared good times and memories together. Finally, I would like to dedicate this thesis to my family. Their help and support were invaluable for the successful completion of my PhD.

Declaration

I declare that the work in this thesis is solely my own except where attributed and cited to another author. Most of the material in this thesis has been previously published by the author. For a complete list of publications, please refer to the next page.

List of Publications

The following is a list of publications that has been produced during the course of my research.

2006

- Xiao Bai, Edwin Hancock ”A Spectral Generative Model for Graph Structure” Syntactical and Structural Pattern Recognition(SSPR), Portugal, pages 173-181 , 2006
- Xiao Bai, Edwin Hancock ”Trace Formula Analysis of Graphs” Syntactical and Structural Pattern Recognition(SSPR), Portugal, pages 306-313 , 2006

2005

- Xiao Bai, Edwin Hancock ”Recent Results on Heat Kernel Embedding of Graphs” 5th Graph Based Representation on Pattern Recognition, France, pages 373-382 , 2005
- Xiao Bai, Edwin Hancock ”Graph Clustering using Heat Content Invariants” 2nd Iberian Conference on Pattern Recognition and Image Analysis, Portugal, pages 123-130, 2005
- Xiao Bai, Edwin Hancock ”Clustering Shapes using Heat Content Invariants” The International Conference on Image Processing(ICIP), 2005

- Xiao Bai, Edwin Hancock ”Geometric Characterization of Graphs” 13th International Conference on Image Analysis and Processing (ICIAP 2005), Italy, pages 471-478, 2005

- Xiao Bai, Richard Wilson, Edwin Hancock ”Manifold Embedding of Graphs Using the Heat Kernel” Mathematics of Surfaces 2005, pages 34-49, LNCS 3604

- Xiao Bai, Edwin Hancock ”Characterizing Graphs using the Heat Kernel” British Machine Vision Conference, U.K., 315-324, 2005

2004

- Xiao Bai, Hang Yu and Edwin Hancock ”Graph Matching using Spectral Embedding and Alignment” International Conference on Pattern Recognition, Cambridge, pages 398-401, 2004

- Xiao Bai, Edwin Hancock ”Heat Kernels, Manifolds and Graph Embedding” Syntactical and Structural Pattern Recognition(SSPR), Portugal, pages 198-206, 2004

- Xiao Bai, Hang Yu and Edwin Hancock ”Graph Matching using Embedding and Semidefinite Programming” British Machine Vision Conference, Kingston, 2004

- Xiao Bai, Hang Yu and Edwin Hancock ”Graph Matching using Manifold Embedding” International Conference on Image Analysis and Recognition, Portugal, pages 352-359, 2004

- Xiao Bai ”Manifold Learning and Spectral Graph Theory” Ph.D. Thesis Proposal, Department of Computer Science, University of York, 2004

2003

- Richard Wilson, Xiao Bai and Edwin Hancock ”Graph Clustering using Symmetric Polynomials and Locally Linear Embedding” British Machine Vision Conference, East Anglia, 289-298, 2003

- Xiao Bai ”Spectral Graph Theory for Graph Clustering” Ph.D. first year report, Department of Computer Science, University of York, 2003

Submitted Papers

- Xiao Bai, Richard Wilson, Edwin Hancock ”Characterizing Graphs using the Heat Kernel” submitted to Jounral of Image and Vision Computing

- Xiao Bai, Edwin Hancock ”Isotree: Tree Clustering via Metric Embedding” submitted to Journal of Neural Computing

- There are also several journal and conference papers under preparation.

Chapter 1

Introduction

1.1 The Problem

Spectral graph theory (Chung, 1997; Biggs, 1993; Cvetkovic et al., 1995) has been widely applied to solve the problems in the field of computer vision and pattern recognition. Examples include image segmentation(Shi and Malik, 1997), routing (Atkins et al., 1998), image classification (Wilson et al., 2003) and etc. These methods use the spectrum, i.e. eigenvalues and eigenvectors, of the adjacency or Laplacian matrix corresponding to the graph. Several authors have also explored the use of the Laplacian and related operators to map data to a manifold in a low dimensional space (Roweis and Saul, 2000; Belkin and Niyogi, 2000; Tenenbaum et al., 2000).

The Laplacian spectrum is closely related to the heat kernel of a graph. The heat kernel is the solution of the heat equation and is formed by exponentiating the Laplacian eigensystem over time. The heat kernel can be viewed as describing the flow of information across the edges of the graph with time. The thesis addresses the issue of how to extract useful information from the graph heat kernel to characterize the graph. It is strongly linked to the manifold learning theory. We will also explore how we can map the nodes of a graph to points in a vector space. This is achieved by the analysis of the heat kernel. Finally we will combine the graph embedding methods with a statistical generative model to construct a linear deformable model which can be used to capture variations within graph structures.

1.2 Goals

The ultimate goal of this thesis is to use ideas derived from manifold learning theory to investigate the information contained in the graph heat kernel, and to explore whether they can provide a stable and robust way to characterize the graph. To this end, we focus on

- Robust graph feature extraction: A graph can be represented by the heat kernel matrix associated with it. Graph information resides in the graph heat kernel. We explore how to extract stable and robust invariants, which can be used for characterizing the graphs.

- Graph embedding: Here we explore how to embed the nodes of graphs as points to a vector space. In other words, we seek a mapping from the node-set of the graph to point-sets in a vector space. After the embedding we can apply the algorithms in the vector-space on graphs.

- Graph embedding for graph clustering: Once we have embedded the nodes of a graph in a vector space, we aim to solve the graph clustering problem by applying a simple point pattern analysis algorithm to the embedded point-set.

- Heat kernel mapping for graph matching: 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 embed the nodes of a graph in a high dimensional coordinate space, and to use point pattern matching techniques to locate correspondences between nodes.

- A generative model for graph structure: We aim to construct a statistical model that can account for the distribution of mapped point positions for corresponding nodes in a sample graph. We can capture variations in graph structures using the sample covariance matrix and mean point position for the corresponding embedded point positions.

1.3 Thesis Overview

Having described the overall goals of the thesis, we proceed to give a brief introduction to the structure of the thesis. Chapter 2 reviews the literature of background on graph embedding, manifold learning methods, and spectral graph theory in computer vision and pattern recognition.

In chapter 3, we illustrate how invariants from the graph heat kernel can be used for graph characterization and used to perform graph clustering. In this chapter we will explore several different invariants extracted from the heat kernel and compare them on the real world image databases.

In chapter 4, two ways of graph embedding are explored. We use the embedding methods to map the graph from the graph space to point-set in a vector space. Once embedded in the vector space, we can perform graph clustering by characterizing the embedded point-sets.

In chapter 5, we combine the methods from spectral graph theory and the heat kernel mapping to develop method for structure based graph matching. We also show how the node correspondences provided by the matching method can be used to construct a statistical model. This model accounts for the distribution of embedded point positions for corresponding nodes in the graph.

In the final chapter, after describing the contribution of the thesis, we will give a brief conclusion of the advantages and shortcomings of the methods described in the thesis and point out some promising directions for future research.

Chapter 2

Literature Review

In this chapter, we aim to review literature relevant to the thesis. The review ranges from spectral graph theory to manifold learning theory and their related applications in computer vision. The organization of this chapter is as follows.

In Section 2.1, we commence by presenting a review of manifold learning methods. Since our aim in the thesis is to develop methods for graph characterization by combining spectral graph theory and manifold learning theory, we then review the spectral graph theory and its applications in computer vision in Section 2.2. In Section 2.3, the research on graph heat kernel and the previous work are reviewed. We survey graph clustering in Section 2.4. Finally in Section 2.5, we review statistical models for graphs to motivate the methods presented in Chapter 5.

2.1 Graph Embedding and Manifold Learning

One of the problems that arises in the manipulation of large amounts of graph data is how to embed graphs in low-dimensional space so that standard machine learning techniques can be used to perform tasks such as clustering(Luo et al., 2003; Wilson et al., 2003). One way to realize this goal is to embed the nodes of a graph on a manifold and to use the geometry of the manifold as a means of graph characterization(Hjaltason and Samet, 2003).

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 the measure of distortion (Linial et al., 1995a). Broadly speaking there are three ways in which the problem can be addressed. First, the graph can be interpolated by a surface whose genus is determined by the number of nodes, edges and faces of the graph. Second, the graph can be interpolated by a hyperbolic surface which has the same pattern of geodesic (internode) distances as the graph (Busemann, 1955). Third, a manifold can be constructed whose triangulation is the simplicial complex of the graph (Ranicki, 1992). A review of methods for efficiently computing distance via embedding is presented in the recent paper of Hjaltason and Samet (Hjaltason and Samet, 2003).

In the pattern analysis community, there has recently been renewed interest in the use of embedding methods motivated by graph theory. One of the best known of these is ISOMAP (Tenenbaum et al., 2000). Here a neighborhood ball is used to convert data points into a graph, and Dijkstra's algorithm is used to compute the shortest–geodesic distances between nodes. The matrix of geodesic distances is used as input to MDS–Multidimensional Scaling(Cox and Cox, 1994). The resulting algorithm has been demonstrated to locate well-formed manifolds for a number of complex data sets. Related algorithms include locally linear embedding (Roweis and Saul, 2000), which is a variant of PCA (Jollife, 1986) that restricts the complexity of the input data using a nearest neighbor graph, and the Laplacian eigenmap (Belkin and Niyogi, 2000) that constructs an adjacency weight matrix for the data points and projects the data onto the principal eigenvectors of the associated Laplacian matrix(the degree matrix minus the weight matrix) (Belkin and Niyogi, 2000). Recently Lafon and Coifman have proposed the diffusion map (Coifman and Lafon, 2004), which constructs the mapping by raising the Laplacian eigensystem to a negative integer power. This mapping is shown to preserve the distances between nodes under a random walk, or diffusion, on the graph. These methods (Roweis and Saul, 2000; Belkin and Niyogi, 2000; Tenenbaum et al., 2000; Coifman and Lafon, 2004) share the feature of using the spectrum of the Laplacian matrix to map data specified in terms of a proximity matrix to a vector space and are together known as manifold learning theory. Their collective aim is to develop variants of the classical methods of PCA and MDS, which can be used to better capture localized variations in the structure of the data.

However, the study of graph structures using manifold learning theory is less advanced than the study of pattern spaces for images (Murase and Nayar, 1994) or shapes (Klassen et al., 2004; Lee and Small, 1999; Cootes et al., 1995). Here a well established route to construct a pattern space for the data is to use principal components analysis. This commences by encoding the image data or shape landmarks as a fixed length long vector. The data is 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 has lead to the development of more sophisticated analysis methods capable of dealing with quite complex pattern spaces. One thing that may hinder the study of graph structure using manifold learning theory is that graphs are not vectorial in natural and we are unable to use node ordering to transform the graphs into vectors. The reason is twofold. First the nodes of a graph are not ordered or labeled, and so there is no natural way to map the nodes to the components of a vector. Secondly, even if a nodes ordering can be found, different graphs may contain different numbers of nodes. We still need to deal with graph pattern vectors of different lengths.

In an attempt to overcome problems that result from the non-vectorial nature of graphs, Luo, Wilson and Hancock (Luo et al., 2003) have explored how ideas from spectral graph theory (Chung, 1997; Sachs et al., 1980; Biggs, 1993) can be used to construct pattern spaces for sets of graphs. The idea here has been to extract features that are permutation invariants from the adjacency matrices of the graphs under study. Pattern spaces may then be constructed from the feature vectors using techniques such as principal components analysis, or the more recently developed ones from manifold learning theory described above. Related literature includes Shokoufandeh et al. (Shokoufandeh et al., 1999) who have used topological spectra to index tree structures, and recently Wilson and Hancock (Wilson et al., 2003) have used algebraic graph theory to construct permutation invariant polynomials from the eigenvectors of the Laplacian matrix. One way of viewing these methods is to construct a low-dimensional feature space that captures the topological structure of the graphs under study.

2.2 Spectral Graph Theory in Computer Vision and Pattern Recognition

Spectral graph theory (Chung, 1997; Sachs et al., 1980; Biggs, 1993) is a branch of mathematics that is concerned with characterizing the structural properties of graphs using the eigenvectors of the adjacency matrix or the closely related Laplacian matrix (the degree matrix minus the adjacency matrix). There are many applications of using spectral graph theory in the physics (Bell and Rowlinson, 1990) and chemistry (Dias, 1993) areas. The use of spectral graph theory in computer vision and pattern recognition is a recent development. In this subsection we will review some problems in computer vision and pattern recognition that have been solved using spectral graph theory.

Pattern matching is a task of pivotal important in high-level vision since it provides a means by which abstract pictorial descriptions can be matched to one another. In recent years there have been many attempts to use spectral graph theory both in graph matching and in point-set matching problems. One of the earliest attempt was done by Umeyama (Umeyama, 1988), who developed a singular value decomposition method to find the permutation matrix between the adjacency matrices of the two graphs to be matched. The method commences by performing singular value decomposition on the adjacency matrices of the two graphs separately. The permutation matrix is found by taking the outer products of the eigenvector matrices for the adjacency matrices of the graphs being matched. The method can cope with both weighted and unweighted graphs, but it can not handle graphs which have a different number of nodes. For the point-sets matching, Scott and Longuett-Higgins (Scott and Longuett-Higgins, 1991) have shown how to recover correspondence by maximizing the inner product of the pairing matrix and the proximity matrix of the two point-sets. Shapiro and Brady (Shapiro and Brady, 1992) have extended Scott and Longuett-Higgins's idea by computing the eigenvectors of the proximity matrices of the two point-sets being matched. By doing so they overcome the shortcoming of Scott and Longuett-Higgins's method, which fails to find the correct correspondence when the rotation angle between the point-sets becomes large. Both Scott and Longuett-Higgins', Shapiro and Brady's methods can only match point-sets and they can not be applied directly to graph matching problems. However, there have been many attempts to overcome these limitations. Luo and Hancock (Luo and Hancock, 2001) have improved Umeyama's method by incorporating the EM algorithm. This allows Umeyama's method to render robustness to the differences in graph size and structural errors. However, the resulting algorithm is time consuming because of its iterative character. Recently, Robles-Kelly and Hancock

(Robles-Kelly and Hancock, 2002) have proposed a new spectral graph matching method by aligning the leading eigenvectors of the adjacency matrices of two graphs, where the leading eigenvector corresponds to the steady-state Markov chain. Based on Shapiro and Brady's point-set matching algorithm, Carcassoni and Hancock (Carcassoni and Hancock, 2003) have shown that by using the EM algorithm, which can incorporate the structure of the point-sets, the confidence of point correspondence can be computed by probabilities using the proximity matrix. Kosinov and Caelli (Kosinov and Caelli, 2002) have improved Shapiro and Brady's method by allowing for scaling in the eigenspace.

Another application for spectral graph theory in computer vision and pattern recognition is for image segmentation. Several authors (Shi and Malik, 1997; Costeira and Kanade, 1995; Perona and Freeman, 1998; Sarkar and Boyer, 1996; Scott and Longuett-Higgins, 1990) have explored how to use the eigenvectors of the affinity matrix to solve the image segmentation problems. Their approaches are attractive because their methods are based on simple eigendecomposition algorithms whose stability is well understood. Scott and Longuett-Higgins (Scott and Longuett-Higgins, 1990) identify point clusters by relocating the normalized eigenvectors of the affinity matrix. The input for this algorithm is an affinity matrix and the same size output matrix can be used to reshuffle the points into different clusters. Costeira and Kanade (Costeira and Kanade, 1995) have applied a variant of the method to 3D motion tracking by using the singular values of the measurement matrix to cluster points. Sarkar and Boyer (Sarkar and Boyer, 1996) used the leading eigenvector of the affinity matrix to locate point clusters and have applied the method to line segments grouping. The algorithm developed by Shi and Malik (Shi and Malik, 1997) uses the second eigenvector (Fiedler vector) of the Laplacian matrix(i.e. the degree matrix minus the affinity matrix) as a solution to a continuous formulation of a minimum normalized cut problem. In this way they locate a segmentation that minimizes the affinity between groups. Perona and Freeman (Perona and Freeman, 1998) have a similar segmentation algorithm based on thresholding the first eigenvector of the affinity matrix. Weiss (Weiss, 1999) gives a review on these spectral based segmentation algorithms. He showed the similarities and differences between the algorithms. The similarity is that they all use the eigenvectors of an affinity matrix. The differences lie in which eigenvectors are used and how to construct the affinity matrix.

There are many other applications that make use of spectral graph theory in computer vision and pattern recognition. One of the most important of these is graph clustering, which we will review in detail in Section 2.4. Another interesting application is routing, which can be used to simplify the graph structure. In (Atkins et al., 1998) Atkins, Bowman and Hendrikson have shown how to use the Fiedler vector of the Laplaican matrix to sequence relational data. Robles-Kelly and Hancock (Robles-Kelly and Hancock, 2005) have shown how to use the first eigenvector to transform the graphs into strings and have used the strings to match graphs and compute string edit distance. Yu and Hancock (Yu and Hancock, 2005) have further extended the method by using semidefinite programming to find the spectral string ordering.

2.3 Heat Kernel and Spectral Geometry

One of the most important properties of the Laplacian spectrum is its close relationship with the heat equation. Recent developments in this area have established a link between graph spectra and the geometry of the manifold on which the embedded graph resides (Grigor'yan, 2001; Grigor'yan, 2003; Lafferty and Lebanon, 2004; Coulhon et al., 2000; Smola and Kondor, 2004; Barlow, 1998;

Smola et al., 1998). According to the heat equation the time derivative of the heat kernel is determined by the graph Laplacian. The solution to the heat equation is obtained by exponentiating the Laplacian eigensystem over time. The heat kernel hence encapsulates the way in which information flows through the edges of the graph over time, and is closely related to the path length distribution on the graph. Moreover the heat kernel also encapsulates the distribution of geodesic distances at the embedded manifold (Yau and Schoen, 1988). As a result the graph can be viewed as residing on a manifold whose pattern of geodesic distances is characterized by the heat kernel. For short times the heat kernel is determined by the local connectivity or topology of the graph as captured by the Laplacian, while for long times the solution gauges the global geometry of the manifold on which the graph resides.

One way to use the heat kernel is to extract invariants that can be used to characterize the corresponding graph or the manifold. Asymptotically for a short time, the trace of the heat kernel (Chung, 1997; Gilkey, 1984)(or the sum of the Laplacian eigenvalues exponentiated with time) can be expanded as a rational polynomial in time. Here the spectrum of the Laplace-Beltrami operator is used to construct a trace function, and the co-efficients of the leading terms in the series are directly related to the geometry of the manifold. In spectral geometry, the heat kernel trace has also been used to characterize the differential geometry of manifolds (Yau and Schoen, 1988; Gilkey, 1984). This function can be expanded as a polynomial series in time, and the co-efficients of the series can be related to the Ricci curvature tensor of the manifold. For instance, the leading co-efficient is the volume of the manifold, the second co-efficient is related to the Euler characteristic, and the third co-efficient is related to the Ricci curvature. However, the relationships between the elements of the Ricci curvature tensor and the co-efficients are difficult to determine, and are only tabulated up to the third order

(Gilkey, 1984). For large graphs, the Laplacian can be viewed as a discrete approximation of the Laplace-Beltrami operator and this analysis can be carried over from manifolds to graphs (Hein et al., 2005). The zeta function(i.e. the sum of the eigenvalues raised to a non-integer power) for the Laplacian also contains geometric information. For instance its derivative at the origin is related to the torsion tensor for the manifold. Colin de Verdiere has shown how to compute geodesic invariants from the Laplacian spectrum (de Verdiere, 1998). In a recent paper McDonald and Meyers (McDonald and Meyers, 2002) have shown that the heat content of the heat kernel is a permutation invariant. The heat content is the sum of the entries of the heat kernel over the nodes of the graph, which may be expanded as a polynomial in time. It is closely related to the trace of the heat kernel, which is also known to be an invariant. This field of study is sometimes referred to as spectral geometry (Gilkey, 1984; Rosenberg, 2002).

As noted above, for a Riemannian manifold, the heat kernel is determined by the pattern of geodesic distances, and can provide a means of analyzing both the local and global differential geometry of the manifold. Spectral geometry also has close links with K-theory and Morse theory. It has topical interest in particle physics, since Witten (Witten et al., 1988) has demonstrated how such theories can be used to understand the geometries of space time that underpin superstrings. Recently, in the pattern analysis area Lebanon and Lafferty (Lafferty and Lebanon, 2004) have used the heat kernel to construct statistical manifolds that can be used for inference and learning tasks.

2.4 Graph Clustering

Graph structures have been proved important in high level-vision since they can be used to represent structural and relational arrangements of objects in a scene.

Structural abstractions of 2D and 3D objects have been proved to be powerful tools for the recognition and learning of shape classes (Shokoufandeh et al., 1999). One of the problems that arises in the analysis of structural abstractions of shape is graph clustering (Bunke and Shearer, 1998). Traditionally, there are two approaches to the problem. The first of these is to maintain a class prototype, and to cluster by iteratively merging graphs together (Lozano and Escolano, 2003; Bunke et al., 2003; Jain and Wysotzki, 2004). The second approach, which avoids the need to maintain a class prototype, is to apply pairwise clustering methods to the edit distance between graphs (Sanfeliu and Fu, 1983; Tsai and Fu, 1983; Bunke, 1997; Bunke, 1999; Torsello and Hancock, 2001). Unfortunately, both of these methods involve computing correspondence between nodes, and since this is potentially an NP-hard problem, the computational overheads can be large.

Turning our attention to the first approach, there has been some research aimed at applying central clustering techniques to cluster graphs. However, rather than characterizing them in a statistical manner, a structural characterization is adopted. For instance, both Lozano and Escolano (Lozano and Escolano, 2003), and Bunke et al. (Bunke et al., 2003) summarize the data using a supergraph. Each sample can be obtained from the supergraph using edit operations. However, the way in which the supergraph is learned or estimated is not statistical in nature. Munger, Bunke and Jiang(Bunke et al., 1999) have recently developed a genetic algorithm for searching for the median graph. Jain and Wysotzki adopt a geometric approach which aims to embed graphs in a high dimensional space by means of the Schur-Hadamard inner product (Jain and Wysotzki, 2004). Central clustering methods are then deployed to learn the class structure of the graphs.

As far as the second approach is concerned, many authors have employed the concept of graph edit distance to measure the structural similarity of graphs. The idea here is to perform elementary editing operations on a graph, such as edge or node insertion and deletion, to make pairs of graphs isomorphic. Each operation has an associated ”cost”, and the minimum total cost of the set of edit operations can be used to gauge the similarity of the graphs. For example, Fu et al (Sanfeliu and Fu, 1983; Tsai and Fu, 1983) have computed similarities using separate edit costs for relabeling, insertion and deletion on both nodes and edges. A search is necessary to locate the set of operations which have minimal cost. More recently, Bunke (Bunke, 1997; Bunke, 1999) has established a relationship between the minimum graph edit distance and the size of the maximum common subgraph. Torsello and Hancock (Torsello and Hancock, 2001) have exploited this relationship to cast the problem into a continuous optimization framework. Shapiro and Haralick (Shapiro and Haralick, 1981) have exploited a similarity measure based on the number of consistent structural relationships in pairs of graphs. Again a search is exploited to locate the best correspondence mapping between the nodes.

There are two main conclusions to draw from this literature on graph clustering. Firstly, the calculation of graph similarity requires the solution of the correspondence problem as a prerequisite. As mentioned before this problem is typically NP-hard and therefore computationally expensive. Secondly, there is no clear consensus on how to compute the similarity of graphs.

An alternative way to capture graph structure has been to use the spectrum of the Laplacian matrix (Chung, 1997; Sachs et al., 1980; Biggs, 1993). Many authors have used the spectral graph theory to characterize the similarity between graphs for pairwise clustering and extract feature vectors from graphs to which centred clustering techniques can be applied. Horaud and Sossa (Horaud and Sossa, 1995) have applied the spectral graph theory to image database indexing by comparing the co-efficients of the polynomials of the Laplacian matrix of the weighted graph extracted from the image. This representation was used for indexing a large database of line drawings. In related work, Sengupta and Boyer (Sengupta and Boyer, 1998) have proposed a spectral method for line-pattern graph database partitioning. Shokoufandeh, Dickinson and Siddiqi (Shokoufandeh et al., 1999) have used topological spectra to index tree structures. Luo, Wilson and Hancock (Luo et al., 2003) have used the spectrum of the adjacency matrix to construct feature vectors for graph characterization, and Wilson and Hancock (Wilson et al., 2003) have extended this earlier work by using the algebraic graph theory to construct permutation invariant polynomials from the eigenvectors of the Laplacian matrix. One way of viewing these methods is to construct a low-dimensional feature space that captures the topological structure of the graphs under study.

2.5 Statistical Model for Graphs

In this section we will describe a number of attempts aimed at developing probabilistic models for variations in graph structure. Some of the earliest work was that of Wong, Constant and You (Wong et al., 1990), who capture the variation in graph structure using a low order probability distribution. By introducing an entropy measure and using an appropriate threshold, clusters of attributed graphs can be synthesized. The probability distributions of the random graphs can be used to characterize the statistical variations of the contextual and structural characteristics of each cluster. Bagdanov and Worring (Bagdanov and Worring, 2003) have overcome some of the computational difficulties associated with this method by using continuous Gaussian distributions to model the densities of the elements in a random graph. Since the pattern graph plays an important role in structure pattern recognition (Pavlidis, 1977; Miclet, 1986), there are potentially many applications for statistical models of graph structure. Examples include character recognition (Kim and Kim, 2001; Wong et al., 1990), occluded face recognition (Alquezar et al., 1998), and recently document classification (Cesarini et al., 1999; Doermann et al., 1997).

For point patterns, Cootes and Graham (Cootes et al., 1995) introduced point distribution models, which can be used to capture the variation within a set of shapes. Each shape is described by a long vector that contains the coordinates of the landmarks. A sample of shapes represented in this way is characterized using the mean and the covariance for the long vectors. The modes of the variation for this point distribution model are computed by performing PCA on the sample covariance matrix. However, this method can only capture the variation within point-sets extracted from the images. It can not be directly applied to capture the variations within graphs.

When it comes to problems of graph matching Hancock and Kittler (Hancock and Kittler, 1990) have developed a probabilistic relaxation method for image labeling using a Bayesian framework to construct a support function for different object arrangements. Christmas, Kittler and Petrou(Christmas et al., 1995) extended this work by making it more applicable to general attributed relational graph matching problems. Wilson and Hancock (Wilson and Hancock, 1997) have developed a Bayesian framework for discrete relaxation while modeling the probabilistic distribution for matching errors using an exponential function of Hamming distance. These methods can be used to transform the graph matching problem into the statistical optimization problems. Optimization may then be achieved using genetic algorithm, iterative search or mean field annealing. There is also a considerable body of related literature in the graphical models community concerned with learning the structure of Bayesian networks from data (Heckerman et al., 1995).

2.6 Conclusion

Based on the review of the related literature, we may draw several conclusions. First, although the heat kernel is an important component of spectral graph theory, it has not been thoroughly investigated or applied for the purposes of solving problems from computer vision and pattern recognition. In this thesis, we will investigate the feasibility of extracting useful information from the graph heat kernel as a means of graph characterization. We will demonstrate that the resulting methods outperform traditional spectral methods for graph clustering.

Second, currently manifold learning theory (Roweis and Saul, 2000; Tenenbaum et al., 2000; Belkin and Niyogi, 2000)etc. is restricted to dimensional data reduction. Using a distance function deduced from the heat kernel analysis, ISOMAP (Tenenbaum et al., 2000) can be used for the purpose of graph embedding. We map the nodes of a graph to point-sets in a vector space. The distribution of the embedded node position can be used for graph characterization.

The third point from the literature review is that graph embedding can be combined with a point-set distribution model (Cootes et al., 1995) to construct a generative model for graph structure. Here we introduce the heat kernel embedding. The nodes of the graph are mapped to high dimensional vectors in a new manifold. We solve the node correspondence between two graphs by performing alignment on the node position vectors in the mapped space. For a set of graphs, the heat kernel mapping and alignment are performed to compute the sample mean and covariance matrix for the embedded node vector position. Variations within the graph sample are computed from the eigendecomposition on the covariance matrix.

Chapter 3

Heat Kernel Invariants

3.1 Introduction

Graph clustering is an important issue in computer vision and pattern recognition, since graphs can be used for the high-level abstraction of scene and object structure. The idea of graph clustering is to divide graphs into different groups based on the structural properties of the sample of graphs. Although graph structures have been proved useful in both the low-level and high-level vision, they are computationally cumbersome because of the need to establish reliable correspondence between nodes. Standard graph clustering methods (Lozano and Escolano, 2003; Bunke et al., 2003; Jain and Wysotzki, 2004; Sanfeliu and Fu, 1983; Tsai and Fu, 1983; Bunke, 1997; Bunke, 1999; Torsello and Hancock, 2001) need to solve the correspondence problems between nodes first. Again this is a potentially NP-hard problem and the computational overheads can be large. Recently spectral graph theory has been applied to graph clustering (Horaud and Sossa, 1995; Shokoufandeh et al., 1999; Sengupta and Boyer, 1998). Luo, Wilson and Hancock (Luo et al., 2003; Wilson et al., 2003) have proposed permutation invariant spectral features for graph clustering. These methods for graph clustering do not need to solve the nodes correspondence problems, but they are restricted to using only part of information from the spectrum of the Laplacian matrix.

The heat kernel plays an important role in spectral graph theory. The heat kernel matrix of the graph encapsulates the way in which information flows through the edges of the graph over time. However, the research into the heat kernel is at an early stage. It has not been applied widely in computer vision and pattern recognition until recently. The aim in this chapter is to investigate whether we can extract useful and stable permutation invariants from the heat kernel as a means of graph clustering.

The outline of this chapter is as follows. We will first explore whether the trace of the heat kernel (Yau and Schoen, 1988; Chung and Yau, 1997) can be used for the purpose of characterizing graphs. The trace of the heat kernel is found by summing a series of terms, each of which is the result of exponentiating a Laplacian eigenvalue with time. As a result the heat kernel trace is a function whose parameters are the Laplacian eigenvalues and whose argument is time. Our aim is to explore whether the shape of this function can be used to characterize the corresponding graph. Our idea is to measure the shape of the heat kernel trace by taking moments with respect to time. Using the Mellin transform it is straightforward to show that the moment generating function is related to the zeta function of the graph, which is a series found by exponentiating and summing the reciprocals of the non-zero eigenvalues of the Laplacian. We construct a feature vector whose components are the values of the zeta function with integer argument.

Moreover, the derivative of the zeta function at the origin is related to the product of the non-zero Laplacian eigenvalues. We will also explore the use of the derivative of the zeta function as a means of characterizing graph structure for the purposes of clustering.

We then turn to the heat content, i.e. the sum of the elements of the heat kernel. The heat content can be expanded as a polynomial over time, and the co-efficients of the polynomial are known to be permutation invariants. We demonstrate how the polynomial co-efficients can be computed from the Laplacian eigensystem. Graph clustering is performed by applying principal components analysis to vectors constructed from the polynomial co-efficients.

For our experiments we use publically available datasets, namely the COIL and Oxford-Caltech databases. We show how the manifold learning theory and spectral methods can be combined to solve the image classification problem.

3.2 Heat Kernel on Graphs

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 the elements are

$$A(u,v) = \begin{cases} 1 & \text{if} \quad u, v \in E \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

We also construct the diagonal degree matrix D, whose elements are given by $D(u,u) = \sum_{v \in V} A(u,v)$. From the degree matrix and the adjacency matrix we construct the Laplacian matrix $L = D - A$, i.e. the degree matrix minus the adjacency matrix,

$$L(u,v) = \begin{cases} d_v & \text{if} \quad u = v \\ -1 & \text{if} \quad u \quad \text{and} \quad v \quad \text{are} \quad \text{adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

d_v is D matrix's $D(v,v)$ value. The normalized Laplacian is given by

$$\hat{L}(u,v) = \begin{cases} 1 & \text{if} \quad u = v \quad \text{and} \quad d_v \neq 0 \\ -\frac{1}{\sqrt{d_u d_v}} & \text{if} \quad u \quad \text{and} \quad v \quad \text{are} \quad \text{adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

and can be written as $\hat{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. The spectral decomposition of the normalized Laplacian matrix is $\hat{L} = \Phi \Lambda \Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})(\lambda_1 < \lambda_2 < ... < \lambda_{|V|})$is the diagonal matrix with the ordered eigenvalues as elements and $\Phi = (\phi_1|\phi_2|....|\phi_{|V|})$ is the matrix with the ordered eigenvectors as columns. Since $\hat{L}$ is symmetric and positive semi-definite, the eigenvalues of the normalized Laplacian are all non-negative. The number of zero eigenvalues is the number of isolated cliques in the graph. For a connected graph, there is only one eigenvalue which equals zero. The eigenvector associated with the smallest non-zero eigenvector is referred to as the Fiedler-vector (Chung, 1997).

We are interested in the heat equation associated with the Laplacian, and this is given by.

$$\frac{\partial h_t}{\partial t} = -\hat{L} h_t \tag{3.4}$$

where h_t is the heat kernel and t is time. The ”heat kernel” is the fundamental solution of the heat equation. It can be viewed as describing the flow of information across the edges of the graph with time. The rate of flow is determined by the Laplacian of the graph. The solution to the heat equation is

$$h_t = e^{-t\hat{L}} \tag{3.5}$$

From (Chung, 1997) we can proceed to compute the heat kernel on a graph by exponentiating the Laplacian eigenspectrum, i.e.

$$h_t = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i \phi_i^T = \Phi \exp[-\Lambda t] \Phi^T \tag{3.6}$$

The heat kernel is a $|V| \times |V|$ size matrix, and for the nodes u and v of the graph G the resulting element is

$$h_t(u, v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i(u)\phi_i(v) \tag{3.7}$$

When t tends to zero, then $h_t \simeq I - \hat{L}t$, i.e. the kernel depends on the local connectivity structure or topology of the graph. If, on the other hand, t is large, then $h_t \simeq \exp[-\lambda_2 t]\phi_2 \phi_2^T$, where λ_2 is the smallest non-zero eigenvalue and ϕ_2 is the associated eigenvector, i.e. the Fiedler vector. Hence, the large time behavior is governed by the global structure of the graph.

3.3 Heat Kernel Invariants

In this section we will describe a set of invariants that can be computed from the heat kernel matrix of the graph. These include the heat kernel trace, the zeta function, heat content invariants, and the derivative of zeta function at origin.

3.3.1 Heat Kernel Trace

The trace of the heat kernel is

$$Z(t) = Tr[h_t] = \sum_{i=1}^{|V|} \exp[-\lambda_i t] \tag{3.8}$$

and it is the sum of the diagonal elements of the heat kernel matrix of the graph, where the λ_i is the eigenvalue of the normalized Laplacian matrix.

To provide an illustration of the potential utility of the trace formula, in Figure 3.1 we show four small graphs with rather different topologies. Figure 3.2 shows the trace of the heat kernel as a function of t for the different graphs. From the plot it is clear that the curves have a distinct shape and could form the basis of a useful representation to distinguish graphs. For instance, the more "dumbbell" shaped the graph the more strongly peaked the trace of the heat kernel at the origin. This is due to the fact that the spectral gap, i.e. the size of λ_2, determines the rate of decay of the trace with time, and this in turn is a measure of the degree of separation of the graph into strongly connected subgraphs or "clusters", and this is again due to the fact that for a connected graph, $\lambda_1 = 0$

$$Z(t) = 1 + \exp[-\lambda_2 t] + \exp[-\lambda_3 t] + ... + \exp[-\lambda_N t]$$

.

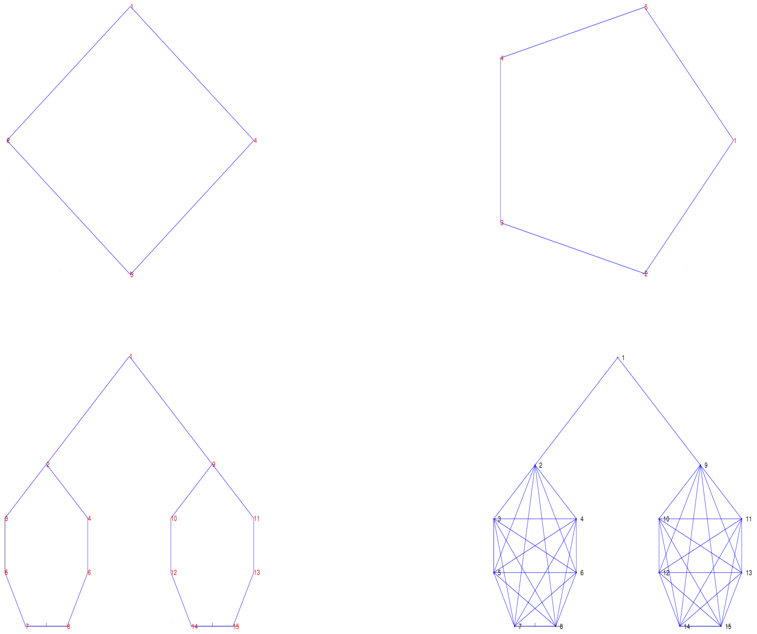

Figure 3.1: Four graphs used for heat kernel trace analysis.

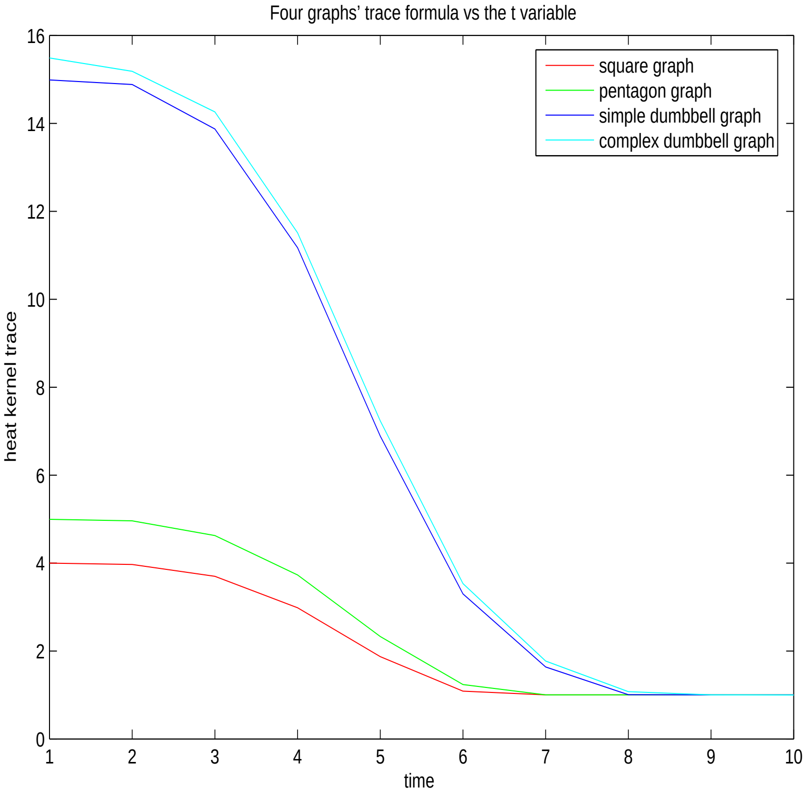

Figure 3.2: Heat kernel trace as a function of t for four graphs from Figure 3.1.

3.3.2 Zeta Function and Heat Kernel Trace Moments

The aim in this section is to use the shape of the heat kernel trace function as a means of characterizing graph structure. Our characterization is found by taking moments of trace function over time. Here we introduce the zeta function, which has a strong relation with the heat kernel trace.

To commence our development, we consider the zeta function associated with the Laplacian eigenvalues. The zeta function is defined by

$$\zeta(s) = \sum_{\lambda_i \neq 0} \lambda_i^{-s} \tag{3.9}$$

In other words, it is the result of exponentiating and summing the reciprocal of the non-zero Laplacian eigenvalues.

To establish the link between the zeta function and the trace of the heat kernel we make use of the Mellin transform. The Mellin Transform of the function $f(t)$ is

$$w(s) = \int_0^\infty t^{s-1} f(t) dt \tag{3.10}$$

From the Mellin Transform

$$\lambda_i^{-s} = \frac{1}{\Gamma(s)} \int_0^\infty t^{s-1} f(t) dt \tag{3.11}$$

and

$$f(t) = \exp(-\lambda_i t) \tag{3.12}$$

In the equation above $\Gamma(s)$ is the gamma function and

$$\Gamma(s) = \int_0^\infty t^{s-1} \exp[-t] dt \tag{3.13}$$

Hence, we can write the zeta function as a moment generating function

$$\zeta(s) = \frac{1}{\Gamma(s)} \int_0^\infty t^{s-1} \sum_{\lambda_i \neq 0} \exp[-\lambda_i t] dt \tag{3.14}$$

The sum of exponentials inside the integral is clearly linked to the trace of the heat kernel. To show this we make use of the fact that

$$Tr[h_t] = C + \sum_{\lambda_i \neq 0} \exp[-\lambda_i t] \tag{3.15}$$

where C is the multiplicity of the zero eigenvalue of the Laplacian, or the number of connected components of the graph. Substituting this result back into the Mellin transform, we have

$$\zeta(s) = \frac{1}{\Gamma(s)} \int_0^\infty t^{s-1} \Big\{ Tr[h_t] - C \Big\} dt \tag{3.16}$$

As a result the zeta function is related to the moments of the heat-kernel trace. It is hence a way of characterizing the shape of the heat kernel trace.

3.3.3 Zeta Function and Torsion

The zeta function is also linked to the determinant of the Laplacian. To show this, we re-write the zeta function in terms of the natural exponential with the result

$$\zeta(s) = \sum_{\lambda_i \neq 0} \lambda_i^{-s} = \sum_{\lambda_i \neq 0} \exp[-s \ln \lambda_i] \tag{3.17}$$

The derivative of the zeta function is given by

$$\zeta'(s) = \sum_{\lambda_i \neq 0} \{-\ln \lambda_i\} \exp[-s \ln \lambda_i] \tag{3.18}$$

At the origin the derivative takes on the value

$$\zeta'(0) = \sum_{\lambda_i \neq 0} \{-\ln \lambda_i\} = \ln \left\{ \prod_{\lambda_i \neq 0} (\frac{1}{\lambda_i}) \right\} \tag{3.19}$$

According to Rosenberg (Rosenberg, 2002) the derivative at the zeta function at the origin is linked to the torsion T, and

$$\zeta'(0) = -\ln Det[T] \tag{3.20}$$

In our experiments, we will use the value of $\zeta'(0)$ as a graph characteristic, and explore whether it can be used to distinguish different graphs.

3.3.4 Unitary Attributes with Symmetric Polynomials

Because of the zeta function, the derivative of zeta function at the origin can provide important features for graph characterization. In this section we illustrate its relationship with the symmetric polynomial method (Wilson et al., 2003) introduced by Wilson and Hancock.

In their recent paper Wilson, Hancock and Luo (Wilson et al., 2003) have reported a family of invariants that can be computed by applying symmetric polynomials to the elements from the graph spectral matrix. The spectral matrix is obtained by performing eigendecomposition on the Laplacian matrix L of the graph, $L = \sum_{i=1}^{n} l_i u_i u_i^T$, where l_i is the ith eigenvalue u_i is the corresponding eigenvector of the symmetric Laplacian L. The spectral matrix C is defined as

$$C = (\sqrt{l_1} u_1, \sqrt{l_2} u_2, ..., \sqrt{l_n} u_n)^T \tag{3.21}$$

It can also be written as $C = \sqrt{\Lambda_L}\Phi_L$, where Λ_L and Φ_L are the eigen matrices of the Laplacian matrix corresponding to the eigenvalues and eigenvectors.

We turn our attention to the symmetric polynomials. A symmetric polynomial on n variables $v_1, v_2, ...v_n$ is a function that is unchanged by any permutation of its variables. In other words, the symmetric polynomials satisfy

$$f(y_1, y_2, ..., y_n) = f(v_1, v_2, ..., v_n) \tag{3.22}$$

where $y_i = v_{\pi(i)}$ and π is an arbitrary permutation of the indices $1, 2, ..., n$.

There are two sets of common symmetric polynomials, the elementary symmetric polynomials and power-sum symmetric polynomials. For a set of variables $\{v_1, v_2 \ldots v_n\}$ the elementary symmetric polynomials are defined as

$$\begin{aligned}
S_1(v_1, \ldots v_n) &= \sum_{i=1}^{n} v_i \\
S_2(v_1, \ldots v_n) &= \sum_{i=1}^{n} \sum_{j=i+1}^{n} v_i v_j \\
&\vdots \\
S_r(v_1, \ldots v_n) &= \sum_{i_1 < i_2 < \cdots < i_r} v_{i_1} v_{i_2} \ldots v_{i_r} \\
&\vdots \\
S_n(v_1, \ldots v_n) &= \prod_{i=1}^{n} v_i
\end{aligned}$$

The output for the n set input variable $\{v_1, v_2 \ldots v_n\}$ is $S_1, S_2, ..., S_n$.

The power-sum symmetric polynomial functions, which are defined as below,

$$\begin{aligned}
P_1(v_1, \ldots v_n) &= \sum_{i=1}^{n} v_i \\
P_2(v_1, \ldots v_n) &= \sum_{i=1}^{n} v_i^2 \\
&\vdots \\
P_r(v_1, \ldots v_n) &= \sum_{i=1}^{n} v_i^r
\end{aligned}$$

$$\vdots$$

$$P_n(v_1, \ldots v_n) \quad = \quad \sum_{i=1}^{n} v_i^n$$

also form a basis set over the set of symmetric polynomials. Any polynomial function which is invariant to permutation of the variable indices $(v_i, v_2, ..., v_n)$ can be expressed in terms of one of these sets of polynomials. The two sets of polynomials are related to one another by the Newton-Girard formula:

$$S_r = \frac{(-1)^{r+1}}{r} \sum_{k=1}^{r} (-1)^{k+r} P_k S_{r-k} \tag{3.23}$$

where we have used the shorthand S_r for $S_r(v_1, ..., v_n)$ and P_r for $P_r(v_1, ..., v_n)$. As a consequence, the elementary symmetric polynomials S_r can be efficiently computed using the power-sum symmetric polynomials. (Wilson et al., 2003) Wilson, Luo and Hancock proposed to use the elements of the spectral matrix $\sqrt{\Lambda_L}\Phi_L$ as input for the symmetric polynomial to compute the invariants for graph characterization.

There are of course alternative invariants that can be computed from the spectrum of the Laplacian. The most obvious of these is the sum of eigenvalues or the trace of the normalized Laplacian eigenvalue matrix of the graph.

$$S_1(\lambda_1, \lambda_2, ..., \lambda_n) = \sum_{i=1}^{n} \lambda_i \tag{3.24}$$

Then if we use the eigenvalues of the normalized Laplacian matrix as input for the symmetric polynomial S_r, the first symmetric polynomial is also the trace of the normalized Laplacian and the last, which is $S_1(\lambda_1, \lambda_2, ..., \lambda_n) = \prod_{i=1}^{n} \lambda_i$, is related to the derivative of the zeta function at the origin. Moreover, (Wilson et al., 2003) Wilson, Hancock and Luo found it necessary to perform logarithmic squashing of the polynomials to bring them into a convenient dynamic range. This property arises naturally through the analysis of the zeta function 3.17.

In our experiments, we will use the symmetric polynomial to construct unitary attributes for graph characterization. The input for the symmetric polynomial are the columns of the spectral matrix and the eigenvalues of the normalized Laplacian matrix.

3.3.5 Heat Content Invariants

In this chapter, we continue considering the heat kernel trace and have explored several techniques for graph characterization based upon it. However, the trace of the heat kernel has limited use for characterizing graphs, since for each value of time it provides only a single scalar attribute. Hence, it must either be sampled with time or a fixed time value has to be selected. In a recent paper McDonald and Meyers (McDonald and Meyers, 2002) have described a set of differential invariants that can be derived from the heat content of the heat kernel. The heat content is the sum of the entries of the heat kernel over the nodes of the graph and is given by

$$Q(t) = \sum_{u \in V} \sum_{v \in V} h_t(u,v) = \sum_{u \in V} \sum_{v \in V} \sum_{k=1}^{|V|} \exp[-\lambda_k t] \phi_k(u) \phi_k(v) \tag{3.25}$$

McDonald and Meyers (McDonald and Meyers, 2002) have shown that the heat content can be expanded as a polynomial in time, i.e.

$$Q(t) = \sum_{m=0}^{\infty} q_m t^m \tag{3.26}$$

Using the MacLaurin Series $\exp[-\lambda_k t]$ can be expanded as

$$\exp[-\lambda_k t] = 1 + (-\lambda_k)t + \frac{(-\lambda_k)^2 t^2}{2!} + ... + \frac{(-\lambda_k)^m t^m}{m!} + ... = \sum_{m=0}^{\infty} \frac{(-\lambda_k)^m t^m}{m!}$$

Substituting the MacLaurin expansion with equation 3.25, we have

$$Q(t) = \sum_{u\in V}\sum_{v\in V}\sum_{k=1}^{|V|} \exp[-\lambda_k t]\phi_k(u)\phi_k(v) = \sum_{m=0}^{\infty}\sum_{u\in V}\sum_{v\in V}\sum_{k=1}^{|V|} \phi_k(u)\phi_k(v)\frac{(-\lambda_k)^m t^m}{m!}$$

As a result the co-efficients q_m appearing in the polynomial expansion of $Q(t)$ are given by

$$q_m = \sum_{k=1}^{|V|}\Big\{(\sum_{u\in V}\phi_k(u))^2\Big\}\frac{(-\lambda_k)^m}{m!} \tag{3.27}$$

The q_m can be treated as a set of invariants which can be used for graph characterization since the co-efficient values are a set of unique values for each graph. In this part, we will explore the use of the polynomial co-efficients for the purpose of graph clustering. To do this we construct a feature vector $\vec{B}= (q_1, q_2, ..., q_k)^T$ from the k leading co-efficients of the heat content polynomial to compare our method with a standard spectral representation. We also compare the use of the vector of leading Laplacian eigenvalues $\vec{B}= (l_1, l_2, ..., l_k)^T$ as a feature vector.

3.3.6 Principal Components Analysis

After constructing the feature vector $\vec{B}$, our next aim is to construct a pattern-space for a set of graphs with pattern vectors $\vec{B}_k$, $k = 1, ..., M$. There are a number of ways in which the graph pattern vectors can be analyzed. Here, for the sake of simplicity, we use principal components analysis (PCA). We commence by constructing the matrix $S = [\vec{B}_1 \mid \vec{B}_2 \mid ... \mid \vec{B}_k \mid ... \mid \vec{B}_M]$ with the graph feature vectors as columns. Next, we compute the covariance matrix for the elements of the feature vectors by taking the matrix product $\sum = \hat{S}\hat{S}^T$, where $\hat{S}$ is computed by substracting the mean of the feature vectors from each column of the matrix S. We extract the principal components directions by performing the eigendecomposition $\sum = \sum_{i=1}^{M} l_i \vec{u}_i \vec{u}_i^T$ on the covariance matrix $\sum$, where the l_i are the eigenvalues and the $\vec{u}_i$ are the eigenvectors. We use the first s leading eigenvectors (3 in practice for visualization purposes) to represent the graphs extracted from the images. The co-ordinate system of the eigenspace is spanned by the s orthogonal vectors $\vec{U}= (\vec{u}_1, \vec{u}_2, ..., \vec{u}_s)$. The individual graphs represented by the vectors B_k, $k = 1, 2, ..., M$ can be projected onto this eigenspace using the formula $\vec{B}_k = \vec{U}^T \vec{B}_k$. Hence each graph G_k is represented by an s-component vector $\vec{B}_k$ in the eigenspace.

3.4 Experiments

In this section we will apply our methods on the COIL and Oxford-Caltech databases and explain whether these invariants can be used for graph characterization. In the first part of this section we will describe the databases and show how to extract graph representations from the images. Second, we will show the results obtained for the databases using the heat kernel trace, symmetric polynomials and heat content invariants.

3.4.1 Database Description

In this and the next chapter we will use two databases, which are widely used in the object recognition literature. It might, therefore, be worthwhile to describe these databases.

The first one is the COIL database (Nene et al., 1996). We choose several objects from the database. For each object there are 72 images taken from the equalized rotated position, 5 degrees per image. Figure 3.4 shows eight sample objects from the COIL database. For the purpose of demonstrating our method on graph clustering, we need to extract the graph representation from the image of the database. For the COIL database, we first extract the feature points using the method of Harris and Stephens (Harris and Stephens, 1994). The feature points or corners can be located by first computing the locally averaged moment matrix using the image gradient. The eigenvalues of the moment matrix can be used to compute a corner ”‘strength”’. The locations of maximum values of the strength indicate the corner positions. We treat these feature points as nodes of a graph. The edge relations are computed by computing the Voronoi tessellations of the feature points, and constructing the region adjacency graph of the Voronoi regions. This process is called Delaunay triangulation. In Figure 3.5, superimposed on the images are the detected feature points and their associated Delaunay graphs. The extracted Delaunay graphs from the images are undirected and unweighted graphs.

The second dataset is the Oxford-Caltech database (Fergus et al., 2003). Compared with the COIL database the Oxford-Caltech is more realistic and complex. The database contains different images of each object or images of the same object from different viewpoints. A sample view is shown in Figure 3.6. For the images in this database we extract relational graphs as belows. First, for each image we use the Canny edge detector(Canny, 1986) to locate edges. From the detected edges we extract the line segments using contour polygonalization. We treat each line segment as a node in the relational graph. The edge weights for each pair of nodes are computed using the relative distance and relative angles attributes between the line segments using the method of Huet and Hancock (Huet and Hancock, 1999). In Figure 3.3 the relative distance d_{gap} between two line segments e_i and e_j is computed by considering the distance of the straight line joining the nearest end points of the two segments. The relative angle θ is found by computing the angle between the connecting straight line at the two end points of the two line segments. The weight $w(e_i, e_j)$ between two line segments e_i and e_j is given by

$$w(e_i, e_j) = E_{Proximity}(e_i, e_j) \times E_{Continuity}(e_i, e_j)$$

where $E_{Proximity}(e_i, e_j)$ represents the significance of proximity between two line segments and $E_{Continuity}(e_i, e_j)$ represents the continuity relationship between two line segments. Here $E_{Proximity}(e_i, e_j)$ is defined as

$$E_{Proximity}(e_i, e_j) = 1 - \min(1, \frac{d_{gap}}{\min(l_i, l_j)})$$

where l_i is the length of the line segment e_i and d_{gap} is the minimum distance between the end points of two line segments. The quantity $E_{Continuity}(e_i, e_j)$ is defined as

$$E_{Continuity}(e_i, e_j) = \min(\frac{l_i + l_j}{l_i + l_j + d_{gap}}, \cos^2(\theta))$$

The weighted links between the line segments aim to capture regular Gestalt inspired relationships of proximity and continuity (Sarkar and Boyer, 1993). The graphs obtained in this way are undirected.

3.4.2 Heat Kernel Trace Experiments

We have experimented with the two characterizations of the heat kernel trace. The first of these are the values of the zeta function with integer argument. As shown above the zeta function is related to the moments of the heat kernel trace, and is a measure of its shape. The second characterization is the slope of the zeta function at the origin.

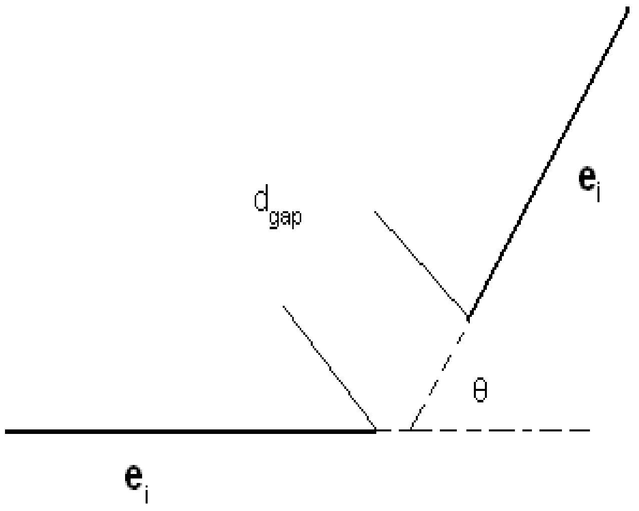

Figure 3.3: Relationship between two line segments.

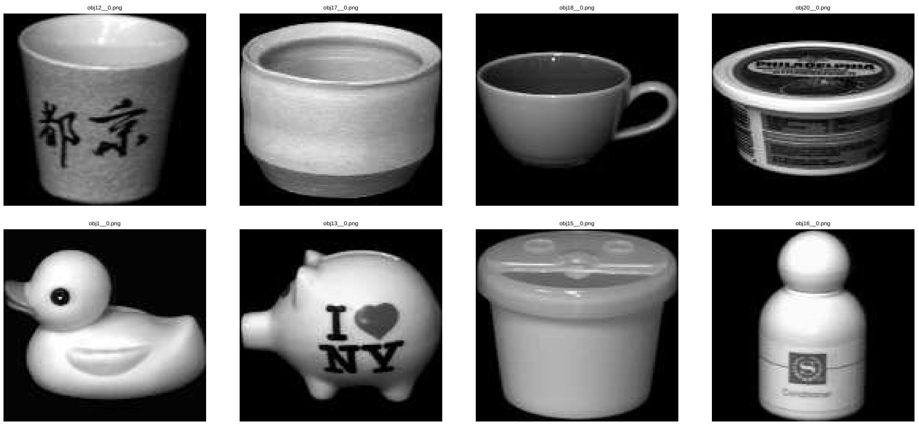

Figure 3.4: Example images of objects from the COIL database.

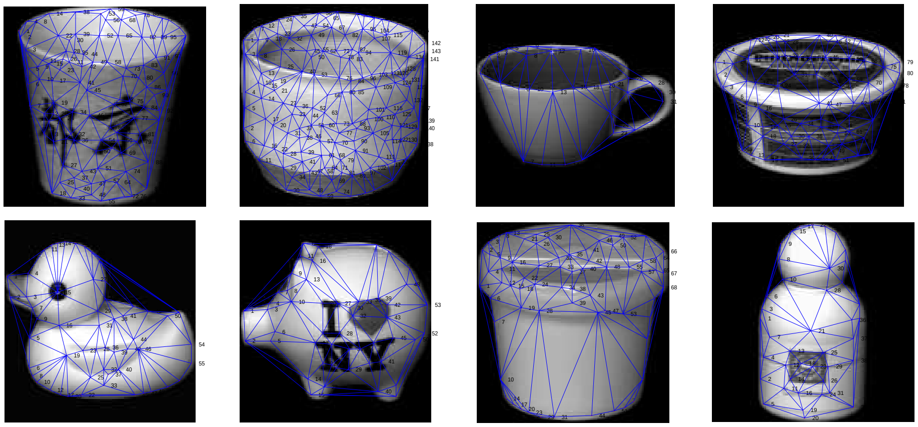

Figure 3.5: Eight objects with their Delaunay graphs overlaid.

We commence by illustrating the behavior of the zeta function for the images of objects from the COIL database. In this experiment we will choose 8 objects from the COIL database. From left-to-right and top-to-bottom in Figure 3.7 we show the values of $\zeta(1)$, $\zeta(2)$, $\zeta(3)$ and $\zeta(4)$ as a function of view number for the eight objects. The different curves in the four plots correspond to the different objects. The main feature to note is that the curves for the different objects are well separated, and that the individual values of the zeta function do not vary significantly with view number. For small integer input, i.e. $s = 1, s = 2$, the fluctuations in the outputs in Figure 3.7 are generally small. However, the fluctuation is large when $s = 4$. This is because when s becomes large the small eigenvalues will play a dominant role in the ζ function.

In Figure 3.8 we show the result of performing principal components analysis on a feature vector $B = (\zeta(1), \zeta(2)....., \zeta(N))^T$ which has as its components the zeta function evaluated at the integers 1 to N. In order for the computation to be convenient and the experiments to be synchronized, we choose $N = 6$ to construct the feature vector for each graph. In the following experiments in this Chapter, for different graph characterizations we will construct the

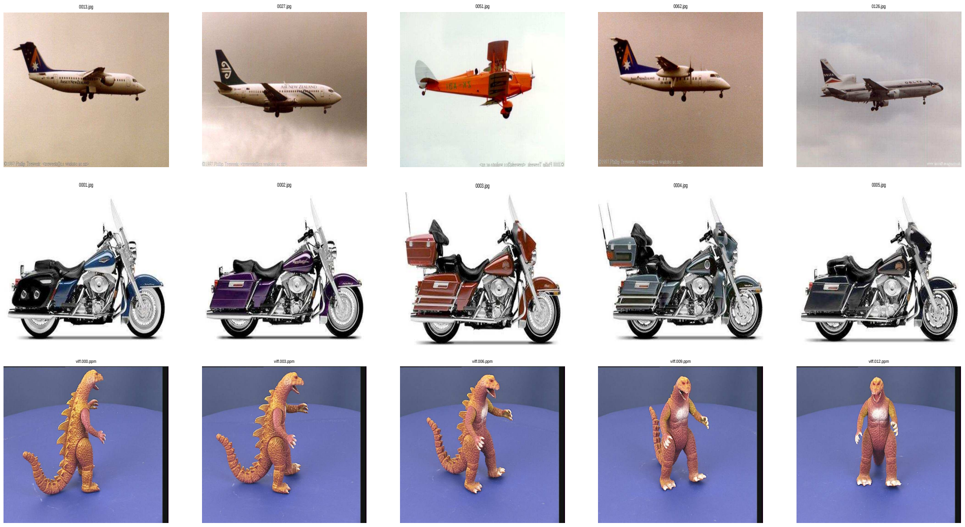

Figure 3.6: Example images from the Oxford-Caltech database.

feature vector, which has a length of six, for each graph. Here we project the graphs onto the eigenspace spanned by the first three eigenvectors of the feature vector covariance matrix as explained in section 3.3.6. The different objects are denoted by points of a different color. The different objects are well separated in the eigenspace. For comparison in Figure 3.9 we show the result of repeating this analysis on a vector of leading eigenvalues of the Laplacian matrix $B = (l_1, l_2...., l_6)^T$. In the eigenspace, the objects overlap, and the clustering is poorer.

In Figure 3.10 we repeat the analysis of the zeta function for the objects from the Caltech-Oxford database. For the graphs extracted from the Oxford-Caltech database we use the six leading zeta function values to construct the feature vector. We then perform PCA to project the graphs onto a three dimensional space. Again, good clusters are obtained. The graphs for the three objects are well separated.

We now turn our attention to the derivative of the zeta function at the origin. In Figures 3.12 and 3.11 we show the curves $\zeta'(0)$ as a function of view number for the Caltech-Oxford and COIL databases respectively. The curves are well separated for the different objects, and the fluctuations are relatively small. In fact in the case of the COIL data the derivative of the zeta function separates the objects better than the individual zeta function values. Figures 3.13 and 3.14 show histograms of $\zeta'(0)$ for the different objects. These are well separated and do not overlap.

To show how the use of $\zeta'(0)$ compares with the use of alternative unary graph features, in Figure 3.15 and 3.16 we show the node and edge frequencies for the COIL and Oxford-Caltech databases. The left plot in each figure is the histogram of the number of nodes for each graph and the right plot is the histogram of the number of edges for each graph. The different objects in these histograms overlap more than in the $\zeta'(0)$ histograms for the databases. As a result the $\zeta'(0)$ appears to be a powerful unary feature for graph clustering.

3.4.3 Unitary Attributes from Symmetric Polynomials

We have compared the slope of the zeta function at the origin, i.e. $\zeta'(0)$, with the symmetric polynomials as unary features for discriminating graphs. The data used for this experiment are again the COIL and Oxford-Caltech databases.

We will first use the symmetric polynomial with the normalized Laplacian eigenvalues as input arguments to compute the unitary attributes. When the symmetric polynomial takes as its arguments the normalized Laplacian eigenvalues, then the lowest order polynomial is the sum of the eigenvalues, i.e. the trace of the normalized Laplacian, and the highest order polynomial is the product of non-zero normalized Laplacian eigenvalues. Hence,

$$S_{|V|}(\lambda_1, \lambda_2, ..., \lambda_{|V|}) = \exp[(\zeta'(0))^{-1}]$$

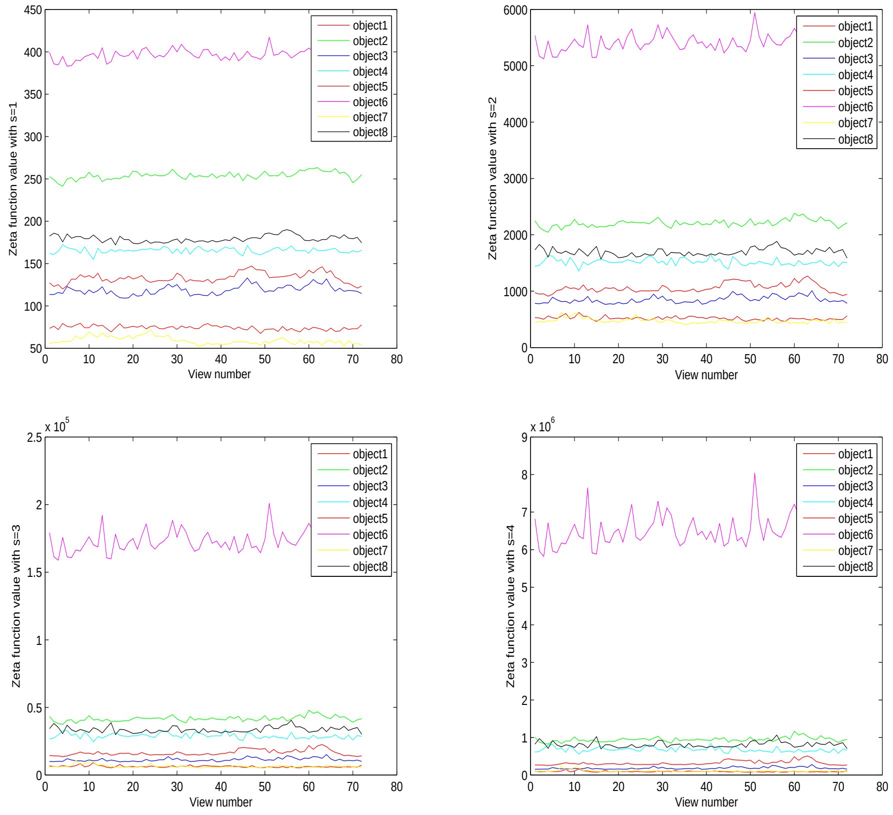

Figure 3.7: Zeta function $\zeta(s)$ with view number(from left to right, and top to bottom, $s = 1, 2, 3$ and 4 respectively).

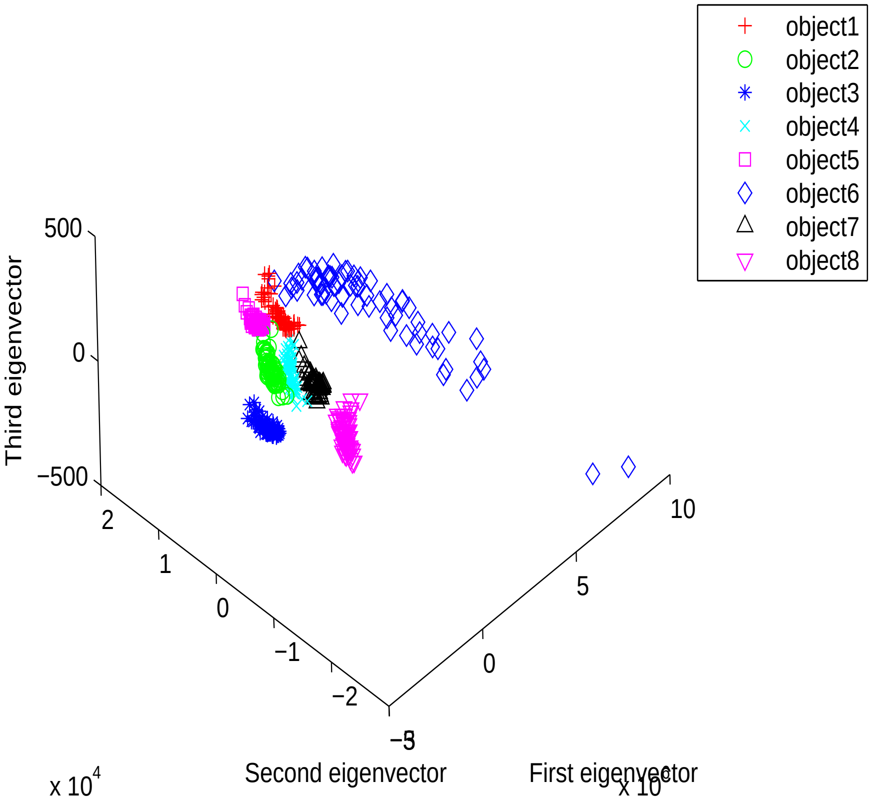

Figure 3.8: Zeta function clustering for the COIL database.

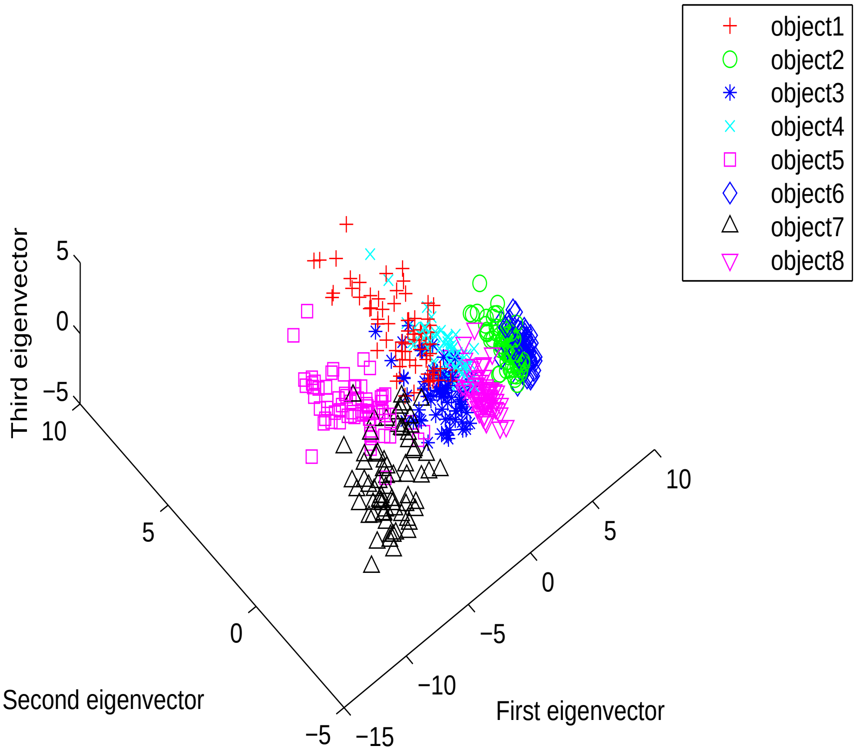

Figure 3.9: Spectral clustering for the COIL database.

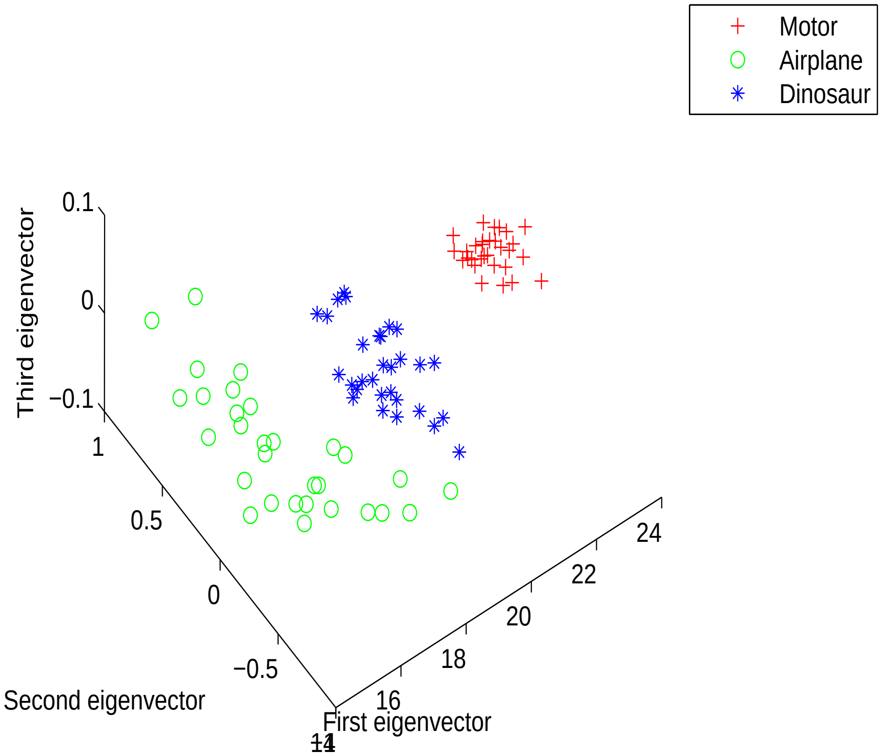

Figure 3.10: Zeta function clustering for the Oxford-Caltech database.

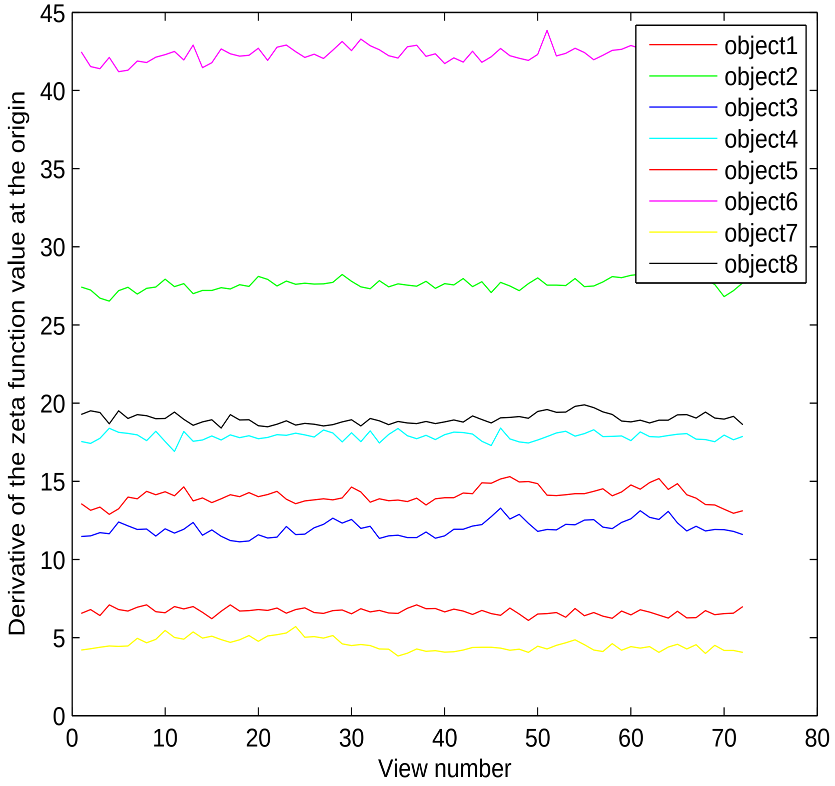

Figure 3.11: Derivative of the zeta function at the origin for the COIL database.

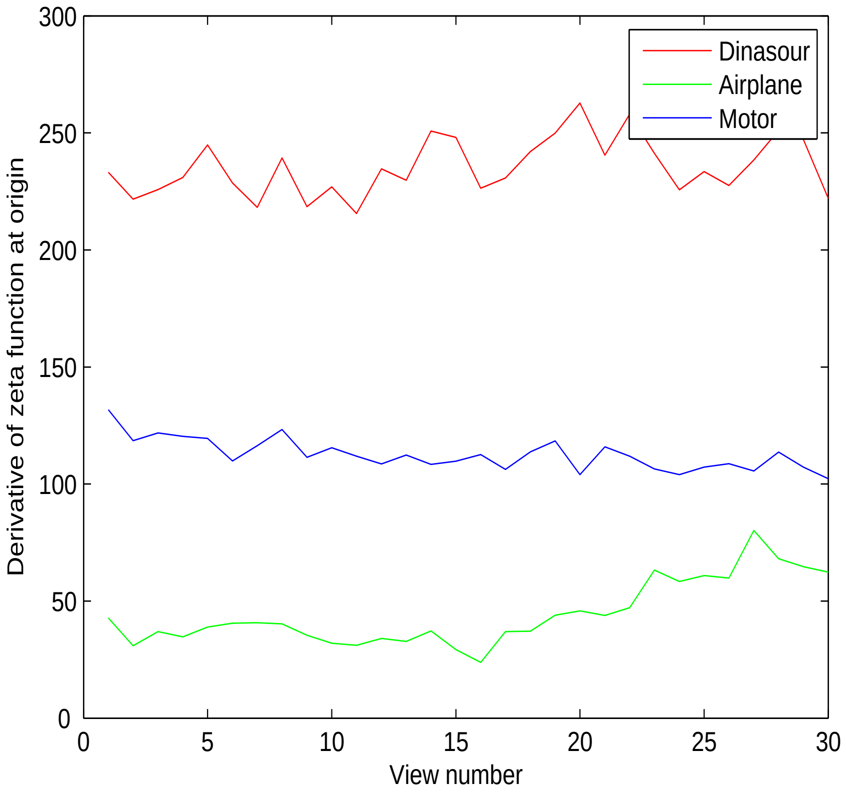

Figure 3.12: Derivative of the zeta function at the origin for the Oxford-Caltech database.

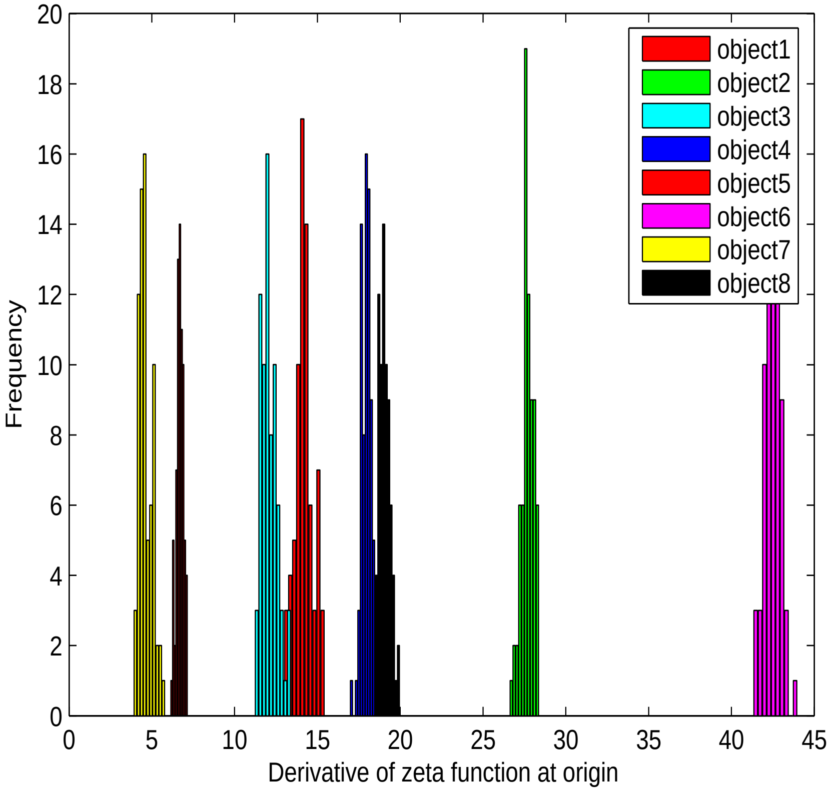

Figure 3.13: Histogram of the derivative of the zeta function at the origin for the objects from the COIL database.

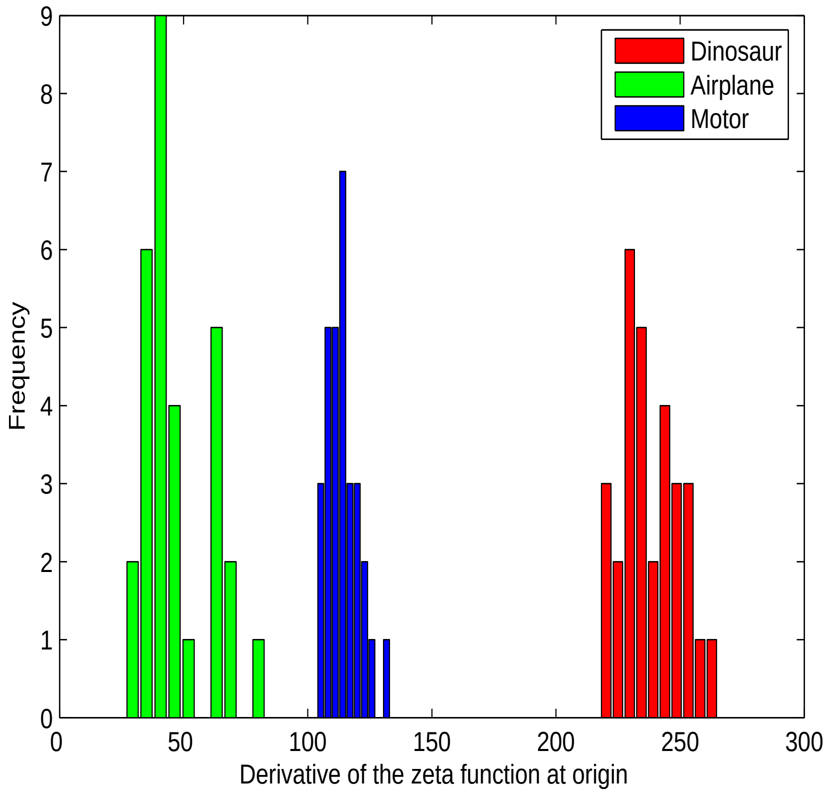

Figure 3.14: Histogram of the derivative of the zeta function at the origin for the objects from the Oxford-Caltech database.

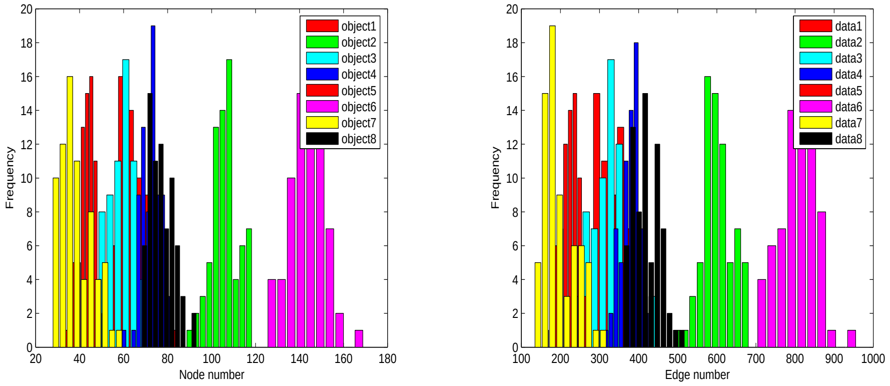

Figure 3.15: Node number(left) and edge number(right) histograms for the COIL database.

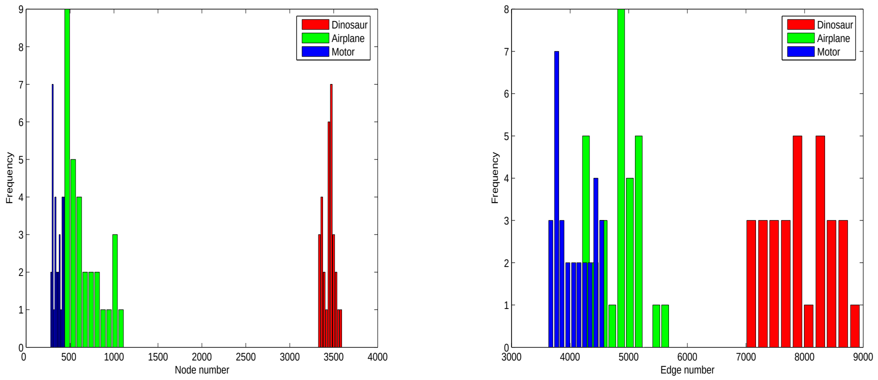

Figure 3.16: Node number(left) and edge number(right) histograms for the Oxford-Caltech database.

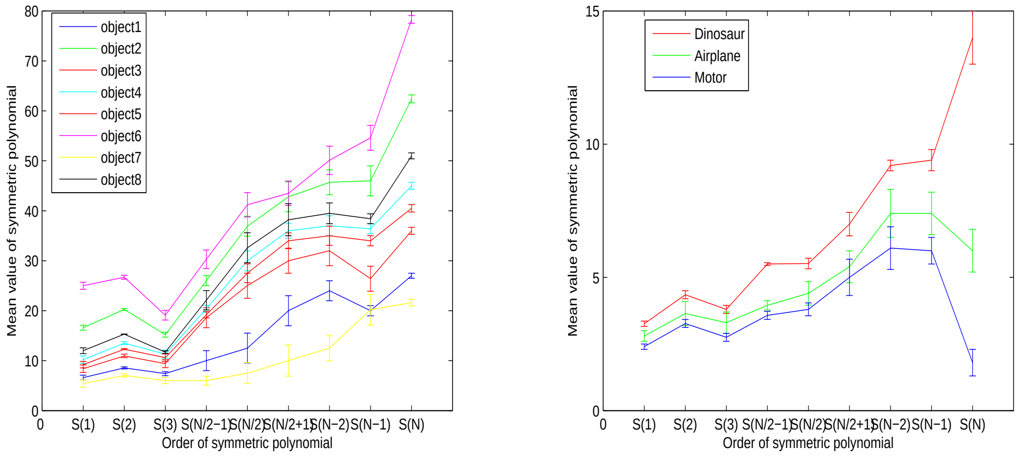

Figure 3.17: Symmetric polynomials using spectral matrix elements for COIL(left) and Oxford-Caltech(right) databases.

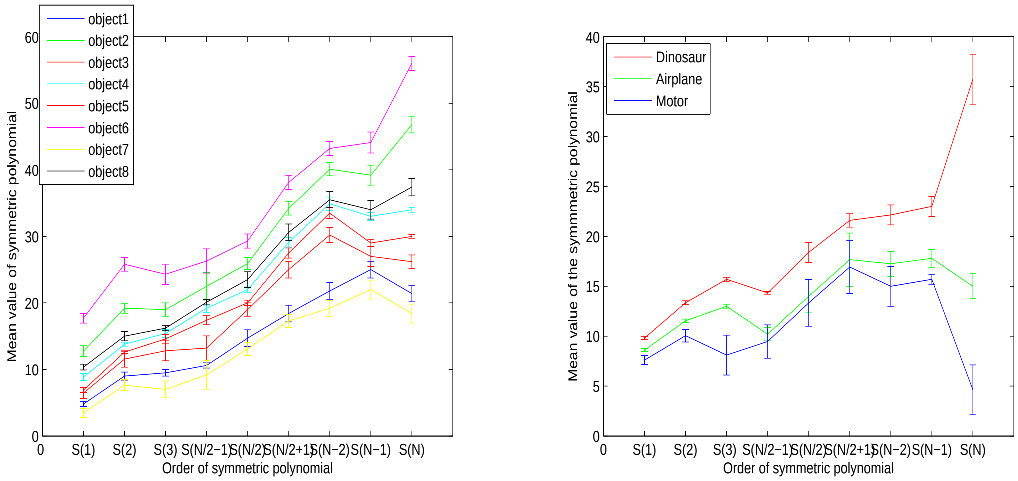

Figure 3.18: Symmetric polynomials using normalized Laplacian eigenvalues for COIL(left) and Oxford-Caltech(right) databases.

where $|V|$ is the number of nodes in the graph.

In Figure 3.17 we plot the mean values from the symmetric polynomials over the set of object views as a function of their order. Here the arguments of the polynomials are the column elements of the spectral matrix $\sqrt{\Lambda_L}\Phi_L$. The different curves in the plot are for different objects. We choose 9 different symmetric polynomials for experiment. These are the first three symmetric polynomials S_1, S_2, S_3, the middle three symmetric polynomials $S_{fix(\frac{N}{2})-1}, S_{fix(\frac{N}{2})}, S_{fix(\frac{N}{2})+1}$ and last three symmetric polynomials S_{N-2}, S_{N-1}, S_N, where N is the number of node in the graph. The error bars show the standard deviation over the different sets of views of the same object. The left-hand plot is for the COIL database and the right-hand plot is for the Oxford-Caltech database. Figure 3.18 repeats the analysis using the normalized Laplacian spectrum as the arguments of the polynomials.

From Figures 3.18 and 3.17 it is clear that the first three and last three symmetric polynomials computed using the normalized Laplacian eigenvalues and spectral matrix elements as inputs can be used for object separation. We also get a better separation of objects when using the polynomials from the normalized Laplacian spectrum rather than the elements of the spectral matrix.

3.4.4 Heat Content Invariants Experiments

We now turn our attention to the invariants computed from the heat content characterization described in section 3.3.5. This time we will use the eight objects in Figure 3.4 for testing.

In Figure 3.19 we plot the four leading heat content invariants co-efficients q_1, q_2, q_3 and q_4 separately as a function of the view number for the eight objects selected from the COIL database. The co-efficients are relatively stable with viewpoint. In the left-hand panel of Figure 3.20 we show the result of performing PCA on the vectors of six leading polynomial co-efficients. For comparison, the right-hand panel in Figure 3.20 shows the corresponding result when we apply PCA to the vector of leading eigenvalues of the Laplacian matrix as we have described in section 3.4.2. The main qualitative feature is that the different views of the eight objects overlap more than when the heat content polynomial co-efficients are used.

3.4.5 Heat Kernel Invariants Comparison

To investigate the behavior of these methods i.e. heat kernel trace, heat content invariant, in a more quantitative way, we have computed the Rand index (Rand, 1971) for the different methods. The Rand index is defined as

$$R_I = \frac{C}{C+W} \tag{3.28}$$

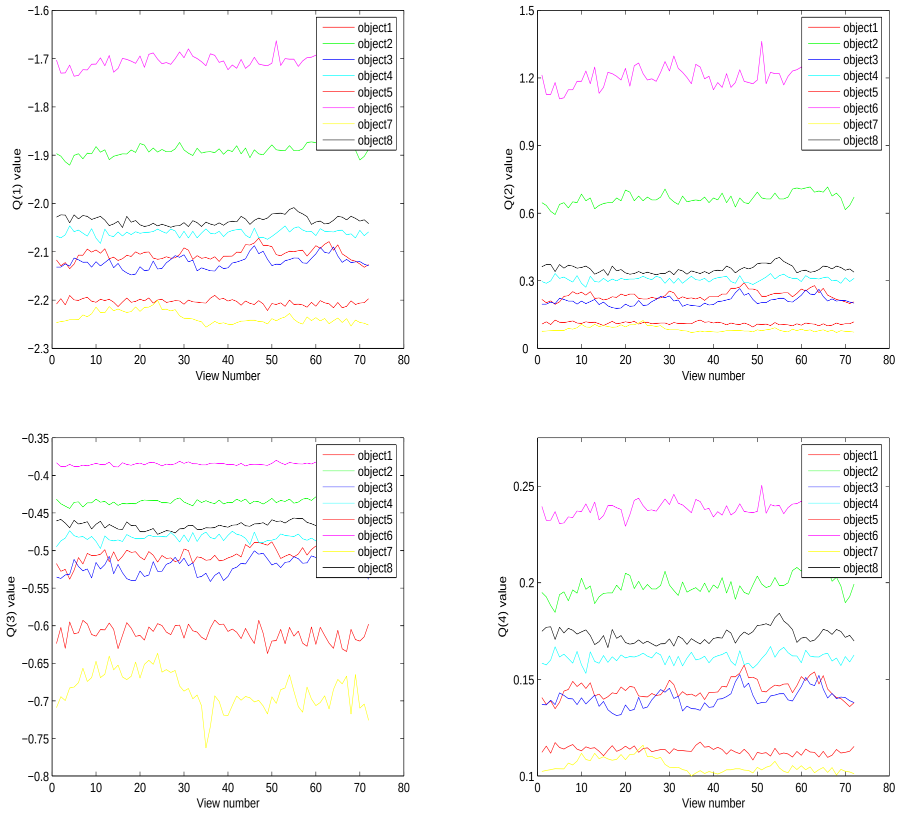

Figure 3.19: Individual heat content invariants as a function of view number.

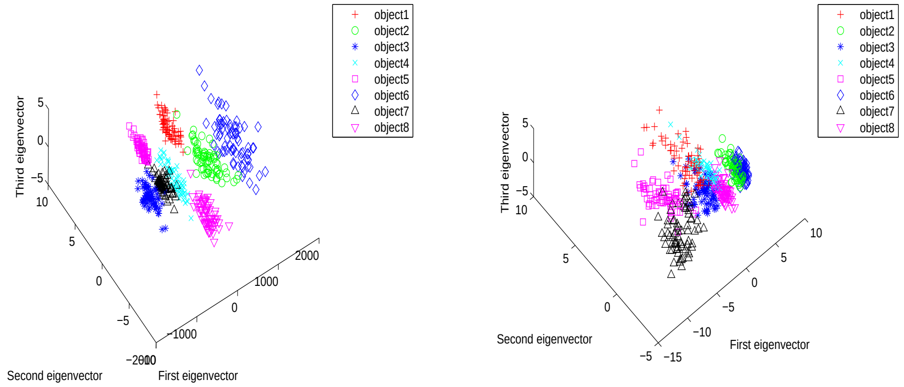

Figure 3.20: Principal component analysis results of the heat content differential co-efficients(left) and Laplacian spectrum (right).

where C is the number of agreements and W is the number of disagreements in cluster assignment. If two objects are in the same cluster in both the ground truth clustering and the clustering from our experiment, this counts as an agreement. If two objects are in the same cluster in the ground truth clustering but in different clustering from our experiment, this counts as a disagreement. The index is hence the fraction of views of a particular class that are closer to an object of the same class than to one of another class. The Rand index takes a value in the interval $[0, 1]$, where 1 corresponds to a perfect clustering.

Since we have already known the ground truth clustering before the experiments, finding out the clustering results from the low-dimensional embedded coordinates can use the original ground truth clustering results. In this part and the rest of the thesis, we use the K-nearest algorithm. The idea is for each embedded coordinate we search its K nearest neighbors. We then check the ground truth cluster numbers of these K neighbors to see if over half of these neighbors belong to the same category as its ground truth one. In our experiment we set K to 5. For example, a graph where in the ground truth it belongs to the first object class. In the embedded coordinate space, we check its five nearest neighbors. If three or more of its neighbors also belong to the first object class in their ground truth clustering, then we say it is an agreement. Otherwise it is a disagreement. We repeat this process for all graphs used in the experiments and compute the Rand Index value according to the equation 3.28.

In the first experiment we use the Rand index to compare five different characterizations for graph clustering. We use the COIL database, which contains eight different objects and more than five hundred graphs, so it is more complex than the Oxford-Caltech database. The five different methods are the zeta function ζ, symmetric polynomial computed using Laplacian eigenvalues and the spectral matrix elements as inputs, heat content invariants q_m, and traditional spectral clustering. Since the derivative of the zeta function at the origin is a unary feature and the five characterizations above are not, we will not compare them with this. In Figure 3.8, the Rand index for zeta ζ function characterization clustering is 0.92. For the symmetric polynomial clustering we use the first and last three symmetric polynomial values $S_1, S_2, S_3, S_{N-2}, S_{N-1}, S_N$ as feature vector for graph clustering. The Rand index value for the symmetric polynomial using Laplacian eigenvalues as inputs is 0.90 and the Rand index value for the symmetric polynomial using spectral matrix elements as inputs is 0.88. In Figure 3.20, the Rand index for the heat content co-efficients clustering is 0.96 while for the Laplacian eigenvalues it is 0.78. This experiment shows that the best clustering result is from the heat content co-efficient characterization.

In the second experiment we evaluate the stability of the different graph characterizations. We now use six different methods for graph characterization. These are the five methods used above, together with the derivative of the zeta function at the origin. We generate a random graph (Zhu, 2006) with thirty five vertices and one hundred and twenty eight edges. We compute the Euclidean distance between the coordinate vector from the principal component analysis (PCA) projection of the graph feature vector and its modified variant obtained by deleting between 3 and 27 edges from the random graph. The edge to be deleted is chosen at random. For each deletion, we perform 30 trails in order to obtain an average distance and a standard deviation.

In Figure 3.21, the x-axis in each plot is the deleted edge number, while the y-axis is the mean Euclidean distance between the original random graph and the modified variant graph together with the standard deviation. These plots show that the five characterizations we proposed in this chapter give a relationship between the Euclidean distance and deleted edge number much close to a linear one, while the tradition spectral seems marginally less linear. In Table 3.1, we

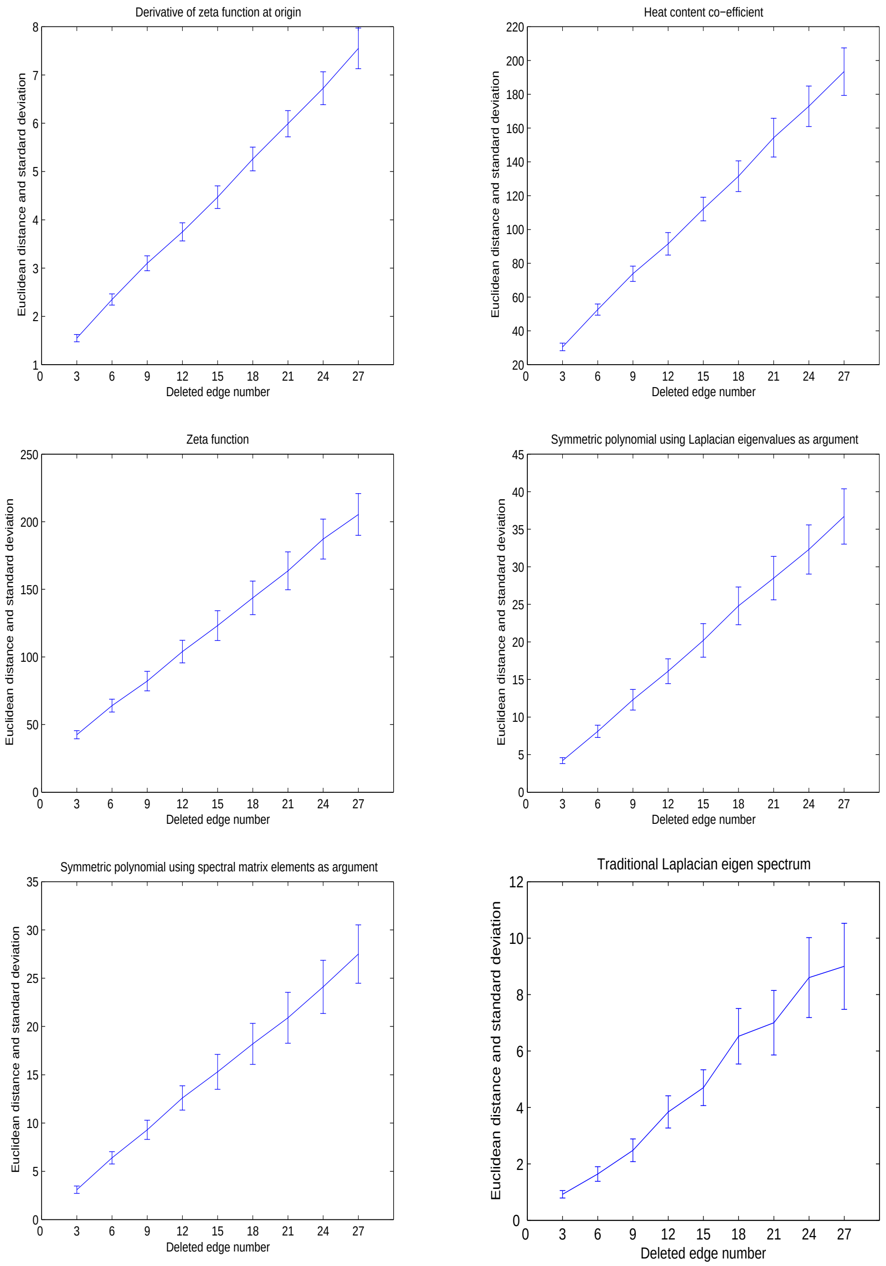

Figure 3.21: Euclidean distance and standard deviation of the random graph experiments.

Graph characterization method	9 edge deletion	15 edge deletion	21 edge deletion	27 edge deletion
Derivative of zeta function at origin	0.0644	0.0607	0.0545	0.0527
Heat content co-efficient	0.0677	0.0650	0.0621	0.0592
Zeta function	0.0847	0.0830	0.0801	0.0758
Symmetric polynomial with Laplacian eigenvalues	0.1103	0.0978	0.0947	0.0911
Symmetric polynomial with spectral matrix	0.1221	0.1165	0.1132	0.1025
Traditional Laplacian eigen spectrum	0.1627	0.1608	0.1544	0.1503

Table 3.1: Relative deviations for six different graph characterizations.

show the relative deviation value, which is the standard deviation divided by the mean, for the six graph characterizations. The value gives an indication of the stability of the different graph characterization methods. From the result, it is clear that the invariants computed from the graph heat kernel can be used as stable ways for graph characterization. The five different graph characterizations proposed in this chapter outperformed the traditional spectral method in terms of both accuracy and stability. The derivative of the zeta function can be used as a unary feature for graph characterization.

3.5 Conclusion

In this chapter we have used several invariants computed from the heat kernel to characterize the graphs. First we have explored the use of the zeta function as a means of characterizing the shape of the heat kernel trace. Using the Mellin transform, we have shown that the zeta function is linked to the moment generating function of the heat kernel trace. We have also shown that the derivative of the zeta function at the origin is related to the determinant of the Laplacian matrix. Both characterizations work well on the COIL and Caltech-Oxford databases. Encouragingly, the derivative of the zeta function at the origin seems to be a unary feature that can be used to measure graph similarity.

The slope of the zeta function at the origin is related to the product of the non-zero Laplacian eigenvalues, and is hence an invariant to the permutation of the node order of the graph. We have further investigated alternative unary permutation invariants by using the symmetric polynomial.

Finally we have explored how the use of heat content can lead to a series of invariants that can be used as an important feature to characterize the graphs. We have shown the effectiveness of this method on the real world images.

Chapter 4

Geometric Characterization of Graphs using Kernel Embedding

4.1 Introduction

In the previous chapter we have shown that we can extract useful and stable invariants from the heat kernel to characterize the graphs. In this chapter we will further explore the geometry of the graphs which resides on a manifold. To do so we need to develop a way to embed the nodes of graphs to points in a vector space on a manifold. In other words, we are seeking a mapping from the node-set of the graph to point-set in a vector space. In this chapter we will introduce two methods for graph embedding. Both of them depend on the analysis of the heat kernel of the graph.

One way is to use the ISOMAP (Tenenbaum et al., 2000) algorithm and multi-dimensional scaling to locate a low-distortion embedding of the Euclidean distance. We call this process parametric distance embedding. This is a technique that allows data specified in terms of relative distances rather than ordinal values to be embedded in a low-dimensional space in a way that minimizes the distortion(or stress) of the distance pattern. To apply this technique to graphs we require a method of assigning a distance measure to the graph edges. Our distance function is furnished by an analysis of the heat kernel. When the manifold on which the nodes of the graph reside is locally Euclidean, then the heat kernel may be approximated by a Gaussian function of the geodesic distance between nodes. By equating the spectral and Gaussian forms of the kernel, we can estimate the Euclidean distances between the nodes of the graph under study. The geodesic distance (i.e the shortest distance on the manifold) is given by the floor of the path-length distribution, and this may be computed from the Laplacian spectrum. The manifold may be approximately reconstructed by using multidimensional scaling to locate a low-distortion embedding of the Euclidean distances.

Another way to embed the node-set of the graph to point-set in a vector space is to perform Young-Householder decomposition (Young and Householder, 1938) on the graph heat kernel. Without computing any distances we only use values contained in the heat kernel as the weights between nodes in the graph. Young-Householder eigendecomposition is then performed on the matrix to map the node of the graph to a high dimensional vector. We call this process heat kernel embedding.

Once embedded in the vector space by using these two methods, we can perform a number of graph manipulation tasks by applying simple point pattern analysis algorithms to the mapped node positions, since the node distributions reflect the geometry information contained in the graph. Here we focus on the problem of graph clustering. Our approach is to extract feature vectors to characterize the point distributions that result from embedding the nodes of the graphs. We introduce two ways to characterize the embedded point vectors. The first method is to use statistical moments. The second method is to use a spectral characterization. Then each graph has an associated feature vector that characterizes how its nodes are distributed in the embedded space. Sets of graphs can be projected into a pattern space by performing principal components analysis (PCA) on the feature vectors.

We will introduce a third way, the sectional curvature, to characterize the graphs geometries based on the Euclidean distances deduced from the two embedding methods. The difference between the geodesic and Euclidean distances can be used to compute the sectional curvatures associated with the edges of the graph. The sectional curvature can also be used for graph characterization. To characterize the manifold on which the graph resides, we use the normalized histogram of sectional curvatures. By performing principal component analysis on long vectors representing the histogram bin-contents, we can construct a pattern space for sets of graphs.

4.2 Parametric Distance Embedding

In this section we introduce the first graph embedding method, which we refer to as parametric distance embedding. We begin with the Euclidean distance deduced by equating the spectral and Gaussian forms of the heat kernel. From the Euclidean distance we can construct a distance matrix for each graph. Isomap (Tenenbaum et al., 2000) is performed on the distance matrix to embed the nodes of graphs to points in a vector space.

4.2.1 Euclidean Distance from Heat Kernel

In the previous chapter, we gave the definitions on the adjacency matrix, Laplacian matrix and normalized Laplacian matrix of a graph. We also gave the solution of the heat kernel on the graph. It is interesting to note that the heat kernel is also related to the path length distribution on the graph. To show this, consider the matrix $P = I - \hat{L}$, where I is the identity matrix. The heat kernel can be rewritten as $h_t = e^{-t(I-P)}$. We can perform the MacLaurin expansion on the heat kernel to re-express it as a polynomial in t. The result of this expansion is

$$h_t = e^{-t}(I + tP + \frac{(tP)^2}{2!} + \frac{(tP)^3}{3!} + ...) = e^{-t} \sum_{k=0}^{\infty} P^k \frac{t^k}{k!} \tag{4.1}$$

For a connected graph, the matrix P has elements

$$P(u,v) = \begin{cases} 0 & \text{if} \quad u = v \\ \frac{1}{\sqrt{d_u d_v}} & \text{if} \quad u \neq v \quad \text{and} \quad (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

As a result, we have that

$$P^k(u,v) = \sum_{S_k} \prod_{i=1}^{k} \frac{1}{\sqrt{d_{u_i} d_{u_{i+1}}}} \tag{4.3}$$

where the walk S_k is a sequence of vertices $u_0, ..., u_k$ of length k such that $(u_i, u_{i+1}) \in E$. Hence, $P^k(u,v)$ is the sum of weights of all walks of length k joining nodes u and v. In terms of this quantity, the elements of the heat kernel are given by

$$h_t(u,v) = \exp[-t] \sum_{k=0}^{|V|^2} P^k(u,v) \frac{t^k}{k!} \tag{4.4}$$

We can find a spectral expression for the matrix P^k using the eigendecomposition of the normalized Laplacian. Writing $P^k = (I - \hat{L})^k$ it follows that $P^k = \Phi(I - \Lambda)^k \Phi^T$. The element associated with the nodes u and v is

$$P^k(u,v) = \sum_{i=1}^{|V|} (1 - \lambda_i)^k \phi_i(u) \phi_i(v) \tag{4.5}$$

The geodesic distance between nodes, i.e. the length of the walk on the graph with the smallest number of connecting edges, can be found by searching for the smallest value of k for which $P^k(u,v)$ is non zero, i.e. $d_G(u,v) = floor_k P_k(u,v)$

Here we are interested in using the heat kernel to compute an approximate Euclidean distance between nodes. This is the shortest distance between nodes in the vector space in which the manifold resides. On a Riemannian manifold the heat kernel can be approximated by the so-called paramatrix (Yau and Schoen, 1988; Grigor'yan, 2003)

$$h_t(u,v) = [4\pi t]^{-\frac{n}{2}} \exp[-\frac{1}{4t} d_G^2(u,v)] \tag{4.6}$$

where $d_G(u,v)$ is the geodesic distance between the nodes u and v on the manifold, n is the dimensionality of the space. In our experiment, we choose $n = 3$, since we have finally embedded the graph in a three dimensional space. We may equate the heat kernel definition on the manifold and on the graph. Hence, to approximate the Euclidean distance between the embedded nodes we can equate the spectral and Gaussian forms for the kernel

$$h_t(u,v) = \sum_{i=1}^{|V|} \exp(-\lambda_i t)\phi_i(u)\phi_i(v) = [4\pi t]^{-\frac{n}{2}} \exp[-\frac{1}{4t} d_G^2(u,v)] \tag{4.7}$$

We can approximate the Euclidean distance d_E

$$d_E^2 \cong d_G^2 = -4t \ln\{(4\pi t)^{\frac{n}{2}} \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i(u)\phi_i(v)\} \tag{4.8}$$

4.2.2 Manifold Embedding of Graphs using the Euclidean Distance

In the previous section we have shown how to deduce the Euclidean distance from the heat kernel. This leads us to the spectral form for the distance matrix

$$d_E^2 = -4t\ln\{(4\pi t)^{\frac{n}{2}}\sum_{i=1}^{|V|}\exp(-\lambda_i t)\phi_i(u)\phi_i(v)\} = -\ln[(4\pi t)^{2nt}\Phi\exp[-4\Lambda t^2]\Phi^T] \tag{4.9}$$

Hence, the distance matrix is just the negative logarithm of a matrix found by applying a transformation to the eigenvalue spectrum. Such a procedure has been suggested by Smola and Kondor (Smola and Kondor, 2004) as a means of regularizing graphs. However, the function suggested above is more complex than the alternatives explored for regularization.

It is interesting to consider the behavior of the distance function for small and large values of t. As t tends to zero, then d_E^2 also tends to zero. For small values of t we can write

$$d_E^2(u,v) \simeq -4t\left\{\frac{n}{2}\ln 4\pi t + \ln[1-\hat{L}(u,v)t]\right\} \simeq 4\hat{L}(u,v)t^2 - 2nt\ln 4\pi t \tag{4.10}$$

In other words, the distance is proportional to t and governed by the corresponding element of the normalized Laplacian matrix. For large t, and as noted before, the behavior is dominated by the smallest non-zero eigenvalue of the Laplacian. If λ_m is the smallest non-zero eigenvalue of $\hat{L}$ and ϕ_m is the corresponding eigenvector, or Fiedler vector, then

$$d_E^2(u,v) \simeq -4t\left\{\frac{n}{2}\ln 4\pi t - \lambda_m t + \ln\phi_m(u)\phi_m(v)\right\} \tag{4.11}$$

Hence, the large t behavior of the distance function is again proportional to t and is governed by the eigenvalue λ_m.

4.2.3 Metric Embedding using ISOMAP

The problem of embedding a finite metric space into a Euclidean space, or other normed spaces, that approximately preserve the metric is one that has received considerable attention in recent years (Hjaltason and Samet, 2003). A number of ways have been proposed for measuring the quality of an embedding procedure. A review of the methods is presented by Hjaltason and Samet (Hjaltason and Samet, 2003). The *distortion* has been widely accepted as a measure of the quality of the embedding. For a finite metric space (X, d) and $c \geq 1$, there is an embedding φ of X into Y where for every two points $x_1, x_2 \in X$ satisfy the condition

$$c_1 d(x_1, x_2) \geq \|\varphi(x_1) - \varphi(x_2)\| \geq \frac{1}{c_2} d(x_1, x_2) \tag{4.12}$$

Such an embedding is said to be have *distortion* $\leq c$, where c is the lowest value for all c_1 and c_2 (Linial et al., 1995b). Recently low-distortion embedding has provided powerful tools for designing efficient pattern analysis algorithms. This is because they enable us to reduce problems defined over difficult metrics to problems over much simpler ones.

The starting point for most metric embedding methods is Bourgain's (Bourgain, 1985) Lemma:

> Any finite metric(X, d) can be embedded into l_2^p with $p < \infty$ with distortion $O(log|X|)$.

We denote $\Re^n$ equipped with l_q norm by l_q^n. The Euclidean norm is l_2. The l_q norm is defined as $||(x_1, ..., x_n)||_q = (\sum |x_i|^q)^{1/q}$. The original bound on p

proved by Bourgain was exponential with n and too large to be of practical use. We seek to introduce an embedding with a much lower distortion.

We first define a suitable metric for the graphs. For a given graph $G = (V, E)$, V represents the nodes in the graph and E represents the edge relations between the nodes. Suppose that d is a metric on the graph G. The metric must satisfy the condition that for any three vertices u, v and $w \in V$, if $d(u, v) = d(w, v) \geq 0$, then $d(u, u) = 0$ and $d(u, v) \leq d(u, w) + d(w, v)$. There are many ways to define metric distances on a graph. In this thesis, we use the Euclidean distance deduced from the heat kernel as a measure of distance between the nodes of the graph. We construct a distance matrix for the graph. The graph embedding process is performed on this distance matrix.

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 (Tenenbaum et al., 2000) as a way to solve the low-distortion graph embedding problem. The idea behind Isomap is to apply classical MDS (Cox and Cox, 1994) to map data points from their high-dimensional input space to low-dimensional coordinates 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. 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 (Tenenbaum et al., 2000). The distortion in this embedding is nearly 1. The idea underpinning Isomap is to apply the MDS on the shortest distance matrix computed from the input data. The novel contribution here is hence to apply MDS to the pairwise Euclidean distance matrix deduced from the heat kernel analysis.

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

4.2.4 Multidimensional Scaling

The pairwise Euclidean distances between nodes $d_E(u,v)$ are used as the elements of an $|V| \times |V|$ dissimilarity matrix S, whose elements are defined as follows

$$S(u,v) = \begin{cases} d_E(u,v) & \text{if} \quad u \neq v \\ 0 & \text{if} \quad u = v \end{cases} \tag{4.13}$$

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

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

4.3 Heat Kernel Embedding

In this section we will introduce the second graph embedding method, which is heat kernel embedding. We use the heat kernel to map nodes of the graph to points in the vector space. In other words, we perform Young-Householder decomposition on the graph heat kernel. We provide an analysis which shows how the eigenvalues and eigenvectors of the covariance matrix for the point distribution resulting from the kernel mapping can be used for graph characterization.

4.3.1 Co-ordinate Embedding

To commence, suppose that the graph under study is denoted by $G = (V, E)$ where V is the set of edges and $E \in V \times V$ is the set of nodes. From the adjacency matrix A of the graph we can compute the corresponding Laplacian matrix L and normalized Laplacian matrix $\hat{L}$. As we have noted above the heat kernel on the graph is $h_t = \Phi \exp[-\Lambda t]\Phi^T$, where Λ and Φ are the eigenvalue and eigenvector matrices of the normalized Laplacian $\hat{L}$.

We use the heat kernel to map the nodes of the graph into a vector space. Let Y be the $|V| \times |V|$ matrix with the vectors of co-ordinates as columns. The vector of co-ordinates for the node index u is hence the u^{th} column of Y. The co-ordinate matrix is found by performing the Young-Householder decomposition $h_t = Y^T Y$ on the heat-kernel. Since $h_t = \Phi \exp[-\Lambda t]\Phi^T$, $Y = \exp[-\frac{1}{2}\Lambda t]\Phi^T$. Hence, the co-ordinate vector for the node indexed u is

$$y_u = (\exp[-\frac{1}{2}\lambda_1 t]\phi_1(u), \exp[-\frac{1}{2}\lambda_2 t]\phi_2(u), ..., \exp[-\frac{1}{2}\lambda_{|V|} t]\phi_{|V|}(|V|))^T \tag{4.14}$$

The kernel mapping $\mathcal{M} : \mathcal{V} \rightarrow \mathcal{R}^{|\mathcal{V}|}$, embeds each node on the graph in a vector space $\mathcal{R}^{|\mathcal{V}|}$. The heat kernel $h_t = Y^T Y$ can also be viewed as a Gram matrix, i.e. its elements are scalar products of the embedding co-ordinates. Consequently, the kernel mapping of the nodes of the graph is an isometry. The squared Euclidean distance between nodes u and v is given by

$$d_E(u,v)^2 = (y_u - y_v)^T(y_u - y_v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t](\phi_i(u) - \phi_i(v))^2 \tag{4.15}$$

The mean co-ordinate vector for the heat kernel embedding is

$$\hat{y} = \frac{1}{|V|}\exp[-\frac{1}{2}\Lambda t]\Phi^T e \tag{4.16}$$

where $e = (1, 1, ..., 1)^T$ is the all ones vector of length $|V|$. The matrix of centred co-ordinates is

$$Y_C = \exp[-\frac{1}{2}\Lambda t]\Phi^T(I - \frac{1}{|V|}ee^T) = \exp[-\frac{1}{2}\Lambda t]\Phi^T M^T \tag{4.17}$$

where $M^T = (I - \frac{1}{|V|}ee^T)$. The covariance matrix is

$$\sum = \exp[-\frac{1}{2}\Lambda t]\Phi^T M^T M \Phi \exp[-\frac{1}{2}\Lambda t] \tag{4.18}$$

4.4 Characterizing the Embedded Point Distribution

Once the nodes of a graph have been embedded, we can attempt to characterize the structure of the graph by summarizing the distribution of points associated with the nodes. Here we explore the three different ways, the statistical moments, the graph spectral characterization and the sectional curvature, associated with the edge of the graphs.

4.4.1 Statistical Moments

We use spatial moments to characterize the embedded point sets. Moment invariants are properties of connected regions in binary images that are invariant to translation, rotation and scale. They are useful because they define a simply calculated set of region properties that can be used for shape classification and part recognition. For $N \times 2$ size point-sets X the general moment is defined to be

$$\mu_{pq} = \sum_{i=1}^{|N|}\sum_{j=1}^{|N|}(X_{i,1} - \hat{X}_1)^p(X_{j,2} - \hat{X}_2)^q \tag{4.19}$$

where $\hat{X}_k = \frac{1}{|N|}\sum_{i=1}^{|N|} X_{i,k}$, and $\hat{X}_1$ and $\hat{X}_2$ are the co-ordinates of the region's center of gravity. From the raw moment μ_{pq}, we compute the four affine invariants suggested by Flusser and Suk (Flusser and Suk, 1993):

$$I_1 = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4} \tag{4.20}$$

$$I_2 = \frac{\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 + 4\mu_{21}^3\mu_{03} - 3\mu_{21}^2\mu_{12}^2}{\mu_{00}^{10}} \tag{4.21}$$

$$I_3 = \frac{\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu 11(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)}{\mu_{00}^7} \tag{4.22}$$

$$
\begin{aligned}
I_4 = (&\mu_{20}^3\mu_{03}^2 - 6\mu_{20}^2\mu_{11}\mu_{12}\mu_{03} - 6\mu_{20}^2\mu_{02}\mu_{21}\mu_{03} + 9\mu_{20}^2\mu_{02}\mu_{12}^2 \\
&+12\mu_{20}\mu_{11}^2\mu_{21}\mu_{03} + 6\mu_{20}\mu_{11}\mu_{02}\mu_{30}\mu_{03} - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} \\
&-8\mu_{11}^3\mu_{30}\mu_{03} - 6\mu_{20}\mu_{02}^2\mu_{30}\mu_{12} + 9\mu_{20}\mu_{02}^2\mu_{21}^2 \\
&+12\mu_{11}^2\mu_{02}\mu_{30}\mu_{12} - 6\mu_{11}\mu_{02}^2\mu_{30}\mu_{21} + \mu_{02}^3\mu_{30}^2)/\mu_{00}^{11}
\end{aligned}
\tag{4.23}
$$

The invariants I_1, I_2, I_3, I_4 are composed only from central moments up to the third order. Zero order moment μ_{00} is the area of object G. Second order moments express the distribution of ”matter” around the center of gravity. Third order moments express the basic properties of symmetry of object G (Flusser and Suk, 1993).

Thus to compute the statistical moments characterization, we use the two leading elements in the embedded vector for each graph node as the point coordinate to represent the node. Then for the graph node indexed i, the embedded vector of coordinate is chosen as $\vec{x}_i = (X_{i,1}, X_{i,2})^T$. The graph is embedded to a point set of size $|V| \times 2$, where $|V|$ is the number of nodes. We first compute the general moment μ_{pq} then from these general moments we compute the four moment invariants as shown above. The four moment invariants are used to compute the graph feature vector $\vec{B} = (I_1, I_2, I_3, I_4)^T$.

4.4.2 Spectral Characterization

One very simple way to characterize the embedded point-set is to study the properties of the covariance matrix of the point-set generated by the embedding methods. To construct the covariance matrix, we commence by computing the mean coordinate vector. The components of the mean co-ordinate vector are found by averaging the elements in the rows of Y. The mean co-ordinate vector is given by

$$\hat{y} = \frac{1}{|V|} Y e \tag{4.24}$$

where $e = (1, 1, ..., 1)^T$ is the all ones vector of length $|V|$. Subtracting the mean from the embedded co-ordinates, the matrix of centred co-ordinates is

$$Y_C = Y - \frac{1}{|V|} Y e e^T \tag{4.25}$$

where $M^T = (I - \frac{1}{|V|} e e^T)$ is the data centering matrix. The covariance matrix is

$$\sum = Y_C Y_C^T \tag{4.26}$$

The spectral decomposition of the covariance matrix is $\sum = \sum_{i=1}^{n} l_i \vec{u}_i \vec{u}_i^T$, where l_i is the ith eigenvalue and $\vec{u}_i$ is the corresponding eigenvector of the covariance matrix. Our spectral characterization of the graph is based on the vector of N leading covariance eigenvalues $\vec{B} = (l_1, ..., l_N)^T$. We can perform pattern analysis on sets of graphs by applying clustering or dimensionality reduction techniques such as principal component analysis (PCA) to the $\vec{B}$ vectors.

4.4.3 Sectional Curvature

In the previous sections we have introduced the Euclidean distance and geodesic distance deduced from the two embedding methods. From the Euclidean distance and geodesic distance we can compute the sectional curvature on the edges of the graph. In this section we will explore an attractive way of characterizing graphs, using sectional curvature associated with the edge for the purposes of characterizing and clustering graphs. To do this, we can make numerical estimation of the sectional curvature for the edges connected between pairs of nodes.

The sectional curvature is determined by the degree to which the geodesic bends away from the Euclidean chord. Hence for a geodesic on the manifold, the sectional curvature can be estimated easily if the Euclidean and geodesic distances are known. Suppose that the geodesic can be locally approximated by a circle. Let the geodesic distance between the pair of points u and v be $d_G(u,v)$ and the corresponding Euclidean distance be $d_E(u,v)$. Further let the radius of curvature of the approximating circle be $r_s(u,v)$ and suppose that the tangent vector to the manifold undergoes a change in direction of $\theta_{u,v}$ as we move along a connecting circle between the two points. We show an illustration of the above in Figure 4.1. In terms of the angle $\theta_{u,v}$, the geodesic distance, i.e. the distance traversed

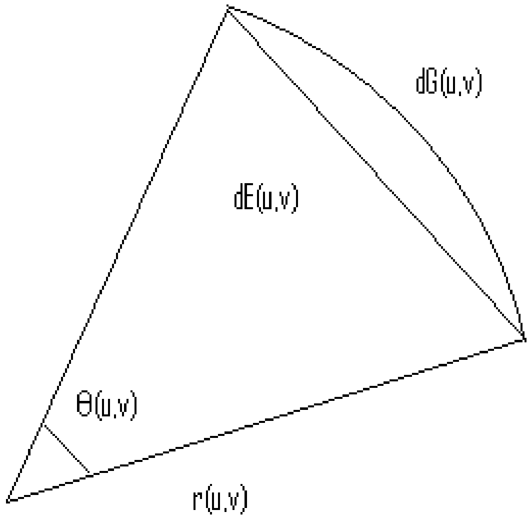

Figure 4.1: Illustration of relationship between the geodesic distance, Euclidean distances and the sectional curvature.

along the circular arc, is $d_G(u,v) = r_s(u,v)\theta_{u,v}$, and as a result we find that $\theta_{u,v} = d_G(u,v)/r_s(u,v)$. The Euclidean distance, on the other hand, is given by $d_E(u,v) = r_s(u,v)\sin\theta_{u,v}$, and can be approximated using the MacLaurin series

$$d_E(u,v) = r_s(u,v)\{\theta_{u,v} - \frac{1}{6}\theta_{u,v}^3 + ...\} \tag{4.27}$$

Substituting for $\theta_{u,v}$ obtained from the geodesic distance, we have

$$d_E(u,v) = d_G(u,v) - \frac{d_G(u,v)^3}{6r_s^2(u,v)} \tag{4.28}$$

Solving the above equation for the radius of curvature, the sectional curvature of the geodesic connecting the nodes u and v is approximately

$$k_s(u,v) = \frac{1}{r_s(u,v)} = \frac{\sqrt{6}(d_G(u,v) - d_E(u,v))^{\frac{1}{2}}}{d_G(u,v)^{\frac{3}{2}}} \tag{4.29}$$

Since for an edge of the graph, we find that $d_G(u,v) = 1$, the squared sectional curvature associated with an embedded edge is $k_s^2(u,v) = 6(1-d_E(u,v))$. As a result we can construct the squared sectional curvature matrix.

$$k_s^2 = 6\{A + \ln[(4\pi t)^{2nt}\Phi\exp[-4\Lambda t^2]\Phi^T]\} \tag{4.30}$$

To characterize the geometry of the graph embedding we construct a histogram of sectional curvatures. The sectional curvatures are assigned to M bins and the normalized contents of the jth bin is denoted by $h(j)$. The feature vector for the graph is constructed from the normalized bin-contents and $B = (h(1), h(2), ...h(N))^T$. Our aim is to explore the structure of a set of graphs with pattern vectors B_k, $k = 1, ..., M$ extracted using sectional curvature histograms.

4.5 Experiments

In this section of the chapter, we provide some experimental evaluation of the graph embedding methods for graph clustering. We will compare the parametric distance embedding and the heat kernel embedding. We will use both the moments method and the spectral method to characterize the embedded point-sets from the two embedding methods. We will also demonstrate that the sectional curvature deduced from the Euclidean distance and the geodesic distance from the graph heat kernel can be used as a method of characterizing graphs.

4.5.1 Experiments on Parametric Distance Embedding

In this subsection we show results from the parametric distance embedding. We applied our method to images from the COIL database for image clustering. Figure 3.4 shows some example images. In this experiment, we first use two ways to characterize the embedded point-sets from the graphs. The first of these is the moments characterization and the second is the spectral characterization of the covariance matrix of the embedded point-sets. We commence by investigating the behavior of the moments extracted from the embedded points. By doing so we need to keep the embedded coordinate sets to be $|V| \times 2$ size point-sets, where $|V|$ is the graph node number. For each node u we represent the embedded coordinate vector, i.e. as $\vec{x}_i = (X_{i,1}, X_{i,2})^T$, the first two elements from the coordinate vector for the corresponding node. We then continue by constructing the feature vector (I_1, I_2, I_3, I_4) and apply the PCA on the feature vectors. In Figure 4.2 we plot the four moments as a function of the time parameter t. For this experiment, we have used the duck graph in the top left in Figure 3.4 for our experiment graph. The main effect to note here is that as the time parameter increases then the four moments become indistinguishable. In Figure

4.3 we plot the four moments separately as a function of the view number for the images of the eight objects studied in the COIL database. From the top-left, and clockwise, the sequence shows the first, second, third and fourth moments respectively. The individual moments appear relatively stable with view number when t equals 0.01. It is clear that although the individual moments could not be used to distinguish the objects, when combined they are sufficient to do so.

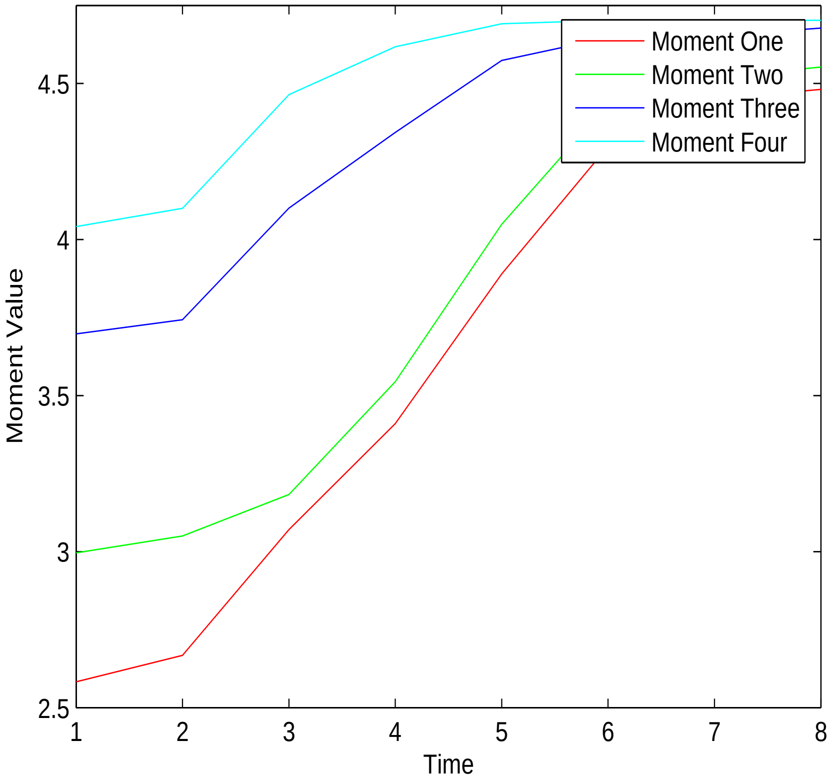

Figure 4.2: Moments as a function of t for a graph from the COIL database for the parametric distance embedding.

Based on the study of the moments, it appears that they may provide the basis for a stable clustering of the graphs. We have therefore performed principal component analysis on vectors whose components are the different moments.

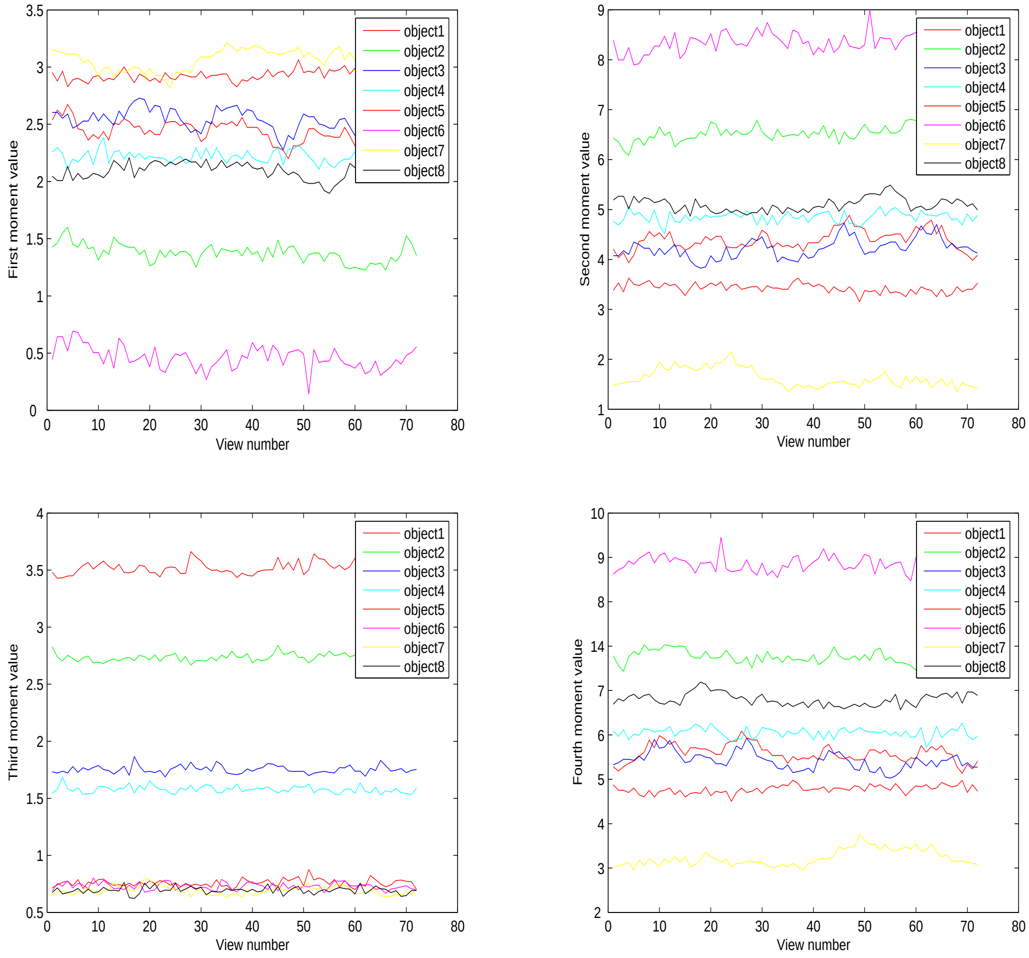

Figure 4.3: Individual moment(moment one, moment two, moment three and moment four) for the eight objects from COIL database as a function of view number.

The data has been projected onto the space spanned by the leading three eigenvectors for the moment vectors. We then investigate the effect of varying the time parameter t. In Figure 4.4 we show the effect on the graph embeddings when we vary t from 0.01 to 1000. From left-to-right and top-to-bottom, we show the clustering results obtained when t equals 0.01,0.1, 1, 10, 100 and 1000. In the figures the different views of the same object are displayed as differently colored symbols. In Figures 4.6 we show corresponding plots for the pairwise distances for the embedded graph nodes. In Figures 4.5 we show the spectral characterization for the embedded point-sets by choosing $\vec{B} = (l_1, l_2, ..., l_6)^T$, which are eigenvalues of the covariance matrix of the embedded point-set. We repeat the clustering results obtained when t equals 0.01, 0.1, 1, 10, 100 and 1000 and Figure 4.7 are the corresponding plots for the pairwise distances figures. The main feature to note from these plots is that by choosing the proper t values, the method can be used for the purposes of graph clustering. As the value of t increases, then so the clusters corresponding to the different objects become overlapped.

We will continue to show how to use the sectional curvature with the parametric distance embedding to cluster images from the COIL database 3.4. In Figure 4.8 we show example histograms for the twenty views for the eight objects in the COIL database. Here the histograms are stacked behind each other, and are ordered by increasing view number. There are a number of conclusions that can be drawn from the histograms. First, the histograms for the same object are relatively stable with view number. Second, the histograms for the different objects have different shapes. The plots shown were obtained with $t = 0.01$. The results of applying PCA to the vectors of histogram bin contents are shown in

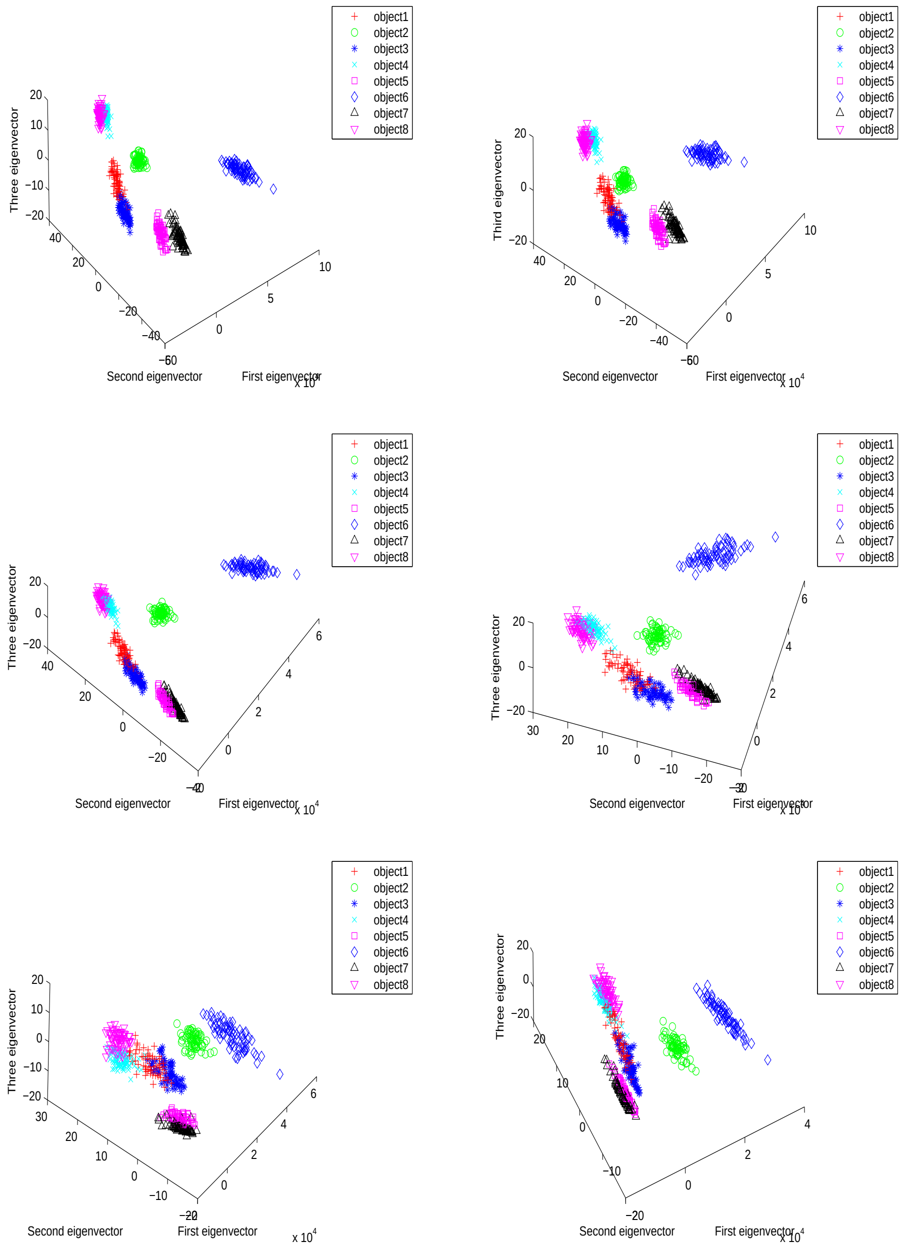

Figure 4.4: Parametric distance embedding varying with t – moments characterization (from left to right, top to bottom, the results obtained when t equals $0.01, 0.1, 1, 10, 100$ and 1000 respectively).

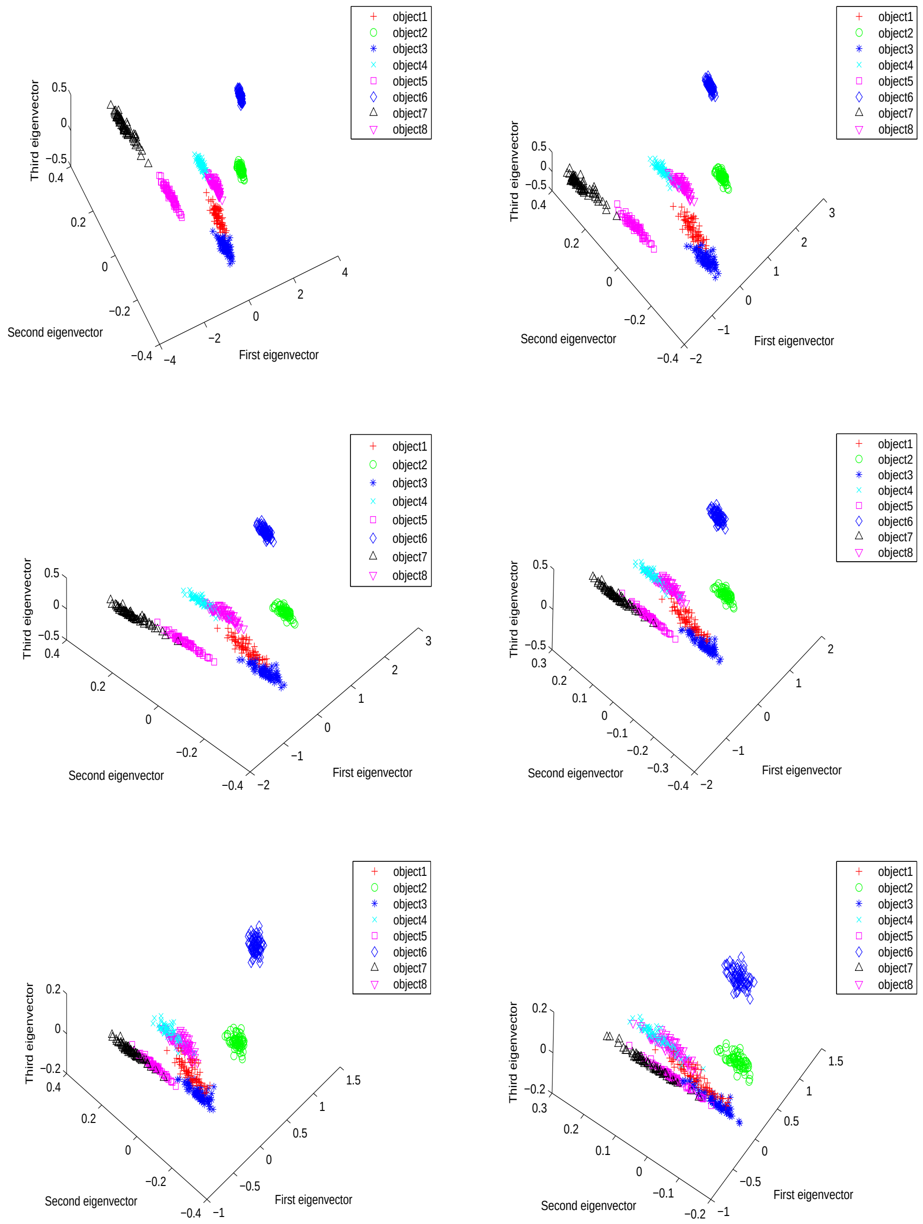

Figure 4.5: Parametric distance embedding varying with t – spectral characterization (from left to right, top to bottom, the results obtained when t equals $0.01, 0.1, 1, 10, 100$ and 1000 respectively).

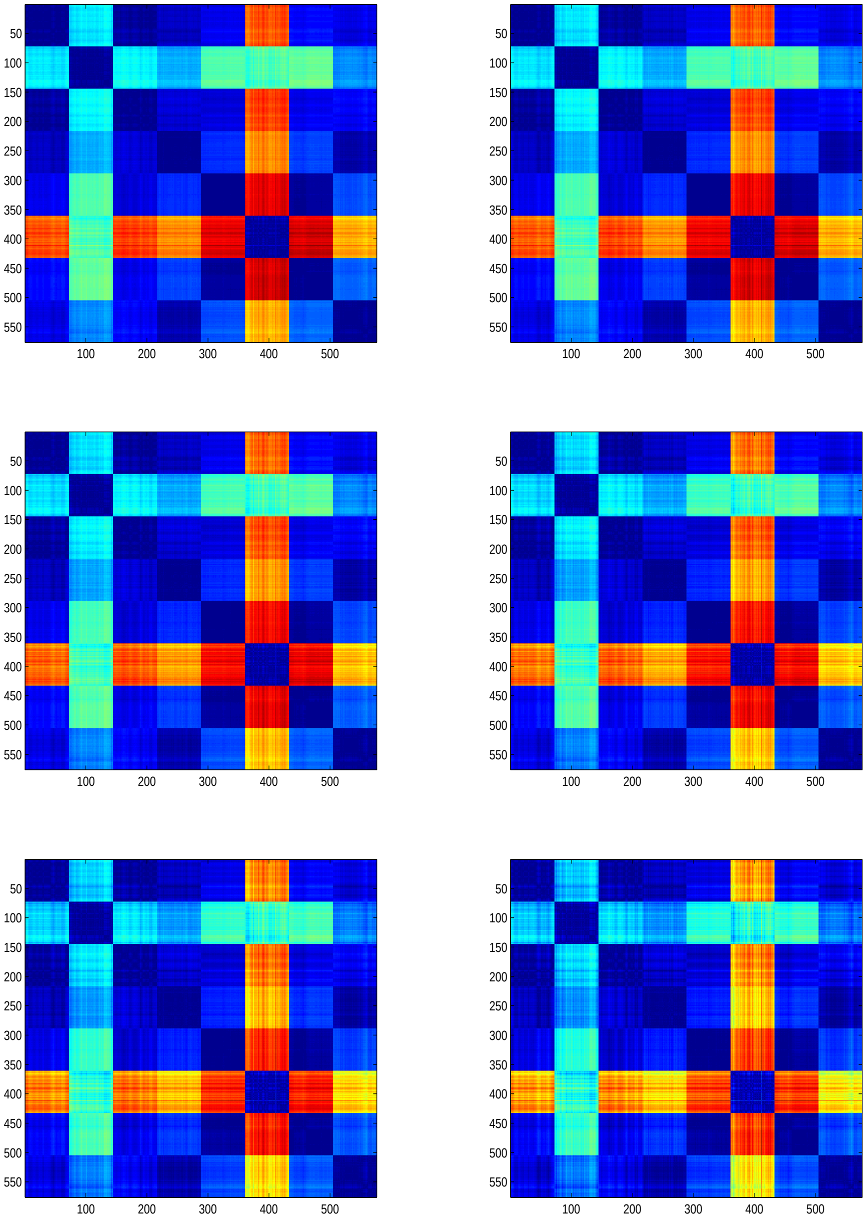

Figure 4.6: Parametric distance embedding with moments characterization distance matrices varying with t (from left to right, top to bottom, the results obtained when t equals $0.01, 0.1, 1, 10, 100$ and 1000 respectively).

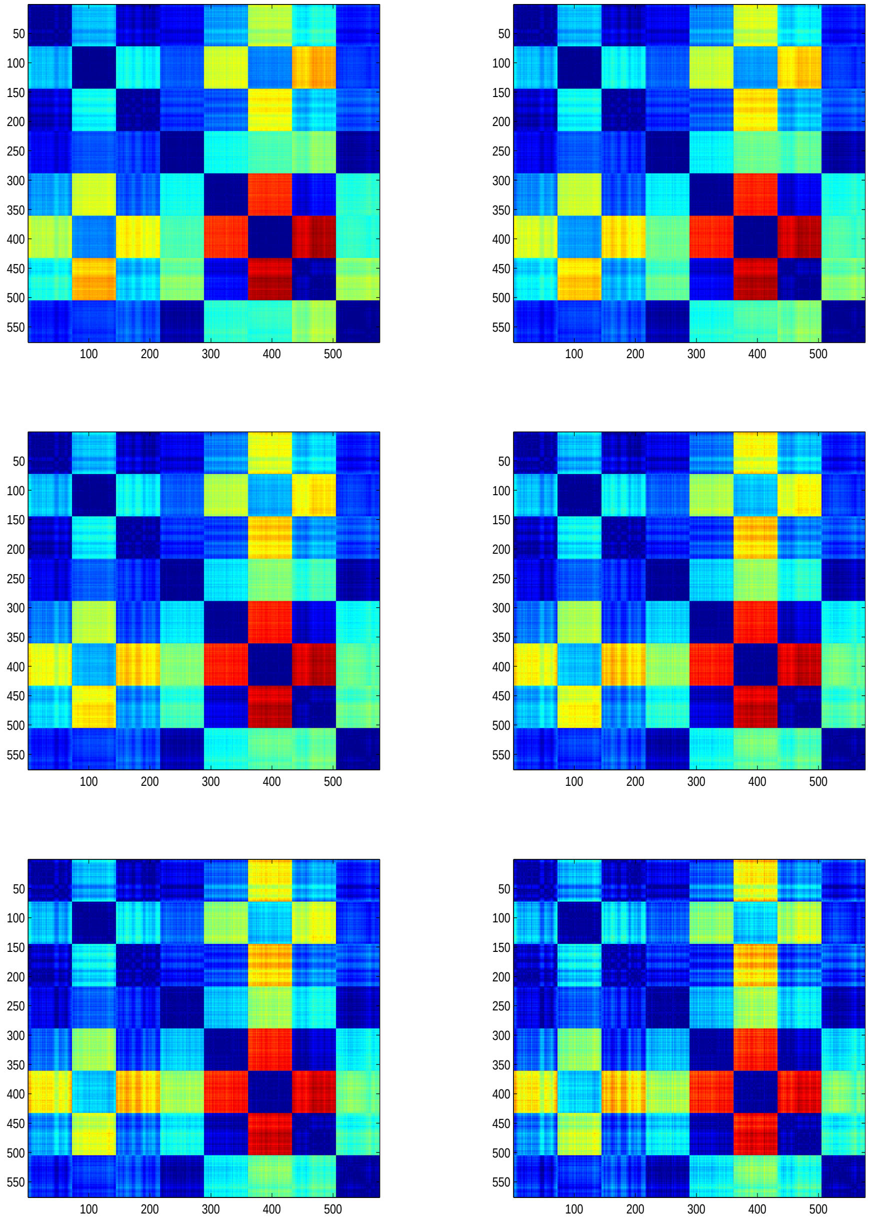

Figure 4.7: Parametric distance embedding with spectral characterization distance matrices varying with t (from left to right, top to bottom, the results obtained when t equals $0.01, 0.1, 1, 10, 100$ and 1000 respectively).

the right hand panel of Figure 4.9 for different values of t. We obtain reasonably well defined clusters of objects.

For comparison, Figure 4.10 shows the corresponding result when spectral clustering is used. The main qualitative feature is that the different views of the eight objects overlap more than when the parametric distance embedding method is used with a low value of t.

To investigate the behavior of the two methods in a more quantitative way, we have plotted the Rand index (Rand, 1971) for the different objects. The solid curve in Figure 4.11 shows the Rand index as a function of t. The performance of the method drops off once t exceeds 10. The performance of the spectral method is significantly poorer, which is 0.78, than that for the heat kernel method for small values of t.

4.5.2 Experiment on Heat Kernel Embedding based Graph Clustering

In this section we will apply our heat kernel embedding method to images from the COIL database. We use the moments method described above to characterize the heat kernel embedding. The embedded coordinate vector for each node of the graph has two components. As a result the node u has coordinate vector $\vec{x}_u = (\exp[-\frac{1}{2}\lambda_1 t]\phi_1(u), \exp[-\frac{1}{2}\lambda_2]\phi_2(u))^T$, which is from the leading two columns of the coordinate matrix Y. In Figure 4.12 we repeat the experiment for the four moments as a function of the time parameter t for the heat kernel embedding. For this experiment, we use the same graph as we have used for the parametric distance embedding experiment in Figure 4.2. The main effect to note here is that for small values of t the moments can provide a useful graph characterization. In Figure 4.13, we show the moments characterization for the embedded coordinates. We repeat the clustering results obtained in Figure 4.13

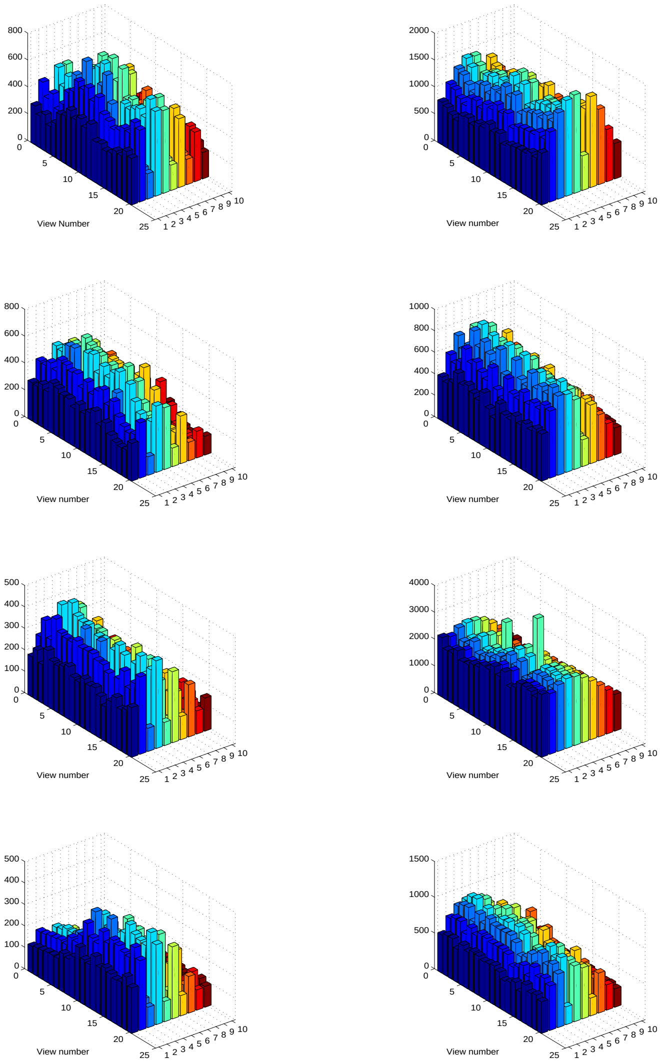

Figure 4.8: 3-D view of the histogram of the sectional curvature matrices–parametric distance embedding.

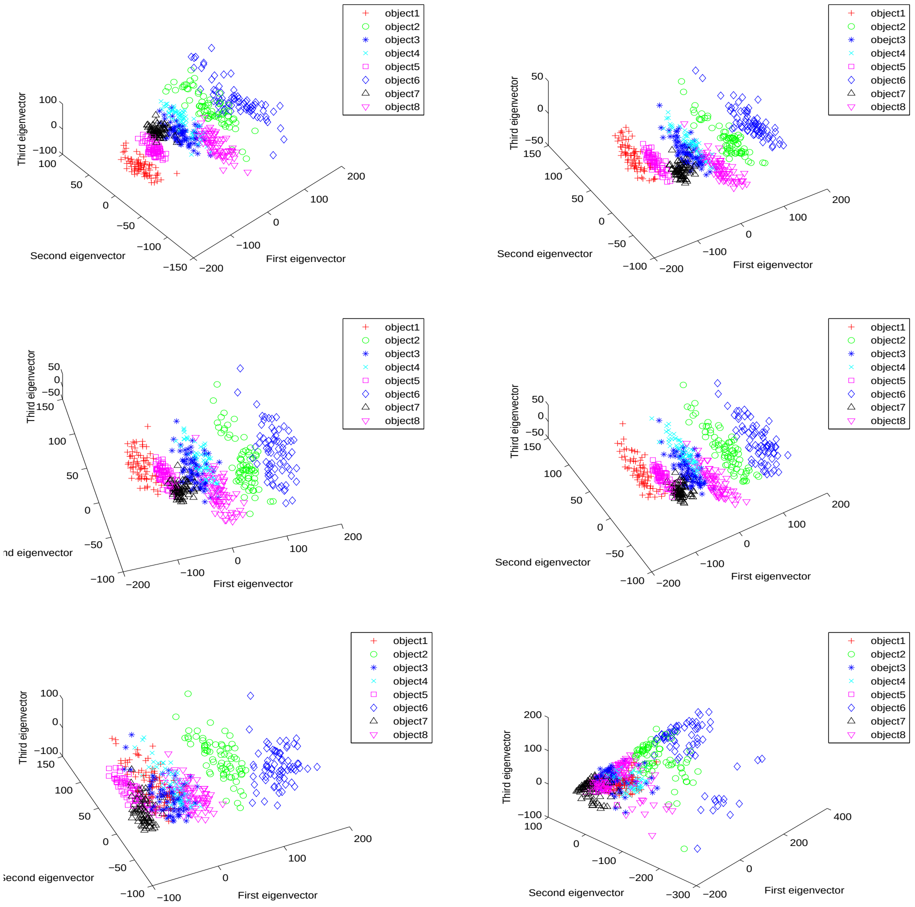

Figure 4.9: Parametric distance embedding with sectional curvature clustering by varying t (from left to right, top to bottom, the results obtained when t equals $0.01, 0.1, 1, 10, 100$ and 1000 respectively).

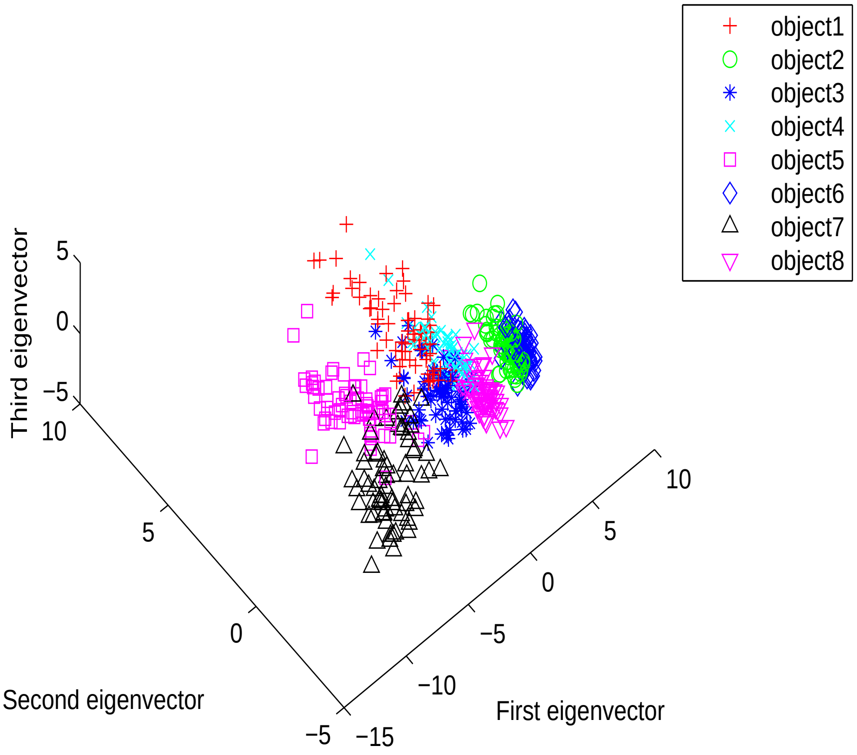

Figure 4.10: Spectral clustering result.

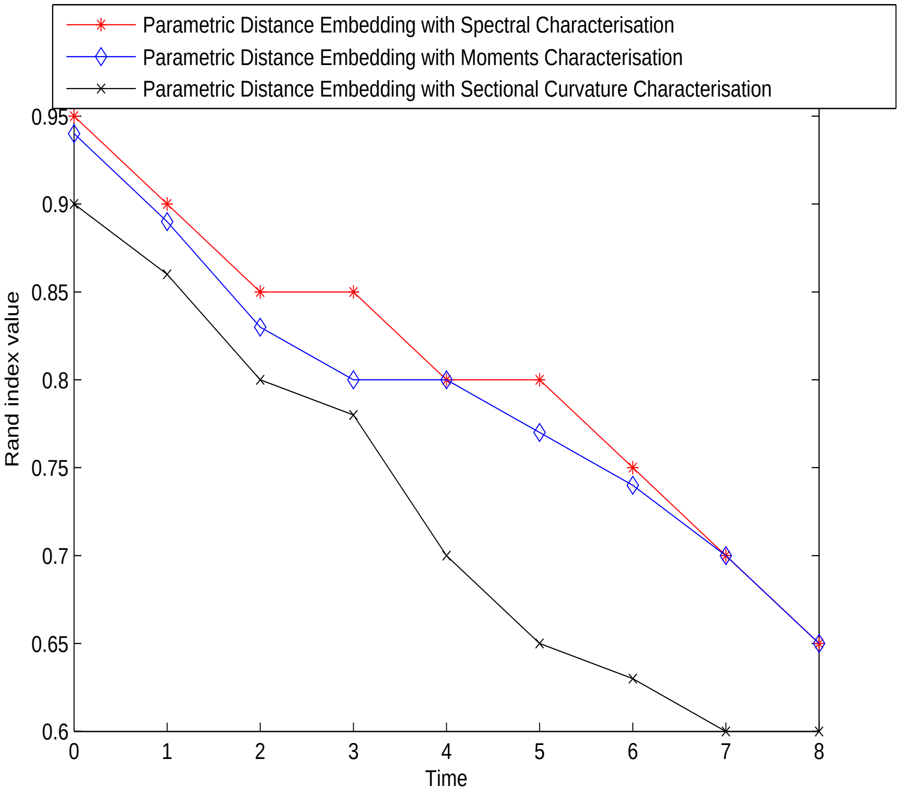

Figure 4.11: Rand index for parametric distance embedding methods.

when t equals 0.5,1, 5, 10, 100 and 1000. The different colors in the plot distinguish the different objects. The object clusters evolve in an interesting way as t increases. Initially, they are very compact but elongated, and distinct. At large values of t they are more dispersed and overlapping.

We can also construct the feature vector by taking the eigenvalues of the covariance matrix of the embedded point-set. Then we perform PCA on the feature vectors. In Figure 4.14 we show the clustering result where the parameter t varies from 0.5 to 1000. In Figures 4.16 and 4.15 we show the corresponding plots for the pairwise distances for the embedded graph nodes with heat kernel eigenvalues and the moments characterization respectively. We continue to show the clustering result by using sectional curvature with heat kernel embedding for graph clustering. In Figure 4.17, we repeat the histograms from twenty views for the eight objects in the COIL database. Figure 4.18 shows the result of applying PCA to the vectors of histogram bin contents.

Compared with the traditional spectral clustering method in Figure 4.10, the heat kernel embedding method gives a better result when proper values of t are chosen. We have plotted the Rand index for the different characterization heat kernel embedding. The results of the comparison are shown in Figure 4.19. The curve in the plot shows the Rand index as a function of t. For low values of t the performance is more than 95 $\%$. When the spectrum of the normalized Laplacian eigenvalues is used then the Rand index is 0.78, which is considerably lower than the result obtained with the heat kernel embedding method.

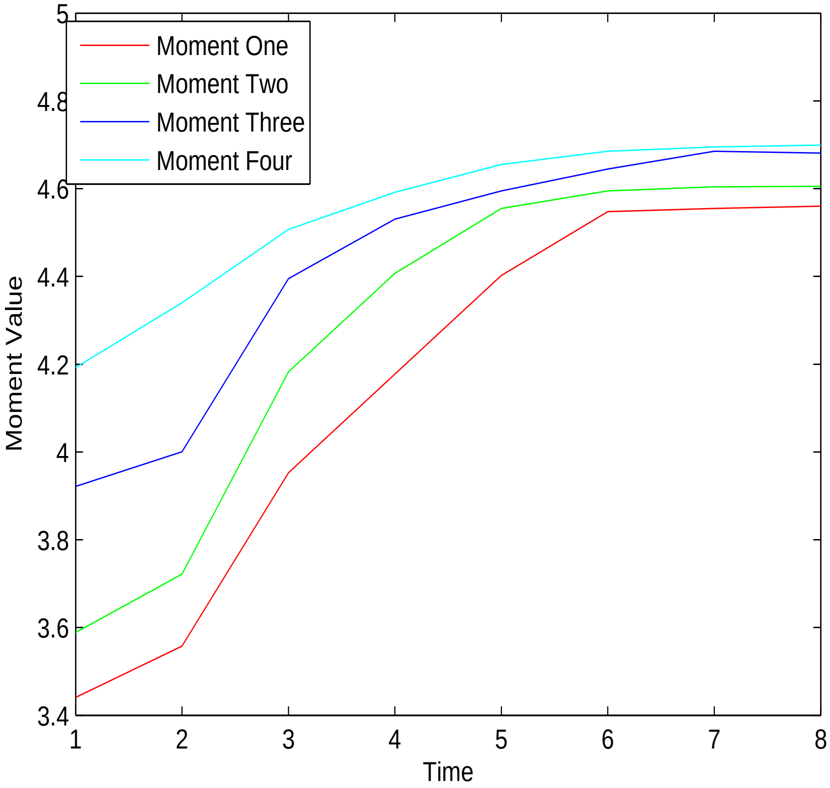

Figure 4.12: Moments as a function of the t parameter for a graph from the COIL database for the heat kernel embedding.

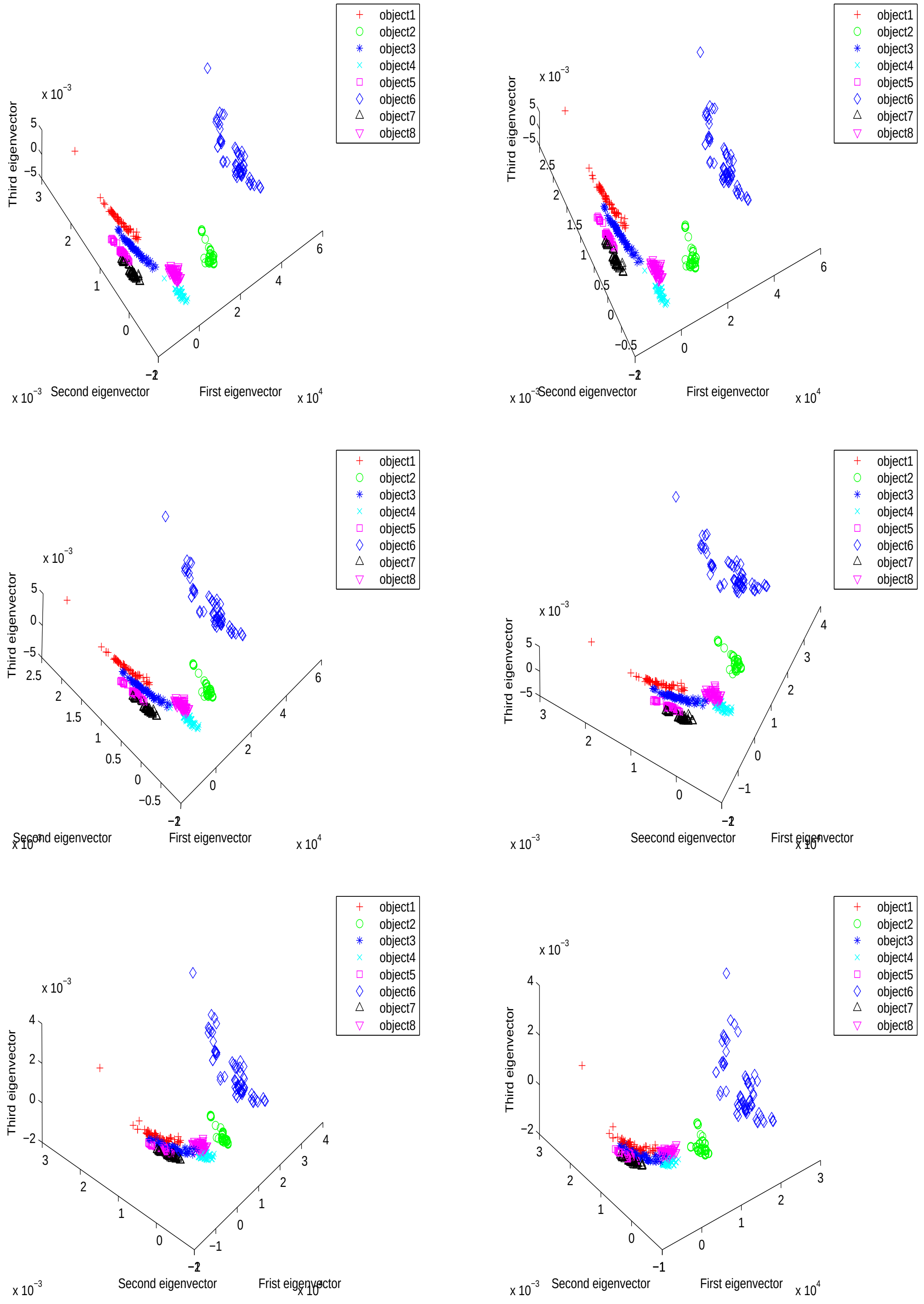

Figure 4.13: Heat kernel embedding varying with t – moments characterization (from left to right, top to bottom, the results obtained when t equals $0.5, 1, 5, 10, 100$ and 1000 respectively).

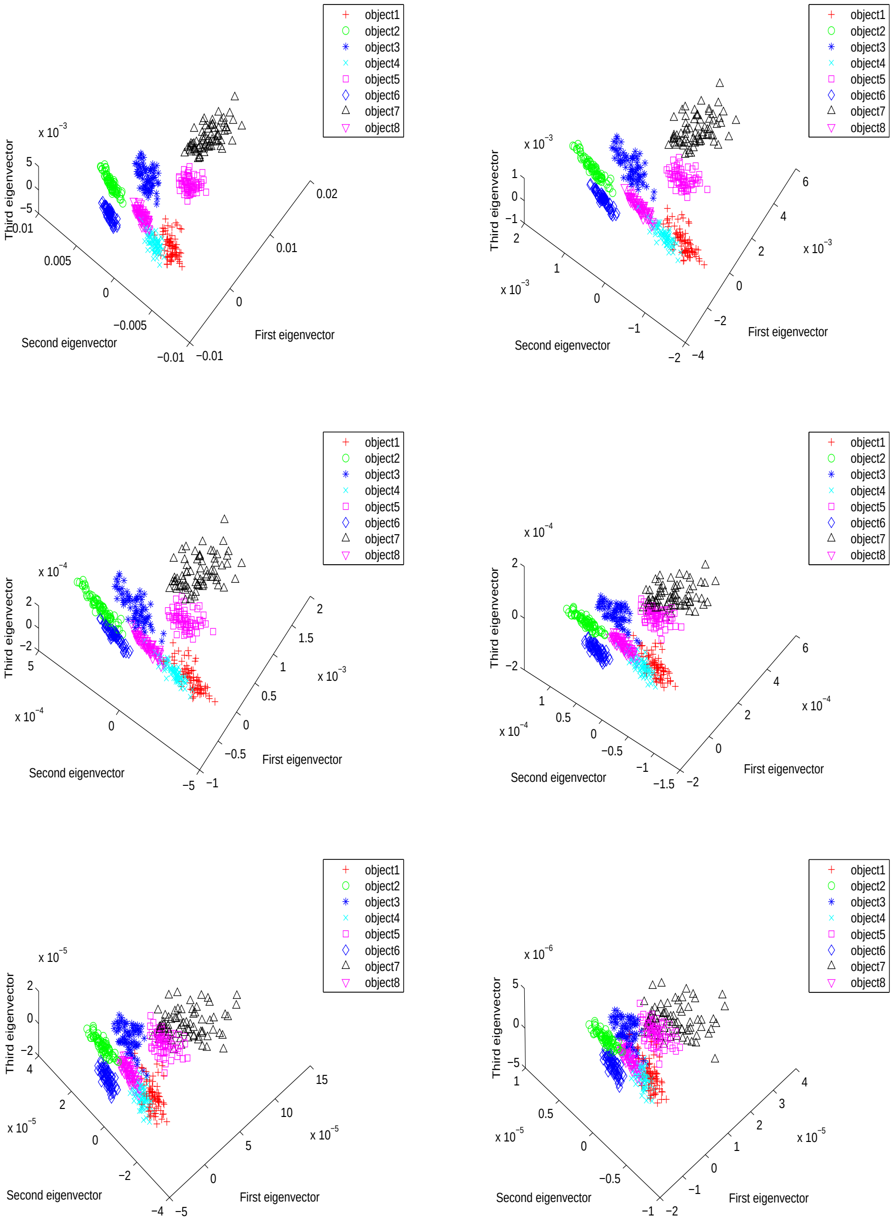

Figure 4.14: Heat kernel embedding varying with t parameter – spectral characterization (from left to right, top to bottom, the results obtained when t equals $0.5, 1, 5, 10, 100$ and 1000 respectively).

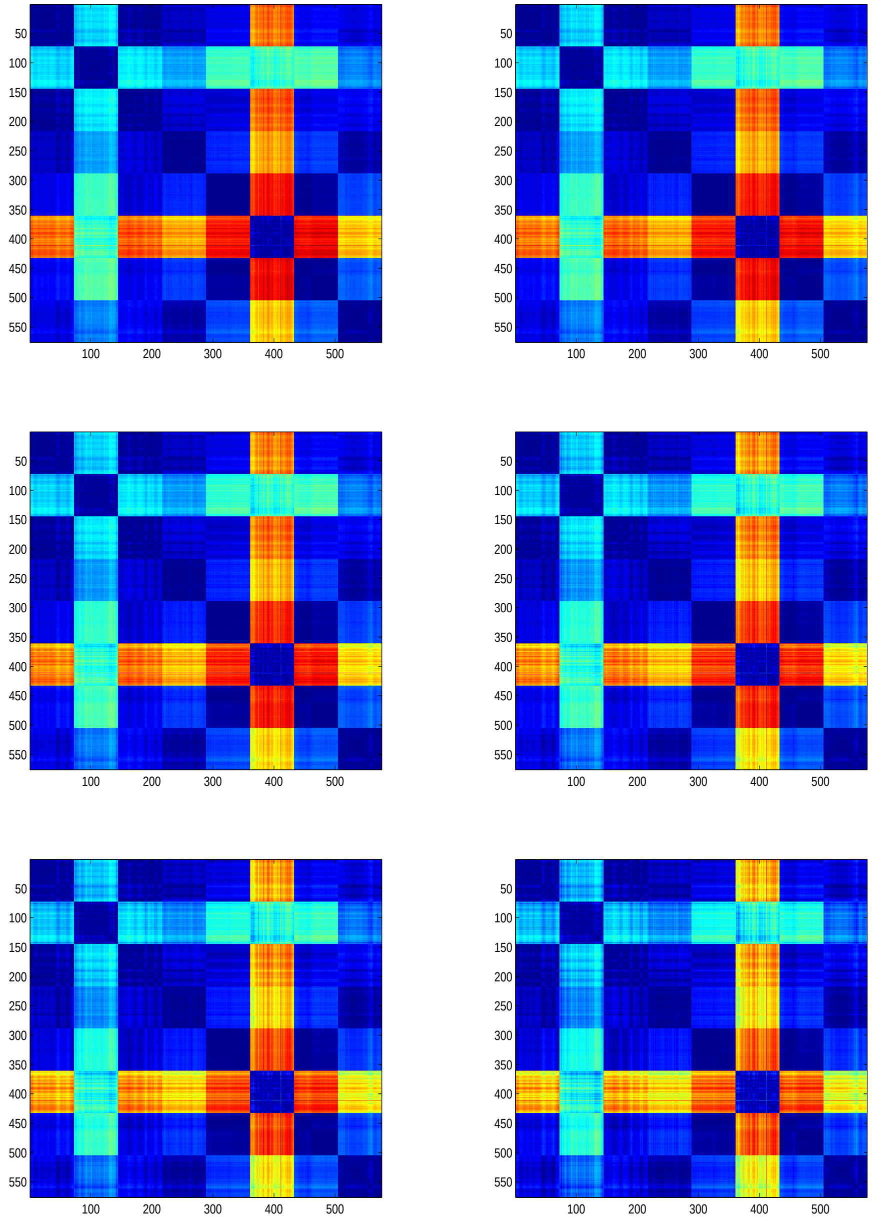

Figure 4.15: Heat kernel embedding with moments characterization distance matrices varying with t parameter (from left to right, top to bottom, the results obtained when t equals $0.5, 1, 5, 10, 100$ and 1000 respectively).

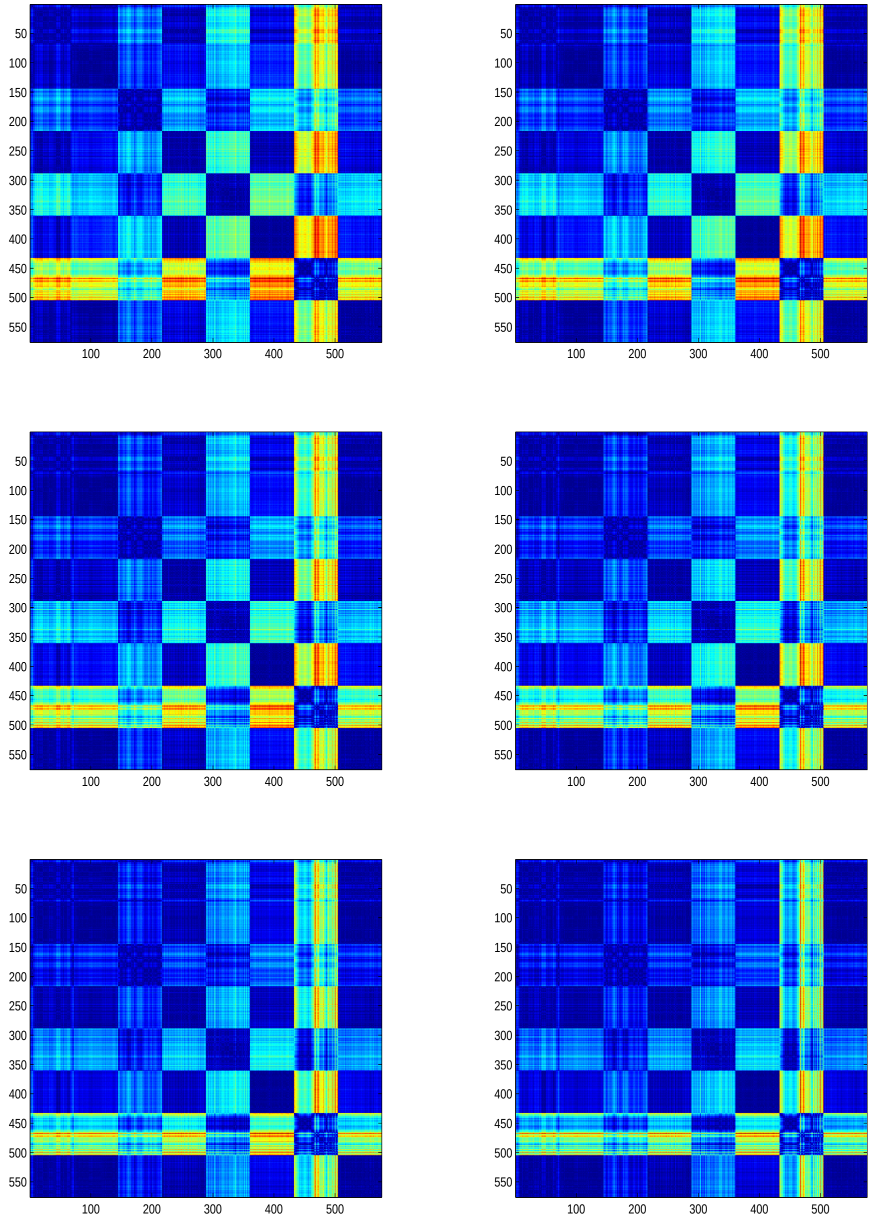

Figure 4.16: Heat kernel embedding with spectral characterization distance matrices varying with t (from left to right, top to bottom, the results obtained when t equals $0.5, 1, 5, 10, 100$ and 1000 respectively).

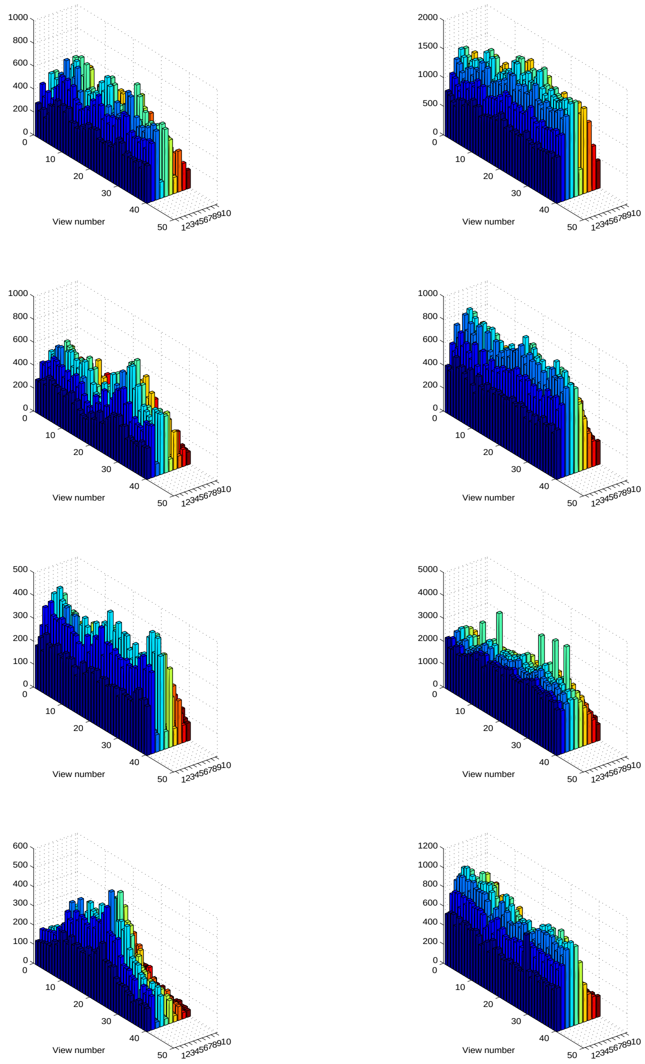

Figure 4.17: 3-D view of the histogram of the sectional curvature matrices – heat kernel embedding.

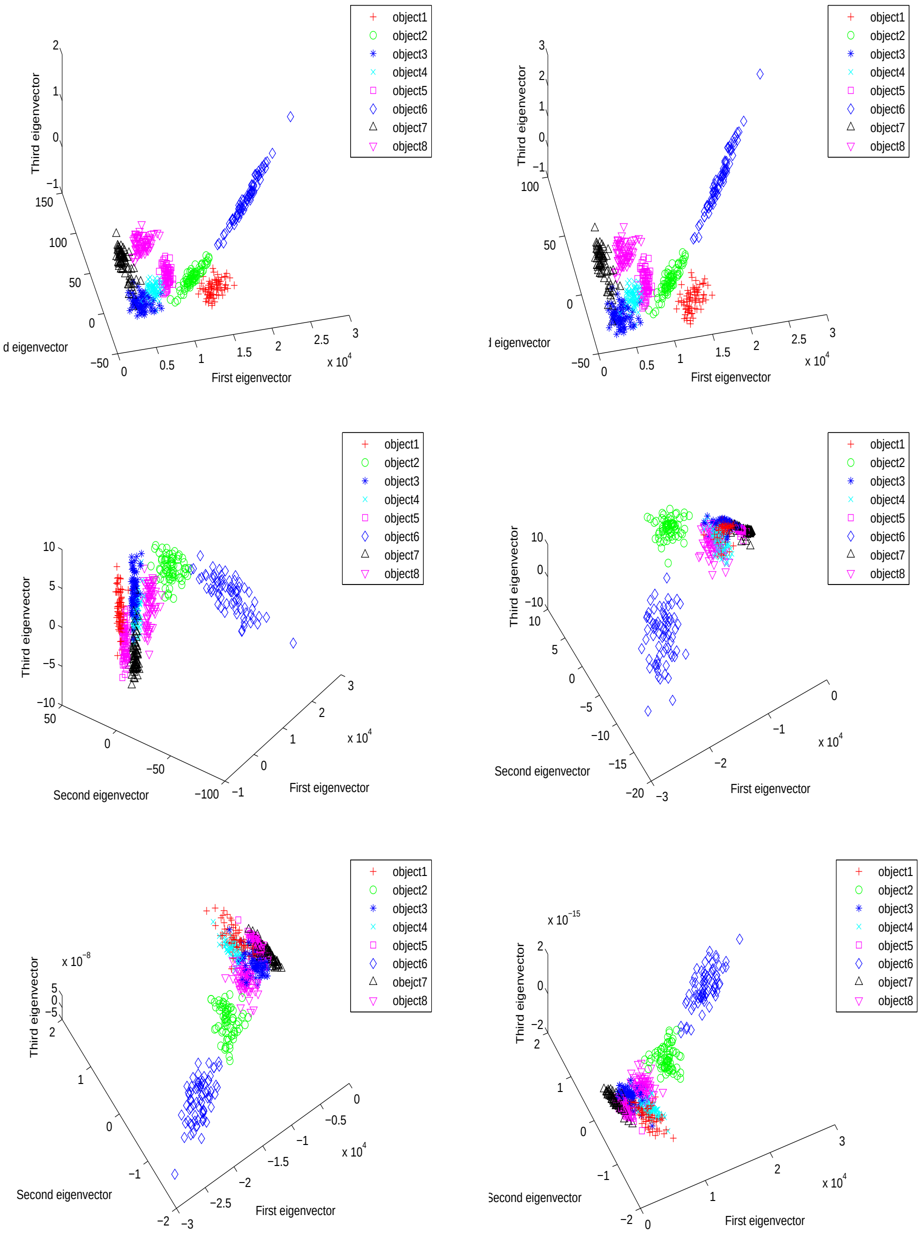

Figure 4.18: Heat kernel embedding with sectional curvature characterization by varying t (from left to right, top to bottom, the results obtained when t equals $0.5, 1, 5, 10, 100$ and 1000 respectively).

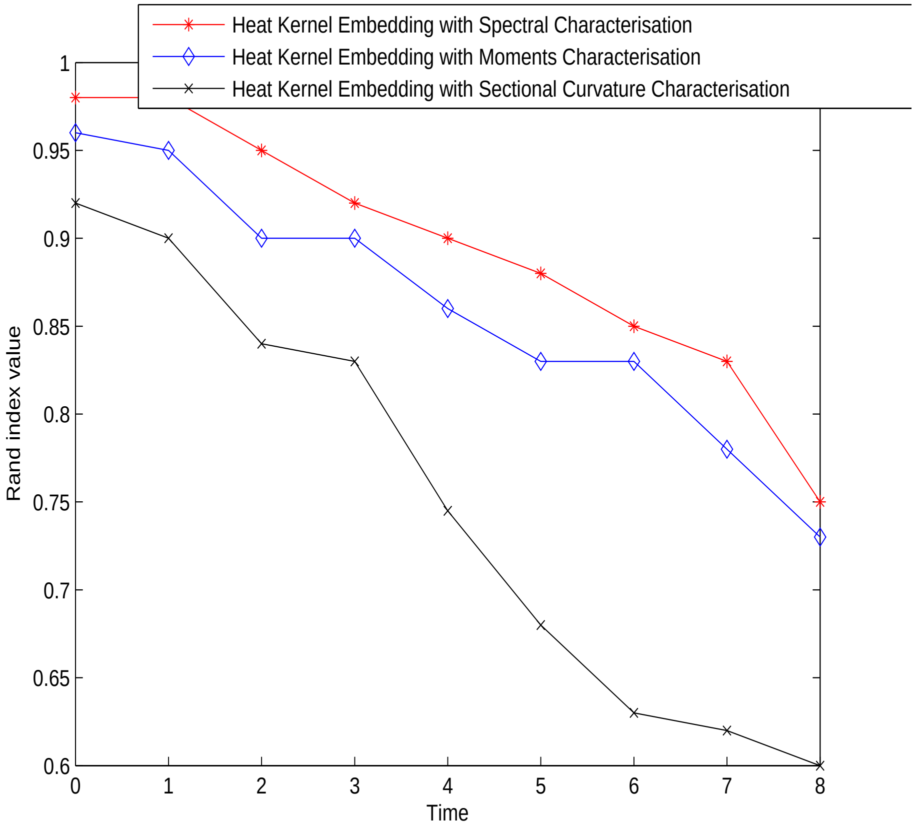

Figure 4.19: Rand index for different characterizations of the heat kernel embedding.

4.6 Conclusion

In this chapter we have introduced two ways for graph embedding and related algorithms for graph characterization using the embeddings. In the first section we have explored how the use of heat kernels can lead to a measure of Euclidean distance that can be used for the purpose of embedding graphs in low dimensional Euclidean spaces. The distance measure is found by equating the spectral and Gaussian forms of the heat kernel. We show how MDS can be used for embedding the distance by satisfying low distortion requirements.

In the second section we have shown a second method for graph embedding, which we refer to as the heat kernel embedding. We have explored how the use of the heat kernel can lead to a useful characterization of the structure of a graph. Our method is based on preforming Young-Householder decomposition on the heat kernel. This allows graph nodes to be mapped to points in a high dimensional space.

After the graph embedding process, we can pose the problem of clustering as that of characterizing the embedded point-set distribution. We have introduced three ways for the point-sets characterization. These are moments characterization, spectral characterization and sectional curvature characterization. For the last of these methods the sectional curvature associated with the edge of the graph is computed by using the difference between the Euclidean and geodesic distances deduced from the heat kernel analysis.

In the experimental section, results are provided for the two heat kernel based graph embedding methods and the three different characterization methods for graph clustering. The database used in this section is the COIL database. From the experiment results, both the parametric distance and heat kernel embedding can provide relatively good results for graph clustering. By choosing proper t values the Rand index value for graph clustering by using graph embedding methods can reach 0.95, which is much higher than the traditional spectral method. In both embedding methods the Rand index value for spectral characterization is higher than the moments characterization. This is due to the fact that the moments characterization only takes the two leading elements in the embedded vector for each graph node.

Chapter 5

Generative Model for Graph Structure

5.1 Introduction

In this chapter we extend the heat kernel embedding method by using a point distribution model to construct a generative model for variance in graph structure. The outline for this chapter is as follows.

In the first part we will investigate whether methods from spectral graph theory can be combined with the heat kernel embedding to develop effective tools for graph structure matching. In the previous chapter we showed that the heat kernel embedding can be used for graph clustering. The embedded point position vectors for the nodes of a graph contain geometry information that can be used for graph characterization. Our first aim is to explore whether the spectral point pattern matching techniques can be used to find correspondence between embedded nodes. We proceed as follows. For the two graphs to be matched, we construct the heat kernel matrix. Then we perform heat kernel embedding on the heat kernel matrices. We map two graphs to two vector sets. Once embedded in the vector space, nodes of the graphs are matched by using point alignment methods. To do this we develop a variant of the Scott and Longuett-Higgins (Scott and Longuett-Higgins, 1991) algorithm. Our method overcomes the problems of structural differences in the graphs. Our experimental part compares the method with some alternatives described in the literature (Umeyama, 1988; Scott and Longuett-Higgins, 1990; Shapiro and Brady, 1992).

Once we have solved the matching correspondence problem, we draw on ideas from linear deformable models to construct a simple and explicit generative model for graph structure. One of the problems that limits the use of the structural clustering methods (Lozano and Escolano, 2003; Bunke et al., 2003; Bunke, 1999; Bunke and Vento, 1999) 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 proved to be a powerful way of capturing the variations in sets of landmark points for 2D and 3D objects (Cootes et al., 1995).

Our second contribution is to use the heat kernel embedding to construct a generative model for graph structure. Using the heat kernel embedding described in section 4.3, we map the nodes of a graph to point position vectors. 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 spectral 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 and how to fit the model to potentially noisy graphs to reconstruct the Laplacian matrix. We illustrate the utility of the resulting method for shape analysis. Here we perform experiments on the Caltech-Oxford and COIL databases, and illustrate how the model can be used both to construct pattern spaces for sets of graphs and to cluster graphs.

5.2 Graph Matching using Manifold Embedding

In the previous chapter we described heat kernel embedding, which can be used to embed the nodes of a graph into a high dimensional vector space. In this section we will show how the heat kernel embedding method combined with the spectral point-set alignment can be used for structure based graph matching. For two graphs G_I and G_J, we construct the heat kernel matrices h_I and h_J separately based on the definition:

$$h_t(u,v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i(u)\phi_i(v) \tag{5.1}$$

We then perform the Young-Householder decomposition on the two kernel matrices, i.e. $h_I = Y_I^T Y_I$ and $h_J = Y_J^T Y_J$, to give two embedded co-ordinate matrices Y_I and Y_J with the vectors of co-ordinates as columns. We present a spectral method to align the two co-ordinates matrices Y_I and Y_J for the purpose of structure matching.

5.2.1 Singular Value Decomposition for Point-sets Alignment

After we have the Y_I and Y_J from the embedding process, we follow a method similar to that of Scott and Longuett-Higgins for point set matching. We first transform the vectors in the co-ordinates matrices to the same size. Suppose that Y_I and Y_J are $m \times m$ and $n \times n$ size respectively, we choose the $max(m, n)$

which is the max number in m and n as the extended vector length. For the co-ordinate matrix whose vector length is smaller than $max(m, n)$ we just extend the length of the vector by adding the 0 elements to the end. The algorithm uses the distances between two graph nodes to compute an affinity matrix. Let y_I^i be the ith column vector of the co-ordinate matrix Y_I, i.e. the co-ordinates of the node $i \in V_I$. For the node i of the sample graph G_I and the node j of the reference graph G_J the affinity matrix element is

$$R_{I,J}(i,j) = \exp[-\frac{1}{\sigma^2}(y_I^i - y_J^j)^T(y_I^i - y_J^j)]$$

where σ is a scaling parameter. In the paper (Scott and Longuett-Higgins, 1990), they treat σ as an appropriate unit of distance and set to be 10, so here we choose σ equal to 10.

If $R_{I,J}$ is a positive definite $|V_I| \times |V_J|$ matrix, then the $|V_I| \times |V_J|$ orthogonal matrix $R^*_{I,J}$ that maximizes the quantity $Tr[R_{I,J}(R^*_{I,J})^T]$ may be found by performing singular value decomposition. To do this we perform the matrix factorization $R_{I,J} = V\Delta U^T$, where V is a $|V_I| \times |V_I|$ orthogonal matrix, U is a $|V_J| \times |V_J|$ orthogonal matrix and Δ is a $|V_I| \times |V_J|$ matrix whose off-diagonal elements $\Delta_{i,j} = 0$ if $i \neq j$ and whose "diagonal" elements $\Delta_{i,i}$ are non-zero. Suppose that E is the matrix obtained from Δ by making the diagonal elements $\Delta_{i,i}$ unity. The matrix $R^*_{I,J}$ which maximizes $Tr[R_{I,J}(R^*_{I,J})^T]$ is $R^*_{I,J} = VEU^T$. The element $R^*_{I,J}(i,j)$ indicates the strength of association between the node $i \in V_I$ in the graph G_I and the node $j \in V_J$ in the reference graph. The rows of $R^*_{I,J}$, index the nodes in the graph G_I, and the columns index the nodes of the reference graph G_J. If $R^*_{I,J}(i,j)$ is both the largest element in row i and the largest element in column j then we regard these nodes as being in one to one correspondence with one another. We record the state of correspondence using the matrix $C_{I,J}$. If the pair of nodes (i,j) satisfies the row and column correspondence condition, then we set $C_{I,J}(i,j) = 1$, otherwise $C_{I,J}(i,j) = 0$ The $C_{I,J}(i,j) = 0$ is a $m \times n$ size matrix.

5.3 A Generative Model for Graph Structure

We perform heat kernel embedding on the graph heat kernel to map the graph nodes in a high dimensional space. In this section we continue to construct a linear deformable model to characterize the graph structure. This is done by constructing the mean and covariance matrix for the corresponding mapped point positions. In the experiment part we will show the results on tracking and clustering by using the linear deformable model.

5.3.1 Generative Model

Our aim is to construct a generative model that can be used to represent the statistical variations in a sample of graphs. Let the sample be $T = \{G_1, G_2, ..., G_T\}$ where the kth graph $G_k = (V_k, E_k)$ has node set V_k and edge set E_k. The result of performing heat kernel embedding of the nodes of the kth graph is a matrix of co-ordinates Y_k.

Our aim is to construct a generative 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 may extend the mapped vector to the same size by adding 0 to the end of the smaller size vector as we have described in the previous section.

To construct the generative model, 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 samples with the largest number of nodes. If there are more than one graph which have the largest number of nodes in the graph set, we can simply randomly choose one as the reference graph. This graph has index $k^* = \arg\max_{G_k \in T} |V_k|$. To locate the correspondences between the nodes of each sample graph G_k and those of the reference graph G_k^*, we use the matching algorithm we described above. We first transform the vectors in the co-ordinates matrices to the same size. Then we perform the SVD point-set spectral alignment algorithm to locate the correspondence matrix C_{k,k^*}.

Embedded Point Distribution Model

Once we have correspondences to hand, we then can construct the generative model for the set of graphs. We first to align the graph co-ordinates matrices to the same size. For graph G_k which has a $n \times n$ size co-ordinates matrix Y_k, we first extend the co-ordinates size to m by adding zeros at the end of the co-ordinates matrix as we have done at the matching step5.2. Then we have a extended co-ordinates matrix Y_k^e which is $m \times n$ size. The second step is to compute a aligned co-ordinate matrix $\tilde{Y_k} = Y_k^e C_{k,k^*}$, where the aligned co-ordinates matrix $\tilde{Y_k}$ is $m \times m$ size. We then can convert the co-ordinates matrix to form a long vector. We do this by staking the columns of the co-ordinates matrix to form a long vector. The long vector is represented as

$$\hat{Y_k} = (\tilde{Y_k}(1,1), \tilde{Y_k}(1,2), ..., \tilde{Y_k}(m,m))^T$$

We model variations in the positions of the stacked vectors using a point distribution model. The process is similiar to the Pricipal Component Analysis(PCA) procedure 3.3.6. We commence by computing the mean vector positions. The matrix of mean position co-ordinates is given by

$$\hat{X} = \frac{1}{T} \sum_{k \in T} \hat{Y}_k$$

, where $\hat{X}$ is a $m \times m$ length long vector, and the covariance matrix for the point positions is

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

To construct the point distribution model, we perform the eigendecomposition

$$\Sigma = \Psi \Gamma \Psi^T$$

where $\Gamma = diag(\gamma_1, \gamma_2, ..., \gamma_K)$ is the diagonal matrix of ordered eigenvectors and $\Psi = (\psi_1|.....|\psi_K)$ is the matrix with the correspondingly ordered eigenvectors as columns.

We deform the mean embedded vector 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. The deformed point-set position is given by

$$\tilde{X} = \hat{X} + \tilde{\Psi} b$$

An observed configuration of graph G_w with embedded co-ordinates matrix Y_w may be fitted to the model. To do this we first align the co-ordinates matrix Y_w to get $\tilde{Y}_m$ with the model graph co-ordinates matrix. Then we stack the aligned co-ordinates matrix to a long vector $\hat{Y}_w$. The best fit parameters are estimated using the least squares procedure

$$b^* = \arg\min_b (\hat{Y}_w - \hat{X} - \tilde{\Psi} b)^T (\hat{Y}_w - \hat{X} - \tilde{\Psi} b)$$

The best-fit parameter vector is solved by

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

and the reconstructed set of embedded point positions is

$$\hat{Y}_w^* = \hat{X} + \tilde{\Psi}\tilde{\Psi}^T(\hat{Y}_w - \hat{X})$$

From the reconstructed point positions we can recover the Laplacian matrix for the corresponding graph. We first pack $\hat{Y}_w^*$ back to $m \times m$ size co-ordinate matrix $\tilde{Y}_w^*$. Then the extended co-ordinates matrix can be achieved by $Y_w^{*e} = \tilde{Y}_w^* C_{w,k^*}^T$. We can then get the origin co-ordinates matrix Y_w^* by truncating the end rows of the matrix Y_w^{*e}. The heat kernel for the reconstructed embedded graph is

$$h_t^* = (Y_w^*)^T Y_w^* = \exp[-\hat{L}^* t]$$

and the Laplacian is hence

$$\hat{L}^* = -\frac{1}{t}\ln\left\{(Y_w^*)^T(Y_w^*)\right\}$$

From the reconstructed Laplacian we can compute the corresponding adjacency matrix

$$A^* = D - D^{\frac{1}{2}}\hat{L}^* D^{\frac{1}{2}} \tag{5.2}$$

5.3.2 Graph Similarity

With the generative model to hand, we can use it to measure graph similarity. We adopt two different approaches to the problem. The first involves computing the Mahalonobis distance between embedded vector positions. The second is based on the Euclidean distance between the fitted parameters of the linear point distribution model.

Mahalanobis Distance

The generative model can be used to match and assess the similarity of graphs. The aim here is to construct a model that can capture the local variance of the embedded vector positions that are in correspondence with a given reference graph node. For the node indexed $j \in V_{k^*}$ of the reference graph the mean position vector of the corresponding nodes is

$$x^j = \frac{1}{|T|} \sum_{k \in T} \tilde{Y_k}^j \tag{5.3}$$

and the matrix of mean point positions is

$$X = (x^1 | x^2 | .... | x^{|V_{k^*}|})$$

The covariance matrix for the positions of nodes of the embedded graphs that are in correspondence with the reference graph node j is

$$\Sigma_j = \frac{1}{|T|} \sum_{k \in T} (\tilde{Y_k}^j - x^j)^T (\tilde{Y_k}^j - x^j) \tag{5.4}$$

The Mahalanobis distance between the aligned embedded position of the node $i \in V_k$ from graph G_k, which is in correspondence with the node $j \in V_{k^*}$ from the reference graph is

$$d_M^2(j) = (\tilde{Y_k}^j - x^j)^T \Sigma_j^{-1} (\tilde{Y_k}^j - x^j)$$

The distance between two graphs G_k and G_{k^*} can be defined as

$$d_M^2(G_k, G_{k^*}) = \sum_{j \in V_{k^*}} d_M^2(j)$$

Graph Similarity in the Eigenspace

The similarity of a pair of graphs can also be measured using the difference in their best-fit parameter vector. Since the parameter vector is just the projection of the corresponding graph into the eigenspace of the model, the difference between 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

$$d^2(k_1, k_2) = (b^*_{k_1} - b^*_{k_2})^T (b^*_{k_1} - b^*_{k_2})$$

In terms of the embedded node co-ordinates

$$d^2(k_1, k_2) = (\hat{Y}^*_{k_1} - \hat{Y}^*_{k_2})^T \tilde{\Psi} \tilde{\Psi}^T (\hat{Y}^*_{k_1} - \hat{Y}^*_{k_2})$$

5.4 Experiments

In this section, we first provide some experimental evaluation of the new graph-matching method. With the matching results we continue to construct the generative model for graphs which can capture the variation within a set of graphs. In the second part of this section, we demonstrate the effectiveness of the model by experiments from both the synthetic data and real world data.

5.4.1 Experiments on Graph Matching

In this section we will show how structural graph matching can be solved by using heat kernel embedding and spectral alignment. To commence we investigate the matching results obtained by changing the t parameter. We choose two graphs from the COIL database. We denote the two graphs by G_I and G_J. We map them to the vector space Y_I and Y_J and compute the correspondence matrix C_{IJ} based on the algorithm described above. We then show the matching result by projecting the Y_I and $\tilde{Y}_J$ to the same coordinate space. For visualization purposes we need to truncate the Y_I and $\tilde{Y}_J$ to two dimensional vectors. So we take the first two columns of the matrices and project them to two dimensional space. In the plots of Figure 5.1 the red points represent the node from G_I while the blue ones represent the aligned nodes from graph G_J. From left-to-right and top-to-bottom the t parameter takes on the value of $0.5, 1, 5, 10, 100, 1000$. The conclusion is that if we choose a good value of the t parameter we can obtain good correspondence matching results.

So we take the t value as 0.5 in the rest of the experiments in this part. We now embark on a delicate analysis of the graph matching algorithm. We compare our method with some alternative methods by using synthetic data. Second, we evaluate our method on real world data. We commence with some synthetic data experiments. The aim is to evaluate how the new method works under controlled structural corruption and to compare it with some alternative methods. These alternatives are Shapiro and Brady (Shapiro and Brady, 1992) and Scott and Longuett-Higgins' (Scott and Longuett-Higgins, 1991) 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 (Umeyama, 1988) method. In our method we are concerned with matching the Delaunay triangulation of corner features. Beside the coordinates of the feature points the Delaunay graph will incorporate important structural information.

Our first experiment is based on synthetic data. We use a seed point-set that contains 30 nodes. We construct the Delaunay graph of the Voronoi tessellation of these nodes. We have kept the number of points fixed and have added

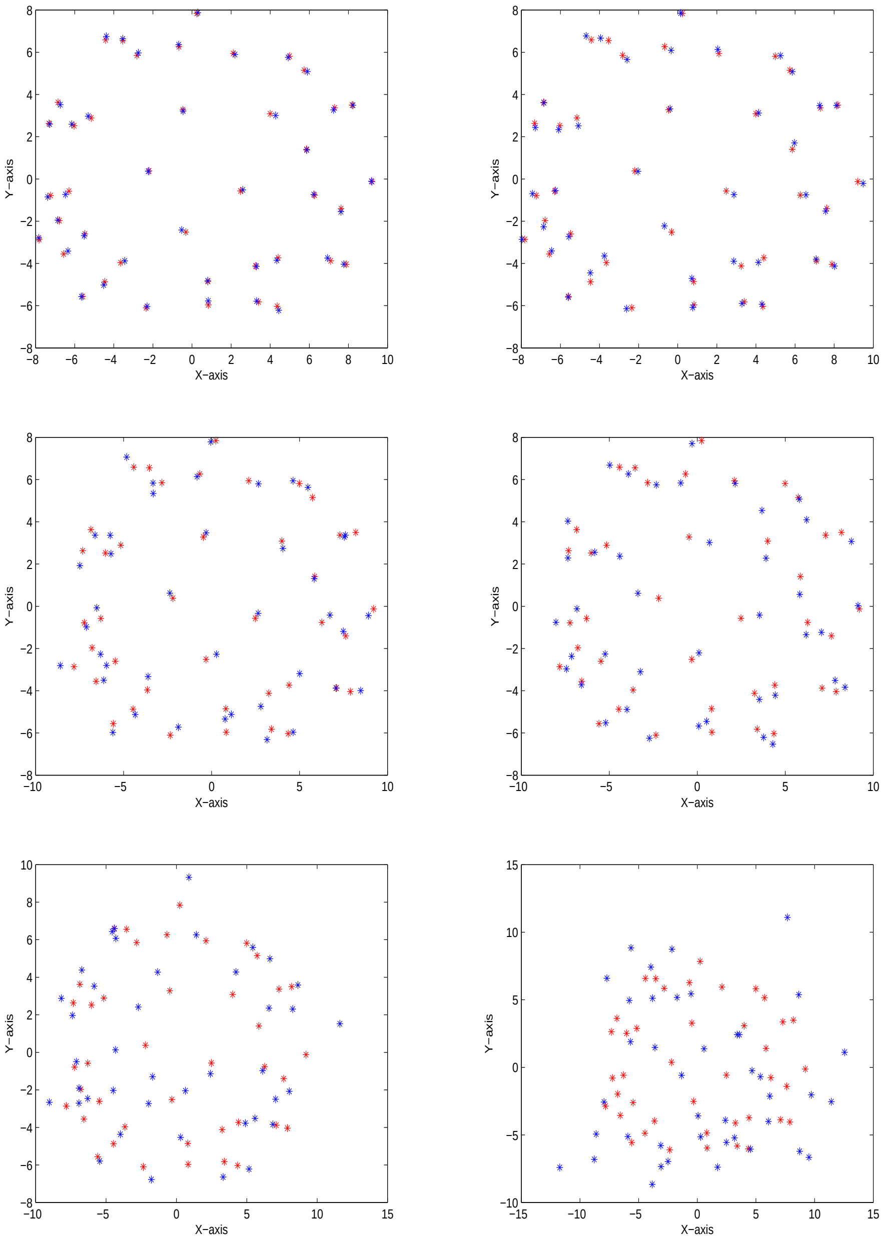

Figure 5.1: Aligned vector coordinates for two embedded graphs by varying the t (from left to right, top to bottom, the results obtained when t equals 0.5, 1, 5, 10, 100 and 1000 respectively).

median Gaussian errors to the nodes positions. The parameter of the noise process is the standard deviation of the positional jitter. When the nodes positions change we reconstruct the Delaunay graph based on the new point-set position. In Figure 5.2, we show the fraction of correct correspondences as a function of the noise standard deviation for our method, Shapiro and Brady's (Shapiro and Brady, 1992) method, Umeyama's method and Scott and Longuett-Higgins' method (Scott and Longuett-Higgins, 1991). To take this study one step further in Figure 5.3, we investigate the effect of structural noise. Here we have added a controlled fraction of additional nodes at random positions and have recomputed the Delaunay triangulations. We plot the fraction of correct correspondences as a function of the fraction of added nodes. The plot compares the result of applying our method to the data, the results obtained using Scott and Longuett-Higgins' method and Shapiro and Brady's method. Since Umeyama's algorithm can not handle graphs of different size, we have not used it this time. The main feature to note is that our method outperforms the two alternatives. This means our method can solve the structure matching problem when the graphs are of different size.

To take this study one step further, we perform some real-world data experiments. We apply our matching method to two image sequences (MOVI and Desk). There are rotation, scaling, and perspective distortion present. Example images from these sequences are shown in Fig 5.4 and correspond to different camera viewing directions. The detected feature points and their Delaunay triangulations are overlaid 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 5.5, we test our method on some pairs of images. In Table 5.1 we summarize the matching results for the MOVI house images. Here we list the number of nodes in the Delaunay graphs, the number of correct correspondences, the number of correspondence errors, and the number of points without correspondence. We also selected a pair of images which contain the same number of corner points (image 1 and image 4 from MOVI sequence 140 nodes). Although the number of corners is the same, there are differences in both the identities

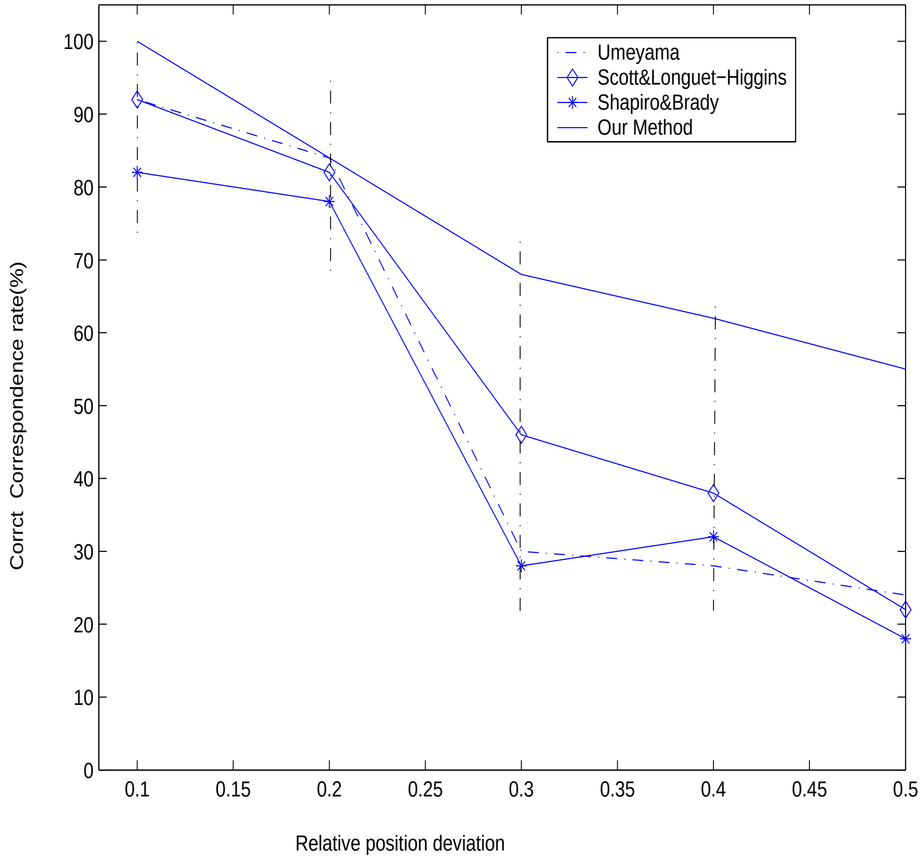

Figure 5.2: Comparison of four methods for matching with the same number of nodes.

of the detected points and their structural arrangement. We compared these image matching results by using our algorithm, Umeyama's algorithm, Scott and Longuett-Higgins' algorithm and Shapiro and Brady's method. The results are summarized in Table 5.2. From these results, it is clear that our new method outperforms the alternatives.

5.4.2 Experiments on Generative Model for Graph Structure

In this section we show how to construct the generative model with the matching results solved by the algorithm described above. The experiments are based on

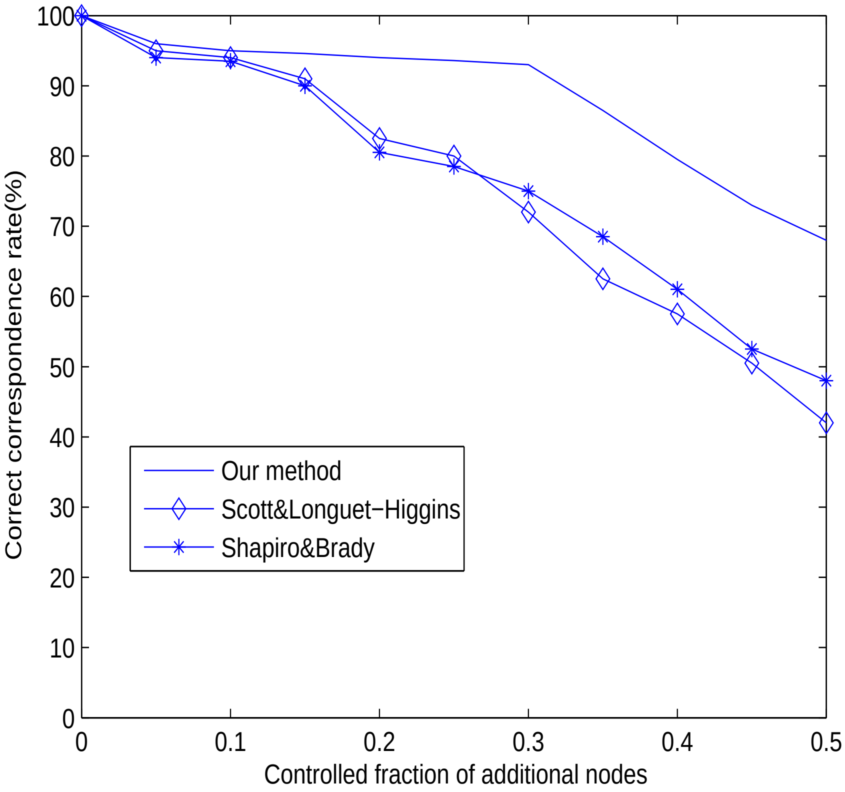

Figure 5.3: Comparison of three matching methods of different number of nodes.

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 5.1: Experiment results for MOVI house sequence images.

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

Table 5.2: Summary of comparison of the four matching algorithms.

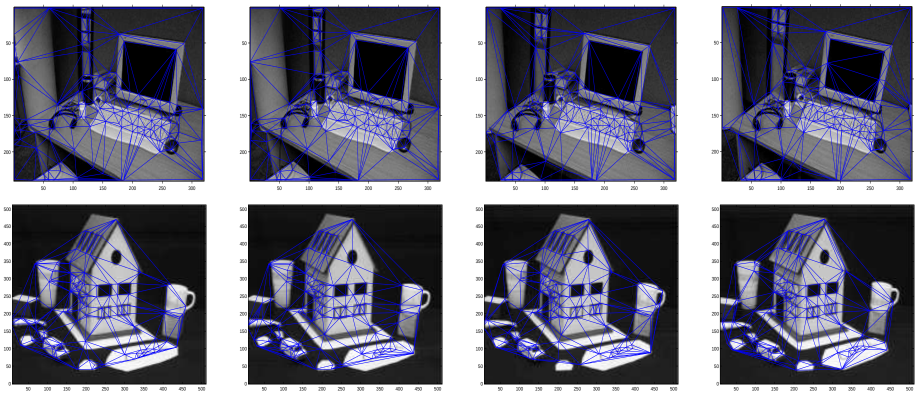

Figure 5.4: Delaunay graphs overlaid on the images.

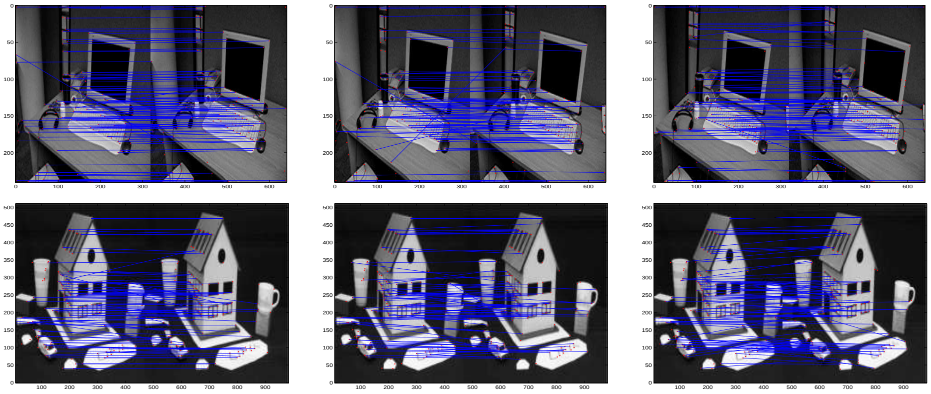

Figure 5.5: Our algorithm for CMU and MOVI house sequences.

synthetic data and real world data.

Synthetic Data

We commence our experimental study with an example based on synthetic data. We use the set of dumbbell shaped graphs in Figure 5.6 to construct the generative model. Here the number of nodes is fixed but the edge structure varies. We construct the covariance matrix Σ and mean shape $\hat{X}$ for the embedded nodes. The rows in Figures 5.7 show the variation modes along the three eigenvector directions corresponding to the largest three eigenvalues of the covariance matrix Σ. The different panels in the rows are obtained by varying the relevant component of the parameter vector b_i from $\sqrt{-3}$ to $\sqrt{3}$. There are clear differences

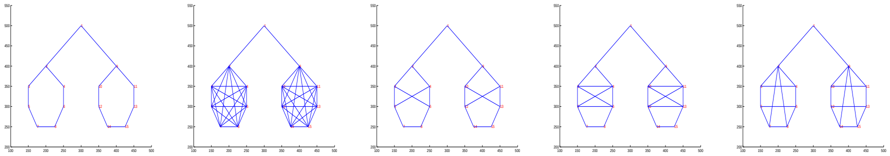

Figure 5.6: Some examples of the dumbbell shape graphs.

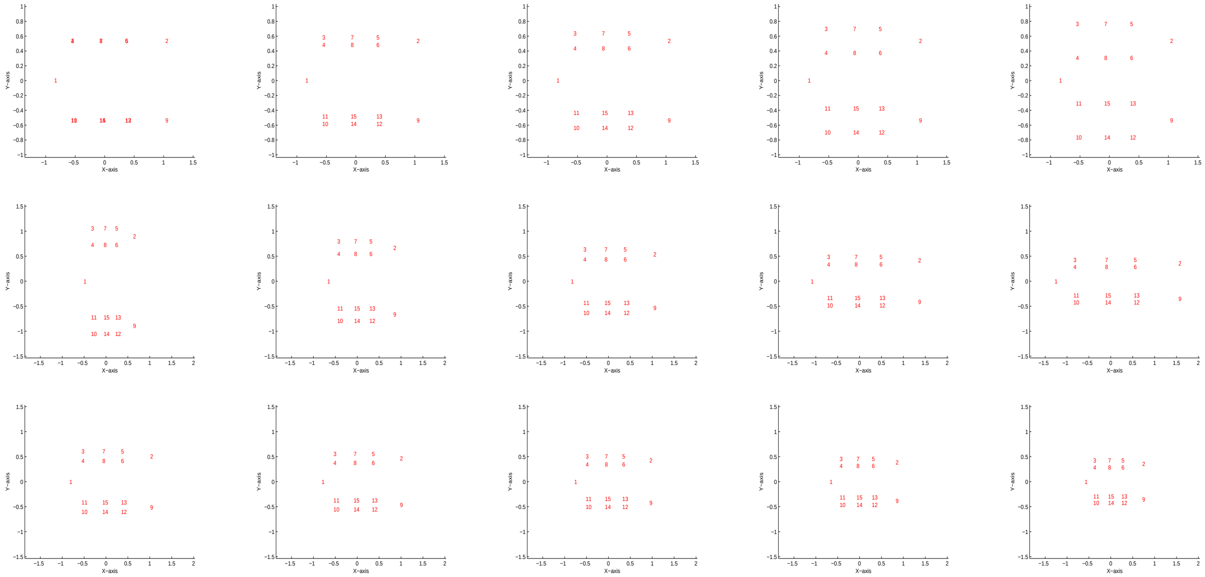

Figure 5.7: Graph eigenvector variation.

in the structures captured by the different eigenmodes. In Figure 5.8 we plot the panels in each row in Figure 5.7 in one plot. This illustrates how the parameter vector can be used to control the reconstructed point positions.

Real-world Data

In this section we provide some experimental evaluation of our generative model on real-world data. We use two data sets for the evaluation. The first of these is the COIL database Figure 3.4. The second is the Caltech-Oxford database Figure 3.6.

In Figure 5.9 we show the result of projecting the nodes into the space

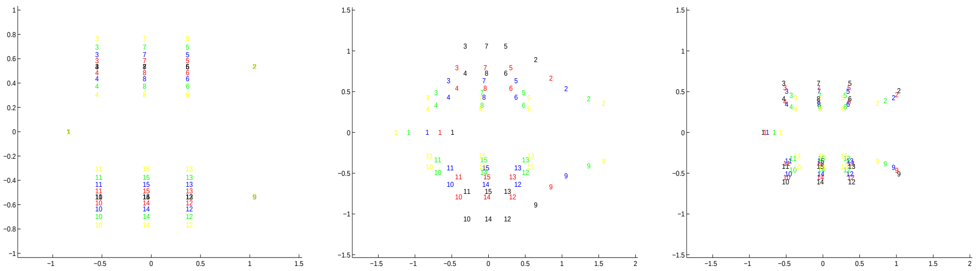

Figure 5.8: Graph eigenvector variation overlaid together.

spanned by the leading two eigenvectors of the heat kernel. The different panels in the figure are for different values of t. From left-to-right and top-to-bottom the values of t are $0.5, 1, 5, 10, 100$ and 1000. For this experiment we have taken 15 images from the duck sequence of the COIL database. Each blue point in the embedding corresponds to a single node of one of the 15 sample graphs. Superimposed on the node-positions as red points are the locations of the mean node positions. Around each mean node position we have drawn an ellipse. The major and minor axes of the ellipse are in the principal directions of the eigenvectors of the node position covariance matrix and the lengths of the semi-major axes are the corresponding eigenvalues. There are a number of features to note from this figure. First, for small values of t they form relatively compact clusters. Second, there is a significant variation in the size and directions of the ellipses. The compactness of the clusters supports the feasibility of our embedding approach and the variation in the ellipses underpins the need for a relatively complex statistical model to describe the distribution of embedded point positions. As the value of t increases the overlap of the ellipses also increases.

To investigate the role of the number of eigenmodes in the reconstruction of the graph structure we have examined the value of the Frobenius norm $F = ||A - A^*||$ between the original graph adjacency matrix A and the reconstructed adjacency matrix A^* computed by fitting the generative model. In Figure 5.18

we show the value of F as a function of the number of eigenmodes used. In this experiment we choose a graph from the duck sequence which contains 35 nodes. The different curves in the plot are for different values of t. The best reconstructions are obtained with small values of t and an increasing number of eigenmodes.

In Figures 5.10, 5.11, 5.12 and 5.13 we show the result of projecting the embedded node vectors for the graphs extracted from the COIL and Oxford-Caltech database onto the eigenvectors of the embedded node position covariance matrix Σ. Here we choose $t = 0.5$ and we use six leading eigenmodes of the covariance matrix of the embedded point position. In Figure 5.10 we use the duck sequence from the COIL database and 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. In Figures 5.11, 5.12 and 5.13 we show the projections of the images from the Oxford-Caltech dataset. The main feature to note from Figures 5.10 and 5.13 is that neighboring images in the sequence are close together in the eigenspace.

We have also experimented with the generative model as a means of clustering graphs. We have selected 60 images of 3 different objects in the Oxford-Caltech database, and have used these to construct the generative model. Using the model we compute graph similarity using the methods outlined in the previous part. In Figure 5.14 we show the matrix of Mahalanobis distances between graphs, in Figure 5.15 the distance matrix between the best fit parameter vectors and in Figure 5.16 the Euclidean distances between embedded points. The main feature to note from these distance matrices is that the Mahalanobis distance between embedded points is less noisy than the alternative two distance measures. Based on this observation, in Figure 5.17 we show the result of performing multidimensional scaling (MDS) on the Mahalanobis distances between the set of graphs. The different objects are shown in different colors and are well separated. For comparison, Figure 5.19 shows the result of projecting the graphs onto the eigenspace spanned by the leading three eigenvectors of the point position covariance matrix. This also gives good clusters.

5.5 Conclusion

In this chapter we have used the heat kernel embedding of graphs to construct a generative model for graph structure. The embedding allows nodes of the graphs under study to be mapped as points in a vector space. In the first section we have introduced the heat kernel embedding based graph matching. After we have mapped the nodes of the graph to the vectors we can perform a variant of Scott and Longuett-Higgins' spectral point-sets alignment algorithm to compute the correspondence matrix between the embedded point position vectors. By doing so we have overcome the limitation of the structural differences in the graphs. We have experimented with matching algorithms on both synthetic and real world image datasets. A comparison with other methods is provided. From the experimental results, it is clear that our method is better than the alternatives both in accuracy and stability to noise.

With the correspondence results between graphs we continue to construct the generative 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. The method proves to be effective for capturing structure variations between graphs and also for clustering graphs.

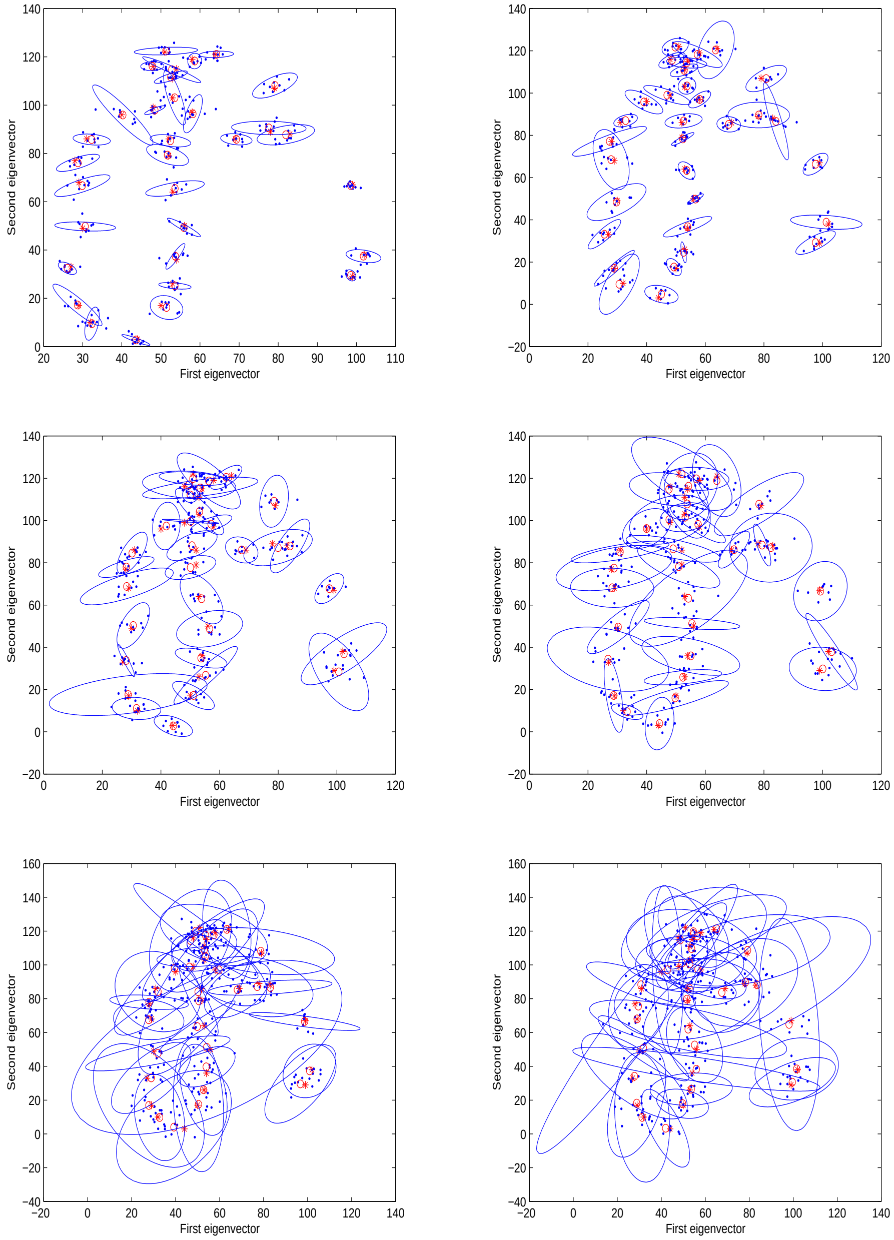

Figure 5.9: Embedded point positions and fitted covariance ellipsoids varying with t (from left to right, top to bottom $t = 0.5, 1, 5, 10, 100$ and 1000 respectively) for the heat kernel.

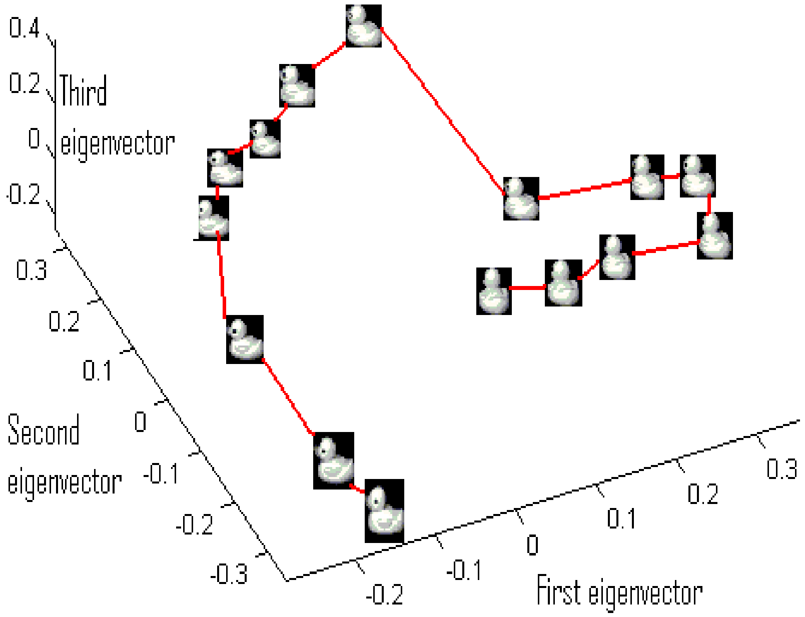

Figure 5.10: Eigenprojection of 15 images of duck sequence from COIL database.

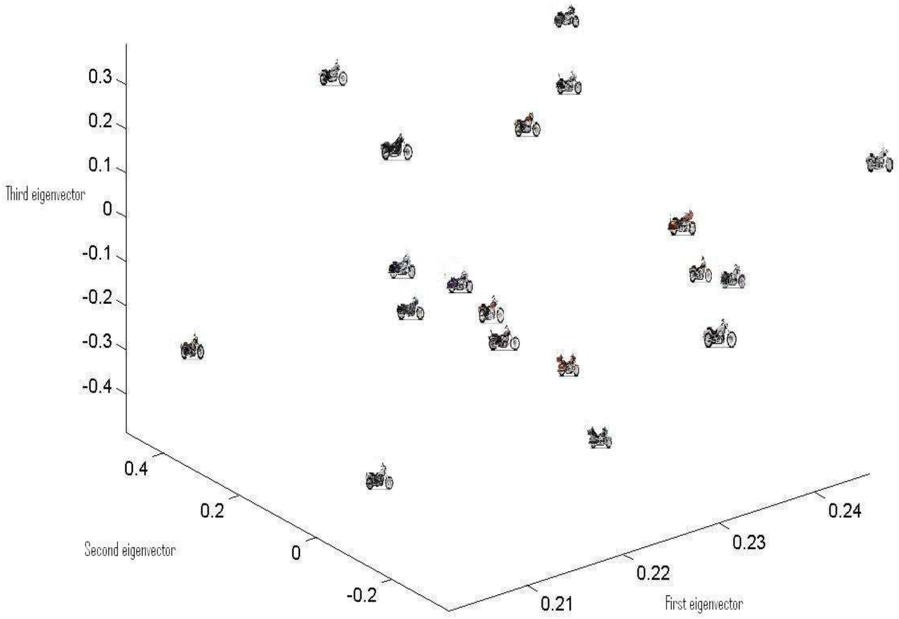

Figure 5.11: Eigenprojection of motorcycle images from Oxford-Caltech database.

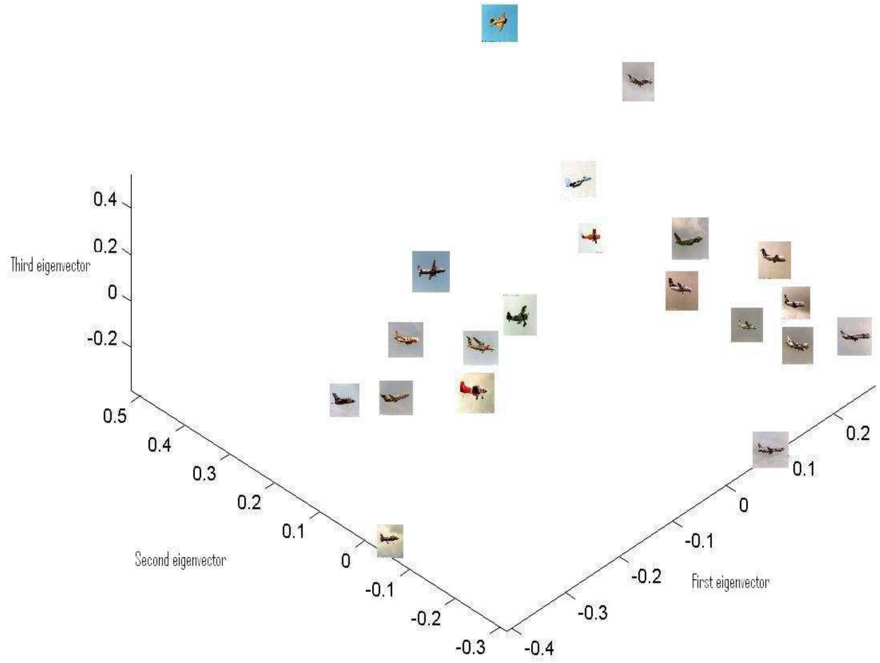

Figure 5.12: Eigenprojection of airplane images from Oxford-Caltech database.

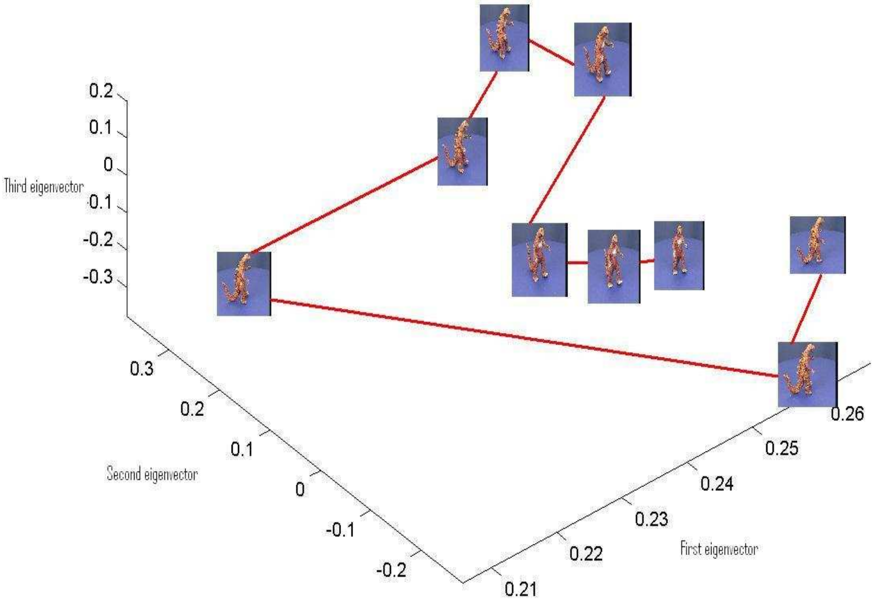

Figure 5.13: Eigenprojection of dinosaur images from Oxford-Caltech database.

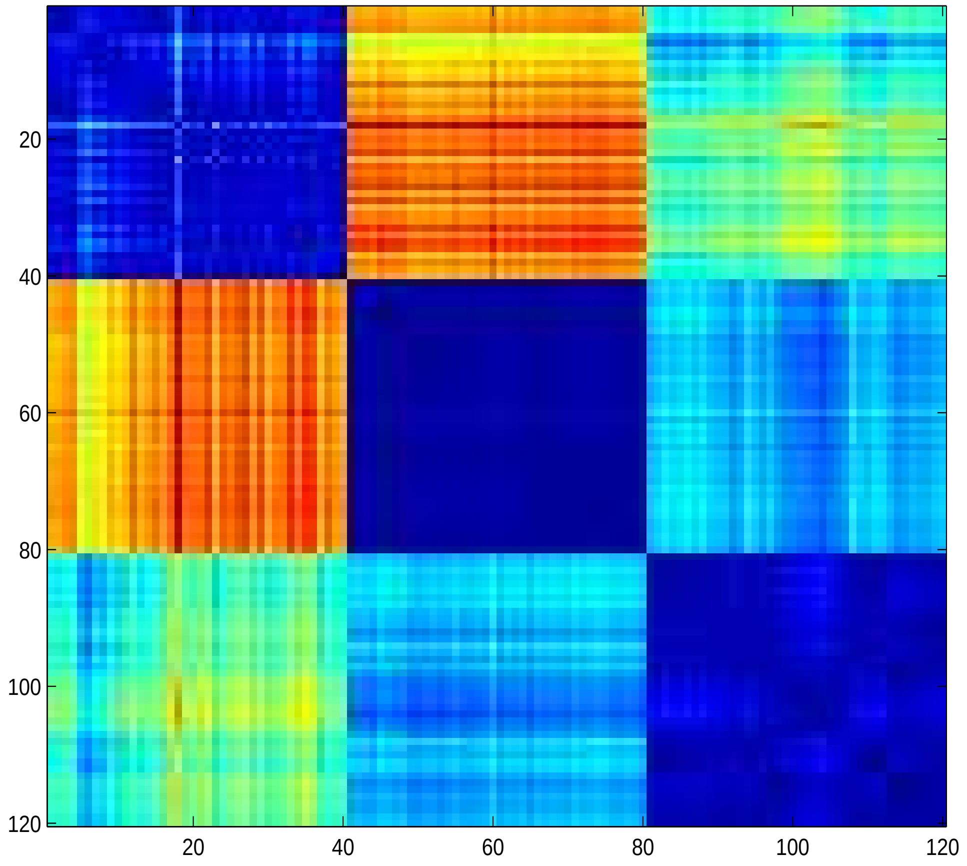

Figure 5.14: Distance matrix for Mahalanobis distance between embedded points.

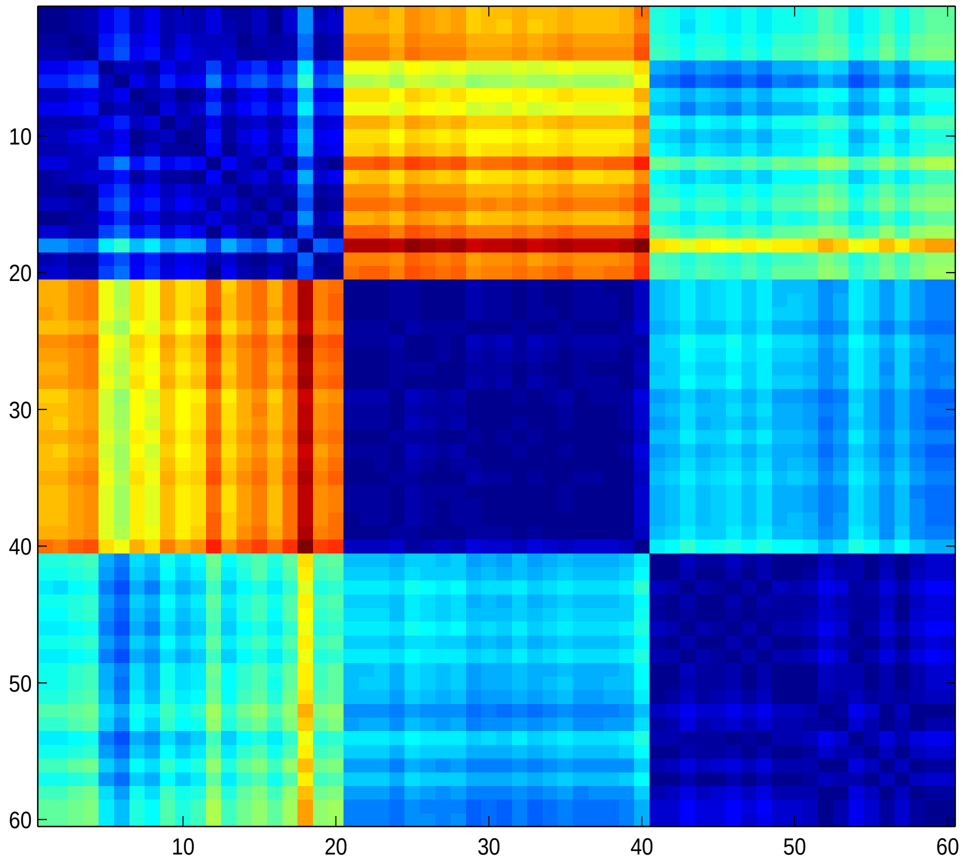

Figure 5.15: Distance matrix for the best fit parameter vectors.

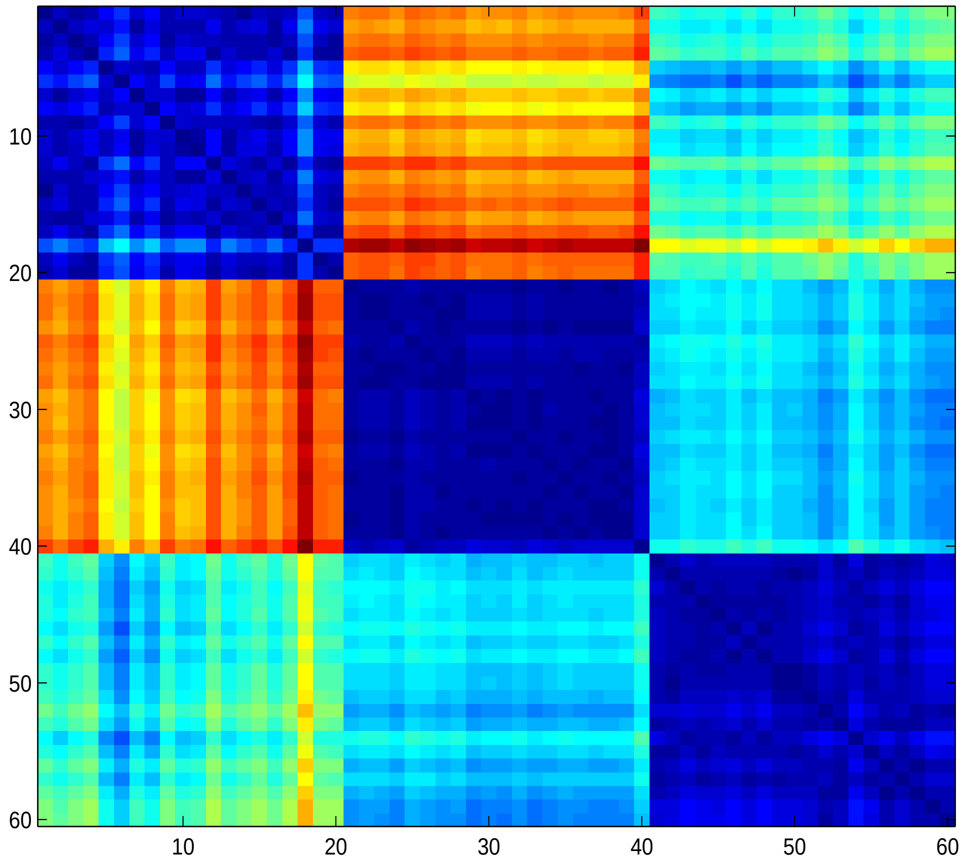

Figure 5.16: Distance matrix for Euclidean distance between embedded points.

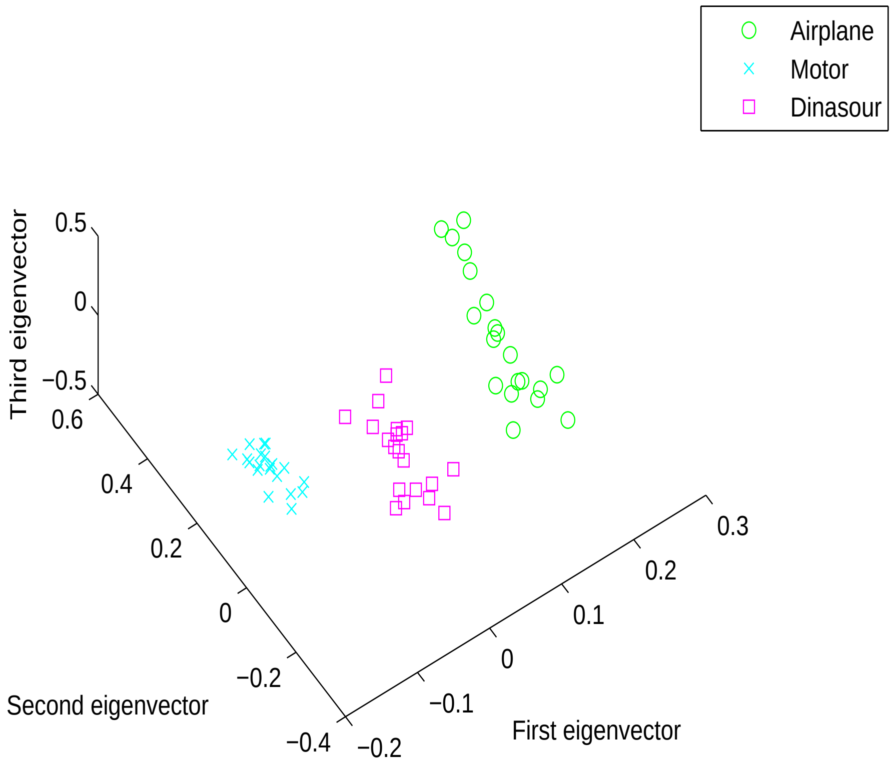

Figure 5.17: Graph clustering using Mahalanobis distances deduced from the graph generative model.

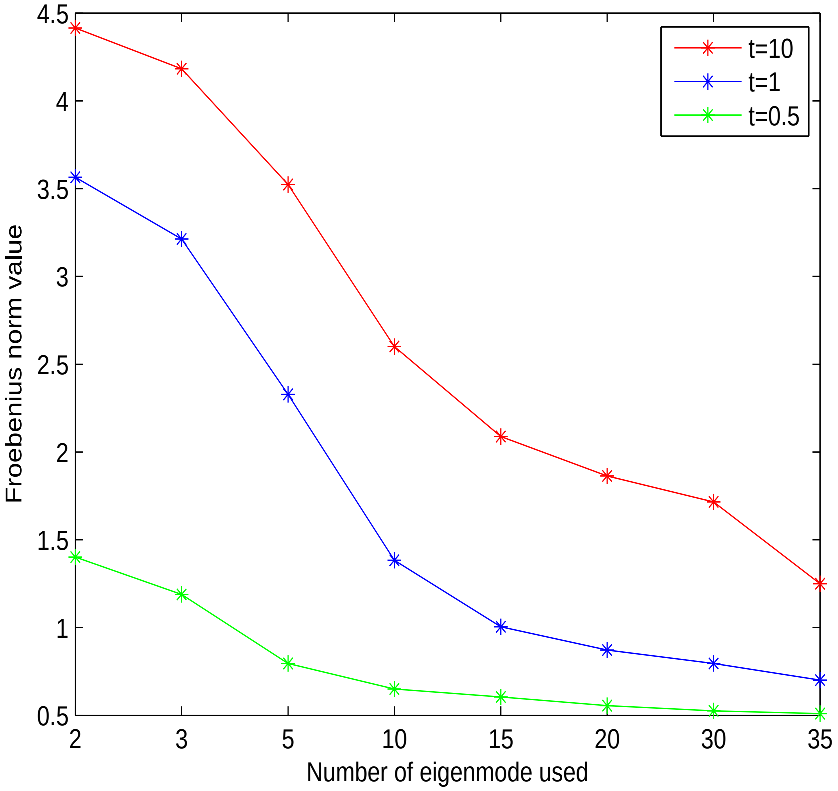

Figure 5.18: Frobenius norm as a function of numbers of eigenmodes.

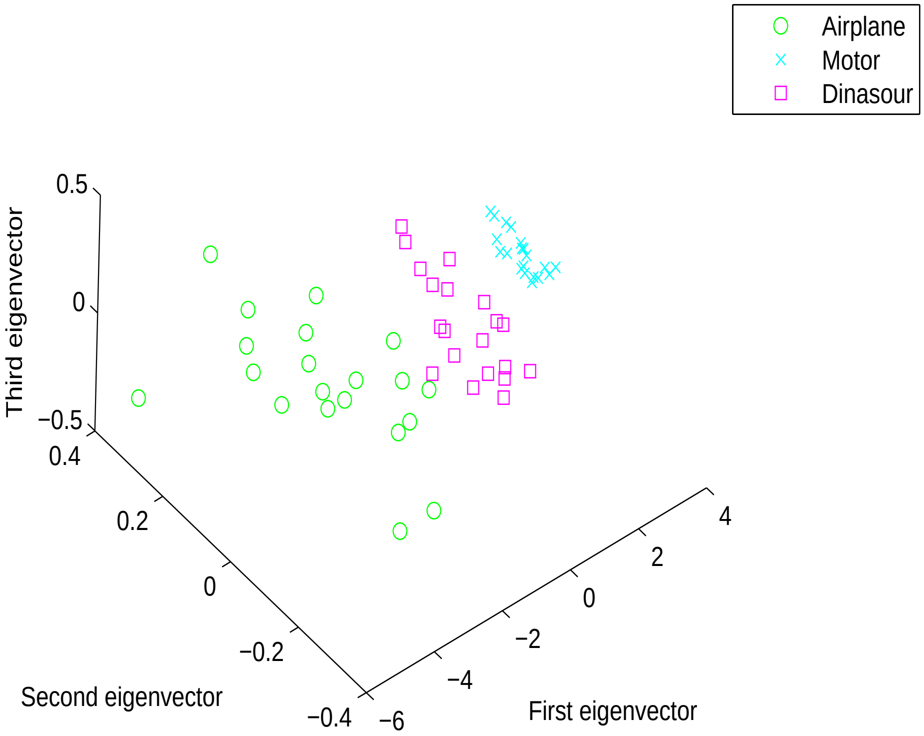

Figure 5.19: Spectral analysis of the Oxford-Caltech database.

Chapter 6

Conclusions and Future Work

In this chapter we first summarize the main contribution of the thesis. This includes the novel ideas on feature extraction, graph embedding, the graph structure generative model and their related applications in real world problems. Secondly we draw some of the limitations and possible extensions of the work.

6.1 Contribution

The theoretical and practical contributions of the thesis are as follows:

- Stable and robust features were extracted from the graph heat kernel for the purposes of graph characterization. We proved experimentally that these methods can be used to solve the image classification problems effectively. (Chapter 3)

- Graph embedding methods were introduced. We map the graph from the graph space to the manifold whose metric is characterized by the heat kernel. We explored two graph embedding methods based on the heat kernel analysis. Graph characterization problems can be solved by analyzing the embedded point position vectors. The applications for the method are graph matching and clustering. (Chapter 4 and Chapter 5)

- We combined the heat kernel embedding with the linear deformable model to construct the generative model for graph structure. Variation within graphs can be captured by using the mean and covariance matrix of the embedded point position vectors. The experimental results show the effectiveness of the method in capturing the variation within a set of moving image sequences. The method can also be applied on image clustering. (Chapter 5)

6.1.1 Feature Invariants from Graph Heat Kernel

The first contribution of this thesis is to extract robust and stable features from graph heat kernels to characterize graphs. Spectral graph characterization has been widely applied in computer vision for matching(Umeyama, 1988), clustering(Wilson et al., 2003), segmentation(Shi and Malik, 1997) and etc. The heat kernel is closely related with the Laplacian spectrum of the graph. Our starting point is to explore whether we can extract useful and stable invariants from the heat kernel of the graph to characterize the graph.

Our first investigation comes from the heat kernel trace, which is the sum of the diagonal elements of the heat kernel matrix of the graph. Mathematically, it is a function whose parameter is the Laplacian eigenvalues and whose argument is time. The shape of this function can be used to characterize the graph. We then turned our attention to the zeta function. By using the Mellin transformation, the zeta function can be proved to have a close relation with the heat kernel trace. It is related to the moments of the heat kernel trace. We may compute a set of invariants from the zeta function by inputing different integer arguments. With these invariants at hand, we can proceed to construct a feature vector which can be used for graph clustering. We also showed the relationship between the symmetric polynomial and the derivative of the zeta function at origin. Both of them can be used as invariants for graph characterization.

Finally in Chapter 3, we turned our attention to the heat content invariants. McDonald and Meyers (McDonald and Meyers, 2002) have shown that a set of differential invariants can be derived from the heat content of the heat kernel. Our investigation shows the feasibility of the heat content invariants as a way of characterizing graphs. We demonstrated experimentally the effectiveness of our methods. Real world examples were provided to show that these features can be used for graph clustering and image classification. A comparison with the standard spectral graph clustering algorithm (Luo et al., 2003; Sengupta and Boyer, 1998) shows our method outperforms the alternatives in terms of correctness for clustering.

In our opinion, the work described in Chapter 3 has two advantages. Firstly we explored the feature invariants from the graph heat kernel, which plays an essential role in spectral graph theory. Secondly, the features extracted from the graph heat kernel can provide more stable and robust characterization for graphs.

6.1.2 Graph Embedding for Graph Matching and Clustering

In Chapter 4 we took the graph heat kernel investigation a step further and explored the geometries of the graphs which reside on a manifold. The second contribution of this thesis is to explore graph embedding methods which can be used to transform the graphs to point-sets in vector space by analyzing the heat kernel matrix of the graph. 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.

We proposed two heat kernel based graph embedding methods. For the first method, we commenced by assigning a distance measure to the graph edges. The distance measure is furnished by the analysis of the heat kernel. We used the Isomap method (Tenenbaum et al., 2000) to find a low-distortion embedding to map the nodes of the graph to a low-dimensional vector space. We refer to this method as the parametric distance embedding. The second method is the heat kernel embedding, performed by applying heat kernel mapping to the graph heat kernel matrix. The embedded point position vector was found by performing the Young-Household decomposition on the heat kernel matrix of the graph.

Once the graphs were embedded to point-sets in vector space, we performed a number of point pattern analysis algorithms. In Chapter 4 we used statistical moments, spectral features from the covariance matrix and sectional curvature to characterize the point-sets. We use vector features to perform graph clustering. Experiments showed that the best performance was obtained by using heat kernel embedding with a spectral characterization, where the image classification rate can reach 98%.

6.1.3 Generative Model for Graph Structure

In Chapter 5 we combined the heat kernel embedding with the spectral point-sets alignment algorithm, to develop a method to match the correspondence between the embedded point position vectors. By doing so we develop a new graph matching algorithm which can be used to match graphs that contain structural differences. A comparison with standard algorithms (Scott and Longuett-Higgins, 1991; Shapiro and Brady, 1992; Umeyama, 1988) shows our algorithm outperforms the alternatives in terms of both the accuracy and robustness to noise and corruption. The linear deformable model (Cootes et al., 1995) was combined with the heat kernel embedding to construct a generative model for graphs to capture structural variation within a set of graphs. After the heat kernel embedding, we performed correspondence matching. With the matching results, we computed the mean and covariance matrix of the point position, which capture the variation information within the set of graphs. A graph is represented by a parameter vector, which can be used to reconstruct the adjacency and Laplacian matrix of the graph. Experiments showed the effectiveness of the model for tracking the variation within a sequence of moving images and clustering.

6.2 Future Work

There are a number of shortcomings in this thesis which can be amended with further work and further explored. In Chapter 3, we have explored a number of invariants which can be used for graph characterization. However, some extensions of the current work still exist. These include the use of features which have a direct geometrical meaning such as the Euler characteristic, the torsion of the mean and the Gaussian curvatures of the manifold.

In Chapter 4, there are clearly a number of ways in which the work reported can be extended. First it would be interesting to cast the graph clustering process into a mixture model which can be used for hierarchal clustering. Second, a possibility exists to explore the use of sectional curvature as a means of directly embedding the nodes of the graphs on a manifold. One of the possibilities that exists here is the variant of MDS reported by Lindman and Caelli (Lindman and Caelli, 1978). A third line of investigation would be to use the Euclidean distances or sectional curvature associated with the edges as attributes for the purposes of graph matching. Finally, it would be interesting to investigate whether the distances and curvatures could be used to aid the process of visualizing or drawing graphs.

In Chapter 5, our future plans revolve around 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 matching algorithm to trees and try to solve the correspondence problem for trees.

For the experimental evaluation in all Chapters, both the COIL and Oxford-Caltech databases contain just a single object and have very little background structure. It would therefore be interesting to apply our methods to images containing complex objects and backgrounds. By doing so, we plan to incorporate a statistical framework. Image recognition or clustering becomes a hierarchical process. A mixture model is used at each level to produce a probability that an object exists in the image. The second experimental shortcoming is the methods we have used to extract the graph representation from images. Currently, we have used Delaunay triangulations and relational graphs. However, the graphs extracted from the images can not exactly represent the actual structure of the original object. This has limited the application of our methods. This is due to the fact that both methods are based on the low-level vision algorithms. To solve this problem we plan to use image representation ideas from the high-level vision (Hall et al., 2002).

Bibliography

Alquezar, R., Sanfeliu, A., and Serratosa, F. (1998). Synthesis of function-described graphs. Structural and Syntactic Pattern Recognition.

Atkins, J. E., Bowman, E. G., and Hendrickson, B. (1998). A spectral algorithm for seriation and the consecutive ones problems. SIAM J.Comput, 28:297–310.

Bagdanov, A. D. and Worring, M. (2003). First order gaussian graphs for efficient structure classification. Pattern Recognition, 36:1311–1324.

Barlow, M. T. (1998). Diffusions on fractals. *Lectures on Probability Theory and Statistics*, Lecture Notes Math. 1690:1–121.

Belkin, M. and Niyogi, P. (2000). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396.

Bell, F. K. and Rowlinson, P. (1990). On the index of tricyclic hamiltonian graphs. Proc. Edinburgh Math. Soc., 33:233–240.

Biggs, N. L. (1993). Algebraic graph theory. Cambridge University Press.

Bourgain, J. (1985). On lipschitz embedding of finite metric spaces into hilbert space. Israel Journal of Mathematics, 52:4652.

Bunke, H. (1997). On a relation between graph edit distance and maximum common subgraph. Pattern Recognition Letters, 18:689–694.

Bunke, H. (1999). Error correcting graph matching: On the influence of the underlying cost function. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21:917–922.

Bunke, H., Foggia, P., Guidobaldi, C., and Vento, M. (2003). Graph clustering using the weighted minimum common supergraph. Graph Based Representations in Pattern Recognition, LNCS 2726:235–246.

Bunke, H., Munger, A., and Jiang, X. (1999). Combinatorial search vs. genetic algorithms: A case study based on the generalized median graph problem. Pattern Recognition Letters, 20:12711279.

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

Bunke, H. and Vento, M. (1999). Benchmarking of graph matching algorithms. Proc. 2nd Workshop on Graph-based Representations.

Busemann, H. (1955). The geometry of geodesics. *Academic Press*.

Canny, J. F. (1986). A computational approach to edge detection. IEEE Transaction on Pattern Analysis and Machine Intelligence, 8(6):679–698.

Carcassoni, M. and Hancock, E. R. (2003). Correspondence matching with modal clusters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26:1609–1615.

Cesarini, F., Francesconi, E., Gori, M., and Soda, G. (1999). A two level knowledge approach for understanding documents of a multi-class domain. The Proceeding of the International Conference on Document Analysis and Recognition.

Christmas, W., Kittler, J., and Petrou, M. (1995). Structural matching in computer vision using probabilistic relaxation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17:31–48.

Chung, F. R. K. (1997). Spectral graph theory. *American Mathematical Society*.

Chung, F. R. K. and Yau, S. T. (1997). A combinatorial trace formula. Tsinghua lectures on geometry and analysis.

Coifman, R. R. and Lafon, S. (2004). Diffusion maps. Applied and Computational Harmonic Analysis.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models - their training and application. Computer Vision and Image Understanding, 61(1):38–59.

Costeira, J. and Kanade, T. (1995). A multibody factorization method for motion analysis. In Proc. International Conference on Computer Vision.

Coulhon, T., Barlow, M., and Grigor'yan, A. (2000). Manifolds and graphs with slow heat kernel decay. Imperial College Print.

Cox, T. and Cox, M. (1994). Multidimensional scaling. Chapman and Hall.

Cvetkovic, D. M., Doob, M., and Sachs, H. (1995). Spectra of graphs. Johann Ambrosius Barth.

de Verdiere, C. (1998). Spectra of graphs. Math of France., 4.

Dias, J. R. (1993). Molecular orbital calculations using chemical graph theory. Springer-Verlag, Berlin.

Doermann, D., Rosenfeld, A., and Rivlen, E. (1997). The function of documents. The Proceeding of the ICDAR97.

Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2:264–271.

Flusser, J. and Suk, T. (1993). Pattern recognition by affine moment invariants. Pattern Recognition, 26:167–174.

Gilkey, P. B. (1984). Invariant theory, the heat equation, and the atiyah-single index theorem. Publish or Perish Inc.

Grigor'yan, A. (2001). Heat kernel on manifold, graphs and fractals. European Congress of Mathematics, I:393–406.

Grigor'yan, A. (2003). Heat kernels on manifolds, graphs and fractals. preprint.

Hall, P. M., Owen, M., and Collomosse, J. (2002). A trainable low-level feature detector. International Conference on Pattern Recognition, 1:708–711.

Hancock, E. R. and Kittler, J. (1990). Edge labeling using dictionary-based relaxation. IEEE Transaction on Pattern Analysis and Machine Intelligence, 12:165181.

Harris, C. G. and Stephens, M. J. (1994). A combined corner and edge detector. Fourth Alvey Vision Conference.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. Machine Learning, 20:197–243.

Hein, M., Audibert, J. Y., and von Luxburg, U. (2005). From graphs to manifold - weak and strong pointwise consistency of graph laplacian. 18th Annual Conference on Learning Theory.

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

Horaud, R. and Sossa, H. (1995). Polyhedral object recognition by indexing. Pattern Recognition, 28(12):1855–1870.

Huet, B. and Hancock, E. R. (1999). Line pattern retrieval using relational histograms. IEEE Transaction on Pattern Analysis and Machine Intelligence, 21:13631370.

Jain, B. J. and Wysotzki, F. (2004). Central clustering of attributed graphs. Machine Learning, 56:169–207.

Jollife, I. T. (1986). Principal component analysis. New York: Springer Verlag.

Kim, H. Y. and Kim, J. H. (2001). Hierarchical random graph representation of handwritten characters and its application to hangul recognition. Pattern Recognition, 34:187–201.

Klassen, Srivastava, A., Mio, W., and Joshi, S. H. (2004). Analysis of planar shapes using geodesic paths on shape spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26:372–383.

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

Lafferty, J. and Lebanon, G. (2004). Diffusion kernels on statistical manifolds. CMU-CS-04-101.

Lee, C. G. and Small, C. G. (1999). Multidimensional scaling of simplex shapes. *Pattern Recognition*, 32:1601–1613.

Lindman, H. and Caelli, T. (1978). Constant curvature riemannian scaling. Journal of Mathematical Psychology, 17:89109.

Linial, N., London, E., and Rabinovich, Y. (1995a). The geometry of graphs and some of its algorithmic applications. Combinatorica, 15:215–245.

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

Lozano, M. A. and Escolano, F. (2003). Acm attributed graph clustering for learning classes of images. *Graph Based Pattern Recognition*, LNCS 2726:247–258.

Luo, B. and Hancock, E. R. (2001). Structural graph matching using the em algorithm and singular value decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23:1120–1136.

Luo, B., Wilson, R. C., and Hancock, E. R. (2003). Spectral embedding of graphs. Pattern Recognition, 36:2213–2230.

McDonald, P. and Meyers, R. (2002). Diffusions on graphs, poisson problems and spectral geometry. Transactions on American Mathematical Society., 354:5111–5136.

Miclet, L. (1986). Structural methods in pattern recognition. North Oxford.

Murase, H. and Nayar, S. K. (1994). Illumination planning for object recognition using parametric eigenspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16:1219–1227.

Nene, S. A., Nayar, S. K., and Murase, H. (1996). Columbia object image library(coil 100). Columbia University.

Pavlidis, T. (1977). Structural pattern recognition. Springer-Verlag.

Perona, P. and Freeman, W. T. (1998). A factorization approach to grouping. In Proc. European Conference on Computer Vision.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering. Journal of the American Statistical Association, 66:846–850.

Ranicki, A. (1992). Algebraic l-theory and topological manifolds. *Cambridge University Press*.

Robles-Kelly, A. and Hancock, E. R. (2002). Ma graph-spectral approach to correspondence matching. *International Conference on Pattern Recognition*.

Robles-Kelly, A. and Hancock, E. R. (2005). Graph edit distance from spectral seriation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27:365–378.

Rosenberg, S. (2002). The laplacian on a riemannian manifold. Cambridge University Press.

Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. Science, 290(5500):2323–2326.

Sachs, H., Cvetkovic, D. M., and Doob, M. (1980). Spectra of graphs. *Academic Press*.

Sanfeliu, A. and Fu, K. S. (1983). A distance measure between attributed relational graphs for pattern-recognition. IEEE Transactions on Systems, Man and Cybernetics, 13(3):353–362.

Sarkar, S. and Boyer, K. L. (1993). Perceptual organization in computer vision. IEEE Trans. Systems, Man and Cybernetics, 23:382–399.

Sarkar, S. and Boyer, K. L. (1996). Quantitative measures of change based on feature organization. In Proc. International Conference on Computer Vision.

Scott, G. and Longuett-Higgins, H. (1990). Feature grouping by relocalisation of eigenvectors of the proximity matrix. In Proc. British Machine Vision Conference.

Scott, G. and Longuett-Higgins, H. (1991). An algorithm for associating the features of two images. Proceedings of the Royal Society of London Series B-Biological, 244:21–26.

Sengupta, K. and Boyer, K. (1998). Modelbase partitioning using property matrix spectra. Computer Vision and Imaging Understanding, 70(2):177–196.

Shapiro, L. G. and Haralick, R. M. (1981). Structural descriptions and inexact matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 3(5):504–519.

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

Shi, J. and Malik, J. (1997). Normalized cuts and image segmentation. International Conference on Computer Vision and Pattern Recognition.

Shokoufandeh, A., Dickinson, S., Siddiqi, K., and Zucker, S. (1999). Indexing using a spectral encoding of topological structure. *International Conference on Computer Vision and Pattern Recognition*.

Smola, A. J., Bartlett, P. L., Schlkopf, B., and Schuurmans, D. (1998). Advances in large margin classifiers. *MIT Press, Cambridge*, 354:5111–5136.

Smola, A. J. and Kondor, R. (2004). Kernels and regularisation of graphs.

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

Torsello, A. and Hancock, E. R. (2001). Efficiently computing weighted tree edit distances using relaxation labeling. Graph Based Pattern Recognition, LNCS 2134:438–453.

Tsai, W. H. and Fu, K. S. (1983). Subgraph error-correcting isomorphism for syntactic pattern-recognition. IEEE Transactions on Systems, Man and Cybernetics, 13(1):48–62.

Umeyama, S. (1988). An eigendecomposition approach to weighted graph matching problems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(5):695–703.

Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. In Proc. International Conference on Computer Vision.

Wilson, R. C. and Hancock, E. R. (1997). Structural matching by discrete relaxation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(6):634–648.

Wilson, R. C., Luo, B., and Hancock, E. R. (2003). Pattern vectors from algebraic graph theory. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27:2220–2237.

Witten, E., Green, M. B., and Schwarz, J. H. (1988). Superstring theory. Cambridge University Press.

Wong, A. K. C., Constant, J., and You, M. L. (1990). Random graphs. Structural and Syntactic Pattern Recognition.

Yau, S. T. and Schoen, R. M. (1988). Differential geometry. Science Publication.

Young, G. and Householder, A. S. (1938). Discussion of a set of points in terms of their mutual distances. Psychometrika, 3(1):19–22.

Yu, H. and Hancock, E. R. (2005). Graph seriation using semi-definite programming. Graph based Pattern Recognition.

Zhu, P. (2006). Stability of graph representation. Master Thesis.