Spectral Representation for Matching and Recognition

MUHAMMAD HASEEB

A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF YORK
DEPARTMENT OF COMPUTER SCIENCE

JANUARY, 2013

Abstract

In this thesis, we aim to use the spectral graph theory to develop a framework to solve the problems of computer vision. The graph spectral methods are concerned with using the eigenvalues and eigenvectors of the adjacency matrix or closely related Laplacian matrix. In this thesis we develop four methods using spectral techniques: (1) We use a Hermitian property matrix for point pattern matching problem; (2) We use coefficients of symmetric polynomials to cluster similar human poses using the skeletal representation acquired from Microsoft Kinect; (3) We use coefficients of the elementary symmetric polynomials to make the direction of the eigenvectors of the proximity matrices consistent with each other for the problem of correspondence matching; (4) We use commute time embedding to construct a 3D shape descriptor for the purpose of 3D shape classification.

In Chapter 3 we address the problem of correspondence matching. We extend the Laplacian matrix to the complex domain by constructing a Hermitian property matrix. We construct a Hermitian property matrix from the spatial locations

of the 2D feature points extracted from a pair of images and the angular information associated with these feature points. We construct the Hermitian property matrix in a way that reflects the Laplacian matrix. The complex eigenvectors of the Hermitian matrix is then used to find the correspondences between pairs of points across two images. We embed the complex eigenvectors of the Hermitian property matrix in the iterative alignment EM algorithm developed by Carcassoni and Hancock to make it robust to rotation, noise and point-position jitter. Experimental results on both synthetic and real world data have been presented.

Chapter 4 develops a clustering method using four different type of feature vectors constructed from the complex coefficients of the elementary symmetric polynomials. These polynomials are computed from the eigenvalues and the complex eigenvectors of a Hermitian property matrix. The feature vectors are embedded into a pattern-space using Principal Component Analysis (PCA) and Multidimensional Scaling (MDS) to cluster similar human poses acquired using the Microsoft Kinect device for Xbox 360. The Hermitian property matrix is constructed from the length of the limbs and the angles subtended by each pair of limbs using the three-dimensional skeletal data produced by the Kinect device. The given skeleton is converted to its equivalent line graph to compute the angles between pairs of limbs. The joints locations are used to compute the limb lengths.

In Chapter 5, we describe a method to correct the sign of eigenvectors of the proximity matrix for the problem of correspondence matching. The signs of the eigenvectors of a proximity matrix are not unique and play an important role in computing the correspondences between a set of feature points. We use the coefficients of the elementary symmetric polynomials to make the direction of the eigenvectors of the two proximity matrices consistent with each other.

Chapter 6 describes a 3D shape descriptor that is robust to changes in pose and topology. The descriptor is based on the D2 shape descriptor developed by

Osada et al, which is essentially the frequency distribution of the Euclidian distance between randomly selected points on the surface of the 3D shape. We use the commute-time distance instead of using the Euclidian distance between randomly selected points. A new and completely unsupervised mesh segmentation algorithm is proposed, which is based on the commute time embedding of the mesh and k-means clustering using the embedded mesh vertices.

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost I would like to thank almighty Allah on successful completion of my PhD and writing up within due time.

I feel very privileged to have Prof. Edwin Hancock as my supervisor. I would like to thank my supervisor for the help and support he has given me throughout my PhD. This thesis would have not been possible without his guidance, encouragement and support. I am very thankful to him for his time, understanding, patience and support. I would like to thank my assessor, Dr. William Smith, for his rigorous assessment of my work. I would like to thank Prof. Richard C. Wilson for his valuable suggestions. My thanks also go to Dr. Adrian Bors and all the friends at York for their help, kindness and friendship. I am grateful to my colleagues and friends, Furqan Aziz, Gul-e-Saman, Abdul Haseeb Malik, Tasawer Khan, Usman Khan, Dr. Ahmad Shahid and Dr. Muhammad Shakeel for giving me strength, self belief and all the enjoyable moments. I owe a great deal to my parents and my sisters for their love, encouragement, support and patience. Especially, I would like to thank my wife Nazneen Haseeb and my

daughter Mariah Haseeb for their patience, sacrifices and support. They have always been there to provide me with all the moral support during the period of my PhD study in York.

I would also like to thank the University of Peshawar, Pakistan and the Higher Education Commission, Pakistan, who provided me with all the financial support I needed.

Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of the University of York. This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references.

Some of the material contained in this thesis has appeared in the following published conference and workshop papers:

- Haseeb, M. and Hancock, E. R. (2011). Feature Point Matching using a Hermitian Property Matrix. In *Proceedings of the First International Workshop on Similarity-Based Pattern Recognition*, SIMBAD'11, (pp. 321–332). Berlin, Heidelberg. Springer-Verlag.

- Haseeb, M. and Hancock, E. R. (2012). 3D Shape Classification using Commute Time. Eigenvector Sign Correction for Spectral Correspondence Matching, SSPR/SPR, (pp. 208–215). Berlin, Heidelberg. Springer-Verlag.

- Haseeb, M. and Hancock, E. R. (2012). Unsupervised Clustering of Human Pose using Spectral Embedding. In *Proceedings of Structural, Syntactic, and Statistical Pattern Recognition – Joint IAPR International Workshop*, SSPR/SPR, (pp. 467–473). Berlin, Heidelberg. Springer-Verlag.

- Haseeb, M. and Hancock, E. R. (2013). Eigenvector Sign Correction for Spectral Correspondence Matching. In *Proceedings of* $17th$ *International Conference on Image Analysis and Processing*, ICIAP. (accepted)

CHAPTER 1

Introduction

1.1 Introduction and Motivation

In this chapter we provide an introduction and motivation for the research work presented in this thesis. Graph spectral methods study the properties of a graph using its spectrum, i.e. the eigenvalues and their corresponding eigenvectors of the adjacency matrix or the closely related Laplacian matrix associated to the graph. Spectral graph theory has been extensively used in the field of computer vision and pattern recognition in recent years and has proved to be a powerful tool for different applications in the field. A large number of spectral methods have been developed in the computer science literature in recent years, appearing in the fields of graph theory, computer vision, visualization, computer graphics and machine learning. In this thesis, we aim to use the graph spectral techniques to solve problems in the field of computer vision. We develop three methods

using spectral technique for (1) point pattern matching, (2) human pose clustering using the skeletal representation acquired from Microsoft Kinect and (3) 3D shape classification using spectral embedding.

Point pattern matching is a fundamental step in many computer vision tasks, for instance, object tracking, object recognition, shape-from-motion and optical flow analysis. The problem of point pattern matching or correspondence matching is to find one to one correspondences between two point sets in a 2D space or in 3D space. The local features of objects in an image are represented by feature points, for instance, the corners or edges of rigid objects. Point pattern matching is used to solve the correspondence and registration problems in a wide range of disciplines, including computer vision, pattern recognition, computational geometry, image registration, molecular biology and computational chemistry. For example, in chemistry point pattern matching is used for protein structures alignment. In biometrics, it is used to match and verify the fingerprints or signatures of employees in automatic personnel identification systems. Point pattern matching is used in 3D scene reconstruction, in automatic cartography from photogrammetric measurements.

There is a vast literature addressing the point pattern matching problem in pattern recognition, which can be divided broadly into two groups i.e. spectral methods and non-spectral methods. In this thesis we focus on the spectral methods. The spectral methods use the eigenvalues and the eigenvectors of the affinity matrix. The spectral methods are very elegant and have been successfully used to solve this problem. However, these methods fail to match the points correctly, especially, when there is a difference in the size of the point set being matched or in the presence of structural noise. The importance of point pattern matching is emphasized by the large amount of work carried out on the subject in the literature (Shapiro & Brady 1992; Scott & Longuet-Higgins 1991; Carcassoni &

Hancock 2003; Sun et al. 2009; Aubry et al. 2011). In the literature, many different methods to address problems related to point pattern matching are presented. The work presented in this thesis has the aim of improving the existing spectral point pattern matching methods by the introduction of a Hermitian property matrix and using the complex eigenvectors of the Hermitian matrix for the purpose of point pattern matching.

Detecting the human pose is an important step in human behaviour analysis, action or gesture recognition. However, human pose detection is a challenging task because of the huge inter-limb and intra-limb feature variability in both still images and image sequences. To acquire the data, we use Microsoft Kinect for Xbox 360. Shotton et al. (Shotton et al. 2011) developed an algorithm for Microsoft Kinect to extract the human body pose from a single depth image. They segment the depth image of human body into its parts and obtain its skeletal model with a set of joint positions. They demonstrate that their algorithm is efficient and effective for reconstructing 3D human body poses even in the presence of partial occlusions, different points of view and under no light conditions. We use the spectral graph technique to cluster similar poses. The technique involves constructing a Hermitian matrix from the input skeleton and then embedding the pattern vectors constructed from the complex coefficients of the elementary symmetric polynomials of the eigenvalues and the complex eigenvectors of the Hermitian matrix into a pattern space for the purpose of clustering similar poses.

Rapid improvement in the digital technology for acquisition and processing of 3D shapes has led to an increase in the number of 3D objects available. The use of 3D shape has become very common in a number of applications including games, engineering, culture and medical research studies. Consequently, the field of 3D shape analysis has attracted the attention of many researchers. The basic aim of the 3D shape analysis is to develop 3D shape descriptors or signatures

that capture the important properties of the 3D shapes. The increasing interest of researchers in different fields motivates the need to develop such 3D shape descriptors because the currently developed descriptors for classification and retrieval of 2D shapes/images cannot be directly extended to 3D shapes. Therefore, new descriptors need to be developed that capture the local and global properties of the 3D shapes. The global properties describe the overall shape while the local properties describe the details of the shape. Unfortunately, developing such descriptors for 3D object processing is not a simple task. Recently, many shape descriptors have been developed based on spectral graph theory. In this thesis, we construct a novel 3D shape descriptor for the purpose of 3D object classification. We embed the shape using commute time embedding and use commute time distance computed from the eigenvalues and the eigenfunctions of the Laplace-Beltrami operator to describe the shape descriptor.

1.2 Goals

The ultimate goal of this thesis is to develop a framework using graph spectral methods and apply it to a variety of applications in computer vision, such as the correspondence matching and graph clustering problems. To achieve this, we focus on:

- Introduction of a graph representation by using a Hermitian property matrix where we associate two type of attributes to the edges and nodes of the graph. Binary attributes are associated to the edges and unary attributes are assigned to the nodes of the graph.

- Using the complex eigenvectors of a Hermitian property matrix for the purpose of point-pattern matching. The distance between each pair of points using a Gaussian weighting function is used as the binary attribute. The

angular information (SIFT orientations for experiments on real world data sets) is used as the unary attribute.

- Embedding the complex eigenvalues and eigenvectors of a Hermitian property matrix into the iterative alignment EM algorithm of Carcassoni and Hancock to make it robust to rotation and point-position jitter.

- Using the complex coefficients of the elementary symmetric polynomials constructed from the spectrum of a Hermitian matrix to establish feature vectors for the purpose of clustering human skeleton poses acquired from the Microsoft Kinect device for Xbox 360.

- Introduction of a 3D shape signature based on the commute-time embedding which is robust to changes in pose and topology.

- Using the coefficients of the elementary symmetric polynomials constructed from the eigenvectors to make the direction of the eigenvectors pair consistent with each other for the purpose of correspondence matching.

1.3 Thesis Overview

The previous section outlined the overall goals of the thesis. Next, the structure of the thesis is presented. In order to set the problem in context, Chapter 2 will review the literature associated with spectral graph theory, correspondence matching, graph embedding and clustering and shape segmentation / classification.

Chapter 3 introduces the problem of correspondence matching and the graph spectral approaches to solve it. The use of a Hermitian property matrix (complex Laplacian) is introduced to improve the performance of two existing correspondence matching methods, and the performance of the new approach is compared

with the existing techniques. In this chapter, a Hermitian property matrix is constructed from the 2D feature point locations extracted from a pair of images and the angular information associated with these points. We use spectra of a Hermitian property matrix to compute the correspondence matching between the pair of point sets. The Hermitian matrix is constructed in such a way that it reflects the Laplacian matrix (degree matrix minus adjacency matrix). The complex eigenvectors of the Hermitian matrix are embedded into the EM framework proposed by Hancock and Carcassoni (Carcassoni & Hancock 2003) to render it robust to the point-position jitter and rotation.

In Chapter 4, we use the spectrum of a Hermitian property matrix and the coefficient of the symmetric polynomials to cluster different human poses taken by an inexpensive 3D camera, the Microsoft Kinect for Xbox 360. A Hermitian property matrix is constructed from the joints and the angles subtended by each pair of limbs using the three-dimensional skeletal data delivered by the Kinect device. To compute the angles between a pair of limbs, a line graph is constructed from the given skeleton. We construct four different types of pattern vectors from the complex coefficients of the elementary symmetric polynomials computed from the complex eigenvectors of the Hermitian property matrix. The pattern vectors are embedded into a pattern-space using two classical embedding methods i.e. PCA and MDS.

In Chapter 5, the problem of eigenvector sign correction for the problem of correspondence matching is addressed. This problem is solved using the coefficient of the symmetric polynomials computed from the eigenvectors of the proximity matrices for the corresponding point sets.

In Chapter 6, a commute-time based 3D shape descriptor is developed that is robust to changes in pose and topology. A new and completely unsupervised mesh segmentation algorithm is proposed, which is based on the commute time

embedding of the mesh and k-means clustering using the embedded mesh vertices.

Finally, Chapter 7 provides conclusions and summarises the work presented in the thesis and the results obtained. Some advantages and shortcomings of the methods described in the thesis are discussed and some possible extensions are proposed.

CHAPTER 2

Literature Review

In this chapter, we review the main literature which is relevant to the research presented in this thesis. The aim of the thesis is to develop efficient methods for four related problems using graph spectral techniques. To comply with this aim, we divide the content of the chapter into six parts. We commence reviewing the spectral graph theory and its applications in the area of image segmentation, graph clustering and graph matching in Section 2.1. We review spectral correspondence matching in Section 2.2, followed by a review on eigenvector sign correction methods for correspondence matching in Section 2.3. We survey graph clustering and graph classification in Section 2.4, followed by a brief review of graph embedding methods in Section 2.5 that we will use to develop our methods in the following chapters. In Section 2.6, we review some methods on 3D shape analysis. Finally, we summarise the chapter.

2.1 Spectral Graph Theory

Spectral graph theory (Chung 1997; Biggs 1974; Mohar 1997; Cvetković et al. 1980) is the branch of mathematics that studies the properties of a graph in relationship to the eigenvalues and eigenvectors of the adjacency matrix or the closely related Laplacian matrix associated to the graph. The earliest literature on algebraic graph theory can be traced back to that of Collatz and Sinogowitz (Collatz & Sinogowitz 1957). This work focused on the co-spectrality of graphs and the fundamental inequalities for bounding the eigenvalues. Since then, a large body of literature has emerged aimed at exploiting the relationship between the spectral and structural properties of a graph. This literature is well documented in several surveys including (Biggs 1974; Cvetković et al. 1980; Chung 1997; Mohar 1992). Spectral graph theory has been extensively used in the field of computer vision and pattern recognition and has proved to be a powerful tool for different applications in the field. The solution of almost every problem using spectral graph theory commences by constructing the adjacency matrix or closely related Laplacian matrix (i.e. the degree matrix minus the adjacency matrix). Once the graph is represented in terms of the adjacency matrices, or the Laplacian matrix, the possibility of using tools from linear algebra to study the properties of graphs is opened up. The graph spectrum refers to the set of eigenvalues of the adjacency or Laplacian matrix of a graph (Biggs 1974). The spectrum can be computed efficiently (Chung 1997) using linear algebra tools. The Laplacian matrix is a positive semi-definite i.e. all of its eigenvalues are non-negative. The spectrum of a graph contains many important properties of the graph. For instance, the isomorphism of two graphs can be determined by their eigen spectra. The multiplicity of the zero eigenvalue of the Laplacian matrix gives the number of connected components of the graph. The corresponding

eigenvector to the smallest non-zero eigenvalue (also called the Fielder vector) of Laplacian matrix can be used to divide the nodes of the graph into two disjoint subset of nodes. A number of data clustering and partitioning algorithms are based on this, for instance, (Shi & Malik 2000). Since these algorithms deal with partitioning the data into exactly two disjoint parts, therefore, these are applied recursively to find k clusters. The spectrum of the Laplacian matrix has recently been used to embed the nodes of a graph into a vector space. Then, the clusters of nodes are found using standard clustering techniques such as k-means. He et al. (He et al. 2005) used the eigenfunctions of the Laplace Beltrami operator on the face manifold for face recognition.

Recently, several authors have attempted to extend the utility of graph spectral methods using the complex property matrices (a Hermitian matrix). This is a natural way of incorporating angular or directional information with the proximity representation. For instance, Wilson, Hancock and Luo (Wilson et al. 2005) extended the Laplacian matrix to the complex domain and used the complex eigen spectrum to cluster similar binary shapes. Leuken et al. (Leuken et al. 2008) developed a shape retrieval method using a complex eigenvector corresponding to the smallest non-zero eigenvalue (Fielder vector) of a Hermitian property matrix. In the next sections we review some of the problems in the field of computer vision which have been solved using spectral graph theory.

2.2 Correspondence Matching

Point/feature matching in 2D images has been well studied in the computer vision community. The point correspondence methods can be broadly categorized into two types. The first type of methods tries to find a transformation matrix which aligns one point-set with another and then find the correspondences

between pair of points. The second type of methods are feature based methods. These methods assign shape descriptors to the points that are invariant to affine or perspective transformations. Correspondences are computed by comparing distances between these descriptors. The feature based methods can be further divided into two groups namely the non-spectral methods (Ling & Jacobs 2007; Chui & Rangarajan 2000; Lowe 2004) and the spectral methods (Shapiro & Brady 1992; Scott & Longuet-Higgins 1991; Umeyama 1988). Graph spectral methods compute the point descriptors using the eigenvalues and eigenvectors of the adjacency matrix or the Laplaican matrix constructed from the input point-sets. In Chapter 3 we address the problem of feature correspondence matching using graph spectral techniques, therefore, this section reviews some of the spectral correspondence matching methods.

Computing correspondence matching using graph-spectral techniques has proved to be a challenging task. Recently, there have been many attempts to use spectral graph theory both in the abstract problem of graph matching (Umeyama 1988) and point-set matching (Shapiro & Brady 1992; Carcassoni & Hancock 2003; Scott & Longuet-Higgins 1991; Mateus et al. 2008) problems. The problem of correspondence matching is often formulated in term of graph matching. The matches are located between the nodes of the graph by comparing the eigenvectors of the corresponding adjacency matrix or Laplacian matrix. The work of Umeyama (Umeyama 1988) is one of the earliest to use eigen-decomposition of the adjacency matrix for graphs of the same size to locate the correspondence matching. His method commences by constructing the adjacency matrices of the two graphs being matched. The singular value decomposition is performed on the adjacency matrices of the two graphs separately. The optimum matching between two weighted graphs is found by locating the least-square permutation matrix. Umeyama's method can be used for graph matching with both directed

and undirected graphs. However, the method can only be used on graphs that have the same number of nodes. The reason for this is that the eigenvectors of the adjacency matrix are unstable under the changes in the number of nodes of the graphs being matched.

Scott and Longuet-Higgins (Scott & Longuet-Higgins 1991) borrowed ideas from structural chemistry and developed an algorithm to match 2D feature-points in two images. Their study relies on the principles of proximity and exclusion, i.e. corresponding points must be close, and each point can have one corresponding point at most. It is believed that the human visual system uses these principles to establish correspondence between points in consecutive frame of a video clip. They used singular value decomposition on a Gaussian-weighted point association matrix between points from two different images. The correspondences are computed by maximizing the inner product of the proximity and exclusion matrices obtained using singular value decomposition. This method copes with 2D translations, expansion and shears, i.e. affine distortions. However, since this algorithm does not include the structural information within the image and gives equal importance to all the feature points, it fails to match the points correctly, especially, when there are large inter-image rotations or large inter-image scaling differences. Weiss (Weiss 1999) suggested using a normalized affinity matrix to improve the performance of the related clustering and matching methods. He concluded that if the matrix is correctly normalised the performance could be improved significantly. Xu and King (Xu & King 2001) developed an algorithm to solve the problem of weighted isomorphism that uses the eigenvalues and eigenvectors along with optimization techniques. They compute a permutation matrix by optimizing an objection function using principal component analysis PCA and the gradient descent. Pilu (Pilu 1997) suggested a modification of the method proposed by Scott and Longuet-Higgins to improve by adding the sim-

ilarity information along with the proximity information to compute the point association matrix. The similarity information is computed as the normalized correlation between neighbourhoods of the feature points.

To overcome the problems of Scott and Longuet-Higgins method, Shapiro and Brady (Shapiro & Brady 1992) developed a method, which uses the intra-image point proximity matrix rather than the inter-image point association matrix. The eigenvectors of the proximity matrices are compared to compute the correspondence across a pair of images. Correspondences are found by minimizing the Euclidean distance between rows of the modal matrices. Caelli and Kosibov (Caelli & Kosinov 2004) have improved Shapiro's method by re-normalizing the eigenvectors and locating the correspondences by maximizing the inner-product of the normalized eigenvectors.

Although spectral methods are robust they are sensitive to noise and structural errors. To cope with this problem several researchers have used the statistical framework of the EM algorithm. One of the earliest examples of using the EM algorithm for feature correspondence matching is the work of Cross and Hancock (Cross & Hancock 1998). They extend the standard EM algorithm by introducing structural consistency constraints to the correspondence matches. This is done by gating contributions to the expected log-likelihood function according to their structural consistency. This so-called dual step EM algorithm simultaneously locates point correspondence and parameters of the affine or perspective transformation matrix underlying the motion. Although their idea is clearly effective and novel, since it uses a dictionary based approach to compute the correspondence probabilities, it is computationally very demanding. Tang et al. (Tang et al. 2007) have used the Gaussian weighted Laplacian property matrix to compute correspondence matching from the eigenvectors of the Laplacian matrix along with the iterative framework of transformation estimation using the

thin-plate spline (TPS) deformation model (Chui & Rangarajan 2000). Luo and Hancock (Luo & Hancock 2001) developed a method using the statistical apparatus of EM algorithm and singular value decomposition SVD. They cast the problem of graph matching into a maximum likelihood framework. They treat the correspondences as hidden variables. Carcassoni and Hancock (Carcassoni & Hancock 2003) later improved the efficiency of the dual step EM algorithm by using the eigenvalues and eigenvectors of the point proximity matrix to compute the gating weights. More recently, Delponte et al. (Delponte et al. 2006) modified the method of Scott and Longuet-Higgins by introducing the scale invariant features (SIFT) to compute the association matrix. They employ affine invariant Harris corner detector to localize the feature point in the input images being matched. The association matrix is established based on the Euclidean distance between SIFT descriptors, completely disregarding the proximity information. Most recently, Sun et al. (Sun et al. 2009) introduced the Heat Kernel Signature (HKS) based on the heat kernel. HKS can characterize the shape up to isometry. However, the HKS is sensitive to low-frequency information. Ovsjanikov et al. (Ovsjanikov et al. 2010) used the HKS to develop the Heat Kernel Maps for shape matching. Aubry et al. (Aubry et al. 2011) have proposed a feature descriptor, the Wave Kernel Signature (WKS), using the Schrödinger equation, for correspondence matching of points on non-rigid 3D shapes.

The RANdom SAmple Consensus (RANSAC) algorithm developed by Fischler and Bolles in (Fischler & Bolles 1981) is an iterative robust parameter estimation procedure. It is designed from within the computer vision community, to cope with a large proportion of outliers in the input data. This is a random sampling technique that determines a coarse solution by using the minimum number observations required to estimate the underlying model parameters. The RANSAC algorithm is often used in the field of computer vision, to solve the

correspondence problem and estimate the fundamental matrix related to a pair of stereo cameras. Torr and Murray (Torr & Murray 1993) were the first to use the RANSAC method to estimate epipolar geometry. Since then, this algorithm has been used in a number of problems in computer vision. For instance, estimating the fundamental matrix to match two images (Schaffalitzky & Zisserman 2001), detecting geometric primitives (Clarke et al. 1996), extracting planes (Cantzler et al. 2002) and correspondence of points across image sequences (Kawakami et al. 2006; Hasler et al. 2003). An advantage of RANSAC is its ability to do robust estimation of the model parameters in the presence of large number of outliers are in the data set. A disadvantage of RANSAC is that it needs a large number of iterations to compute these parameters. When the number of iterations is limited to a small number, the solution obtained may not be optimal.

2.3 Eigenvector Sign Correction

The spectral techniques for correspondence matching use the eigenvectors of the proximity matrices constructed from the input point sets to compute the correspondences. The signs assigned to eigenvectors play a critical role in computing the correspondences. Several authors have proposed different methods to correct the direction of the eigenvectors. For instance, Park et al. (Park et al. 2000) have suggested a method to correct the direction of the eigenvectors by comparing the magnitude of the sum and difference of the two input eigenvectors. If the magnitude of the sum is greater than the magnitude of the difference then the directions of the input pair of eigenvector are consistent with each other, otherwise, sign of one of the eigenvectors needs to be flipped. Umeyama (Umeyama 1988) handles the problem of eigenvector sign correction by taking the absolute values of the components of both the eigenvectors. Caelli and Kosinov (Caelli

& Kosinov 2004) find the number of positive and negative components for each eigenvector. The sign of the eigenvector is flipped if the number of negative components is greater than the number of positive components. This is essentially a dominant sign correction, always ensuring that there are more positive entries in each eigenvector. Shapiro and Brady (Shapiro & Brady 1992) suggested a greedy approach to correct the direction of the eigenvectors.

2.4 Graph Clustering and Classification

In Chapter 4 we develop a method for clustering similar human body poses acquired using the Microsoft Kinect sensor. Here, the eigenvectors of a complex Laplacian matrix are used to compute the coefficients of elementary symmetric polynomials. Form these coefficients pattern vectors are established. The pattern vectors are embedded into a pattern space to cluster similar human body poses. Therefore, in this section we review some of the graph clustering and classification methods in the literature.

Data clustering is one of the important and widely used techniques for exploring the structures of data. It has recently found increasing support and applications in many areas ranging from statistics, computer science, biology to social sciences and psychology. Data clustering is the process of dividing the given set of data into meaningful groups. Clustering is unsupervised classification of data patterns based on some similarity measure (Jain et al. 1999). A good data clustering procedure should cluster the data in such a way that after clustering the data objects within the same group are more similar than those belong to different groups. This is usually done using some similarity or dissimilarity measure between each pair of data. The basic aspects in clustering are the pattern representation and the similarity or dissimilarity measure. The most popular

dissimilarity measure for metric representation is the distance, for instance the Euclidean distance. A large number of techniques have been developed for this problem in the literature under the name of unsupervised classification (Duda & Hart 1973). Early approaches of unsupervised data clustering methods include k-means and minimal spanning trees (Jain et al. 1999). When a small portion of data is already classified, then the semi-supervised classification methods are used. These methods can be broadly divided into two groups, namely statistical methods and graph theory based methods.

Statistical methods can be further divided into two groups i.e. the parametric and non-parametric methods. The parametric methods aim to draw patterns from data having a mixture of distributions, for instance mixture of Gaussian distributions. These methods estimate the parameters of those distributions. These include k-means (MacQueen 1967), the maximum likelihood estimation (Dempster et al. 1977) and the Expectation Maximisation (EM) algorithms. The aim of k-mean algorithm is to cluster the data into k groups by maximising the total inter-class variance. The EM algorithm originally developed by Dempster et al. (Dempster et al. 1977) is an iterative optimisation algorithm which estimates the parameters of a model. The maximum likelihood estimation algorithm finds the parameters of a mixture by maximising the log-likelihood function for the underlying probability distribution to the data. Some examples of non-parametric methods includes the histogram based estimation (Silverman 1986), kernel density estimation (Elgammal et al. 2003) and mean shift (Comaniciu et al. 2002). The histogram based methods tries to cluster data based on the histogram constructed from the data. In (Comaniciu et al. 2002) the authors proposed recursive mean shift estimation method for the analysis of a complex multi-modal feature space and to delineate arbitrarily shaped clusters in it. More recently, Shotton et al. (Shotton et al. 2011) proposed an algorithm to predict 3D positions of human

body joints from a single depth image, quickly and accurately using Microsoft Kinect device for Xbox 360. They use a huge set of human samples to infer pixel labels through Random Forest estimation, and skeletal model is defined as the centroid of mass of the different dense regions using mean shift algorithm resulting in the 3D joint proposals.

In recent years, graph theory based clustering methods have become more popular amongst the computer vision and the machine learning community (Kannan 2000; Ng et al. 2001; Bach & Jordan 2004; Zelnik-manor & Perona 2004). There has been a significant amount of work aimed at using spectral graph theory (Chung 1997) to cluster graphs. The spectral graph theory configures the graph clustering problem as a graph cut where a suitable objective function needs to be optimized. The basic idea behind this framework is to use the information conveyed by the eigenvalues and eigenvectors of the adjacency matrix or the Laplacian matrix of the weighted graph obtained from the data. The data points are represented by the nodes of the graph, while the edges denote the similarity or dissimilarity between each pair of nodes. Thus the clustering problem become graph cut problem. For a large data set, spectral clustering can be used with a sparse similarity matrix.

In the earliest graph spectral clustering method (Donath & Hoffman 1972), the authors suggest the use of eigenvectors of an adjacency matrix to find partitions of the graph representing the data. Later Fiedler (Fiedler 1973) has proposed using the eigenvector corresponding to the smallest non-zero eigenvalue of the Laplacian matrix. Since then, a significant amount of work has been done in this area. Scott and Longuet-Higgins (Scott & Longuet-Higgins 1990) construct a proximity matrix to measure the dissimilarities between image features and then use the eigenvalues and eigenvectors of the proximity matrix to partition the image features into clusters. They relocate the eigenvectors of the affinity mat-

rix to refine its block structure. Sarkar and Boyer (Sarkar & Boyer 1996) used leading eigenvectors of the affinity matrix to locate line segments grouping in images. Weiss (Weiss 1999) suggested the use of normalized affinity matrix to improve the performance of the related clustering methods. Shi and Malik (Shi & Malik 2000) use normalized cut to balance the cut and the association between the nodes of the graph. They use the Fiedler vector recursively (second smallest eigenvector of the Laplacian matrix) for the purpose of image segmentation. Ng et al. (Ng et al. 2001) embed the graph into a vector space and then cluster the point in the space using k-means algorithm. Wilson et al. (Wilson et al. 2005) construct feature vectors which are permutation invariants from a graph by applying elementary symmetric polynomial to elements of the spectral matrix derived from the Laplacian matrix. They used the spectrum of a complex Laplacian matrix to cluster shock graphs extracted from 2D shapes. Qiu and Hancock (Qiu & Hancock 2007) have used commute time for the purpose of image segmentation and show that the commute time method outperforms the normalized cut. A comparison of some spectral clustering methods have been detailed in (Luxburg 2007). More recently, Gangopadhyay et al. (Gangopadhyay et al. 2012) used Laplacian matrix and k-means to study the deterioration of renal functions after kidney transplant. Xiao et al. (Xiao et al. 2009) explored how permutation invariants computed from the heat kernel trace can be used to characterize graphs for the purposes of measuring similarity and clustering. The trace of the heat kernel is given by the sum of the Laplacian eigenvalues exponentiated with time. Ren et al. in (Ren et al. 2011) constructed pattern vectors from coefficients of the Ihara zeta function for the purpose of graph characterization.

2.5 Graph Embedding

In Chapter 6 we develop a 3D shape signature using commute time embedding that is robust with respect to changes in pose and topology. Here we use a combination of conventional and spectral techniques for better shape classification. Therefore, we review some of the data/graph embedding methods in this section. We also review some of the 3D shape analysis methods in Section 2.6.

The aim of graph embedding is to establish a mapping between graph and its vectorial representation. Once a graph is converted into a high dimensional vector, we are able to operate on graph in the vector space using linear algebra tools. For instance, the vector can be projected onto a low dimensional manifold for the purpose of analysis and visualization. In literature, a variety of embedding methods exist based on spectral graph theory. They all share the same principle of using the eigenvectors of the affinity or similarity matrix. For instance, Principal Component Analysis (PCA) (Hotelling 1933) and Kernel Principal Component Analysis (KPCA) (Schölkopf et al. 1998) use the leading eigenvectors of the covariance matrix to determine the projection directions with the maximal variance. Multidimensional Scaling (MDS) (Kruskal & Wish 1978) uses the eigenvectors of pairwise distance matrix to find an embedding of the data that minimized the cost function called stress. Its aim is to preserve the pairwise inner product by minimizing the differences of inner product from the input data and the vectorial data. The classical MDS preserves the inter-point distance if the input dissimilarity data is Euclidean. PCA and MDS are suitable when the low-dimensional manifold is embedded linearly in the vector space. Recently, a number of spectral embedding methods motivated by graph theory have been developed to deal with general non-linear manifolds. The isometric feature mapping (ISOMAP) (Tenenbaum et al. 2000) is an extension of MDS which preserve

the geodesic distance between data points located on a manifold. Some other related embedding algorithms include locally linear embedding (LLE) (Roweis & Saul 2000), Laplacian eigenmap (Belkin & Niyogi 2003), Hessian LLE (Donoho & Grimes 2003) and the diffusion map (Lafon & Lee 2006a). LLE is also a variant of PCA and preserves local structure by using linear coefficients to represent a data point by its neighbour points with coefficients and then attempting to preserve coefficients from the high-dimensional data space to the low dimensional manifold. LLE first finds the coordinates for data points on each local patch and then derives the global coordinates with the alignment of local patches by solving an eigenvalue problem. Laplacian eigenmaps attempts to preserve certain local geometric structure of the data by constructing an adjacency weight matrix from the data points and projecting the data onto the leading eigenvector of the resulting Laplacian matrix. Hessian LLE finds a low-dimensional configuration of points by using the estimated Hessian over neighbourhood as the Laplacian matrix. The diffusion map is a variance of Laplacian eigenmaps and constructs the Laplacian matrix by using a kernel function.

Luo et al. (Luo et al. 2003) proposed using the leading eigenvectors of the graph adjacency matrix to define eigenmodes of the adjacency matrix, which were use to construct a vector representation for the graphs. And then, embed these vectors into eigenspaces with the use of the eigenvectors of the covariance matrix of the vectors for the purpose of graph classification. In a similar approach (Wilson et al. 2005) used the coefficients of symmetric polynomials to construct the vectorial representation of the graphs from the spectrum of the Laplacian matrix. Robles-Kelly and Hancock (Robles-Kelly & Hancock 2007) solve the problem of matching the nodes of a pair of graphs by embedding the nodes of the graph onto a Riemannian manifold. This is done by applying a doubly centred multidimensional scaling technique to the Laplacian matrix com-

puted from the edge-weights. The embedding coordinates are given by the eigenvectors of the centred Laplacian. Then the problem of matching the nodes is viewed as the alignment of the embedded point sets.

2.6 3D Shape Analysis

Three dimensional shapes are being used in a large number of application including games, engineering, archaeology, biometrics, medical imaging etc. The discrete representation of 3D shape in the computer is a mesh, or sometimes a point set. Rapid advancement in the digital technology in 3D shape acquisition and processing has increased the availability of large amount of 3D shapes easily accessible. Consequently, the field of 3D shape analysis has attracted many researchers' attention. Analysis of 3D shapes involves object tracking, object recognition, registration, correspondence matching etc, and it aims to establish shape descriptors or signatures which capture the important properties of the shapes for the purpose of classification, clustering, retrieval and correspondence matching. These descriptors need be developed in a way that captures the local and global shape characteristics of the object. Unfortunately, developing such descriptors/signatures for the processing of a 3D object is not a trivial task. Osada's work reported in (Osada et al. 2001) is one of the earliest works on 3D shape representation for classification and retrieval. They computed a number of different types of global shape distributions to represent 3D objects using different shape functions. They used the angle between three random points on the surface of a 3D shape (A3), the distance between a fixed point and one random point on the surface of the shape (D1), the distance between two random points on the surface of a 3D shape (D2), the square root of the area of the triangle between three random points selected on the surface the 3D shape (D3) and the

cube root of the volume of the tetrahedron between four randomly selected points on the surface the 3D shape (D4). Ohbuchi et al. (Ohbuchi et al. 2005) modified the D2 shape function proposed by Osada et al. to improve its retrieval performance. They used the distance between the randomly selected points along with the mutual angle of the triangles on which the pair of points is located. Unlike the D2, which is a 1D histogram, they used 2D histograms.

Spectral methods for 3D shape analysis use the eigenvalues and eigenvectors/ eigenfunctions of some appropriately defined operator on the shape. In the recent past, a large number of spectral methods have been developed in many related fields including computer vision, machine learning, computer graphics and visualization etc. The methods are developed to solve different problems, for instance, correspondence matching, segmentation, shape smoothing and surface reconstruction etc. Early work on spectral shape analysis can be traced back to that of Taubin in 1995 reported in (Taubin 1995). In this study the author introduced the use of Laplacian operators for the purpose of 3D shape smoothing. In (Kolluri et al. 2004) the authors introduced the use of spectral graph partitioning for surface reconstruction from noisy point cloud data.

Recently, the graph spectral methods defined in the context of clustering have been applied to 3D shape processing. In this context, spectral invariants such as the eigenfunctions of the Laplacian operator can be used for near-isometric shape matching. Cuzzolin et al. (Cuzzolin et al. 2008) and Lee et al. (Lee et al. 2008) have performed segmentation for mesh sequences. The former method computes only protrusions, while the later uses an additional skeleton. In (Cuzzolin et al. 2008), the authors use locally linear embedding (LLE) (Roweis & Saul 2000) to represent a cloud of points and perform segmentation in the LLE space. The segments obtained are then propagated across time to obtain a temporal coherent segmentation of a voxel-sequence into protrusions of the shape. The method

works well for rigid body parts (such as head, hands and legs etc), but cannot be used directly for identifying rigid body-parts (for example, separating the upper-arm from the lower-arm). Graph spectral techniques have also been using in a number of correspondence matching and shape registration algorithms. For instance, Mateus et al. (Mateus et al. 2008) used eigenmaps obtained by the first k eigenfunctions of the Laplace operator as low-dimensional Euclidean representations of non-rigid shapes for the purpose of 3D point registration.

Spectral methods have recently been used in a number of algorithms to measure the similarity of 3D shapes. For instance, Rustamov (Rustamov 2007) has suggested using the eigen-decomposition of the Laplace-Beltrami operator to construct an isometric invariant surface representation. The Global Point Signature (GPS) proposed by Rustamov for shape comparison used the eigenvalues and eigenfunctions of the discrete Laplace-Beltrami operator, which closely resembles the diffusion map. The major drawback of his signature was its ambiguity to sign flips of each eigenfunction. Sun et al. (Sun et al. 2009) introduced a point signature based on the properties of the heat diffusion process on a shape, referred to as the Heat Kernel Signature (HKS). HKS captures all the information about the shape contained in the heat kernel, and can characterize the shape up to isometry. Castellani et al. (Castellani et al. 2011) have used HKS to develop the so-called Global Heat Kernel Signature (GHKS) by accumulating the local heat kernel values at each point into a histogram for a fixed number of scales. Ovsjanikov et al. (Ovsjanikov et al. 2010) used a heat diffusion process to construct the Heat Kernel Maps for 3D shape matching. Aubry et al. (Aubry et al. 2011) have proposed a feature descriptor based on a quantum mechanical approach, for correspondence matching of points on non-rigid 3D shapes.

2.7 Summary

In the previous sections of the chapter, we have reviewed the related literature on spectral graph theory. We also reviewed different methods developed using spectral graph theory to solve various problems from computer vision. Based on the review of the related literature, we may draw several conclusions.

First, although there is an ample amount of research on correspondence matching using graph spectral techniques, developing efficient point/feature matching algorithms is still a challenging problem to solve. Spectral methods offer an elegant approach to the problem of correspondence matching. However, these existing graph matching methods suffer from the curse of expensive computational complexity and their performance decreases in the presence of structural noise i.e. they give poor results when dealing with point-sets of different size. In this thesis, we aim to improve the performance of existing spectral methods for correspondence matching by introducing a Hermitian property matrix in a way that reflects the Laplacian matrix. A Hermitian property matrix captures more information about the input graph. Hence producing better correspondence results. In the literature, complex Laplacian matrix has been used for graph clustering (Wilson et al. 2005) and shape retrieval (Leuken et al. 2008) methods. However, it has not been used before for the correspondence matching.

Second, there is a substantial body of research on graph clustering, embedding and characterization using spectral techniques in the past few decades with lesser focus on the use of the Hermitian matrix which encodes the graph using complex numbers. Additional information in the form of angles between pair of limbs is encoded in the Hermitian matrix. Later, in the thesis, we show how feature vectors can be established by using the eigenvectors of the Hermitian matrix and the coefficients of elementary symmetric polynomials to cluster similar hu-

man body poses acquired from the Microsoft Kinect sensor.

Third, graph spectral methods solve a problem using the eigenvalues and eigenvectors of the adjacency matrix or the Laplacian matrix. However, the directions of the eigenvectors computed by the numerical solver are arbitrary, which causes problems in computing correspondences between pair of points. In the literature many methods have been proposed to solve this problem. However, none of them is robust. Later, we will show how the coefficients of elementary symmetric polynomials can be used to correct the directions of the corresponding eigenvectors for the problem of correspondence matching.

CHAPTER 3

Feature Point Matching using a Hermitian Property Matrix

In this chapter we investigate the spectral approaches to the problem of point pattern matching. We construct a Hermitian property matrix from the point locations and the directional information associated with them. We use spectra of a Hermitian property matrix to compute the correspondence matching between the pair of point sets. We construct a complex matrix which reflects the Laplacian matrix (degree matrix minus adjacency matrix). We embed the spectra of the Hermitian matrix into the EM framework proposed by Hancock and Carcassoni (Carcassoni & Hancock 2003) to render it robust to the point-position jitter and the rotation. The proposed method is compared with Shaprio's (Shapiro & Brady 1992), Scott's (Scott & Longuet-Higgins 1991) and Carcassoni's (Carcassoni & Hancock 2003) original alignment methods. Experiments on both synthetic and real world data are performed, which show that the new approach

gives encouraging results.

3.1 Introduction

Feature-point matching is very fundamental and one of the most important tasks in computer vision. Correspondence matching between 2D images and more recently, 3D objects is the pre-processing step for a number of computer vision algorithms. These include visual object tracking, object recognition using corners and edges, shape-from-motion (Jerain & Jain 1991; Tomasi & Kanade 1992; Costeira & Kanade 1998), stereopsis (Dornaika & Chung 1999), optical flow analysis (Weng et al. 1997) and morphable or deformable models. Such methods have found applications in many fields. For instance, in chemistry, it can be used to align the protein structures, to find the similarity between them. In the field of biometrics it can be used for automatic personal identification based on the finger prints or signatures. Once the correspondences are computed, further analysis can by performed, for instance, recovery of 3D structure of object (Tomasi & Kanade 1992), localization of objects in the image and finding the number of moving objects in the image sequence (Costeira & Kanade 1998).

3.1.1 The Correspondence Problem

The problem of feature correspondence matching is to find a one-to-one correspondence between feature points in a pair of 2D images which represent an object in the image or in 3D shapes. The images can be taken from a different point of view, at different times. In literature many different methods have been presented to address the problem of correspondence matching. These methods can be broadly categorized into two classes namely the non-spectral methods (Ling &

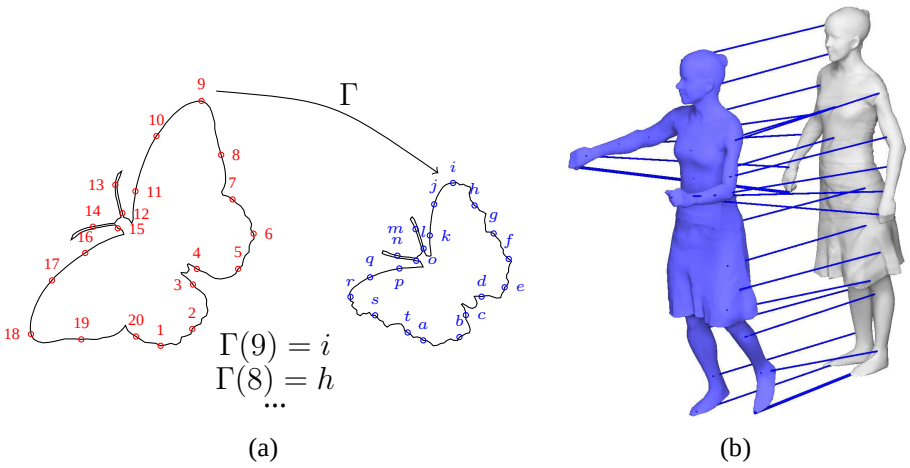


Figure 3.1: Correspondence Problem. a) 2D feature points. b) 3D points on articulated shapes.

Jacobs 2007; Chui & Rangarajan 2000) and the spectral methods (Shapiro & Brady 1992; Scott & Longuet-Higgins 1991; Umeyama 1988). Graph spectral techniques solve the problem using the eigenvectors and eigenvalues of the adjacency matrix or the Laplacian matrix (degree matrix minus the adjacency matrix) for the point set arrangement. Correspondences are computed by embedding the graphs into a common eigenspace using an eigen-decomposition of the point-proximity matrices, where correspondences are computed by the closest points matching in this eigenspace. Hence the feature correspondence matching problem is solved using weighted graph matching technique.

Let $G = (V_1, w_1)$, $H = (V_2, w_2)$ be weighted graphs with n nodes, where V_1 and V_2 are set of the vertices and w_1 and w_2 are the weights defined on the edges of the graphs G and H respectively. The weighted graph matching problem is the problem of finding a one-to-one correspondence Γ between $V_1 = \{v_1, v_2, \ldots, v_n\}$ and $V2 = \{v'_1, v'_2, \ldots, v'_n\}$ which minimizes some cost

function J defined as

$$J(\Gamma) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(w_1(v_i, v_j) - w_2(\Gamma(v_i), \Gamma(v_j))\right)^2$$

if A_G and A_H are the adjacency matrices of the graph G and H respectively then the cost function J can be written as function of a Permutation matrix P.

$$J(P) = ||PA_GP^T - A_H||^2$$

where the permutation matrix P represents the node correspondence Γ and $||.||$ is the Euclidean norm. Thus the problem of weighted graph matching is reduced to the problem of finding the permutation matrix P which minimizes the cost function $J(P)$.

3.2 Graph Spectral Matching

Graph spectral methods solve the feature points correspondence matching problem by constructing a graph representation for the given feature point sets. The feature points are represented by the nodes of the graph. Each edge of the graph corresponds to some spatial relationship between a pair of feature points. This relationship is usually computed using a weighting function of the Euclidean distance between each pair of points. These weights represent the similarities (or dissimilarities) between each pair of points. The most frequently used weighting function is the Gaussian function. However, various other similarity (or dissimilarity) measurement functions can also be used. For instance Carcassoni and Hancock (Carcassoni & Hancock 2003) have used Gaussian, sigmoidal, Euclidean and increasing weighting functions and have shown that the best performance is obtained using the increasing weighting function. The weights computed

for each edge of the graph are put in the form of a weighted adjacency matrix W. In literature both the adjacency matrix and the Laplacian matrix have been used as the proximity matrix to compute the correspondences. For instance, Umeyama (Umeyama 1988) has used an adjacency matrix while Tang et. al. (Tang et al. 2007) have used a Gaussian weighted Laplacian matrix. The elements W_{ij} of the matrix W stores the similarity or dissimilarity relationship between feature point x_i and x_j. Once the property matrix W is to hand, we subject it to the eigen-decomposition, to compute the eigenvalues and eigenvectors of the graph. Correspondences are computed from the eigenvalues and eigenvectors computed for both the input graphs.

The aim of the graph spectral methods is to embed the similarities of the input graph constructed from the feature points into a common eigen-space, where correspondence matching is performed. Since we are dealing with objects subject to different affine geometric transformations including translation, rotation, scaling and also some deformation, therefore, the similarity weighting function should be invariant under these transformations. Since, the Euclidean distance is invariant to translation and rotation, therefore, weighting function used by many methods are functions of the Euclidean distance between pair of feature points. For example, (Shapiro & Brady 1992), and (Scott & Longuet-Higgins 1991) have used a Gaussian weighting function to construct the proximity matrices.

$$W_{ij} = e^{-d_{ij}^2/2\sigma^2}$$

where d_{ij} is the Euclidean distance between the feature points x_i and x_j and σ is a constant parameter which controls the interaction of the feature points. Besides the Euclidean distance, the directional property of the feature points is an interesting example of the invariance of the rigid transformation.

A number of correspondence algorithms analyse the eigenvalues and eigen-

vector of the inter-image proximity matrix or the intra-image proximity matrices to find the correspondence between the points in the given pair of images, for instance (Scott & Longuet-Higgins 1991). A small change in the locations of points in one image will results in changes in the corresponding proximity matrix. These small changes are translated to small changes in the corresponding eigenvalues and eigenvectors computed. However, these changes appear in the trailing eigenvalues and their corresponding eigenvectors. This captures the global similarity between the feature points globally, which enables us to compute the correct correspondence between many of the points. The orthogonal property of the eigen-decomposition ensures that the obtained correspondences are unique.

In this chapter we aim to perform the correspondence matching of point-sets by using the directional/angular information associated with each feature point along with the Euclidean distance between each pair of feature points. We construct a complex Laplacian matrix, in which we encode both the angle and distance information about a feature points in the form of complex numbers. We use the SIFT (Lowe 2004) algorithm to acquire the angles at the extracted feature points from the two images to be matched. We use the point locations and their angles to construct a complex matrix (Hermitian). We compute the complex eigenvectors of the Hermitian property matrix. Correspondence matching is calculated by comparing the complex eigenvectors. We show how to embed the eigenvectors of Hermitian matrix in Carcassoni's EM algorithm for correspondence problem. The proposed method is more robust to noise, rotation and point-position jitter. In the experiment section, we compare our results with Shapiro-Brady's and Carcassoni's original alignment methods.

3.3 Hermitian Property Matrix

A Hermitian matrix H is a square matrix with complex elements that remains unchanged under the joint operation of transposition and complex conjugation of the elements. That is, the element in the i^{th} row and j^{th} column is equal to the complex conjugate of the element in the j^{th} row and i^{th} column, for all indices i and j, i.e. $a_{i,j} = \overline{a}_{j,i}$.

Hermitian matrices are named after Charles Hermite. In 1855 Charles Hermite proved that the eigenvalues of these matrices are always real. Following are a few important properties of a Hermitian matrix.

1. The diagonal elements of a Hermitian matrix are real.

2. The off-diagonal elements of a Hermitian matrix are complex number. Therefore, these can be a 2-component quantities, for instance, angular measurements.

3. The complex conjugate of a Hermitian matrix is a Hermitian matrix.

4. For a Hermitian matrix H, $H^{\dagger} = H$. The operation of transposition and complex conjugation is denoted by the dagger operator †

5. The eigenvalues of a Hermitian matrix are real.

6. The eigenvectors of a Hermitian matrix are complex and form an orthonormal basis. An $n \times n$ Hermitian matrix H has n orthonormal complex eigenvectors $u_1, u_2, ..., u_n$ sitting in the columns of the matrix U. i.e. $H = U\Lambda U^{\dagger}$, where $UU^{\dagger} = U^{\dagger}U = I$ and therefore, $H = \sum_{i=1}^{n} \lambda_i u_i u_i^{\dagger}$, where λ_i are the eigenvalues sitting on the main diagonal of the diagonal matrix Λ.

7. A real symmetric matrix is a special case of a Hermitian matrix.

3.4 Hermitian Matrix Construction

In this section we explain how we construct a Hermitian property matrix from the given images I and I' with m and n feature points respectively. We commence by constructing a complete graph for each set of feature points, where each pair of nodes is connected by an edge. The nodes of the graphs represent the feature points and the edges represent the similarity measurements between each pair of nodes which is some function of the Euclidean distance between the nodes. We use the SIFT (Lowe 2004) feature extraction algorithm to acquire angles at each feature point and assign them to the corresponding node. Once we have the feature point positions and the angles associated with them to hand, we can construct the Hermitian matrix. We construct it in a way that reflects the weighted Laplacian matrix. The Hermitian matrices H and H' for both of the graphs being matched are established.

3.4.1 Complex Laplacian Matrix

To commence, consider an undirected weighted graph denoted by $G = (V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges. The weight adjacency matrix A of the graph G is a $|V| \times |V|$ matrix, which is defined by:

$$A = [a_{ij}] = \begin{cases} w(v_i, v_j) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where $w(v_i, v_j)$ is the weight assigned to the edge between node v_i and v_j. The weight $w(v_i, v_j)$ is usually computed using a Gaussian-weighting function as $e^{-r_{ij}^2/2\sigma^2}$, where $r_{ij}^2 = \|v_i - v_j\|^2$ is the squared Euclidean distance between each pair of feature points.

To construct the weighted Laplacian matrix, we first establish the diagonal weighted degree matrix D, whose diagonal elements D_{ii} are given by the sum of the weights on the edges connected to the node i, i.e. $D_{ii} = \sum_j w(v_i, v_j)$. From the degree matrix and the adjacency matrix we can construct the standard Laplacian matrix. i.e. $L = D - A$. The elements of the Laplacian matrix are given as:

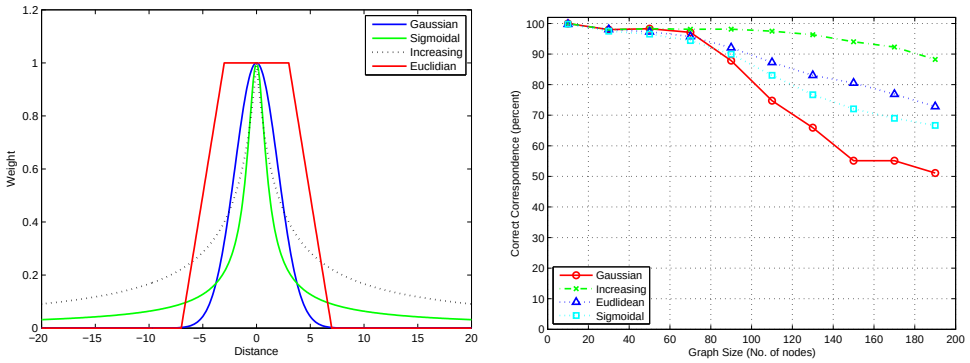
$$L = [l_{ij}] = \begin{cases} deg(v_i) & \text{if } i = j \\ -w(v_i, v_j) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where $w(v_i, v_j)$ is the weight assigned to the edge between node v_i and v_j and $deg(v_i)$ is the degree of the node v_i and is defined as $\sum_i w(v_i, v_j)$. The weight $w(v_i, v_j)$ is usually computed using a Gaussian-weighting function as $e^{-r_{ij}^2/2\sigma^2}$, where $r_{ij}^2 = \|v_i - v_j\|^2$ is the squared Euclidean distance between each pair of feature points.

Complex Laplacian matrix H is a Hermitian matrix which reflects the real weighted Laplacian matrix L. To construct the complex analog of the Laplacian matrix, we add the angular information to each element of the Laplacian matrix in the form of a complex number. The off-diagonal elements of H are calculated using a Gaussian-weighting function as:

$$H_{ij} = -e^{-r_{ij}^2/2\sigma^2} e^{\iota(\theta_i - \theta_j)} \tag{3.3}$$

where $r_{ij}^2 = \|v_i - v_j\|^2$ is the squared Euclidean distance between each pair of feature points. The parameter σ controls the interaction between features and $(\theta_i - \theta_j)$ is the difference between each pair of angles within the same image. The on-diagonal elements are given by the sum of the magnitudes of the elements



(a) (b)

Figure 3.2: Weighting Functions. a) Graph of the four weighting functions b) Performance of the four weighting functions on graphs of different sizes.

in the same row or in the same column of the matrix and hence are real numbers.

$$H_{ii} = \sum_{j \neq i} e^{-r_{ij}^2/2\sigma^2} \tag{3.4}$$

3.4.2 Weighting Functions

In (Carcassoni & Hancock 2003) the authors have suggested using different ways of constructing the weighted point-proximity matrix. They have used four different weighting functions, i.e. Gaussian, sigmoidal, Euclidean and increasing weighting functions and have shown that the increasing weighting function outperforms the others. These weighting functions are defined in the following subsections.

3.4.2.1 Gaussian Weighting Function

Using the Gaussian weighting function is the standard way to represent the adjacency relationship between the points. If i and j are two data points then the

corresponding elements of the proximity matrix H is computed as

$$H(i,j) = \exp(-d_{ij}^2/2\sigma^2)$$

where d_{ij}^2 is the squared distance between the points and σ controls the width of the weighting function.

3.4.2.2 Increasing Weighting Function

The following increasing weighting function can be used to compute the elements of the proximity matrix H.

$$H(i,j) = \frac{1}{1 + \frac{1}{s}||d_{ij}||}$$

where the parameter s controls the width of the function and d_{ij} is distance between points i and j

3.4.2.3 Sigmoidal Weighting Function

The following sigmoidal weighting function is used to construct the proximity matrix H.

$$H(i,j) = \frac{2}{\pi||d_{ij}||}\tanh(\frac{\pi}{s}||d_{ij}||)$$

where the parameter s controls the width of the function and d_{ij} is distance between points i and j

3.4.2.4 Euclidean Weighting Function

The Euclidean weighting function decreases linearly with the distance which can be define as

$$H(i,j) = \begin{cases} 1 & \text{if } d_{ij} < s_1 \\ 1 - \frac{1}{s_2 - s_1} & \text{if } s_1 < d_{ij} < s_2 \\ 0 & \text{otherwise} \end{cases}$$

where s_1 is the half-width of the ceiling of the function, s_2 is the half-width of the base and d_{ij} is distance between points i and j.

The graphs of those weighting functions are shown in Figure 3.2(a). We have empirically confirmed the performance of the those weighting functions on graphs of different sizes. Figure 3.2(b) shows that the performance of all the four weighting functions is same when used with smaller graphs (with less number of nodes). However, with increase in the number of node of graph being matched, the performance of the functions decreases. With larger graphs the best performance is obtained by using the increasing weighting function.

Since the matching performance of the weighting functions mentioned above is same for smaller graphs, therefore, we use Gaussian weighting function for the real-world data where the nodes are extracted from images. However for the synthetic data, with large random graphs of more than 60 nodes, we use the increasing weighting function. In case of the Hermitian property matrix, angular measurements are used to compute the complex elements of the matrix. Those elements are scaled by the weights computed using a Gaussian or increasing weighting function depending on the number of nodes in the graph.

3.4.3 SIFT Feature Orientation

To acquire angles at each node we use the SIFT (Lowe 2004) feature extraction algorithm. The angle/orientation at each feature point is calculated as follows. A gradient orientation histogram is computed in the neighbourhood of the feature point (using the Gaussian image at the closest scale to the feature point's scale). Peaks in the orientation histogram correspond to dominant direction of local gradients. For a point $I(x, y)$ in the image, the orientation $\theta(x, y)$ and the scale of gradient $m(x, y)$ are computed as:

$$m(x,y) = \sqrt{(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2)}$$

$$\theta(x,y) = tan^{-1}(I(x,y+1) - I(x,y-1))/(I(x+1,y) - I(x-1,y)))$$

Some feature points may have more than two or more peaks in their corresponding orientation histograms. In that case, an additional feature point is created at the same spatial location for the angles corresponding to the peaks in the histogram which are 80 percent of the maximum value of the histogram. Therefore, some of the points may have more than one orientation assigned to it.

A feature point in the first image of the sequence may have one angle assigned to it while the corresponding feature point in the second image may have two angle assigned to it or vice versa. This causes to change the number of feature points initially extracted from the image, which could in turn badly affect the computation of the correct correspondence matching. Therefore, we need to remove the extra angle associated with each point in both of the input images.

In figure 3.3 two frames of the CMU/VASC image sequence (left column) and their corresponding SIFT histograms of the extracted points (right column) have been shown . The first image (figure 3.3(a)) is the 1st frame while the second

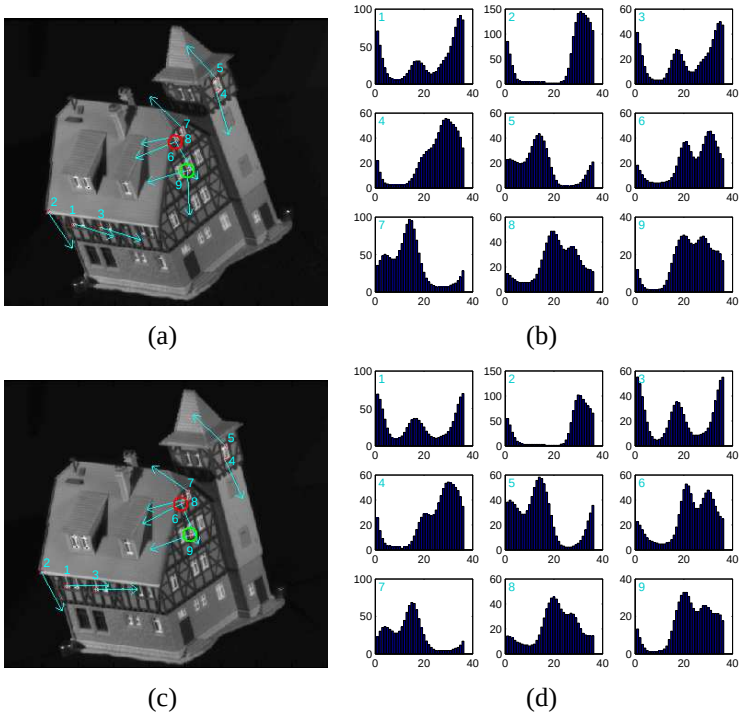


Figure 3.3: Feature points with multiple SIFT angles. a) CMU/VASC house sequence frame 1. b) Local gradient histograms of feature points in figure (a) on the left hand side. c) CMU/VASC house sequence frame 20. d) Local gradient histograms of feature points in figure (c) on the left hand side.

image (figure 3.3(c)) is the 20th frame of the sequence. Note that in both frames each feature point extracted has orientation(s) associated with it. There are two angles associated with one point (labelled as 6) in both of the frames (marked with red circle). However, there is a point (labelled as 9) in the first image to which two angles have been associated but the corresponding point (labelled as 9) in the second image has only one angle assigned to it (marked with green circle). This could cause wrong matching of the points. Therefore, we need to remove one of the angles from the feature point marked with green circle in the first image.

In this subsection we explain how to analyse the local gradient histograms computed at the feature points to remove any extra angle associated. We take a

simple but an effective approach. The local gradient histogram has the angular information stored in a histogram bin indexed by angular interval of ten degree each. We compute the normalized correlation between a pair of histograms to find how similar they are. Suppose the input histograms are A and B. The normalized correlation C between them can be computed as

$$C = \frac{\sum_{i=1}^{N}(A_i - \bar{A})(B_i - \bar{B})}{N \times \sigma(A)\sigma(B)} \tag{3.5}$$

where N is the number of bins in both the histograms A and B, $\bar{A}$ and $\bar{B}$ are the means of histograms A and B respectively, and $\sigma(A)$ and $\sigma(B)$ are the standard deviations of histograms A and B respectively.

First we normalize the gradient histograms by dividing all the bin values by the maximum value, so that in each histogram the maximum value becomes one. Then, we enumerate all those points which have more than one angle associated with it, in first image. We take the first point P_1 with more than one angles in the first image and search for the points having similar histograms within some radius r in the second image. Next, we suppress one of the angles in the histogram of P_1 in the first image and search again for the points having similar histogram within the radius r in the second image. Then, we suppress the other angle in the histogram of P_1 in the first image and find feature point having similar histogram in the same way. Now we have three similarity measures. If the first one is maximum of them, we keep both angles associated with P_1. If the second similarity measure is maximum of the three we remove the first angle at P_1 and keep the second one. In case, the third similarity measure is maximum of the three, we keep the first angle and remove the second angle at point P_1. The same procedure can be generalized to the case where more than two angles are associated to one feature point. We repeat this process for all the points having

more than one angle assigned to it. The extra angles associated to the feature points in the second image are removed by repeating the same procedure.

3.5 Correspondence Matching

In this section we describe the feature point correspondence matching algorithm and show how it uses the eigenvectors of a Hermitian property matrix. The idea behind the graph spectral methods for computing the feature correspondence is to use the eigenvectors of the graph as signature of the points in a high dimensional space. Each row of the modal/eigenvector matrix represents one point. Correspondences are computed by finding the distances between each pair of the rows of modal matrices.

Once we have H and H' to hand we perform the eigen decomposition, i.e. $H = V\Lambda V^T$ and $H' = V'\Lambda' V'^T$ where V and V' are the modal matrices of the images I and I' respectively, with complex eigenvectors as its columns, Λ and Λ' are the diagonal matrices with real eigenvalues along their principal diagonals. Each row of the modal matrix V is a $feature\ vector\ F_i$, while each row of the modal matrix V' is a $feature\ vector\ F'_j$.

$$V = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix}, V' = \begin{bmatrix} F'_1 \\ F'_2 \\ \vdots \\ F'_n \end{bmatrix}$$

The least significant $|m - n|$ eigenvectors and the feature vectors are discarded from the larger modal matrix, in the case where V and V' are of different sizes.

The next step is to calculate the correspondence probabilities matrix ζ from

the $feature\ vectors\ F_i$ of the image I and F'_j of the image I' by taking the Euclidean distances between each pair of the feature vectors of both images using the following binary decision.

$$\zeta_{ij} = \begin{cases} 1, & \text{if } j = \arg min_{j'} \|F_i - F'_{j'}\|^2 \\ 0, & \text{otherwise} \end{cases}$$

$i = 1...|m-n|$, $j = 1...|m-n|$. However, before computing the correspondence probabilities, the eigenvector normalization step is performed. Since, the eigenvector are complex, we add the angles of the eigenvector components in a head to tail fashion, and subtract the resultant angle from each eigenvector component so that $\arg(\sum_i \phi_{ij}) = 0$. Correspondence matches are given by the elements in the matrix ζ which are maximum (one) in their row and column.

3.5.1 Expectation Maximization

An Expectation-Maximization (EM) algorithm originally proposed by Dempster, Laird and Rubin (Dempster et al. 1977) is a method for finding maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. EM is an iterative method which alternates between performing an expectation (E) step, which computes the expectation of the log-likelihood evaluated using the current estimate for the latent variables, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found in the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

Although spectral methods are robust, they are sensitive to noise and structural errors. To cope with this problem several researchers have used the statistical framework of EM algorithm. One of the earliest examples of using EM

algorithm for feature correspondence matching is the work of Cross and Hancock (Cross & Hancock 1998). They extended the standard EM algorithm by introducing structural consistency constraints to the correspondence matches. This is done by gating contributions to the expected log-likelihood function according to their structural consistency. This so-called dual step EM algorithm simultaneously locates point correspondences and parameters of the affine or perspective transformation matrix underlying the motion. Since this method uses a dictionary based approach to compute the correspondence probabilities, it is very time consuming. Carcassoni and Hancock (Carcassoni & Hancock 2003) later improved the efficiency of the dual step EM algorithm by using the eigenvectors and the eigenvalues of the point proximity matrix to compute the gating weights for the expected log-likelihood function.

Here, we use the complex point proximity (Hermitian) matrix in the iterative framework of EM algorithm for point pattern matching proposed by Carcassoni and Hancock (Carcassoni & Hancock 2003). The experimental results show that embedding the Hermitian matrix into Carcassoni's method makes it more robust to the random point-position jitter and rotation.

3.5.2 Carcassoni's EM Algorithm

Suppose $\mathcal{T}^{(n)}$ is the affine geometric transformation matrix that best aligns a set of image feature points $\vec{w}$ with the feature points $\vec{z}$ in a model. Each point is encoded in homogeneous co-ordinates. i.e. $\vec{w}_i = (x_i, y_i, 1)^T$ and $\vec{z}_j = (x_j, y_j, 1)^T$. There are six transformation parameters, which model the translation in x and y directions, the rotation, the scaling, the shear in x and the shear in y direction.

These parameters are combined into the transformation matrix as

$$\mathcal{T}^{(n)} = \begin{pmatrix} t_{1,1}^{(n)} & t_{1,2}^{(n)} & t_{1,3}^{(n)} \\ t_{1,2}^{(n)} & t_{2,2}^{(n)} & t_{2,3}^{(n)} \\ 0 & 0 & 1 \end{pmatrix} \tag{3.6}$$

The new transformed co-ordinates are computed from the pervious co-ordinates using the following matrix multiplication

$$\vec{w}_i^{(n)} = \mathcal{T}^{(n)} \vec{w}_i^{(n-1)} \tag{3.7}$$

here the superscript n shows that the parameters are taken from n^{th} iteration. Carcassoni and Hancock's iterative EM algorithm matches point-features across a pair of images. They have shown how structural constraints can be embedded in an EM algorithm for point alignment under affine and perspective distortion. Graph-spectra are used to compute the required correspondence probabilities. Point correspondence matching and the parameters of the affine transformation matrix underlying the motion are simultaneously computed, so as to maximize the expected log-likelihood function:

$$Q(\mathcal{T}^{(n+1)}|\mathcal{T}^{(n)}) = \sum_{i \in D} \sum_{j \in M} P(\vec{z}_j|\vec{w}_i, \mathcal{T}^{(n)}) \zeta_{i,j}^{(n)} \times \ln p(\vec{w}_i|\vec{z}_j, \mathcal{T}^{(n+1)}) \tag{3.8}$$

where D is the set of data feature points $\vec{w}_i$, M is the set of model feature points $\vec{z}_j$. The measurement densities $p(\vec{w}_i|\vec{z}_j, \mathcal{T}^{(n+1)})$ model the distribution of error-residuals between the two point sets. The log-likelihood contributions at iteration $n+1$ are weighted by the a posteriori measurement probabilities $P(\vec{z}_j|\vec{w}_i, \mathcal{T}^{(n)})$ computed at the previous iteration. The individual contributions to the expected log-likelihood function are gated by the structural matching probabilities $\zeta_{i,j}^{(n)}$.

Under the assignment of Gaussian alignment errors, in the point positions, the correspondence probability matrix is give as

$$\zeta_{i,j}^{(n)} = \frac{\sum_{l=1}^{o} \exp[-\mu \parallel V_D^{(n)}(i,l) - V_M(j,l) \parallel^2]}{\sum_{j' \in M} \sum_{l=1}^{o} \exp[-\mu \parallel V_D^{(n)}(i,l) - V_M(j',l) \parallel^2]} \tag{3.9}$$

where $o = \min(|D|, |M|)$. The resulting matrix ζ has o rows and o columns.

3.5.2.1 E-Step

In the E step of the algorithm the a posteriori probabilities of the points $\vec{z}_j$ are updated. The a posteriori probabilities can be written in terms of the conditional measurement densities using the Bayes rule.

$$P(\vec{z}_j|\vec{w}_i, \mathcal{T}^{(n)}) = \frac{\alpha_j^{(n)} p(\vec{w}_i|\vec{z}_j, \mathcal{T}^{(n+1)})}{\sum_{j' \in M} \alpha_{j'}^{(n)} p(\vec{w}_i|\vec{z}_{j'}, \mathcal{T}^{(n+1)})} \tag{3.10}$$

where the mixing proportions are calculated as $\alpha_j^{(n+1)} = \frac{1}{|D|} \Sigma_{i \in D} P(\vec{z}_j|\vec{w}_i, \mathcal{T}^{(n)})$ The conditional measurement densities $p(\vec{w}_i|\vec{z}_j, \mathcal{T}^{(n)})$ can be defined in terms of a multivariate Gaussian distribution.

$$p(\vec{w}_i|\vec{z}_j, \mathcal{T}^{(n)}) = \frac{1}{2\pi\sqrt{|\Sigma|}} \times \exp\left[-\frac{1}{2}(\vec{z}_j - \mathcal{T}^{(n)}\vec{w}_i)^T \Sigma^{-1} (\vec{z}_j - \mathcal{T}^{(n)}\vec{w}_i)\right] \tag{3.11}$$

3.5.2.2 M-Step

The dual step EM algorithm originally proposed in (Cross & Hancock 1998) iterates between the two interleaved maximization steps. The first step maximizes the a posteriori probability correspondence estimating correspondence assignments. The second one locates maximum likelihood for alignment parameters estimation. The update formula to maximize the a posteriori probability of the

structural match is

$$f^{n+1}(i) = \arg\max_{j \in M} P(\vec{z}_j | \vec{w}_i, \mathcal{T}^{(n)}) \zeta_{i,j}^{(n)} \tag{3.12}$$

The maximum-likelihood affine transformation parameters $a_{k,l}^{(n+1)}$ for k=1,2 and l=1,2,3 are found by solving the following saddle-point equations, which can be solved using matrix inversion.

$$\frac{\partial Q(\mathcal{T}^{(n+1)} | \mathcal{T}^{(n)})}{\partial a_{k,l}^{(n+1)}} = 0 \tag{3.13}$$

$$\begin{aligned} \mathcal{T}^{(n+1)} = & \left[\sum_{i \in D} \sum_{j \in M} P(\vec{z}_j | \vec{w}_i, \mathcal{T}^{(n)}) \zeta_{i,j}^{(n)} \vec{w}_i U^T \vec{w}_i^T \Sigma^{-1} \right]^{-1} \\ & \times \left[\sum_{i \in D} \sum_{j \in M} P(\vec{z}_j | \vec{w}_i, \mathcal{T}^{(n)}) \zeta_{i,j}^{(n)} \vec{z}_j U^T \vec{w}_i^T \Sigma^{-1} \right] \end{aligned} \tag{3.14}$$

where Σ is the variance-covariance matrix for the position errors. The element of the matrix U are the partial derivatives of the affine transformation matrix with respect to the individual parameters, i.e.

$$U = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.15}$$

A set of improved transformation parameters are computed at each iteration. Once the improved parameters are found, the a posteriori measurement probabilities are updated using the Bayes theorem.

3.6 Computational Complexity

Following are the steps of the algorithm developed in this chapter.

3.6.1 Steps

1. N number of feature of points are extracted from the input images using a feature points detector (Harris & Stephens 1988) or (Lucas & Kanade 1981). Angles are computed using using Vedaldi's MATLAB/C implementation (Vedaldi 2006) of the SIFT detector.

2. A Hermitian matrix is constructed using the distances between each pair of feature points and the SIFT angles computed at each feature point.

$$H_{i,j} = e^{(-(d_{ij}^2/2\sigma^2))} \times e^{\iota(\theta_i - \theta_i)}$$

3. The eigenvalues and eigenvectors of the Hermitian matrix are computed.

$$H = \Phi \Lambda \Phi^T$$

4. The correspondence probabilities are computed to construct the association matrix ζ from the eigenvectors of the Hermitian matrices.

$$\zeta_{i,j}^{(n)} = \frac{\sum_{l=1}^{o} \exp[-\mu \parallel \Phi^{(n)}(i,l) - \Phi'(j,l) \parallel^2]}{\sum_{j' \in M} \sum_{l=1}^{o} \exp[-\mu \parallel \Phi^{(n)}(i,l) - \Phi'(j',l) \parallel^2]}$$

   where Φ and Φ' are the corresponding eigenvector matrices constructed using the feature points extracted from the two input images.

5. The association matrix ζ is embedded into the iterative framework of EM

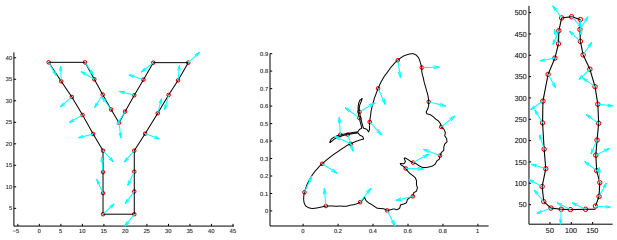
algorithm for point pattern matching proposed by Carcassoni and Hancock (Carcassoni & Hancock 2003).

3.6.2 Complexity

Let M and N be the number of feature points extracted from the two input images respectively. Without the loss of generality, we can assume that $M > N$. Step 2 takes quadratic time to construct the Hermitian matrix. The eigen-decomposition of each matrix takes cubic time in number of feature points, and so the total complexity of this step becomes O(M^3) + O(N^3). Finally computing the association matrix also takes quadratic time. Hence the total running time is O(M^2) + O(N^2) + O(M^3) + O(N^3) = O(M^3).

3.7 Experimental Results

In this section of the chapter, we provide some experimental investigations of the correspondence matching using the complex Laplacian matrix to evaluate its performance. We focus on its use in two different settings. The first is an investigation of using the standard proximity matrix and its Hermitian counterpart in the Shapiro-Brady (Shapiro & Brady 1992) algorithm. The second is a similar investigation for the Carcassoni-Hancock (Carcassoni & Hancock 2003) algorithm. In both settings, we experiment with synthetic and real world data. To compare the performance of using the Hermitian property matrix when deformations are present, experiments are performed on synthetically generated data where 2D translation, rotation and scaling are added. The effect of missing points and random point jitter in terms of 2D Gaussian random matrices with different covariance are also tested.



(a) (b) (c)

Figure 3.4: Synthetic Dataset

3.7.1 Synthetic Data

Here, we perform a number of experiments on the synthetic data to investigate the correspondence matching using the Hermitian property matrix. We perform experiments to evaluate our approach on four point sets which are generated as follows. First, we take 27 points along the border of the English alphabet letter Y. Second, we take 20 points along the silhouette of a butterfly. Third, we take 30 equally spaced points along the silhouette of a bottle. Finally, we take random point sets of size 25 to 500. The first three point sets are shown in the figure 3.4. Note that each feature point has a vector associated with it. We need the difference of angles associated with each pair of feature points to construct the Hermitian property matrix detailed in Section 3.3. We investigate two sources of error. The first of these is random measurement error or point-position jitter. Here we subject the positions of the points to Gaussian measurement error. The second source of error is structural. Here we randomly delete controlled number of points. This type of the error is most destructive for the spectral methods.

In our first experiment, we take a feature point set and make a copy of it. We apply different affine geometric transformation (i.e. translation, rotation and scaling) to the second copy. We construct Hermitian matrices from both of the locations of the feature points and the angles associated with them. We compute the correspondences from the corresponding eigenvectors of the two Hermitian

matrices as detailed in Section 3.5. Some of the results are shown in figure 3.5 which show that the proposed approach has the ability to compute the correct correspondences under different affine geometric transformations.

In the second experiment, we evaluate the performance of the proposed approach on the synthetic data with controlled point-position jitter. We choose 30 feature points taken along the silhouette of a bottle shown in figure 3.4(c). We take a copy of the point set and subject the positions and the angles associated with the feature points to Gaussian measurement error. We then compute the correspondences from the eigenvectors of the Hermitian matrix computed from both point sets. The results are compared with Shapiro-Brady algorithm applied to the same point sets. The correspondence results of both Shapiro & Brady algorithm and its Hermitian counterpart are shown in figure 3.6. The left column (figure 3.6(a) and figure 3.6(b)) shows the point matching using the Hermitian matrix. The right column (3.6(c) and 3.6(d)) shows the point matching using Shapiro-Brady (Shapiro & Brady 1992) method. The upper and lower rows have noise of $\sigma = 0.1$ and $\sigma = 0.2$ added respectively.

In the third experiment, the performance of our method is evaluated on the random point sets. We take random point set of size 25 to 300. We experiment on this data set on the problem of correspondence with random position jitter. Here we compare the results of using Hermitian property matrix, Shapiro & Brady algorithm and Tang et al. (referred to as Laplacian) algorithm with increasing point-position jitter. The results are shown in figure 3.7. The correct correspondence is shown as a function of the standard deviation of the Gaussian noise added. The results are the average of 100 runs for each value of standard deviation used to generate the random jitter.

To test the performance of the proposed approach on the point-set of different sizes we take random point-sets of size 20 to 450. We add a fixed amount of

Gaussian noise to them. The size of the two point sets being matched is same. We compare the correspondence results of the Hermitian property matrix with that of Shapiro & Brady algorithm and Tang et al. (referred to as Laplacian). In figure 3.8, the fraction of correct correspondences is shown as a function of the size of the point sets. The performance of all the three methods decreases with the increase in the size of the point sets because with the increase in the number of points, the average inter-point distance decreases. However, the correspondence matching using the Hermitian property matrix outperforms the other two correspondence matching algorithms. The results shown are the average of 100 runs on each size of the graph.

Next, we introduce the structural noise to the point sets by randomly deleting a controlled proportion of points. The effect of missing points for rigid point matching are shown in figure 3.9. Here, note that with the increase in the number of deleted points the performance of all of the three methods fall down abruptly. With 50% of structural error the performance of all of the three algorithms reach to zero.

We now turn to the use of Carcassoni's EM algorithm. We embed the complex Laplacian matrix into the framework of Carcassoni's algorithm to render it more robust to noise and rotation. We compare the results for the Shapiro & Brady algorithm, the original EM algorithm (referred to as Carcassoni) and the modified version (referred to as Carcassoni + Complex Laplacian) in figure 3.10. The results show that by embedding the complex Laplacian into the EM algorithm, on the average, clearly improves its performance by about 5 to 10%.

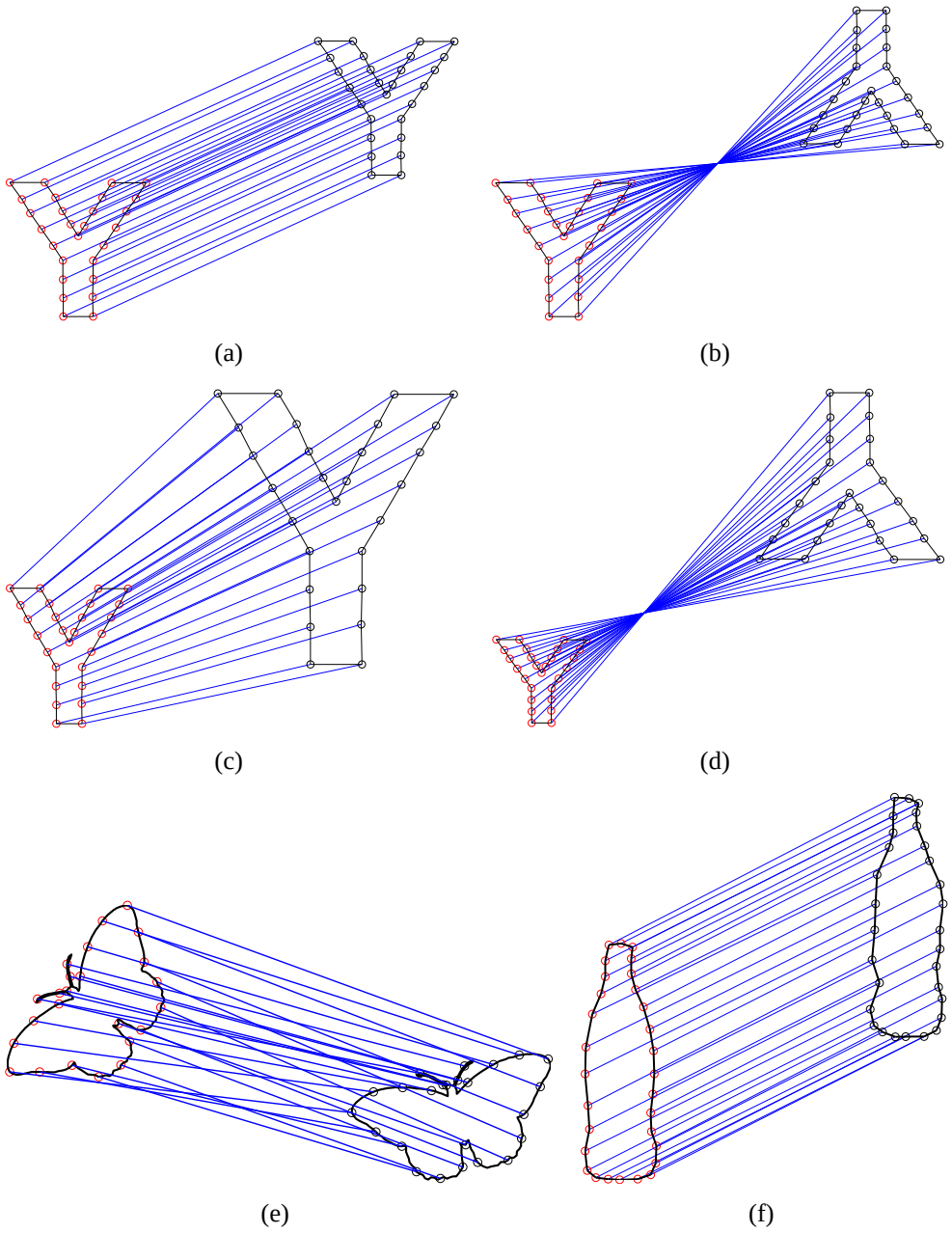


Figure 3.5: Correspondence matching results under different affine geometric transformations, Correspondence under a) Translation b) Rotation c) Scaling d) Scaling, rotation and translation e) Rotation f) Point-position jitter

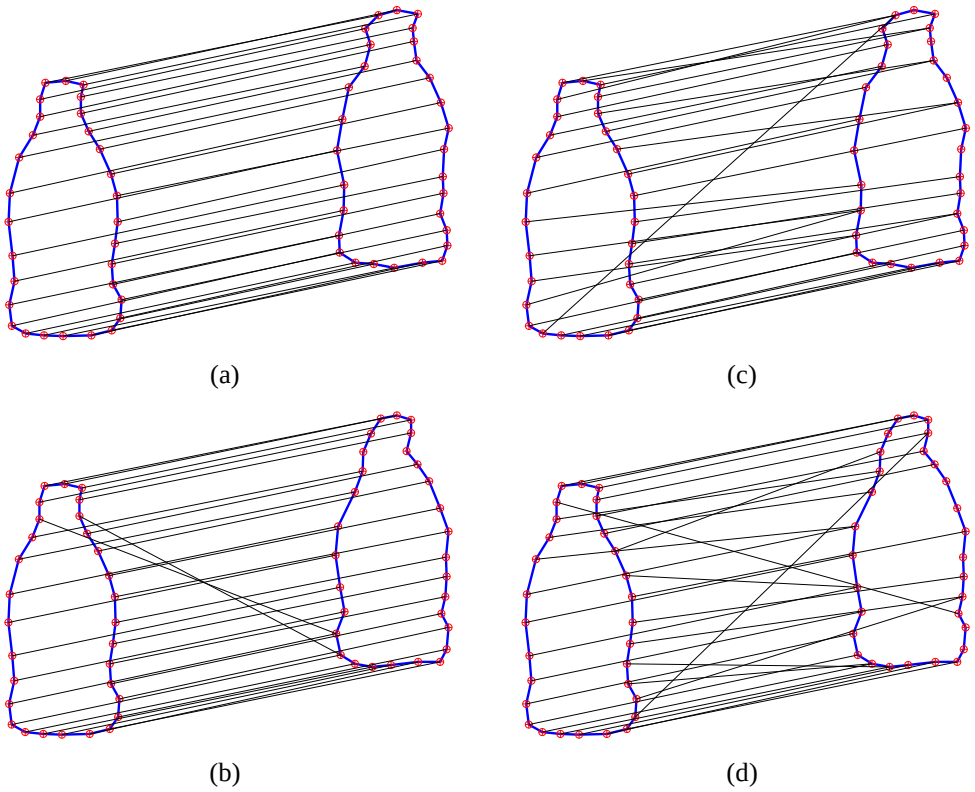


Figure 3.6: Correspondence matching with Gaussian noise added in point positions using (a) Hermitian matrix σ = 0.1 (b) Hermitian matrix $\sigma = 0.2$ (c) Shapiro-Brady method σ = 0.1 (d) Shapiro-Brady method $\sigma = 0.2$

3.7.2 Real-World Data

Our final piece of experimental work focuses on real-world data. For real-world data we evaluate our approach on images from two image sequences, namely, the CMU/VASC model-house sequence and the Swiss chalet model house sequence.

In the first experiment, we use the CMU/VASC model-house sequence. We compare our method (referred to as Hermitian) with other spectral point matching methods i.e. Scott and Longuet-Higgins (referred to as Scott) and Carcassoni's EM point alignment algorithm. Forty feature points are extracted using Kanade-Lucas-Tomasi (Shi & Tomasi 1994) feature point extractor from each

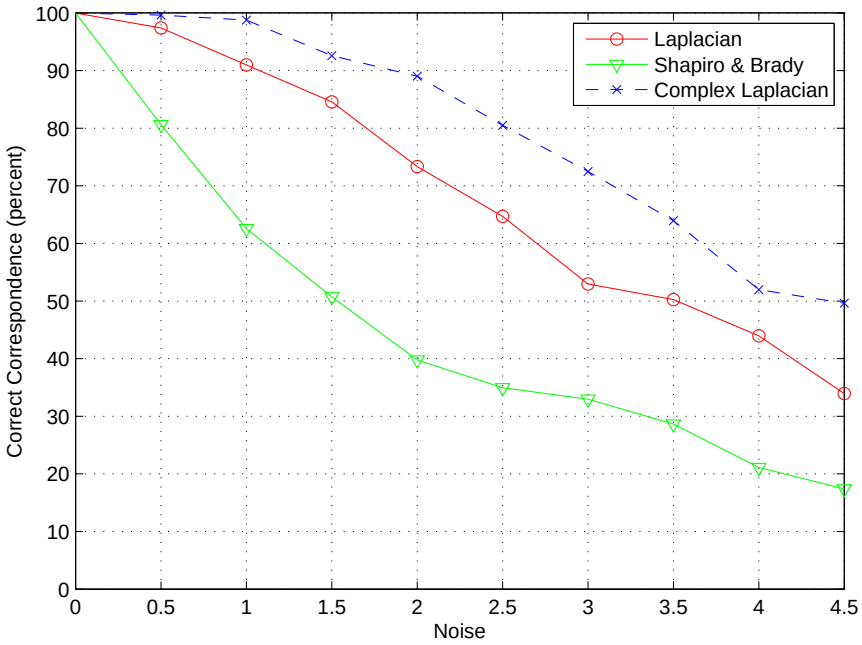


Figure 3.7: Effect of noise in point positions

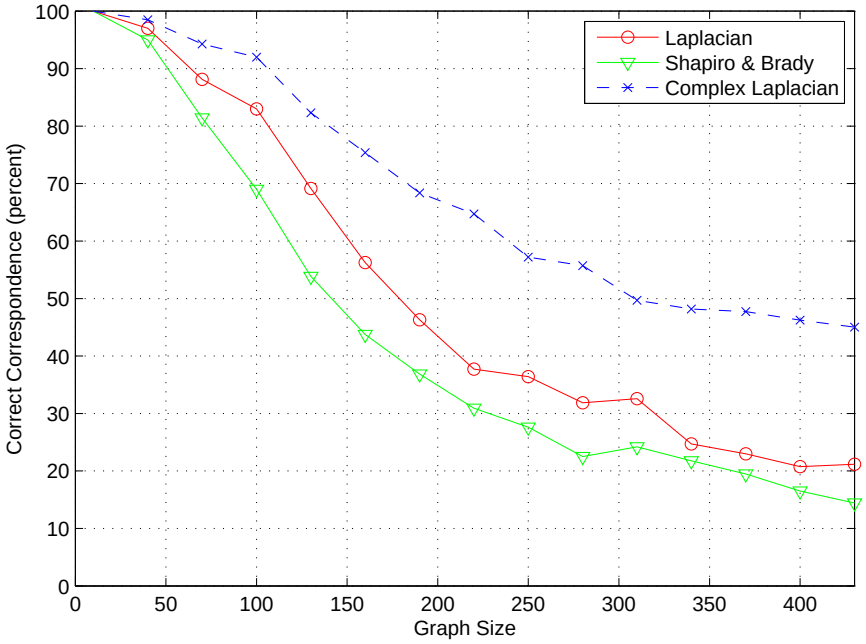


Figure 3.8: Effect of graph-size on correspondence matching

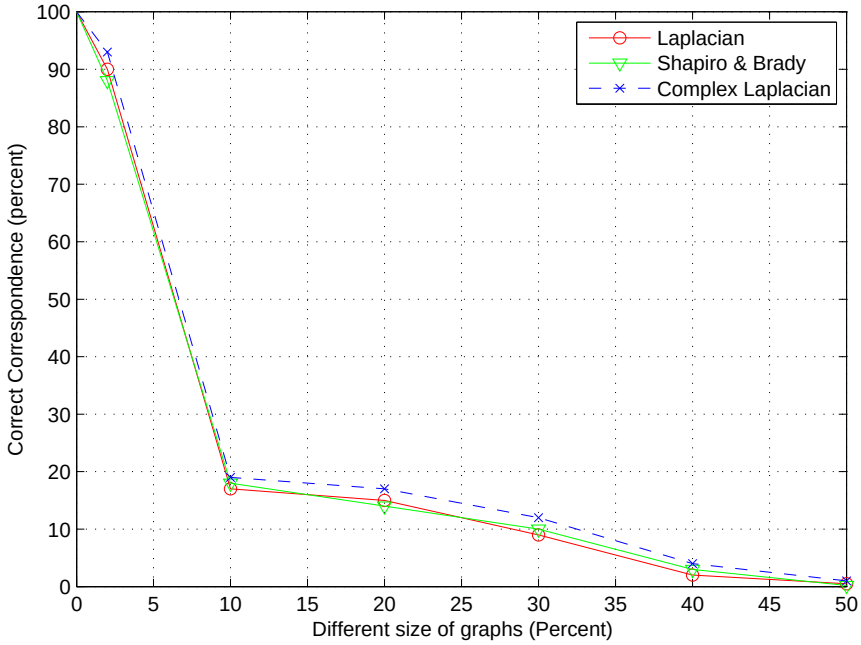


Figure 3.9: Effect of structural noise

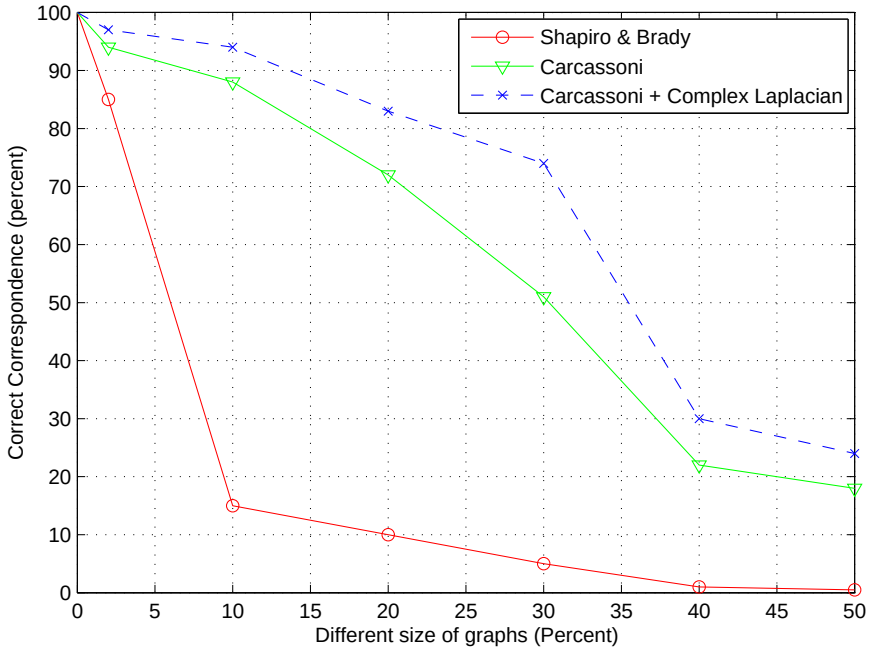


Figure 3.10: Effect of structural noise, using Carcassoni EM + Complex Laplacian

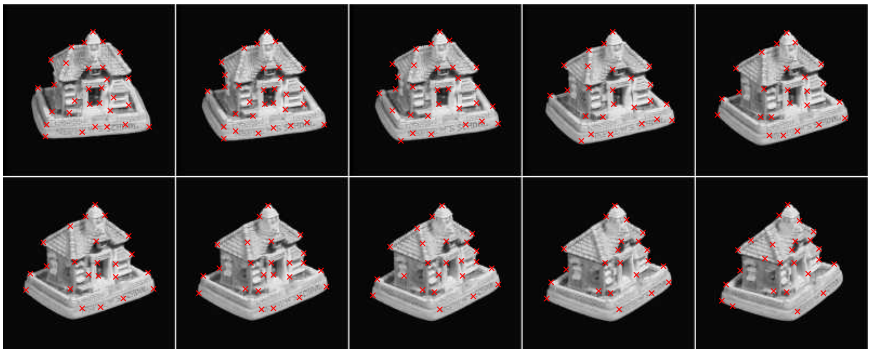
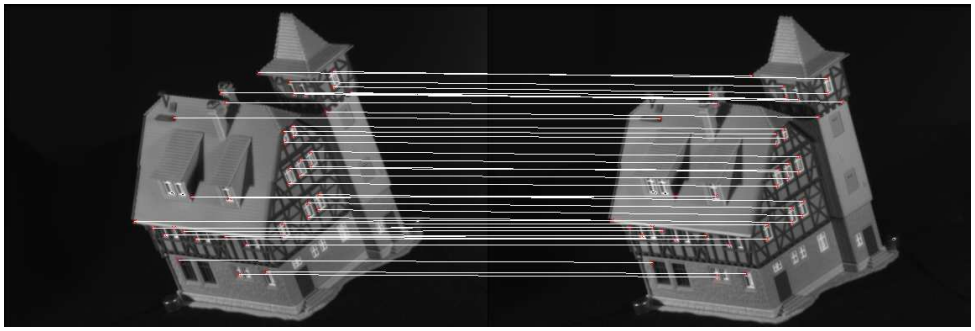
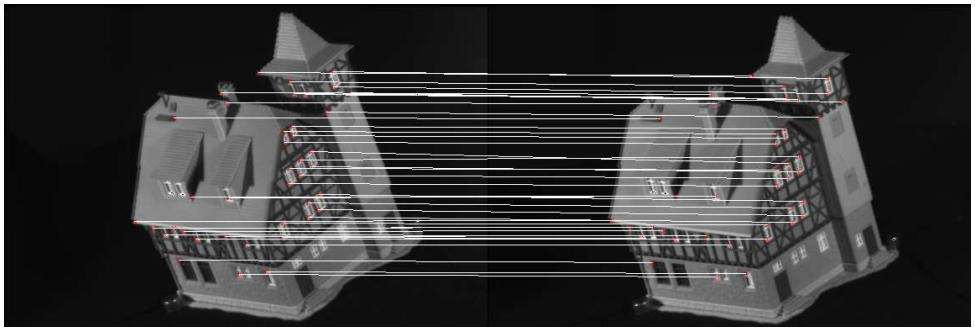


Figure 3.11: The Swiss chalet model house sequence, with the feature points extracted

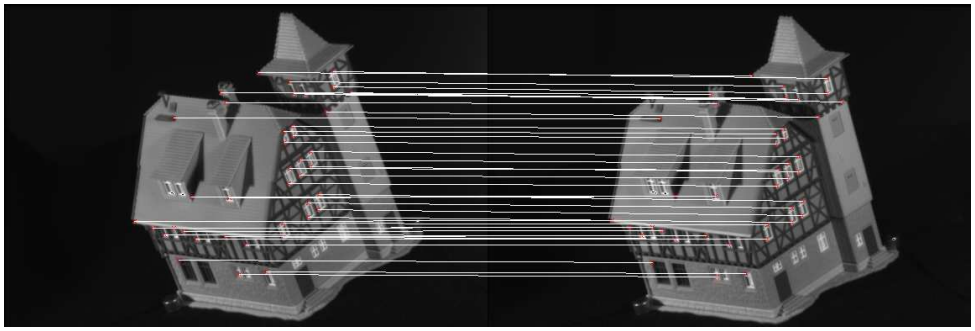
image. Hermitian matrices are constructed using equation 3.3 as explained in Section 3.4. The parameter σ controls the interaction between the feature points. The choice of the value of σ significantly affects the performance of the algorithm. Here we choose maximum of the x and y coordinates of all the feature points in the image as the value of σ. We compute angles, at the feature points localised, using Vedaldi's MATLAB/C implementation (Vedaldi 2006) of the SIFT detector and descriptor. Correspondences are computed between the 1^{st} frame and the 20^{th}, 40^{th}, 60^{th}, 80^{th} and 100^{th} frames. Figure 3.18 shows the correspondence matching results of the three methods mentioned above. The matching results of the 1st frame of the sequence with the other frames are shown in figure 3.12 to figure 3.16. Here, the first pair of frames (top) is the result produced by the Scott and Longuet-Higgins (Scott & Longuet-Higgins 1991) algorithm. The second pair(middle) is the correspondence result of the EM algorithm developed by Carcassoni and Hancock while the third pair (bottom) is the result obtained when the Hermitian matrix is embedded in Carcassoni and Hancock's algorithm. Figure 3.17 shows the matching between the 1^{st} frame and the 10^{th} frame of the CMU/VASC sequence. There are 6 incorrect matches using only spectral information. However, there are not any wrong correspondences when



(a)



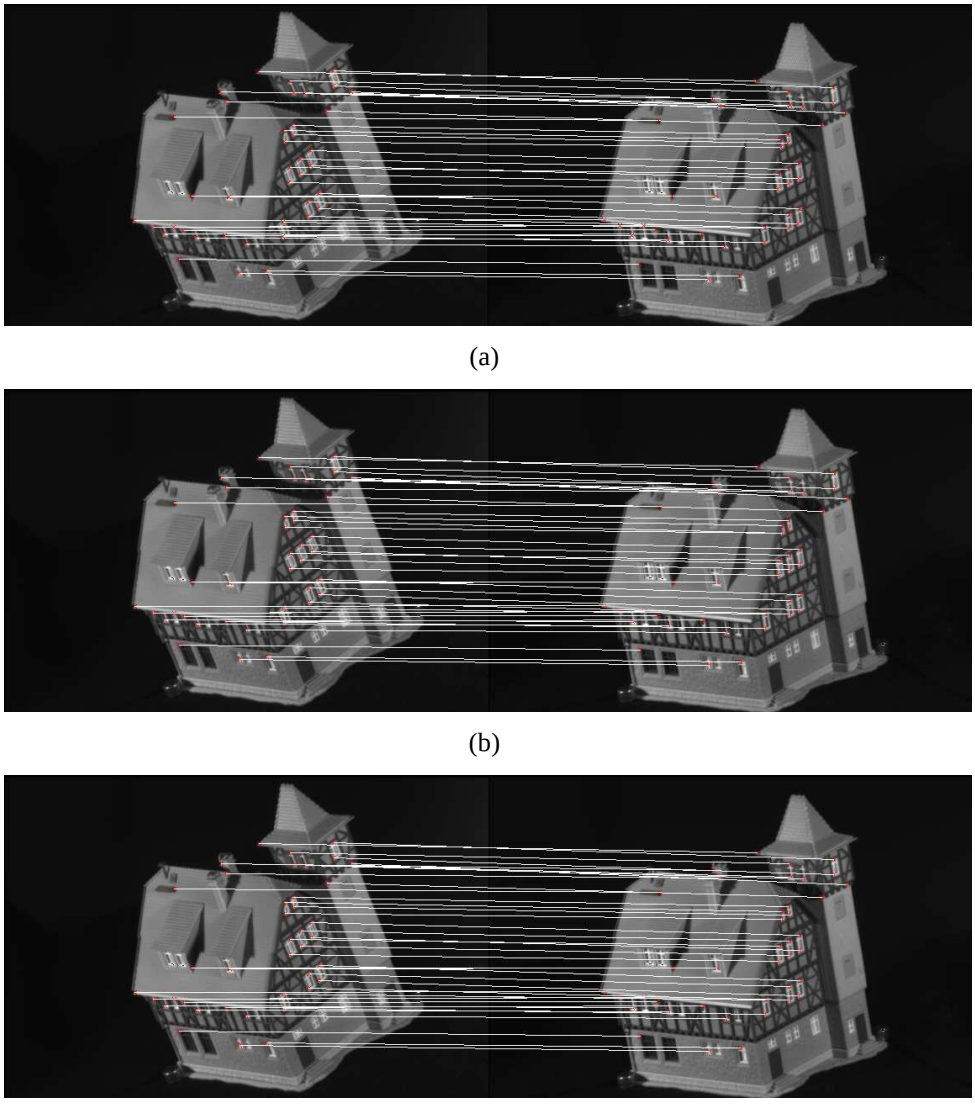
(b)



(c)

Figure 3.12: Comparing different methods, matching the 1^{st} and 20^{th} frame, a) Carcassoni b) Scott & Longuet-Higgins c) Carcassoni + Hermitian

EM alignment algorithm is incorporated along with the complex spectral information. We compare our method with a non-spectral method developed by Chui and Rangarajan (Chui & Rangarajan 2000). We use the same data set i.e. the CMU/VASC model-house sequence. The results are shown in Table 3.1 which shows that the performance of Chui and Rangarajan's method referred to as TSP,



(a)

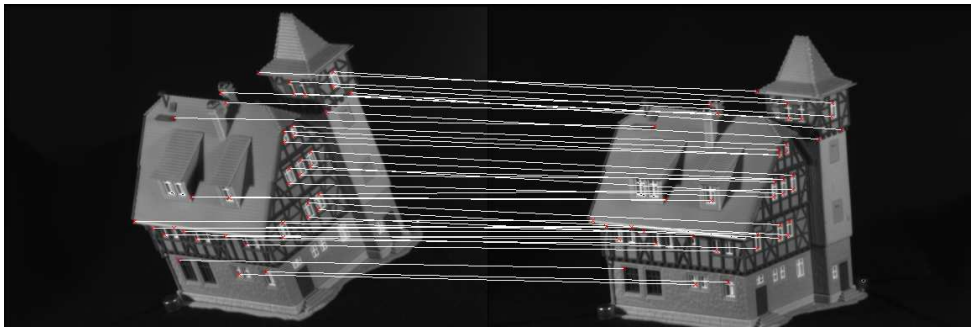
(b)

(c)

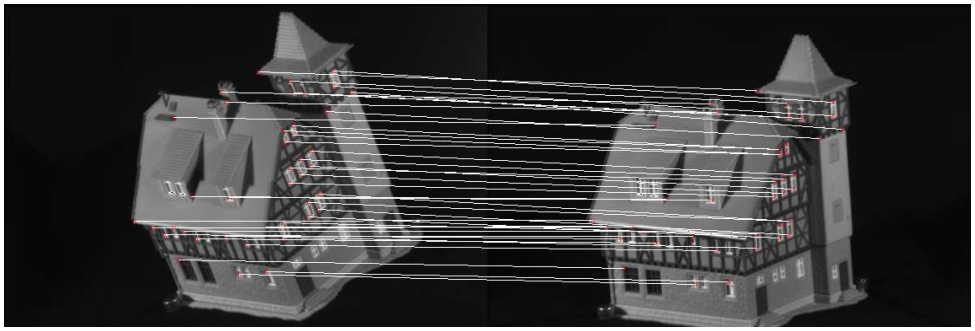
Figure 3.13: Comparing different methods, matching the 1^{st} and 40^{th} frame, a) Carcassoni b) Scott & Longuet-Higgins c) Carcassoni + Hermitian

decreases when the rotation in the input point-sets increases.

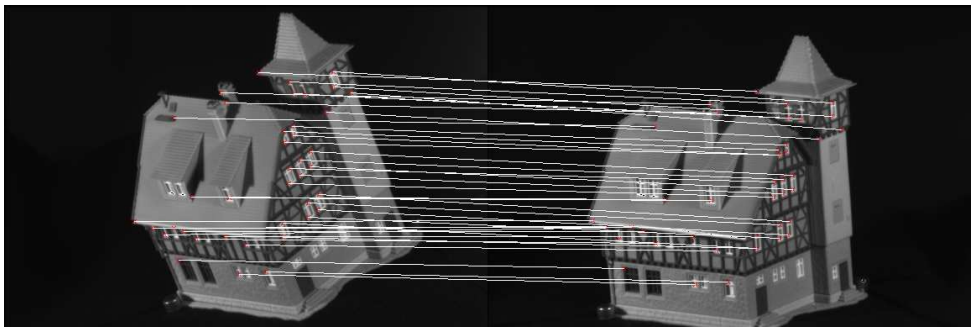
The second experiment we performed is on the Swiss chalet model house sequence. Ten frames of the sequence are shown in figure 3.11 with the extracted feature points. The feature points are extracted using a corner detector (Harris & Stephens 1988) which produces the point-sets of different sizes. For instance,



(a)



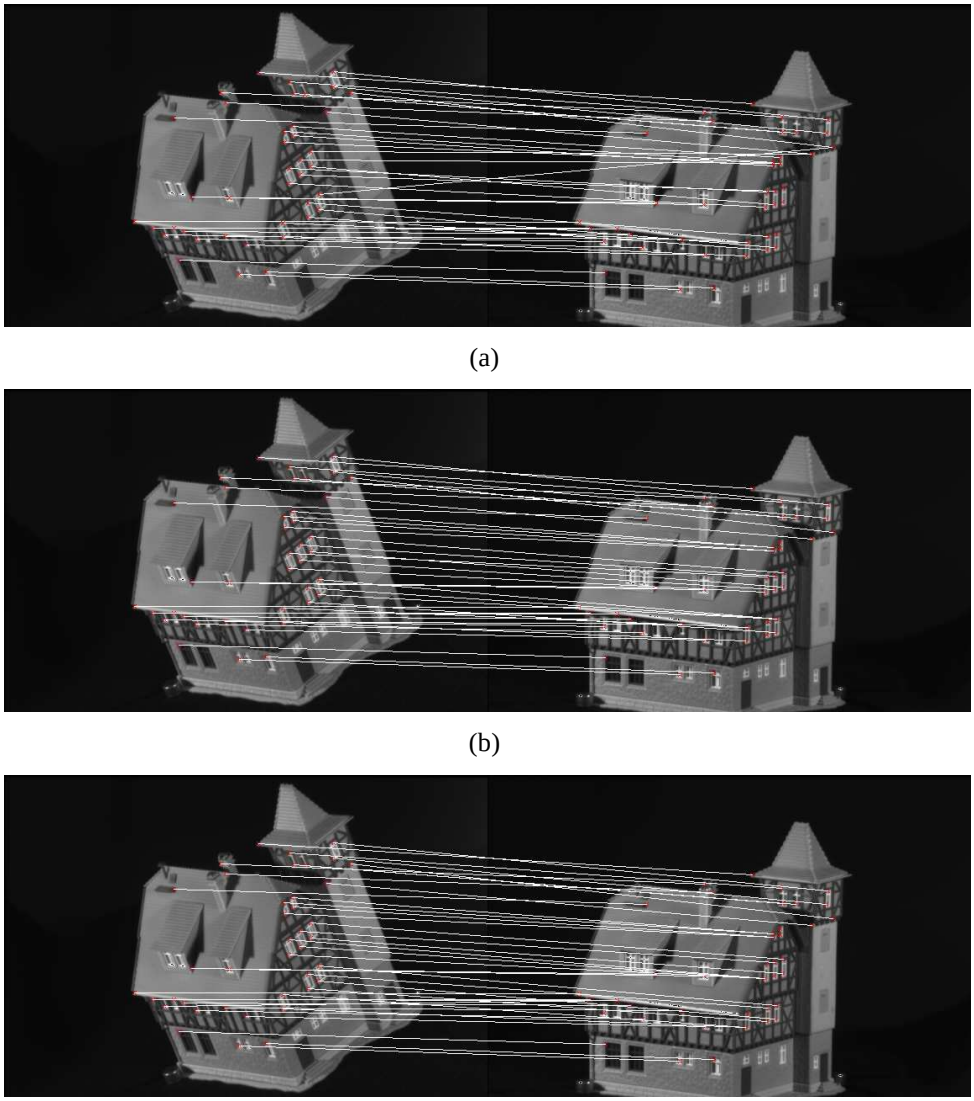
(b)



(c)

Figure 3.14: Comparing different methods, matching the 1^{st} and 60^{th} frame, a) Carcassoni b) Scott & Longuet-Higgins c) Carcassoni + Hermitian

in the frames 01 to 10 of the sequence, the sizes of the point-sets are 30, 32, 30, 25, 25, 23, 24, 24, 22 and 25 respectively. Hermitian matrices are constructed using equation 3.3 as explained in Section 3.4. The parameter σ controls the interaction between the feature points. The choice of the value of σ significantly affects the performance of the algorithm. Here we choose maximum of the x



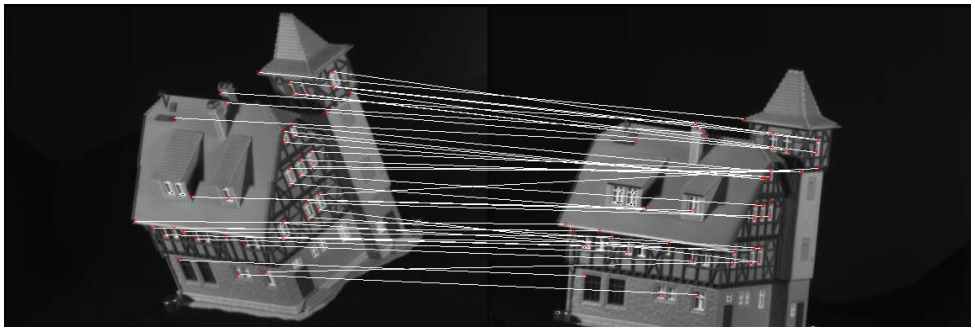
(a)

(b)

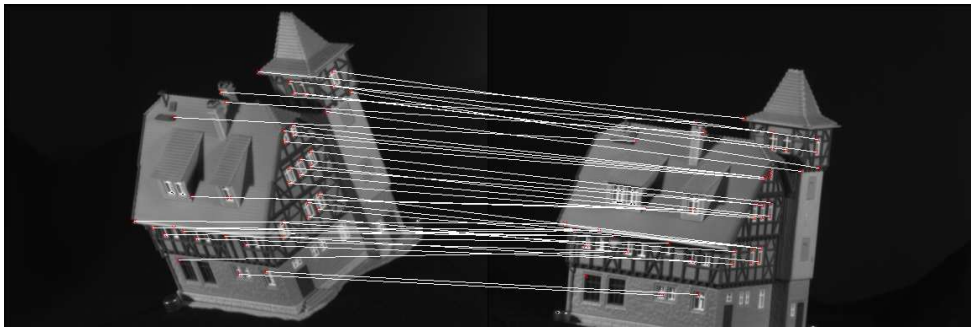
(c)

Figure 3.15: Comparing different methods, matching the 1^{st} and 80^{th} frame, a) Carcassoni b) Scott & Longuet-Higgins c) Carcassoni + Hermitian

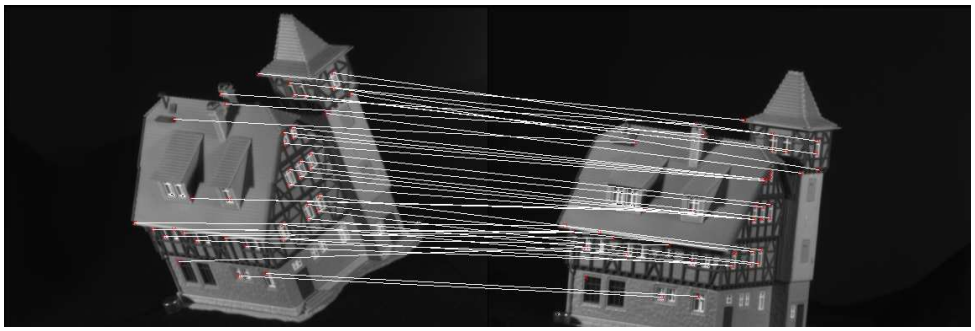
and y coordinates of all the feature points in the image as the value of σ. We compute angles, at the feature points localised, using Vedaldi's MATLAB/C implementation (Vedaldi 2006) of the SIFT detector and descriptor. The results of the corresponding matching using different methods are given in Table 3.2 in terms of the number of correct correspondences. We compare the performance



(a)



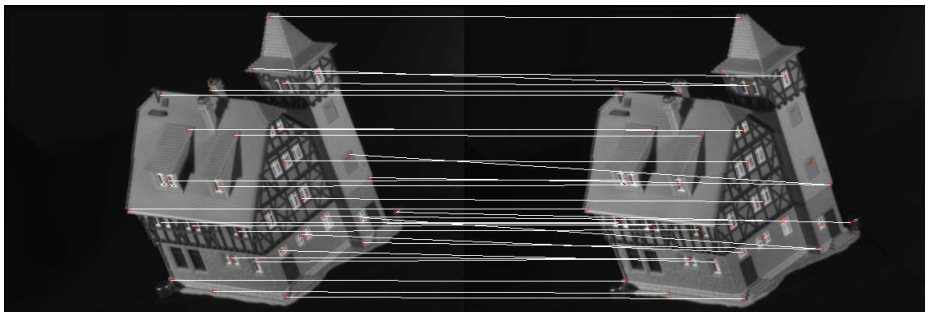
(b)



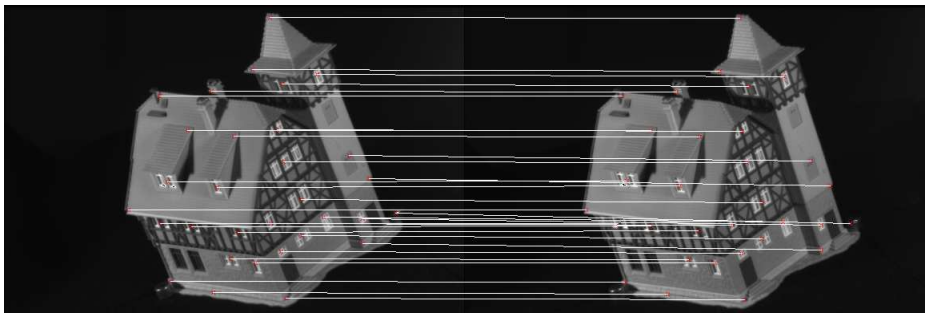
(c)

Figure 3.16: Comparing different methods, matching the 1^{st} and 100^{th} frame, a) Carcassoni b) Scott & Longuet-Higgins c) Carcassoni + Hermitian

of our algorithm with Chui and Rangarajan's method, referred to as TSP, on the Swiss chalet model house sequence. The quantitative results are shown in Table 3.2.



(a)



(b)

Figure 3.17: Experimental results: Correspondence matching of the 1^{st} and 10^{th} frame (a)using spectral information only (b)using EM alignment along with spectral information

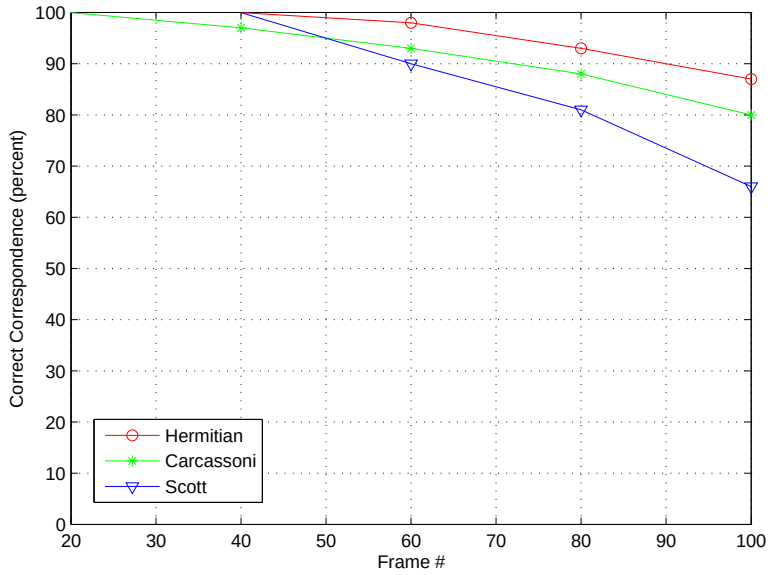


Figure 3.18: Effect of viewing angle on correspondence matching

	Number of incorrect matches (out of 40)				
Frame	1-20	1-40	1-60	1-80	1-100
Scott	0	0	4	7	18
Carcassoni	0	1	3	5	8
Carcassoni+Hermitian	0	0	1	3	5
TSP (non spectral)	0	3	8	13	19

Table 3.1: Performance on the CMU/VASC house sequence. The first image frame has been matched against the $20^{th}, 40^{th}, 60^{th}, 80^{th}$ and 100^{th} frame

	Number of correct correspondences								
Frame	1	2	3	4	5	6	7	8	9
# of points	30	32	30	25	25	23	24	24	22
Scott & Longuet-Higgins	-	28	25	21	20	15	11	7	6
Shapiro & Brady	-	28	26	21	17	14	10	9	5
Carcassoni	-	30	29	27	22	20	20	19	16
Carcassoni + Hermitian	-	30	30	29	24	22	22	21	19
TSP (non spectral)	-	25	22	18	18	19	16	14	12

Table 3.2: Performance of different algorithms on the Swiss Chalet model house sequence. The first image frame is matched against remaining nine frames

3.8 Summary

In this chapter we have investigated how the correspondence matching method of Shapiro and Brady (Shapiro & Brady 1992) can be improved by using complex eigenvectors of Hermitian property matrix. We added the angular information to the proximity matrix used by Shapiro and Brady, to extend it to the complex domain. We constructed a complex analog of a real weighted Laplacian matrix. We used the eigenvector of complex Laplacian for the purpose of correspondence matching. Secondly, we used the complex eigenvectors of the complex Lapla-

cian matrix to calculate the correspondence probabilities matrix and embed it into Carcassoni's EM algorithm to render it more robust to large viewing angle change between the images being matched. We tested the proposed method on both the synthetic data and the real world data. The experimental results on synthetic and real world data both indicate that our approach works with a relatively higher accuracy.

CHAPTER 4

Unsupervised Clustering of Human Pose using Spectral Embedding

4.1 Introduction

Full body human pose analysis is one of the fundamental problems in computer vision. Detecting the human pose is an important step in human behaviour analysis, action or gesture recognition. However, human pose detection is a challenging task because of the huge inter-limb and intra-limb feature variability in both still images and image sequences. It has a wide range of potential applications such as video-gaming, human-computer interaction, security, and healthcare etc. In literature, a significant amount of work has been done on human pose estimation, detection, clustering and classification (Andriluka et al. 2009; Johnson & Everingham 2009; Eichner & Ferrari 2010). Agarwal and Triggs (Agarwal & Triggs 2006) describe a learning-based method for recovering 3D human

body pose from single images and monocular image sequences. In (Rogez et al. 2012), an efficient method to jointly localize and recognize the pose of humans is proposed, using the randomized hierarchical cascades classifier. Here we use a graph clustering approach using spectral methods to cluster similar human poses produced by the Microsoft Kinect device.

Graph partitioning/clustering and classification is one of the most extensively studied topics in computer vision and machine learning community. Clustering is closely related to unsupervised learning in pattern recognition systems. Graphs are structures formed by a set of vertices called nodes and a set of edges that are connections between the pairs of nodes. Graph clustering is grouping similar graphs based on structural similarity within clusters. Bunke et al. (Bunke et al. 2003) proposed a structural method referred to as the Weighted Minimum Common Supergraph (WMCS), for representing a cluster of patterns. There has been significant amount of work aimed at using spectral graph theory (Chung 1997) to cluster graphs. This work shows the common feature of using graph representations of the data for the graph partitioning. Luo et al. (Luo et al. 2002) have used the discriminatory qualities of a number of features constructed from the graph spectrum. Using the leading eigenvalues and eigenvectors of the adjacency matrix they found that the leading eigenvalues have the best capabilities for structural comparison. There are a number of examples of applying pairwise clustering methods to graph edit distances (Pavan & Pelillo 2003). Recently, the properties of the eigenvectors and eigenvalues of the Laplacian matrix of graph have been exploited in many areas of computer vision. For instance, Shi and Malik (Shi & Malik 2000) used the eigenvector corresponding to second smallest (none zero) eigenvalue (also called Fielder vector) of the Laplacian matrix to iteratively bi-partition the graph for image segmentation. The information encoded in the eigenvectors of the Laplacian has been used for shape registration

(Mateus et al. 2008) and clustering. Veltkamp et al. (Leuken et al. 2008) developed a shape retrieval method using a complex Fielder vector of a Hermitian property matrix. Recent spectral approaches use the eigenvectors corresponding to the k smallest eigenvalues of the Laplacian matrix to embed the graph onto a k dimensional Euclidean space (Ng et al. 2001; Yu & Shi 2003).

In this chapter we propose a clustering method using the angular information and the distance between pair of joints, of the skeleton extracted from the Microsoft Kinect 3D sensor (Microsoft 2010). Given the skeleton acquired from Microsoft Kinect, we commence by converting the skeletal graph to its equivalent line graph because we need the angles between pairs of limbs. The angle between adjacent pair of limbs is computed by creating vectors parallel to adjacent limbs and taking the inverse cosine of the dot products of the vectors representing the limbs. For instance, the angle between the upper arm and lower arm is calculated using vectors created by *Elbow* joint to *Wrist* joint and *Elbow* joint to *Shoulder* joint as shown in figure 4.5(b). We construct a Hermitian matrix using the distance as the weights of the edges multiplied by the angles between each pair of limbs in the form of a complex number. We use the spectrum of the Hermitian matrix to cluster similar human poses. The feature vectors are constructed from the eigenvalues and eigenvectors of the Hermitian matrix of the graph. The topology of a graph is invariant under permutation of the node labels. However, if the nodes are relabelled, the adjacency, the Laplacian and the Hermitian matrices undergo a permutation of rows and columns. The corresponding eigenvector matrix undergoes a permutation of rows, i.e. the corresponding elements of the eigenvectors undergo a permutation. To construct feature-vectors which are invariant to the nodes labels, we use sets of symmetric polynomial coefficients. Once the feature-vectors for all the poses are to hand, we subject these vectors to two of the classical embedding methods including Principal Compon-

Figure 4.1: Microsoft Kinect 3D depth device for Xbox 360

ent Analysis (PCA) and Multidimensional Scaling (MDS).

The remainder of the chapter is organized as follows. Section 4.2 explains how a human pose is represented by a graph. In Section 4.3 the Hermitian matrix is defined. The symmetric polynomials are briefly reviewed in Section 4.4. Section 4.5 details the construction of the feature vectors. Experimental results are provided in Section 4.8 and finally Section 4.9 concludes the chapter.

4.2 Human Pose Representation

This section describes the processing of the graph extracted from the skeleton acquired from the Microsoft Kinect 3D depth device for Xbox 360. It's a specialized sensor built by Microsoft that is capable of recognizing and tracking humans in 3D space. The Kinect has three windows at the front as shown in the figure 4.1. The left window on the Kinect is an infrared (IR) projector; the middle window is a colour (RGB) camera while the right window is an infrared sensor. The IR projector and the IR sensor work together to make a 3D depth sensor. The IR projector emits a grid of IR light in front of it. This light is reflected back to the IR sensor. The pattern received by the IR sensor is decoded in the Kinect to determine the depth information. This depth information is very useful in many computer vision applications.

Using this device Shotton et al. (Shotton et al. 2011) developed a method to extract the human body pose from a single depth image. They use the depth data

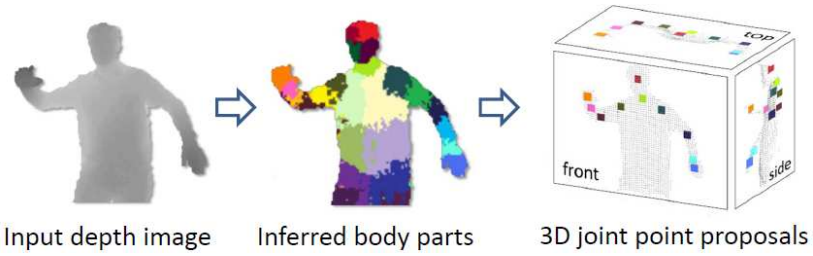


Figure 4.2: 3D Joint Proposals Pipeline

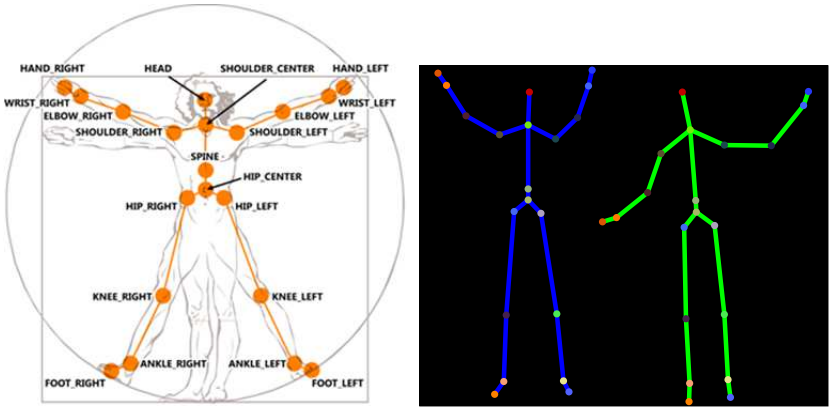


Figure 4.3: Kinect 3D Joints, with skeletal model

in order to perform the segmentation of the human body to obtain its skeletal model which consists of a set of joint positions. They use a huge set of human samples to infer pixel labels through Random Forest estimation, and the skeletal model is defined as the centroid of mass of the different dense regions using mean shift algorithm resulting in the 3D joint proposals. Through experimental results, they demonstrate that their algorithm is efficient and effective for reconstructing 3D human body poses, even in the presence of partial occlusions, different points of view and under no light conditions. The process of joint proposal from the depth image is shown in figure 4.2.

We use the Microsoft Kinect Beta 2 SDK API functions to extract the 3D joint positions of the human skeletal model. The NUI Skeleton API provides information about the location of players standing in front of the Kinect device, with

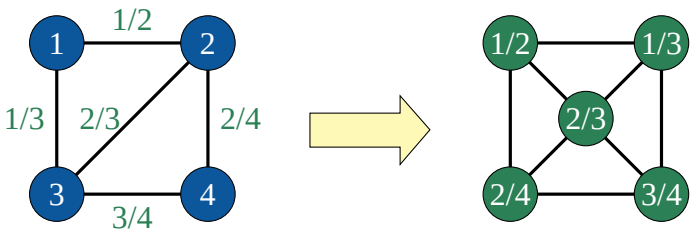
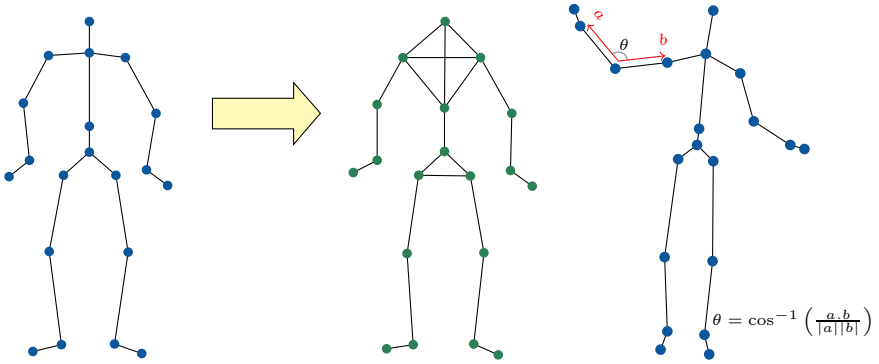


Figure 4.4: Line graph example, Original graph (left) and its equivalent line graph (right), the nodes represent the limbs

detailed position and orientation information. The data is provided to application code as a set of points, called skeleton positions, that compose a skeleton, as shown in figure 4.3. This skeleton represents a user's current position and pose. The skeleton has 20 points that are called Joints in Kinect SDK.

Here, our aim is to cluster similar human poses represented by the skeleton with 20 points acquired from the Kinect sensor using the spectral graph techniques. We commence by constructing a graph representing a human pose, where the nodes of the graph represent the joints and the edges represent the human body limbs. We use the length of the limbs and the angle between a pair of limbs as features. Since, we use the angles between each pair of limbs which are represented by the edges of the graph, therefore, we need to convert that graph to its equivalent line graph so that the angular information is defined on the nodes.

The line graph of undirected graph G is another graph that represents the adjacency between edges of G. The nodes in the line graph represents the edges of the original graph G. For instance, figure 4.4 shows an example graph and its equivalent line graph. The original graph has 4 nodes and 5 edges while the resulting line graph has 5 nodes and 8 edges. Similarly, we convert the human skeleton into its equivalent line graph shown in figure 4.5(a). There are 19 edges in the original skeleton, therefore, the nodes in its equivalent line graph are 19. The Hermitian matrix is established from the difference between the lengths of



(a) (b)

Figure 4.5: Human skeleton graph a) Skeleton captured using MS Kinect (left) and its equivalent line graph (right); b) Skeleton showing the angle θ between upper and lower arm

each pair of edges and the angles subtended by those edges. We use the spectra of a Hermitian property matrix along with the coefficients of symmetric polynomials to construct a feature vector which represents a single human pose.

4.3 Complex Laplacian (Hermitian) matrix

A Hermitian matrix H (or self-adjoint matrix) is a square matrix with complex elements that remains unchanged under the joint operation of transposition and complex conjugation of the elements. That is, the element in the i^{th} row and j^{th} column is equal to the complex conjugate of the element in the j^{th} row and i^{th} column, for all indices i and j, i.e. $a_{i,j} = \overline{a}_{j,i}$. Complex conjugation is denoted by the dagger operator † i.e. $H^{\dagger} = H$. Hermitian matrices can be viewed as the complex number extension of the symmetric matrix for real numbers. The on-diagonal elements of a Hermitian matrix are necessarily real quantities. Each off-diagonal element is a complex number which has two components, and therefore, can represent a 2-component measurement.

To create a positive semi-definite Hermitian matrix of a graph, there should be

some constraints applied on the measurement representations. Let $\{x_1, x_2, ..., x_n\}$ be a set of measurements for the node-set $\mathcal{V}$ and $\{y_{1,1}, y_{1,2}, ..., y_{n,n}\}$ be the set of measurements associated with the edges of the graph, in addition to the graph weights. Each edge then has a pair of observations $(\mathcal{W}_{a,b}, y_{a,b})$ associated with it. There are a number of ways in which the complex number $H_{a,b}$ could represent this information, for example with the real part as $\mathcal{W}$ and the imaginary part as y. However, here we follow Wilson, Hancock and Luo (Wilson et al. 2005) and construct the complex property matrix so as to reflect the Laplacian. As a result the off-diagonal elements of H are chosen to be

$$H_{a,b} = -\mathcal{W}_{a,b} e^{\iota y_{a,b}}$$

. The edge weights are encoded by the magnitude of the complex number $H_{a,b}$ and the additional measurement by its phase. By using this encoding, the magnitude of the number is the same as the original Laplacian matrix. This encoding is suitable when measurements are angles, satisfying the conditions $-\pi \leq y_{a,b} < \pi$ and $y_{a,b} = -y_{a,b}$ to produce a Hermitian matrix. To ensure a positive definite matrix, H_{aa} should be greater than $-\Sigma_{b \neq a} |H_{ab}|$. This condition is satisfied if $H_{aa} = x_a + \Sigma_{b \neq a} \mathcal{W}_{a,b}$ and $x_a \geq 0$. When defined in this way the property matrix is a complex analogue of the weighted Laplacian matrix for the graph.

For a Hermitian matrix there is an orthogonal complete basis set of eigenvectors and eigenvalues i.e. $H\phi = \lambda\phi$. The eigenvalues λ_i of Hermitian matrix are real while the eigenvectors ϕ_i are complex. There is a potential ambiguity in the eigenvectors, in that any multiple of an eigenvector is a solution of the eigenvector equation $H\phi = \lambda\phi$. i.e. $H\alpha\phi = \lambda\alpha\phi$. Therefore, we need two constraints for them. Firstly, make each eigenvector of unit length vector i.e. $|\phi_i| = 1$, and secondly impose the condition $\arg \sum_i \phi_{ij} = 0$.

4.4 Symmetric Polynomials

A symmetric polynomial is a polynomial $S(x_1, x_2, \ldots, x_n)$ in n variables, such that if any of the variables are interchanged, the same polynomial is obtained. A symmetric polynomial is invariant under permutation of the variable indices. There is a special set of symmetric polynomials referred to as the *elementary symmetric polynomial (S)* that form a basis set for symmetric polynomial. The elementary symmetric polynomials are the most fundamental symmetric polynomials. Any symmetric polynomial can be expressed as a polynomial function of the elementary symmetric polynomials. For a set of variables $x_1, x_2, \ldots, x_n$ the elementary symmetric polynomials can be defined as:

$$
\begin{aligned}
S_1(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} x_i \\
S_2(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j \\
&\vdots \\
S_r(x_1, x_2, \ldots, x_n) &= \sum_{i_1 < i_2 < \ldots < i_r} x_{i1} x_{i2} \ldots x_{ir} \\
&\vdots \\
S_n(x_1, x_2, \ldots, x_n) &= \prod_{i=1}^{n} x_i
\end{aligned}
$$

The power symmetric polynomial functions (P) are defined as

$$
\begin{aligned}
P_1(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} x_i \\
P_2(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} x_i^2 \\
&\vdots \\
P_r(x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{n} x_i^r
\end{aligned}
$$

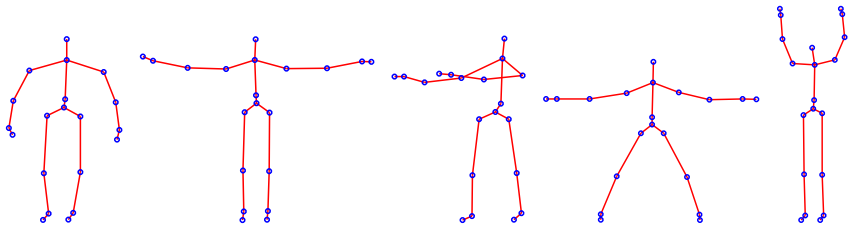


Figure 4.6: Some Examples of Poses for Experiments

$$\vdots$$

$$P_n(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i^n$$

The elementary symmetric polynomials can be efficiently computed from the coefficients of the power symmetric polynomials using the Newton-Girard formula

$$S_r = \frac{(-1)^{r+1}}{r} \sum_{k=1}^{r} (-1)^{k+r} P_r S_{r-k} \tag{4.1}$$

here the shortcut S_r is used for $S_r(x_1, x_2, \ldots, x_n)$ and P_r is used for $P_r(x_1, x_2, \ldots, x_n)$.

4.5 Feature Vectors

The skeleton of human body with twenty, 3-dimensional points representing the joints connected by the lines representing the limbs, is acquired using the Microsoft Kinect SDK. Kinect provides the skeletal data with the rate of 30 frames per second. Figure 4.6 shows some examples of the skeletons captured with the Kinect sensor. Each point in the skeleton is represented by a three dimensional vector $w_i = (x_i, y_i, z_i)^T$.

We used the limb joint angles and the limb length assigned by the Microsoft Kinect SKD. We convert the skeleton into its equivalent line graph. The line graph of undirected graph G is another graph that represents the adjacency between

the edges of G. The nodes in the line graph represents the edges of the original graph G. We construct a Hermitian matrix from the difference between the lengths of each pair of the edges and the angles subtended by those edges, to reflect the Laplacian matrix as detailed in section 4.3. Given two adjacent edges e_i and e_j, with the nodes w_{k-1}, w_k and w_{k+1}, where w_k is the common (middle) node. The angle between the edges e_i and e_j is given by

$$\theta_{ij} = \cos^{-1}\left(\frac{(w_k - w_{k-1})^T(w_k - w_{k+1})}{||w_k - w_{k-1}|| \times ||w_k - w_{k+1}||}\right) \tag{4.2}$$

The Hermitian matrix H has element with row index i and column index j is given by

$$H(i,j) = -\mathcal{W}_{i,j}e^{\iota\theta_{i,j}} \tag{4.3}$$

where $\mathcal{W}_{i,j}$ is the difference of the lengths of the edges e_i and e_j and $\theta_{i,j}$ is the angle between the edges e_i and e_j. To obey the antisymmetric condition $\theta_{i,j} = -\theta_{j,i}$, we multiply $\theta_{i,j}$ by -1 if length of edge $e_i > e_j$.

With the complex matrix H to hand, we compute its eigenvalues and eigenvectors. The eigenvector of a Hermitian matrix are complex and the eigenvalues are real.

$$H = \Phi\Lambda\Phi^T \tag{4.4}$$

where Φ is the eigenvector matrix, with eigenvector sitting in its columns, and Λ is a diagonal matrix with eigenvalues λ_i on its main diagonal. We order the eigenvectors according to the decreasing magnitude of the eigenvalues i.e. $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$. We construct a complex spectral matrix Ψ for the input pose from the eigenvalues and eigenvectors of the Hermitian matrix H by multiplying each eigenvector by the square root of its corresponding eigenvalue as follows

$$\Psi = \left(\sqrt{\lambda_1}\phi_1|\sqrt{\lambda_2}\phi_2|\ldots|\sqrt{\lambda_n}\phi_n\right) \tag{4.5}$$

where λ_i are the eigenvalues and ϕ_i are their corresponding eigenvectors. From the scaled eigenvectors in the columns of the complex spectral matrix Ψ, the symmetric polynomial coefficients are computed. To do so we first compute the power symmetric polynomials. From the power symmetric polynomials the elementary symmetric polynomials are computed using the Newton-Girard formula (equation 4.1) as described in Section 4.4 (Wilson et al. 2005). Each column of the complex spectral matrix Ψ is used as input to the set of symmetric polynomials. For instance, the first column $(\Psi_{1,1}, \Psi_{2,1}, \ldots, \Psi_{n,1})^T$ will produce the polynomial coefficients $S_1(\Psi_{1,1}, \Psi_{2,1}, \ldots, \Psi_{n,1})$, $S_2(\Psi_{1,1}, \Psi_{2,1}, \ldots, \Psi_{n,1})$, ..., $S_n(\Psi_{1,1}, \Psi_{2,1}, \ldots, \Psi_{n,1})$. We put these coefficients in the first column of a matrix $\mathcal{S}$. The second column of the matrix $\mathcal{S}$ is computed from the second column of the spectral matrix. Similarly we can compute the n coefficients for each column of the spectral matrix and put these in the corresponding column of the matrix $\mathcal{S}$. The n^{th} column of the matrix $\mathcal{S}$ is computed from the n^{th} column of the spectral matrix Ψ i.e. the column $(\Psi_{1,n}, \Psi_{2,n}, \ldots, \Psi_{n,n})^T$ will produce $S_1(\Psi_{1,n}, \Psi_{2,n}, \ldots, \Psi_{n,n})$, $S_2(\Psi_{1,n}, \Psi_{2,n}, \ldots, \Psi_{n,n})$, ..., $S_n(\Psi_{1,n}, \Psi_{2,n}, \ldots, \Psi_{n,n})$. Hence, for all n columns we will have n^2 coefficients. These coefficients are invariant to the permutation of the node labels of the input graph.

$$\mathcal{S} = \begin{pmatrix} S_1(C_1) & S_1(C_2) & \ldots & S_1(C_n) \\ S_2(C_1) & S_2(C_2) & \ldots & S_2(C_n) \\ \vdots & \vdots & \ddots & \vdots \\ S_n(C_1) & S_n(C_2) & \ldots & S_n(C_n) \end{pmatrix} \tag{4.6}$$

where $S_1, S_2, ..., S_n$ are the first, second and the n^{th} coefficients of the symmetric polynomial, and $C_1, C_2, ..., C_n$ are the first, second and n^{th} column of the spectral matrix Ψ respectively. The coefficients of high order polynomials tend to zero because of the product terms in high order polynomials, therefore, we construct

the feature vectors using the first 10 coefficients only.

Since, the coefficient of the elementary symmetric polynomials are complex numbers, therefore, we can construct the feature-vectors in a number of ways given below

1. Let $\hat{\mathcal{S}}$ is $2n \times n$ created from the $n \times n$ complex elementary symmetric polynomials matrix $\mathcal{S}$. The real and imaginary components of the coefficients of symmetric polynomials are interleaved. The columns of this $2n \times n$ are stacked to form a long feature vector F_i for the graph representing the pose frame.

$$F_i = (\hat{\mathcal{S}}_{1,1}, \hat{\mathcal{S}}_{2,1}, \ldots, \hat{\mathcal{S}}_{2n,1}, \hat{\mathcal{S}}_{1,2}, \hat{\mathcal{S}}_{2,2}, \ldots, \hat{\mathcal{S}}_{2n,2}, \ldots, \hat{\mathcal{S}}_{1,n}, \hat{\mathcal{S}}_{2,n}, \ldots, \hat{\mathcal{S}}_{2n,n})^T \tag{4.7}$$

2. Let Γ be the matrix whose elements be the magnitude of the components of the complex elementary symmetric polynomial matrix $\mathcal{S}$. i.e. $\Gamma_{i,j} = |\mathcal{S}_{i,j}| = \sqrt{\Re(\mathcal{S}_{i,j})^2 + \Im(\mathcal{S}_{i,j})^2}$, where $\Re(\mathcal{S}_{i,j})$ is the real part of the complex symmetric polynomial coefficient $\mathcal{S}_{i,j}$ and $\Im(\mathcal{S}_{i,j})$ is the imaginary part of $\mathcal{S}_{i,j}$. The columns of the matrix Γ are stacked to form a long feature vector F_i for the graph representing the pose frame.

$$F_i = (\Gamma_{1,1}, \Gamma_{2,1}, \ldots, \Gamma_{n,1}, \Gamma_{1,2}, \Gamma_{2,2}, \ldots, \Gamma_{n,2}, \ldots, \Gamma_{1,n}, \Gamma_{2,n}, \ldots, \Gamma_{n,n})^T \tag{4.8}$$

3. Let $\mathcal{R}$ be the matrix with its elements $\mathcal{R}_{i,j}$ be the real part of the components of the complex elementary symmetric polynomial coefficients $\mathcal{S}_{i,j}$. i.e. $\mathcal{R}_{i,j} = \Re(\mathcal{S}_{i,j})$, where $\Re(\mathcal{S}_{i,j})$ is the real part of the complex symmetric polynomial coefficient $\mathcal{S}_{i,j}$. The columns of the matrix $\mathcal{R}$ are stacked to form a long feature vector F_i for the graph representing the pose frame.

The imaginary part is ignored.

$$F_i = (\mathcal{R}_{1,1}, \mathcal{R}_{2,1}, \ldots, \mathcal{R}_{n,1}, \mathcal{R}_{1,2}, \mathcal{R}_{2,2}, \ldots, \mathcal{R}_{n,2}, \ldots, \mathcal{R}_{1,n}, \mathcal{R}_{2,n}, \ldots, \mathcal{R}_{n,n})^T \tag{4.9}$$

4. Let $\mathcal{I}$ be the matrix with its elements $\mathcal{I}_{i,j}$ be the imaginary part of the components of the complex elementary symmetric polynomial coefficients $\mathcal{S}_{i,j}$. i.e. $\mathcal{I}_{i,j} = \Im(\mathcal{S}_{i,j})$, where $\Im(\mathcal{S}_{i,j})$ is the imaginary part of the complex symmetric polynomial coefficient $\mathcal{S}_{i,j}$. The columns of the matrix $\mathcal{I}$ are stacked to form a long feature vector F_i for the graph representing the pose frame. The real part is discarded.

$$F_i = (\mathcal{I}_{1,1}, \mathcal{I}_{2,1}, \ldots, \mathcal{I}_{n,1}, \mathcal{I}_{1,2}, \mathcal{I}_{2,2}, \ldots, \mathcal{I}_{n,2}, \ldots, \mathcal{I}_{1,n}, \mathcal{I}_{2,n}, \ldots, \mathcal{I}_{n,n})^T \tag{4.10}$$

4.6 Embedding Methods

In this section we explore two different methods of embedding the graph feature vectors in a pattern space, namely Principal Components Analysis (PCA) (Jolliffe 2002) and Multidimensional Scaling (MDS) (Kruskal & Wish 1978). PCA finds the projection which is in the direction of maximum variance in the data. Multidimensional scaling on the other hand, preserves the relative distance between a pair of data. MDS is performed on a set of pairwise distance between each pair of vectors.

4.6.1 Principal Component Analysis

Principal Components Analysis (PCA) is a popular technique for dimensionality reduction. PCA transforms the input data to a new coordinate system such that

the greatest variance by any projection of the data comes to lie on the first coordinate called the first principal component, the second greatest variance on the second coordinate, and so on. We start by constructing a matrix S which has the mean-adjusted feature vectors. $S = [F_1 - \bar{F} | F_2 - \bar{F} | \ldots | F_N - \bar{F}]$ from the feature vectors F_i for the graph representing the pose frame, where $\bar{F}$ is the mean feature vector for the given feature vectors F_i. The next step is to compute the covariance matrix C by taking the product $C = S^T S$. The principal components of the covariance matrix C are computed by subjecting it to the eigen-decomposition $C = U \Lambda U^T$, where U is the eigenvector matrix with the eigenvectors sitting in its columns i.e. $U = (u_1, u_2, \ldots, u_n)$ and Λ is diagonal matrix with eigenvalues sitting on its main diagonal. Here we use the first k leading eigenvectors to represent the feature vectors for the graphs i.e. $U = (u_1, u_2, \ldots, u_k)$. For visualization purpose we take only 2 or 3 leading principal components. Each graph is represented by a feature vector F_i. We project the feature vector F_i onto the eigenspace using the equation $Y_i = U^T (F_i - \bar{F})$

4.6.2 Multidimensional Scaling

Multidimensional Scaling (MDS) is a technique which provides a low dimensional representation of high dimensional data for visualization purpose. The input data should be given in terms of a square, symmetric matrix of pairwise distances between each pair of the data objects. The data object (high dimensional) are represented as points in a low dimensional pattern space, such that the Euclidean distances between the points match the input dissimilarities as closely as possible. To commence we need to compute the pairwise distances between the graphs representing the pose frames. We compute the Euclidean distance between the feature vector F_i corresponding to the pair of graph representing the pose frames. For instance, the distance $d_{1,2}$ between feature vector

F_1 and F_2 is computed as $d_{1,2} = (F_1 - F_2)(F_1 - F_2)^T$. The pairwise distances $d_{r,c}$ are used as the elements of an $N \times N$ dissimilarity matrix U. Then, we need to compute a matrix T whose element with row r and column c is give by $T_{rc} = -\frac{1}{2}[U^2_{rc} - \hat{U}^2_{r.} - \hat{U}^2_{.c} + \hat{U}^2_{..}]$, where $\hat{U}^2_{r.} = \frac{1}{N}\sum_{c=1}^{N} U_{rc}$, is the average dissimilarity value over the r^{th} row in the distance matrix, $\hat{U}^2_{.c}$ is the dissimilarity average value over the c^{th} column in the distance matrix and $\hat{U}^2_{..} = \frac{1}{N^2}\sum_{r=1}^{N}\sum_{c=1}^{N} U_{rc}$ is the average dissimilarity value over all rows and columns of the distance matrix. Then, we subject the matrix T to eigenvector decomposition to obtain a matrix of embedding coordinates X. The number of non-zero eigenvalues we get is equal to the rank of the matrix T. If the rank of T is k, $k \leq N$, then we get k non-zero eigenvalues. We arrange these k non-zero eigenvalues in descending order, i.e. $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_k \geq 0$. The corresponding ordered eigenvectors are denoted by u_i where λ_i is the corresponding eigenvalue. The embedding coordinate system for the graphs is $X = [\sqrt{\lambda_1}u_1, \sqrt{\lambda_2}u_2, \ldots, \sqrt{\lambda_k}u_k]$. The embedded coordinates vector x_i for the graph i is given by $x_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,k})^T$.

4.7 Computational Complexity

Following are the steps of the algorithm developed in this chapter.

4.7.1 Steps

Given the skeleton with 20 nodes

1. Compute the line graph of the Skeletal graph, as we need the angles between pairs of limbs. The angles are computed between adjacent pair of limbs by creating vectors parallel to the limbs and taking the inverse cosine of the dot products of the vectors.

2. A Hermitian matrix is constructed from the line graph using the distances between each pair of joints and the angles between the limbs.

$$H_{i,j} = e^{(-(d_{ij}^2/2\sigma^2))} \times e^{\iota(\theta_{ij})}$$

3. The eigenvalues and eigenvectors of the Hermitian matrix are computed.

$$H = \Phi\Lambda\Phi^T$$

4. Spectral matrix is computed using the eigenvalues and eigenvectors of the Hermitian matrix.

$$\Psi = \left(\sqrt{\lambda_1}\phi_1|\sqrt{\lambda_2}\phi_2|\ldots|\sqrt{\lambda_n}\phi_n\right)$$

5. Elementary Symmetric Polynomials are computed from the matrix Ψ and all the columns are stacked to make a long vector which represents a pose. To construct feature-vectors which are invariant to the node labels, we use coefficients of the symmetric polynomials.

6. Use PCA and MDS to embed those vectors in space to cluster similar poses

4.7.2 Complexity

Since the number of nodes in each graph is fixed (i.e. 20), therefore, the execution time for this algorithm is constant.

4.8 Experimental Results

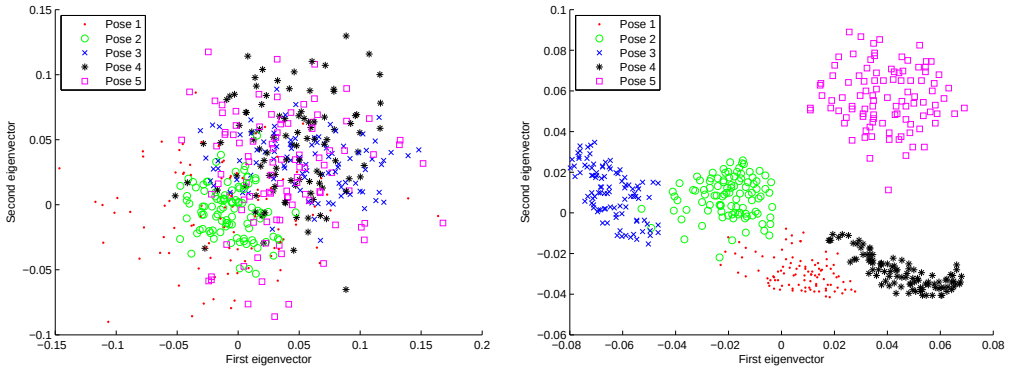
In this section, we provide some experimental investigations of the clustering of different human poses. We focus on two different settings. In the first setting we perform experiments to cluster similar human poses using a Hermitian property matrix and compare the results with that of the real Laplacian matrix. We also perform clustering using the Laplacian spectral pattern vectors (Luo et al. 2003) for comparison. Under the second setting we perform some experiments to investigate which combination of the feature vectors (detailed in section 4.5) and the embedding method (detailed in section 4.6) gives the best set of clustering results. Under both setting we use the human skeleton data taken from the Kinect.

Data set: Our data set consists of the human skeleton poses taken from Microsoft Kinect device, of 15 different subjects, including 7 males, 6 females and 2 children. 10 different poses of each subject were recorded. 150 instances of each pose were recorded, with slight change in the pose, position and angle.

In the first experiment, we take 5 different poses of randomly selected subjects. 100 instances of each pose are used. We construct the weighted Laplacian matrix $L = D - W$ for each pose using the joints as nodes of the graph and the length of the limbs as the edges of the graph, where D is the diagonal degree matrix and W is the weighted adjacency matrix. The entries of the weighted adjacency matrix W are computed using a Gaussian kernel i.e. $W_{ij} = e^{-d_{ij}^2/2\sigma^2}$, where d_{ij}^2 is the squared distance between node i and node j. We subject the Laplacian matrix L to eigen-decomposition i.e. $L = \Phi\Lambda\Phi^T$, where Φ is the eigenvector matrix, with eigenvector sitting in its columns, and Λ is a diagonal matrix with eigenvalues λ_i on its main diagonal. We order the eigenvectors according to the decreasing magnitude of the eigenvalues i.e. $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$.

We construct a spectral matrix from the eigenvectors and the eigenvalues of the Laplacian matrix L by multiplying each eigenvector by the square root of its corresponding eigenvalue i.e. $\Psi = \left(\sqrt{\lambda_1}\phi_1|\sqrt{\lambda_2}\phi_2|\ldots|\sqrt{\lambda_n}\phi_n\right)$ where λ_i are the eigenvalues and ϕ_i are their corresponding eigenvectors. From the scaled eigenvectors in the columns of the spectral matrix Ψ, the symmetric polynomial coefficients are computed. Since the Laplacian is a real matrix with real eigenvalues and real eigenvectors, the symmetric polynomial coefficients are also real. We construct the feature vectors F_i from the real coefficients of symmetric polynomials by stacking the columns of the coefficients matrix. We also construct the feature vectors from the coefficients of complex symmetric polynomials established using a Hermitian property matrix as described in section 4.3 by accounting the angular information along with the distances between each pair of nodes using equation 4.7. We project both sets of feature vectors onto 2 dimensional space using principal component analysis (PCA). Figure 4.7 illustrates the result of this comparison. Figure 4.7(a) shows the result obtained using the symmetric polynomials computed from the eigenvalues and eigenvectors of the Laplacian matrix L. Figure 4.7(b) shows the result obtained using the symmetric polynomials established from the eigenvalues and eigenvectors of the Hermitian matrix H. The Hermitian property matrix produces a better classification results than the Laplacian matrix for the human skeleton data captured from Microsoft Kinect device for Xbox 360, as Hermitian matrix captures more information from the input human skeleton.

In the second experiment, we randomly choose three poses of randomly selected subjects. Some examples of the input poses are shown in Figure 4.6. We take 100 different instances of each pose. We construct the feature vectors F_i according to the steps mentioned in Section 4.5 using equation 4.7. We then embed the feature vectors into a three dimensional pattern-space by performing



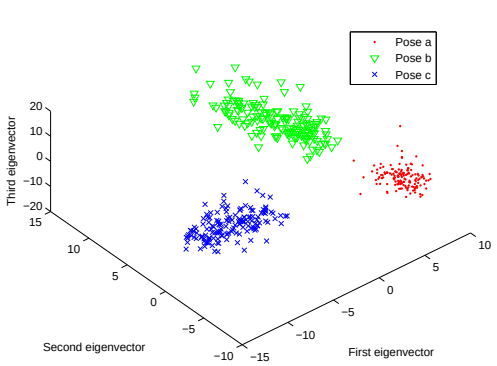
(a) (b)

Figure 4.7: Comparison of clustering results using PCA with a) weighted Laplacian matrix with only the distance measurements b) Hermitian property matrix with additional angular information embedded along with the distance measurements

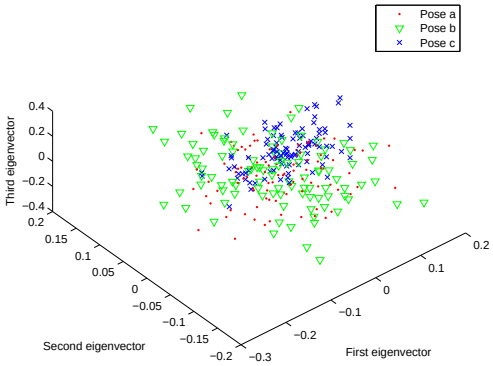
the principal component analysis. Figure 4.8(a) shows the result of the clustering using the first three eigenvectors. For comparison we establish feature vectors using the spectrum of Laplacian matrix of the graph representing the human skeleton. The eigenvalues of the Laplacian has an important role in the graph clustering algorithms. We take the smallest non-zero eigenvalue to the largest eigenvalue of the Laplacian matrix as components of the feature vector, i.e. $F_i = [\lambda_2, \lambda_3, \ldots, \lambda_n]^T$, where $L = \Phi\Lambda\Phi^T$, $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal matrix with eigenvalues sitting on its main diagonal and $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

Similarly, we construct another set of feature vectors using the spectrum of the Hermitian property matrix H described in section 4.3. We take the smallest non-zero eigenvalue to the largest eigenvalue of the Hermitian matrix as components of the feature vector, i.e. $F_i = [\lambda_2, \lambda_3, \ldots, \lambda_n]^T$, where $H = \Phi\Lambda\Phi^T$, $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal matrix with eigenvalues on its main diagonal and $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

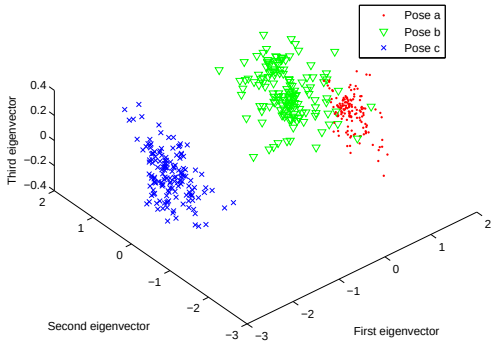
We subject the three sets of feature vectors to principal component analysis



(a) 3 poses, Hermitian (Symmetric Polynomial Coefficients)



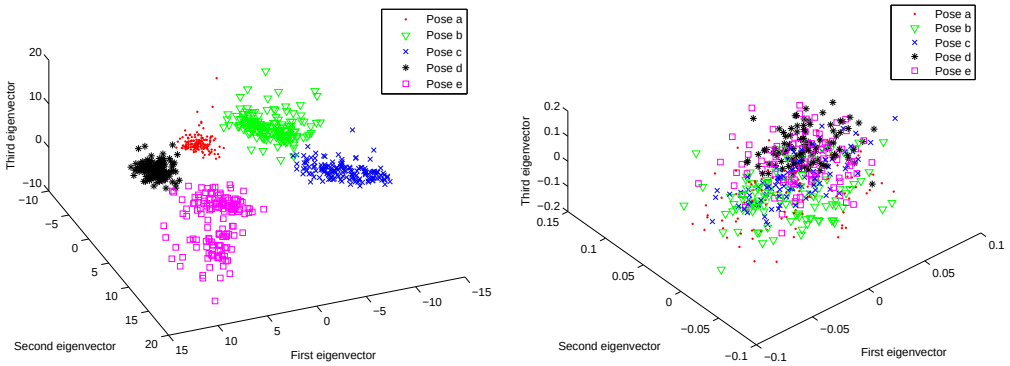
(b) 3 poses, Laplacian Spectra



(c) 3 poses, Hermitian Spectra

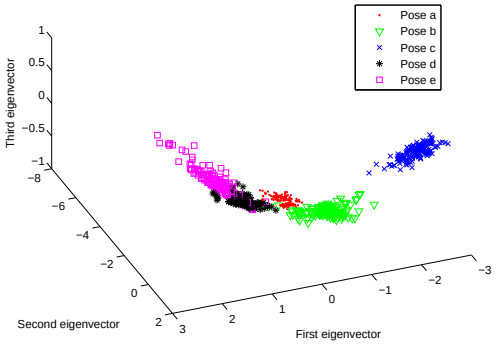
Figure 4.8: Performance of clustering, 3 poses

to embed them in a three dimensional pattern-space for visualisation. The clustering results are show in figure 4.8. Figure 4.8(a) shows the clustering result of the feature vectors constructed from the coefficients of the symmetric polynomials computed from the complex spectral matrix $\mathcal{S}$ of the Hermitian matrix using equation 4.7. Figure 4.8(b) shows the clustering result for the feature vectors constructed from the eigenvalues of the real Laplacian matrix as explained above. Figure 4.8(c) shows the result for the feature vectors constructed from the spectrum of the Hermitian matrix, i.e. the feature vector whose elements are the eigenvalues of the Hermitian matrix. We repeat the same experiment with five different poses. We randomly take five poses of randomly selected subjects.



(a) 5 poses, Hermitian (Symmetric Polynomial Coefficients)

(b) 5 poses, Laplacian Spectra



(c) 5 poses, Hermitian Spectra

Figure 4.9: Performance of clustering, 5 poses

We construct the three set of feature vectors for all the input poses. We project the feature vectors into a three dimensional space for visualization. The results are shown in figure 4.9. The empirical results show that the weighed Laplacian matrix which records the distances only as its edge weights is not suitable for clustering the human pose data obtained from Microsoft Kinect. Both type of feature vectors produced from the Hermitian property matrix gives better class separation than the Laplacian matrix for the human skeletal data.

To evaluate the clustering results we apply k-means algorithm to the embedded points to obtain clusters. Then we compute the Rand indices to assess the clustering results we get using the three type of feature vectors we construct. The

	Rand Indices			
# of poses	2	3	4	5
Hermitian (Symmetric Polynomial)	0.99	0.93	0.90	0.87
Hermitian Spectrum	0.90	0.86	0.72	0.65
Laplacian Spectrum	0.52	0.21	0.13	0.08

Table 4.1: Rand Indices Comparison

Rand index is define as

$$R_I = \frac{X}{X+Y} \tag{4.11}$$

where X is the number of agreements and Y 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 the experiment, this counts as an agreement. If two objects are in the same cluster in the ground truth clustering but are in different clusters from the experiment, this counts as a disagreement. The value of Rand index is always between 0 and 1. Rand index of 1 means a perfect clustering result.

Table 4.1 shows the Rand indices obtained when clustering is attempted using different number of poses. The first row shows the Rand indices obtained using the feature vectors constructed from the symmetric polynomial coefficients of the Hermitian matrix detailed in section 4.3 (referred to as 'Hermitian Symmetric Polynomials'). The second row shows the Rand indices obtained using the feature vectors constructed from eigenvalues of the Hermitian matrix (referred to as 'Hermitian Spectrum'), while the third row shows the Rand indices obtained using the feature vectors constructed from eigenvalues of the Laplacian matrix (referred to as 'Laplacian Spectrum'). The same statistics have been shown in the Figure 4.10 visually which shows that the clustering results using the angular information is better than that of the Laplacian spectral pattern vectors.

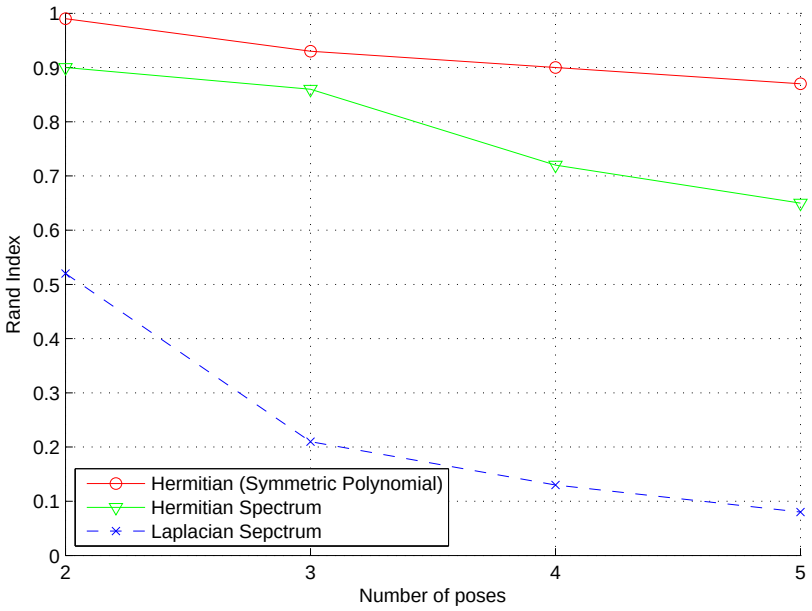


Figure 4.10: Rand Indices Comparison

In the next experiment, we investigate the combination of the feature vectors and the embedding methods gives best results. Here again we randomly choose five poses of randomly selected subjects. We take 150 different instances of each pose. Then we construct four sets of feature vectors F_i according to the steps mentioned in Section 4.5 using equation 4.7, 4.8, 4.9 and 4.10 respectively. The first set of feature vectors is established by interleaving the real and imaginary components of the complex coefficients of the elementary symmetric polynomials and by stacking these to form long feature vectors F_i. The second set of feature vectors is constructed by taking the magnitude of the complex coefficients of the elementary symmetric polynomials and putting them as components of the feature vectors F_i. The third set of feature vectors is constructed by putting the real part of the complex coefficients of the elementary symmetric polynomials into the feature vectors F_i. The imaginary part is ignored. Finally, the fourth set of feature vectors are built using only the imaginary part of complex coef-

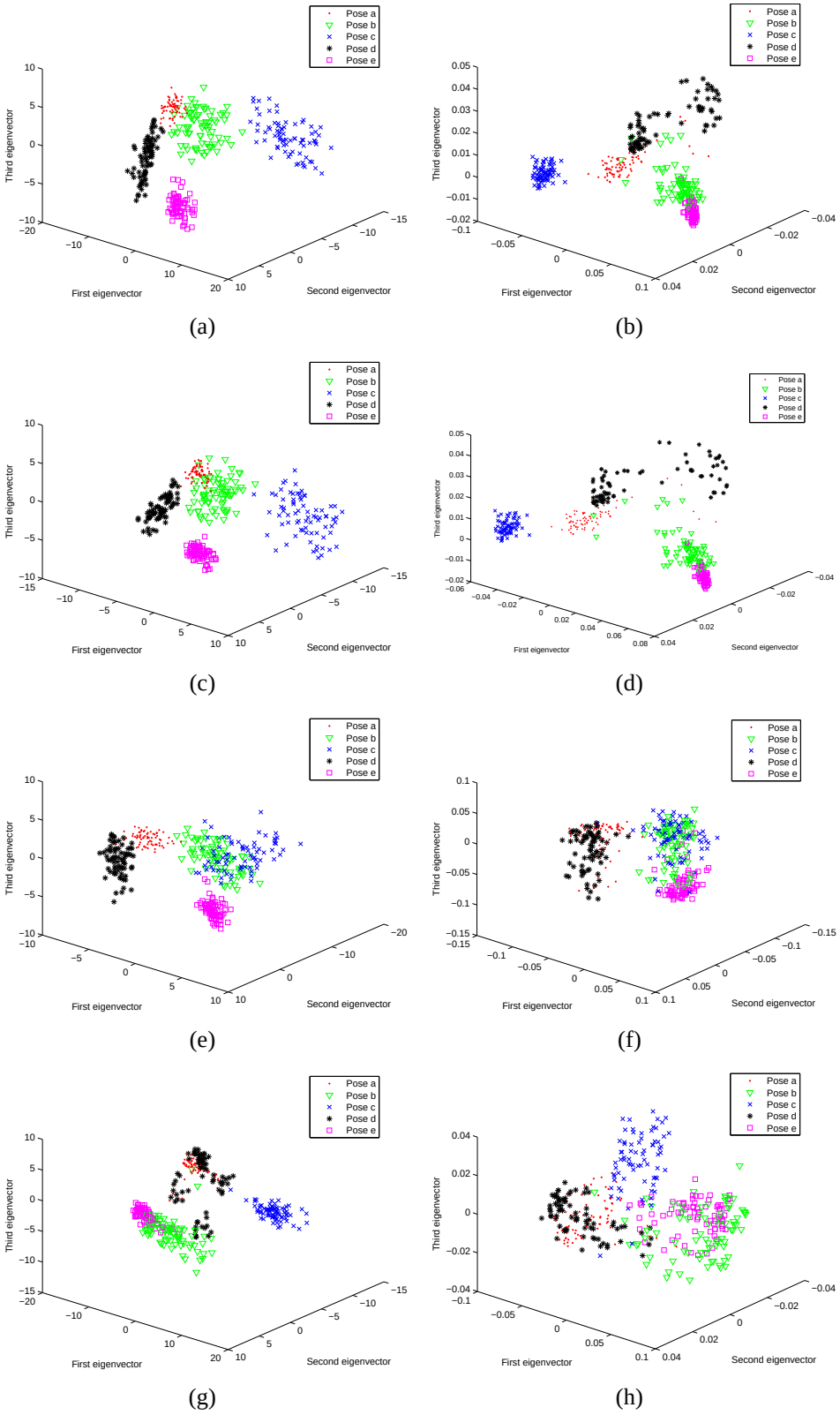


Figure 4.11: Performance of clustering using different feature vectors, with PCA (left-hand column) and MDS (right-hand column) embedding

	Rand Indices	
Feature vector / Embedding method	PCA	MDS
Real+Imaginary part interleaved	0.9982	0.8819
Magnitude	0.9452	0.8616
Real part only	0.8817	0.7972
Imaginary part only	0.7183	0.6586

Table 4.2: Rand Indices Comparison using different feature vectors, with PCA and MDS embedding

ficients of the elementary symmetric polynomials. The real part is disregarded. We then embed the feature vectors into a three dimensional pattern-space by performing the principal component analysis (PCA) and multidimensional scaling (MDS). Figure 4.11 shows the results of the clustering using the first three eigenvectors. The left-hand column shows the results obtained with PCA. The right-hand column shows the results obtained with MDS. The first row shows the results for the first set of feature vectors constructed using equation 4.7. Similarly, second, third and fourth row show the results for the set of feature vectors constructed using equation 4.8, 4.9 and 4.10 respectively. After embedding the feature vectors into a three dimensional pattern space using PCA and MDS, we locate clusters using the k-means algorithm and compute the Rand indices. The Rand indices for all the combinations are shown in Table 4.2. The feature vectors constructed using equation 4.7 with principal component analysis (PCA) gives the best clustering performance. The second best performance is obtained with the feature vectors constructed using equation 4.8 with PCA. The poorest clustering result is given by the feature vectors constructed using equation 4.10 with multidimensional scaling (MDS).

4.9 Summary

In this chapter our aim was to cluster similar human poses represented by the skeleton with 20 points acquired from the Kinect device for Xbox 360 using the spectral graph techniques. We studied how to extract feature vectors from the human skeletal data acquired from the Kinect device. We converted the input graph into its equivalent line graph. We used the spectrum of a Hermitian property matrix employing the angle between the limbs and the lengths of the limbs. The nodes of the graph represented the joints and the edges represented the human body limbs. We used the length of the limbs and the angle between a pair of limbs as features. From the spectrum of the Hermitian property matrix we constructed four different types of feature vectors (detailed in section 4.5) using the complex coefficients of the symmetric polynomials. We embedded those feature vectors into pattern-space using two embedding methods i.e. principal component analysis (PCA) and multidimensional scaling (MDS). For comparison we constructed feature vectors from the eigenvalues of the Laplacian (real) and the eigenvalues of the Hermitian property matrix. Experimental results provided (both quantitative and qualitative) suggest that Hermitian matrix produced best performance with PCA for the human poses clustering problem.

CHAPTER 5

Eigenvector Sign Correction

5.1 Introduction

Correspondence matching between 2D images is the preprocessing step for a number of computer vision algorithms. The problem of feature correspondence matching is to find a one-to-one correspondence between feature points in a pair of 2D images that represent an object in the image. The images can be taken from a different point of view, at different times. In literature many different methods have been presented to address the problem of correspondence matching. These methods can be broadly categorized into two classes namely the non-spectral methods (Ling & Jacobs 2007; Chui & Rangarajan 2000) and the spectral methods (Shapiro & Brady 1992; Scott & Longuet-Higgins 1991; Umeyama 1988). Spectral methods solve the problem using the eigenvalues and eigenvectors of the adjacency matrix or the Laplacian matrix (degree matrix minus the

adjacency matrix) for the point set arrangement. Correspondence matchings are computed by embedding the graphs into a common eigenspace using an eigen-decomposition of the point-proximity matrices, where correspondences are computed by closest point matching in this eigenspace. However, arbitrary determination of the signs of the eigenvectors returned by a numerical solver causes error in correct correspondence matchings. This problem needs to be handled and has already been reported in previous works (Shapiro & Brady 1992; Caelli & Kosinov 2004).

In this chapter we address the problem of eigenvector sign correction for the problem of correspondence matching. We propose a novel method that solves the problem of eigenvector sign flipping by using the co-efficient of the symmetric polynomials.

5.2 Eigenvector Sign Flip Problem

Spectral graph based correspondence matching algorithms commence by constructing the proximity matrices from the given set of points. The structural information present in the proximity matrices of the point sets are used to establish correspondences between the point sets. The work of Shapiro and Brady (Shapiro & Brady 1992) is one of the earliest and state of the art algorithm. Shapiro and Brady proposed an algorithm to match 2D feature points across a pair of images using the eigenvectors of a proximity matrix computed from the intra-image distances between each pair of feature points. As input, the algorithm receives a set of m feature points x_i in image I_1 and n feature points y_j in image I_2. Each image feature point is assigned a coordinate in the higher space i.e. each 2D point in image I_1 is mapped from 2D image-plane into an m dimensional hyper-space, and each 2D point in image I_2 is mapped from 2D image-plane into an

n dimensional hyperspace. This mapping is performed independently for each image, and when the shapes of the images are similar, the corresponding feature points coincide in the hyperspace.

The eigenvectors or the modes of a single image having m features x_i are computed from a square proximity matrix H. The matrix H is created, recording the affinity between each pair of feature points within the image.

$$H_{ij} = e^{-d_{ij}^2/2\sigma^2} \tag{5.1}$$

where $d_{ij}^2 = ||x_i - x_j||^2$ is the squared Euclidian distance between each pair of feature points. H is a symmetric matrix and its diagonal entries are unity. The parameter σ controls the interaction between feature points. For small σ the interaction is local, while for large σ each feature point is more globally aware of its surroundings. The next step is to compute the eigenvalues λ_i and the eigenvectors e_i of the matrix H, i.e. by solving

$$He_i = \lambda_i e_i, i = 1 \ldots m,$$

The eigenvectors form an orthonormal basis as the eigenvectors are of unit length and are mutually orthogonal. In matrix form

$$H = \Phi \Lambda \Phi^T \tag{5.2}$$

where the diagonal matrix Λ contains the eigenvalues along its diagonal in decreasing order. The modal matrix Φ is orthogonal and has the eigenvectors as its column vectors i.e. $\Phi = [e_1| \ldots |e_m]$. Each row of Φ can be thought of a feature

vector F_i, containing the m model coordinates of feature point i.

$$\Phi = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix}$$

This computation is done for both images I_1 (with m feature points) and I_2 (with n feature points). Two sets of *feature vectors* are obtained, i.e., $F_{i,1}$ and $F_{j,2}$ one for each image respectively. The final step is to compute the association matrix Z. The elements of Z_{ij} shows the confidence in the match between the feature points x_i and y_j. The least significant $|m - n|$ eigenvectors and feature vectors are discarded from the larger modal matrix, in the case where the two modal matrices are of different sizes. The best matches are given by the elements of the association matrix Z which are smallest in their row and column. The values Z_{ij} is the Euclidean distance between feature vectors, i.e.

$$Z_{ij} = ||F_{i,1} - F_{j,2}||^2 \tag{5.3}$$

However, before computing the association matrix Z, the direction of the both sets of the eigenvectors must be made consistent. The sign of each eigenvector is not unique as the signs of the eigenvectors returned by a numerical solver are assigned arbitrary and switching its direction does not violate the orthogonality of the basis. When calculating the distance between two feature vectors in equation (5.3), signs play a critical role. In case of inconsistent eigenvector signs, we need to change the sign of feature vector components in one of the two eigenvector matrices to be consistent with the other.

If H_1 represents the proximity matrix of a set of points and H_2 represents

the proximity matrix of the same set of points after reordering the labels of the points, the two proximity matrices will contain the same measures but at different locations. Consequently, the eigenvalues obtained from the two matrices will be the same except that their components will be in different order. When $m \neq n$, the eigenvalues obtained from the two proximity matrices H_1 and H_2 are both order from the largest to the smallest. Similarly, the eigenvectors of H_1 and H_2 are reordered so that their order match the order of their eigenvalues. Then m eigenvectors are used to create the feature vectors, from which the values of matrix Z are calculated.

5.3 Sign Correction Methods

Several researchers have proposed different methods to correct the direction of the eigenvectors. For instance,

1. Park et al. (Park et al. 2000) have suggested a method to correct the direction of the eigenvectors. Let V and V' be the modal matrices with e and e' as its eigenvectors respectively. In (Park et al. 2000) each eigenvector e_i is compared with it counterpart e'_i and the sign of e_i is corrected so that

$$e_i := \begin{cases} e_i, & \text{if } ||e_i + e'_i|| > ||e_i - e'_i|| \\ -e_i, & \text{otherwise} \end{cases}$$

   where e_i and e'_i are the corresponding eigenvectors of the two adjacency matrices computed from the two images respectively.

2. Umeyama (Umeyama 1988) has handled the problem of eigenvector sign correction by taking the absolute values of the components of the eigenvector of both the modal matrices. Umeyama's method works fine under

three conditions. First, when noise is sufficiently low. Second, when the eigenvalues of the proximity matrices are not very close to each other. Third, when the rows of the absolute modal matrices are sufficiently different from each other.

3. Caelli and Kosinov (Caelli & Kosinov 2004) find the number of positive and negative components for each eigenvector. The eigenvector is multiplied by -1 if the number of negative components is greater than the number of positive components. This is essentially a dominant sign correction, always ensuring that there are more positive entries in each eigenvector. However, this is highly unreliable since specific to spectral correspondence, the eigenvectors tend to have about the same number of positive and negative entries due to orthogonality to a constant eigenvector.

4. Shapiro and Brady (Shapiro & Brady 1992) suggested a greedy approach to correct the direction of the eigenvectors. They treat one modal matrix as reference basis and proceed to orient the axes of the other modal matrix one at a time by optimizing for a correspondence cost, choosing for each that direction which maximally aligns the two sets of feature vectors (Shapiro 1991).

5.3.1 Symmetric Polynomials

A symmetric polynomial is a polynomial $S(x_1, x_2, \ldots, x_n)$ in n variables, such that if any of the variables are interchanged, the same polynomial is obtained. A symmetric polynomial is invariant under permutation of the variable indices. Symmetric polynomials arise naturally in the study of the relation between the roots of a polynomial in one variable and its coefficients (Wikipedia 2013). There is a special set of symmetric polynomials referred to as the *elementary symmet-*

ric polynomial (S) that form a basis set for symmetric polynomial. Any symmetric polynomial can be expressed as a polynomial function of the elementary symmetric polynomials. For a set of variables $x_1, x_2, \ldots, x_n$ the elementary symmetric polynomials can be efficiently computed using the power symmetric polynomials using the Newton-Girard formula detailed in section 4.4.

5.3.2 Proposed Method

Our proposed method for eigenvector direction correction is based on the use of the coefficients of the elementary symmetric polynomials. For any two eigenvectors the corresponding odd coefficients have opposite sign if their directions are not consistent with each other. Our approach is similar to that of Shapiro and Brady, i.e. we treat one modal matrix as reference basis and proceed to orient the axes of the other model matrix one at a time, by comparing their corresponding coefficients of symmetric polynomials. If the corresponding odd coefficients for the two eigenvectors have opposite sign then we multiply one of the eigenvectors by -1 to flip its direction. Any odd coefficients will work, for instance, using only the first coefficients should work. Which is essentially the sum of the eigenvector components. i.e. $S_1(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i$. However, if the sum of the eigenvector components is nearly equal to zero then we move to the next odd coefficients to compare.

The Algorithm

The following steps show how to correction the sign of eigenvectors for correspondence matching.

Input: Proximity matrices A and B

1. Find eigen-decomposition, $A = V_A \Lambda V_A^T$ and $B = V_B \Lambda V_B^T$

2. Discard the least significant eigenvectors and the feature vectors are discarded from the larger modal matrix in the case where A and B are of different sizes. Let N be the size of the smaller matrix.

3. Compute the coefficients symmetric polynomial of each column of matrix V_A and V_B, Let S_A and S_B be the matrices containing the coefficients in its columns, computed from the corresponding columns of the model matrices V_A and V_B respectively.

4. for v := 1 to N
   i := 1
   flipflag := False
   while (i $<$ N and not flipflag) do
   if (S_A[i, v] $*$ S_B[i,v]$<$0) then
   Flip the sign of the vth eigenvector in matrix V_B
   flipflag := True
   end if
   i := i + 2
   end while
   end for

5.3.3 Eigenvector Sign Correction for EM Algorithm

In Chapter 3 we have discussed the EM algorithm developed by Carcassoni (Carcassoni & Hancock 2003) for the point pattern matching in detail. The algorithm works very well and offers a powerful means of estimating the transformation parameters. However, there is one problem that restrict the automatic use of the method, which is the need to initialise the parameters. The quality of the corres-

ponding matching results very much depends on a good choice of initial values of the parameters. The authors use classical multidimensional scaling on the pairwise dissimilarity matrices S_D (for data-points) and S_M (for model-points) computed from the given point-sets. MDS embed the dissimilarity matrices onto a low dimensional space using the eigen-decomposition. The embedded coordinates for both the point-sets are used to compute the initial correspondence probabilities for the EM algorithm to proceed.

Since, MDS uses the eigen-decomposition to embed the point-set, therefore the embedded coordinates for both the point-sets may not align properly because of the eigenvector sign flip problem. The EM alignment algorithm developed by Carcassoni produces very bad result if the eigenvectors are not made consistent with each other by correcting their signs. This is illustrated in figure 5.2.

5.4 Computational Complexity

The steps of the algorithm are given in Section 5.3.2.

5.4.1 Complexity

Let M and N be the sizes of the two matrices. Without the loss of generality, we can assume $M > N$. The execution time for the eigen-decomposition step is O(M^3) + O(N^3). Each power symmetric polynomial can be computed in O(N) time. There are N such polynomials for each matrix, so the total running time to compute all the power symmetric polynomials is O(N^2). Finally the elementary symmetric polynomials can be computed in linear time, once the power symmetric polynomials are computed. Hence the total running time of computing symmetric polynomials becomes O(N^2).

In step 4, both the outer loop and the nested loop take O(N) time. Hence the

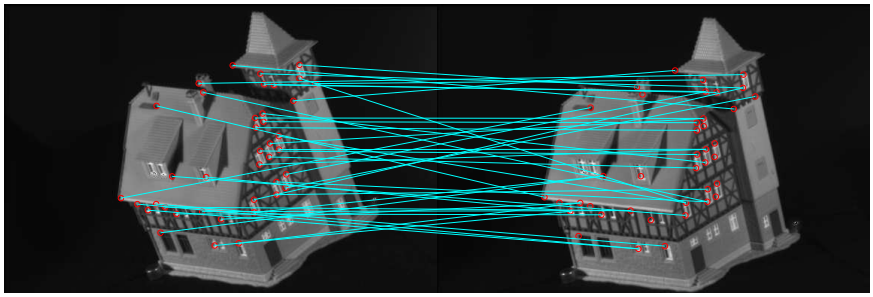
total time is O(N^2) for step 4. Step 3 and 4, each take quadratic time in number of points. Hence the total running time becomes O(M^3) + O(N^3) + O(N^2) = O(M^3), where M is the size of largest matrix.

	Number of incorrect matches					
Frames	1-10	1-20	1-30	1-40	1-50	1-60
Park et al.	21	29	34	37	38	37
Caelli & Kosinov	5	7	16	19	19	21
Umeyamma	0	1	5	8	13	16
Symmetric Polynomials	0	0	0	3	7	9

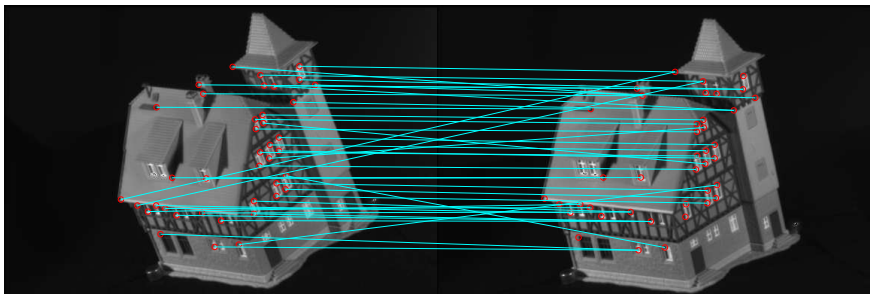
Table 5.1: Performance of sign correction methods on the CMU/VASC house sequence. The first image frame has been matched against the $10^{th}, 20^{th}, 30^{th}, 40^{th}, 50^{th}$ and 60^{th} frame

5.5 Experimental Results

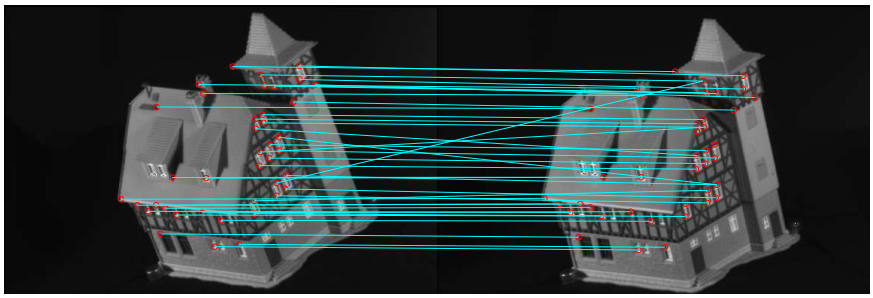
In this section, we provide some experimental results of the correspondence matching affected by the problem of eigenvector sign flipping. We use different techniques for the eigenvector sign correction detailed in Section 5.3. We extract 40 feature points from the 1^{st}, 10^{th}, 20^{th}, 30^{th}, 40^{th}, 50^{th} and 60^{th} frame of the CMU/VASC model house sequence. Table 5.1 shows the number of incorrect correspondences obtained matching the first frame with the other frames mentioned above. Figure 5.1 shows correspondence matching between of the feature points extracted from frame 1 and frame 30 of the CMU/VASC model house sequence, after using four different method to correct the sign of the eigenvectors. The first image (figure 5.1(a)) is the result of the method proposed by Park et al. (Park et al. 2000). The second image (figure 5.1(b)) is the result of the method proposed by Caelli and Kosinov (Caelli & Kosinov 2004), the third image (figure 5.1(c)) shows the result of the correspondence after the eigenvector



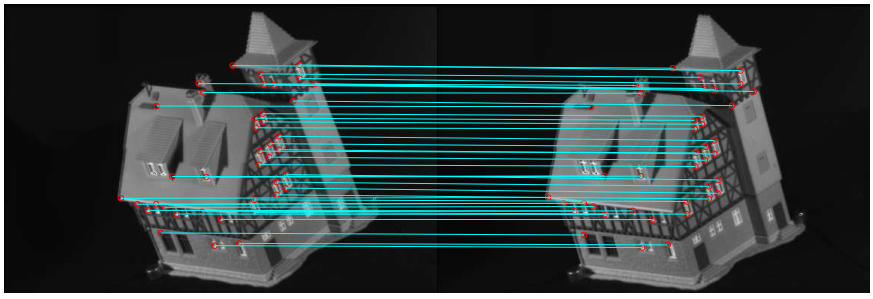
(a)



(b)

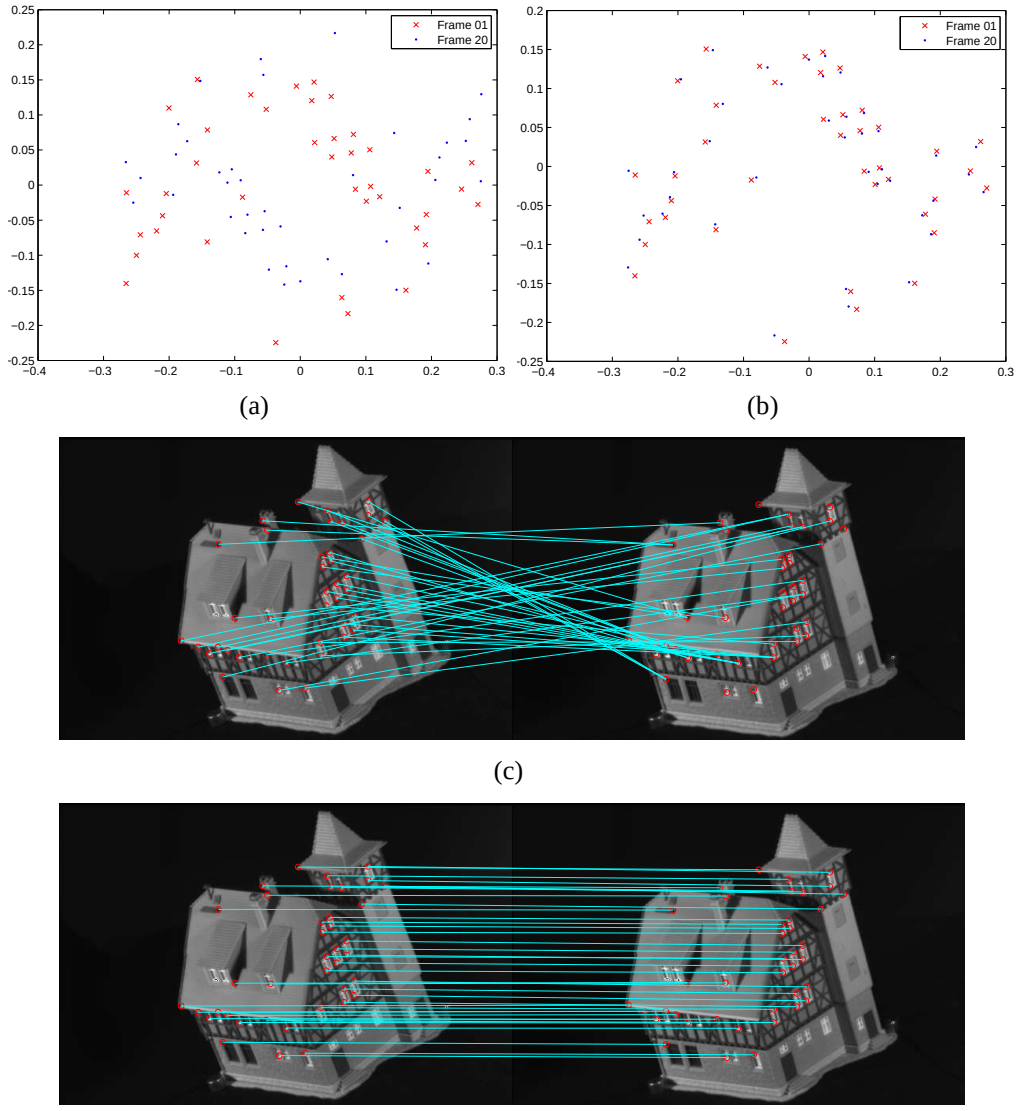


(c)



(d)

Figure 5.1: Comparing different eigenvector direction correction methods, a) Park et al. b) Caelli & Kosinov. c) Umeyamma. d) Symmetric Polynomials



(a) (b)

(c)

(d)

Figure 5.2: Effect of the eigenvector sign correction on Carcassoni's alignment EM algorithm, a) Embedded point without sign corrections. b) Embedded point after sign corrections. c) Correspondence matching without sign corrections. d) Correspondence matching after sign corrections

sign corrections using the method proposed by Umeyamma (Umeyama 1988) and the final image (figure 5.1(d)) shows the result of the correspondence after the eigenvector sign corrections using the coefficients of symmetric polynomials.

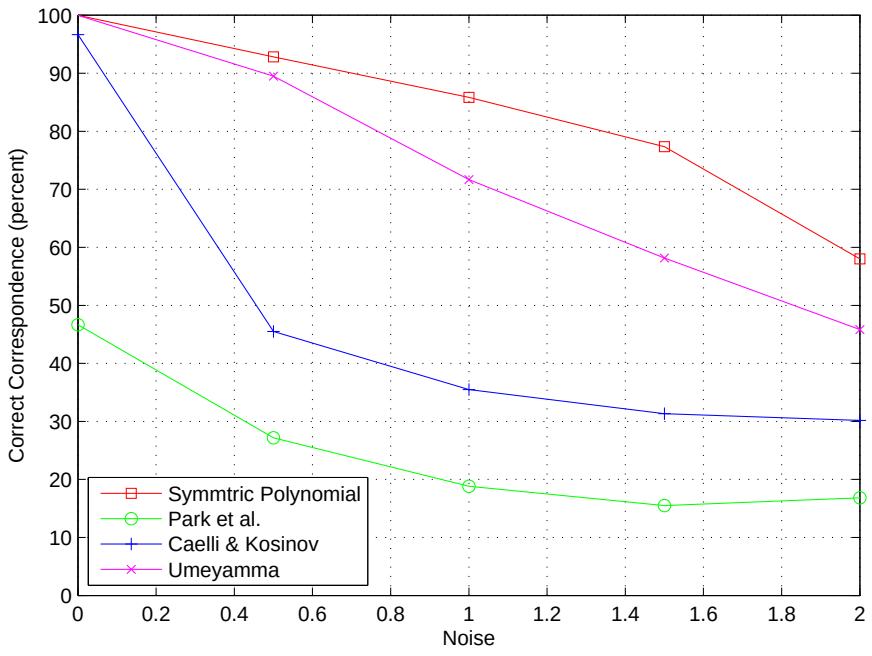


Figure 5.3: Effect of increasing noise on correct correspondences using different eigenvector sign correction strategies

It is clear from the figure that out of 40 correspondences, the method proposed by Park et al. produces 34 wrong correspondences, the method developed by Caelli and Kosinov produces 16 wrong correspondences, the method proposed by Umeyamma produces 5 wrong correspondence while our proposed method (Symmetric Polynomials) produces 100% correct correspondences.

In the next experiment, we compare the performance of the different eigenvectors sign correction strategies against the Gaussian noise added in the point positions. Figure 5.3 shows the fraction of correct correspondences of the four eigenvectors sign correction strategies as a function of the increasing random point-position jitter. Random position jitter is simulated by adding randomly generated position error sampled from a 2D Gaussian distribution to the data point-set. The performance of all the methods decreases with the increase in the noise level. However, the best performance is obtained by using the coef-

ficients of symmetric polynomials. The next best performance is obtained by Umeyamma's strategy. The poorest performance is returned by the method of Park et al.

In the final experiment, we show the results of the correspondence matching by the EM algorithm proposed by Carcassoni (Carcassoni & Hancock 2003) with and without the sign correction. 40 points are extracted from the 1st and 20th frame of the CMU/VASC model house sequence for matching. Figure 5.2(a) and 5.2(b) show the embedding points of the two frames. Red crosses are the embedded points from Frame 1 while the blue dots represent the embedded points from Frame 20. Figure 5.2(a) shows the embedded points when the signs of the eigenvectors have not been corrected. The resulting correspondences are shown in Figure 5.2(c). Figure 5.2(b) and Figure 5.2(d) show the embedded points and the resulting correspondences respectively when the signs of the eigenvectors are corrected. Note that without correcting the signs of the eigenvectors the EM alignment algorithm can not compute the correct correspondences.

5.6 Summary

In this chapter we have investigated the problem of the eigenvector sign correction for correspondences matching. The eigenvector sign correction is an important step in all graph spectral correspondence matching techniques. If the sign of the eigenvector are not corrected properly, the robust alignment algorithms like the one developed by Carcassoni (Carcassoni & Hancock 2003) can fail to produce good results.

We used the coefficients of the symmetric polynomials to solve the problem. We also compared our method to some other methods already proposed in the literature, and found that using the coefficients of the symmetric polynomials

solved the problem better than the others.

CHAPTER 6

3D Shape Analysis using Commute Time

6.1 Introduction

The rapid advancement in the digital technology in 3D shape modelling, digitizing and processing has led to an increasing number of 3D models, both on the internet and in domain specific databases. Computing the similarity between 3D shapes is a fundamental task in shape-based recognition, retrieval, clustering, and classification. The aim of 3D shape analysis is to establish a shape descriptors or signatures which capture the important properties of the shapes for the purpose of classification, clustering, retrieval and correspondence matching. Shape descriptors are mathematical functions which are applied to a shape and produce numerical values which represent the shape.

Spectral methods have recently been used to establish shape descriptors which can also be used to measure the similarity of 3D shapes. For instance, diffusion

geometry methods were used to define low dimensional representations of manifolds. Rustamov (Rustamov 2007) has suggested using the eigen-decomposition of the Laplace-Beltrami operator to construct an isometric invariant surface representation, aiming to measure similarity between non-rigid shapes, rather than for correspondence detection. The Global Point Signature (GPS) suggested by Rustamov (Rustamov 2007) for shape comparison employs the discrete Laplace-Beltrami operator, which globally captures the shape's geometry. The Laplace-Beltrami operator was later employed by many other researchers. For instance, Sun et al. (Sun et al. 2009) defined a point signature based on the properties of the heat diffusion process on a shape, referred to as the Heat Kernel Signature (HKS) for brain classification. HKS is obtained by restricting the well-known heat kernel to the temporal domain. Ovsjanikov et al. (Ovsjanikov et al. 2010) employed a heat diffusion process to construct the Heat Kernel Maps for the shape matching. Castellani et al. (Castellani et al. 2011) have extended the idea of Heat Kernel Signature (HKS). The local heat kernel values observed at each point are accumulated into a histogram for a fixed number of scales leading to the so-called Global Heat Kernel Signature (GHKS). In a recent paper (Aubry et al. 2011), based on quantum mechanical approach, Aubry et al. have developed the Wave Kernel Signature (WKS) for characterizing points on non-rigid 3D shapes. They have shown that their signature performed better than the Heat Kernel Signature (HKS).

Despite significant efforts in the past ten to fifteen years, graph clustering and classification remain an open challenge in the machine learning community. One of the most promising approaches is to use spectral clustering methods which exploit graph representations of the data and locate clusters by partitioning the graph that optimize an edge cut criterion. Early spectral approaches recursively compute the normalized cut (Shi & Malik 2000) over the graph using the first

non-zero Laplacian eigenvector (also known as the Fiedler vector) (Chung 1997) and are referred to as spectral bi-partitioning (SB) methods. Unfortunately, this does not guarantee good clusters as the normalized cut is computed recursively, irrespective of the global structure of the data (Belkin & Niyogi 2003). Qiu and Hancock (Qiu & Hancock 2007) have used commute time for the purpose of image segmentation and have shown that the commute time method outperforms the normalized cut.

Recently, the graph spectral methods defined in the context of clustering have been applied to 3D shape processing. The 3D shape is represented by a mesh that approximates the boundary surface of the shape. In this context, spectral invariants such as the eigenfunctions of the Laplacian operator can be used for near-isometric shape matching. For instance, Mateus et al. (Mateus et al. 2008) used eigenmaps obtained by the first k eigenfunctions of the Laplace operator as low-dimensional Euclidean representations of non-rigid shapes for the purpose of 3D point registration. Cuzzolin et al. (Cuzzolin et al. 2008) and Lee et al. (Lee et al. 2008) have performed segmentation for mesh sequences. However, the former method computes only protrusions, while the later uses an additional skeleton. In (Mateus et al. 2008), the authors use locally linear embedding (LLE) to represent a cloud of points and perform segmentation in the LLE space. The segments obtained are then propagated across time to obtain a temporally coherent segmentation of a voxel-sequence into protrusions of the shape. The method works well to segment rigid body parts (such as head, hands and legs etc), but it cannot be used directly for identifying rigid body-parts (for example, separating the upper-arm from the lower-arm).

In this chapter we construct a novel 3D shape distribution for the purpose of 3D object classification. The method commences from a modification of the 3D shape distribution reported in (Osada et al. 2001). Instead of using Euclidean

distances between pair of points on the shape, we use commute-time distance computed from the eigenvalues and the eigenfunctions of the Laplace-Beltrami operator. The empirical results show that the distribution computed using our method gives a better shape signature than (Osada et al. 2001).

6.2 Laplace-Beltrami operator

Let f be a real valued function defined on a differentiable manifold $\mathcal{M}$ with Riemannian metric. The Laplace-Beltrami operator, like the Laplacian, is the divergence of the gradient of f i.e.

$$\Delta f = \text{div}(\text{grad}(f)) \tag{6.1}$$

where grad and div are the gradient and divergence on the manifold respectively. The Laplace-Beltrami operator is a self-adjoint and semi-positive definite operator (Rosenberg 1997). The Laplacian eigenvalue problem is given by the following equation

$$\Delta f = \lambda f \tag{6.2}$$

where λ is the eigenvalue and f is the eigenfunction. The Laplace-Beltrami operator has an ortho-normal eigensystem, that is a basis of the space of square integrable function, with $\Delta\phi_i = \lambda_i\phi_i$, $\lambda_0 \leq \lambda_1 \leq \ldots, \lambda_i$. where λ_i are the eigenvalues and ϕ_i are the corresponding eigenfunctions.

Most of the techniques (Rustamov 2007; Meyer et al. 2003) for characterizing points on non rigid 3D shapes use the eigenpairs of the Laplace-Beltrami operator. The combinatorial Laplacian is suitable for the meshes only and it does not contain much information about the shape. The discrete Laplacian or Laplace-Beltrami operator captures the geometric and topological properties of

the surface. The solution to the eigenvalue problem (equation 6.2) is approximated by a piecewise linear function over a triangulation with vertices p_i for $i = 1, \ldots, n$. The discrete Laplace-Beltrami operator can be written as

$$\Delta f(p_i) = \frac{1}{s_i} \sum_{j \in N(i)} w_{ij}(f(p_i) - f(p_j)) \tag{6.3}$$

where $N(i)$ are the indices of all the vertices connected to p_i by an edge, s_i are the masses related to vertex i and the w_{ij} are the weights associated with the edges. To write the definition of discrete Laplace-Beltrami operator in equation 6.3 in the matrix form we need to define a vector $\vec{f} = (f(p_1), \ldots, f(p_n))^T$ whose entries/components are the values of the function f defined at different vertices p_i, a weighted adjacency matrix W, whose entries are the weights w_{ij} associated with the edges, a diagonal degree matrix D whose diagonal entries $D_{ii} = \sum_{j \in N(i)} w_{ij}$, a stiffness matrix $A = D - W$ and a diagonal mass matrix S whose entries $S = \text{diag}(s_1, \ldots, s_n)$. Then we can define the Laplacian matrix as $L = S^{-1}A$. Here, $\Delta f(p_i)$ is the i^{th} component of the vector $L\vec{f}$ and equation 6.2 can be written as $L\vec{f} = \lambda\vec{f}$. Since, L is not symmetric due to the fact that each row of the matrix A is divided by different s_i, therefore, equation 6.2 can be written as a generalized eigenvalue problem $A\vec{f} = \lambda S\vec{f}$.

There are a number of ways to select the edge weights w_{ij} and the masses s_i to construct the Laplacian matrix. One way is to take the weights w_{ij} equal to 1 if the vertex p_i is connected with the vertex p_j and 0 if they are not connected, and assume the masses s_i = 1. In this way we get traditional Laplacian that only considers the structure of the mesh and ignores the underlying geometry of the shape. Such approaches are therefore not suitable for 3D shapes. Many schemes have been proposed to construct the discrete Laplacian that estimates the Laplace-Beltrami operator for 3D shapes. The majority of them use the so

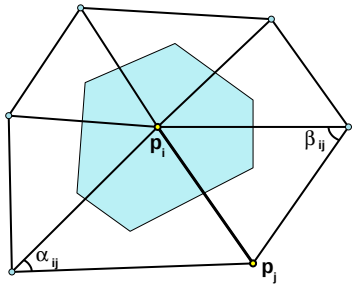


Figure 6.1: Definitions of the angles and the area appearing in the discrete Laplace-Beltrami operator.

called cotangent scheme that uses the angles and the area of the region obtained by joining the circumcenters of all the triangles around the vertex on the shape. For instance, Pinkall and Polthier's (Pinkall et al. 1993) work is one of the early works on the geometric approaches. The weights are computed from the cotangents of the angles opposite to the edge between vertex p_i and p_j as

$$w_{ij} = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2}$$

where α_{ij} and β_{ij} are the angles opposite to the edge between p_i and p_j as shown in Figure 6.1. Since this methods does not include the masses, the weights computed form the cotangents are very much dependent on the mesh sampling. Desbrun et al. (Desbrun et al. 1999) solved this problem by taking the average area of the triangles at the vertex i as the masses s_i. Meyer at al. (Meyer et al. 2003) modify the method of Desbrun et al. by taking the masses s_i equal to the area obtained by joining the circumcenters of all the triangles around the vertex i, shown in the Figure 6.1.

Xu (Xu 2006) modified the method proposed by Meyer et al. This modification gives better convergence properties. In this chapter we will follow Xu's method to construct the discrete Laplacian (Laplace-Beltrami operator).

6.2.1 The generalized eigenvalue problem

For a function f defined on the surface, the Laplacian Δf is approximated as

$$\Delta f \approx \frac{1}{s_i} \sum_{j \in N(i)} w_{ij}[f(p_j) - f(p_i)]$$

where $N(i)$ are the neighbours for the vertex p_i and w_{ij} is the weight assigned to the edge between point p_i and p_j . The above formula can be written as $\Delta f \approx Lf$. Here L is the discrete Laplacian matrix. The weight w_{ij} of the edge is given by

$$w_{i,j} = \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} \tag{6.4}$$

The angles appearing in this formula i.e. α_{ij} and β_{ij} are shown in the figure 6.1. The area s_i is also shown as the shaded region in the same figure. We compute the Laplacian, which has the entries as follows

$$L(i,j) = \begin{cases} \sum_k w(i,k)/s_i & \text{if } i = j \\ -w(i,j)/s_i & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

The standard eigenvalue problem for L is $L\phi = \lambda\phi$, where λ is the eigenvalue of L and ϕ is the corresponding eigenvector. The area s_i at each vertex is computed as

$$s_i = \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{8} ||p_i - p_j||^2 \tag{6.5}$$

Since the areas s_i computed at the vertices of the mesh are different, hence, the discrete Laplacian matrix L computed is not symmetric. This may cause the eigenvalues and eigenfunction to be complex. Therefore, we solve the generalized eigenvalue problem. Let S be the diagonal matrix with entries $S_{ii} = s_i$ and

$W_{ij} = w_{ij}$ be the symmetric weight matrix. Since $L = S^{-1}W$, therefore, we can rewrite the equation $L\phi = \lambda\phi$ as $S^{-1}W\phi = \lambda\phi$ or

$$W\phi = \lambda S\phi \tag{6.6}$$

Once we have the eigenvalues and eigenfunction of L to hand, we can compute the commute time matrix using the eigenvalues and eigenfunction.

6.3 Commute Time

In this section, we briefly review how to compute the commute time and describe its relationship to the graph Laplacian. Commute time is the average time taken by a random walker on a graph walking from a node u to node v and then back to node u. The commute time can be computed from the Laplacian spectrum as it has a close relationship with the graph Laplacian and heat kernel.

Consider a weighted graph by the triple $\Gamma = (V, E, \Omega)$, where V is the set of nodes, $E \subseteq V \times V$ is the set of edges, and Ω is the weighted adjacency matrix.

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

where $w(u,v)$ is the weight on the edge $(u,v) \in E$. Furthermore, let $T = diag(d_u; u \in V)$ be the diagonal weighted degree matrix with elements given by the degrees of the nodes, $d_u = \sum_{v=1}^{|V|} w(u,v)$. The *unnormalized* weighted Laplacian matrix is given by $L = T - \Omega$ and the *normalized* weighted Laplacian

matrix is defined to be $\mathcal{L} = T^{-1/2}LT^{-1/2}$ and has elements

$$\mathcal{L}(u,v) = \begin{cases} 1 & \text{if } u = v \\ -\frac{w(u,v)}{\sqrt{d_u d_v}} & \text{if } u \neq v \text{ and } (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$$

The spectral decomposition of the normalized Laplacian is $\mathcal{L} = \Phi\Lambda\Phi^T$ where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the diagonal matrix with the ordered eigenvalues as the elements satisfying the condition $0 = \lambda_1 \leq \lambda_2 \leq ..., \leq \lambda_{|V|}$ and $\Phi = (\phi_1|\phi_2|...|\phi_{|V|})$ is the matrix with the ordered eigenvectors as columns.

The *hitting time* $O(u,v)$ of a random walk on a graph is defined as the expected number of steps before node v is visited, commencing from node u. The *commute time* $CT(u,v)$, on the other hand, is the expected time for the random walk to travel from node u to reach node v and then return. As a result $CT(u,v) = O(u,v) + O(v,u)$. In terms of the eigenvectors of the *normalized* Laplacian the commute time matrix is given by

$$CT(u,v) = vol \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \left(\frac{\phi_i(u)}{\sqrt{d_u}} - \frac{\phi_i(v)}{\sqrt{d_v}} \right)^2 \tag{6.7}$$

where $vol = \sum_{v \in V} d_v$ is the volume of the graph.

The commute time embedding is a mapping from the data space into a Hilbert subspace, which preserves the original commute times. It has some properties similar to existing embedding methods including principal component analysis (Jolliffe 2002) (PCA), the Laplacian eigenmap (Belkin & Niyogi 2003) and the diffusion map (Lafon & Lee 2006b). The embedding of the nodes of the graph

into a vector space that preserves commute time has the co-ordinate matrix

$$\Theta = \sqrt{vol}\Lambda^{-1/2}\Phi^T T^{-1/2} \tag{6.8}$$

The columns of the matrix are vectors of embedding co-ordinates for the nodes of the graph.

6.4 Shape Clustering and Classification

The commute time embedding gives a deformation-independent embedding of a 3D shape into a high dimensional space. In this chapter, we compute a shape descriptor from the commute time embedding. We use Laplace-Beltrami operator detailed in Section 6.2 to estimate the Laplacian of the shape. From the eigenvalues and eigenvectors of the Laplacian obtained, we compute the commute time matrix using the procedure given in Section 6.3. We use a modification of *D2* distributions introduced in (Osada et al. 2001). *D2* distribution is essentially, the histogram of pairwise Euclidean distance between the points uniformly sampled from the surface. To compute our new shape descriptor, we use the commute time distance instead of the Euclidean distance. The commute time matrix is computed using the equation 6.7. Where we replace the degree of the nodes (i.e. d_u and d_v) by the area associated with the vertices (i.e. s_i. and s_j respectively). We replace the vol in the original equation by $\sum_i s_i$.

6.5 Computational Complexity

Following are the steps of the algorithm developed in this chapter.

6.5.1 Steps

Given 3D shape with N vertices

1. Compute the Laplacian, which has the entries as follows

$$L(i,j) = \begin{cases} \sum_k w(i,k)/s_i & \text{if } i = j \\ -w(i,j)/s_i & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

2. Compute the Commute Time matric using the eigenvalues (λ_i) and eigenvectors (ϕ_i) of L as

$$CT(u,v) = vol \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \left(\frac{\phi_i(u)}{\sqrt{d_u}} - \frac{\phi_i(v)}{\sqrt{d_v}} \right)^2$$

   where $vol = \sum_{v \in V} d_v$ is the volume of the graph

3. Take random samples (pair of points) and compute the commute time distance between them to construct a histogram (64 bins)

4. Use Bhattaharyya distance to construct the distance matrix from the given set of histograms

5. Use MDS to embed the distance matrix in space to cluster shapes.

6.5.2 Complexity

The running time to the algorithm is dominated by the eigen-decomposition of the Laplacian matrix. Since we are using MATLAB's *eigs* function, this time is always less than O(N^3), where N is the number of vertices in the mesh. Each of

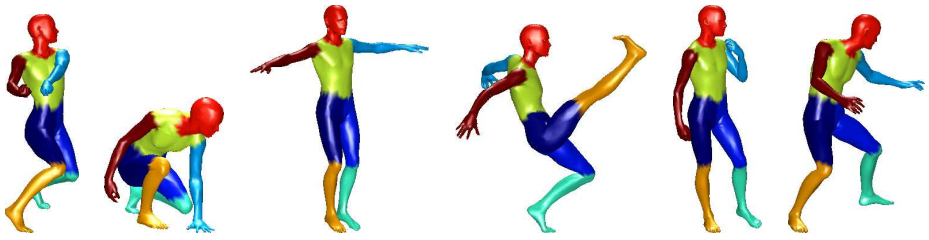


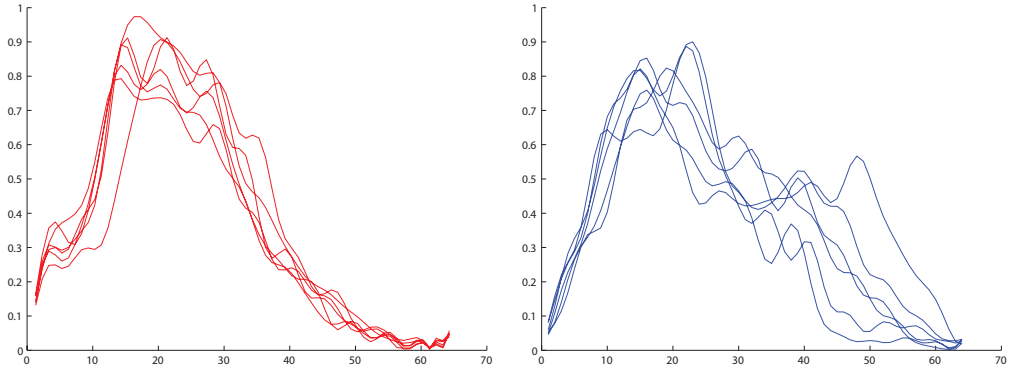
Figure 6.2: The k-means clustering on the Commute Time coordinates results in segmentation of six deformations of a 3D shape.

step 3 and 4 takes quadratic time in number of vertices. Finally MDS depends on the eigen-decomposition, which takes cubic time in the number of vertices. Hence, the worst case time of the proposed method is bounded by O(N^3).

6.6 Experimental Results

In this section, we provide some experimental investigations of the proposed method. We focus on the use of commute time embedding of 3D shapes in two different settings. The first is an investigation of using the commute time embedding for the purpose of partitioning the 3D shape into its parts. The second investigation is about using the modified shape distribution of Osada et al (Osada et al. 2001) computed by employing the commute time distance instead of the Euclidean distance.

In our first experiment we use the commute time embedding coordinates computed using equation 6.8 to partition six deformations of a human body selected from the Nonrigid world 3D database (Alexander & Bronstein 2009) shown in figure 6.2. The database contains a total of 148 objects, including 9 cats, 11 dogs, 3 wolves, 17 horses, 15 lions, 21 gorillas, 1 shark, 24 female figures, and two different male figures, containing 15 and 20 poses. The database also contains 6 centaurs, and 6 seahorses for partial similarity experiments. Each object contains approximately 3500 vertices. Figure 6.2 shows the result



(a) (b)

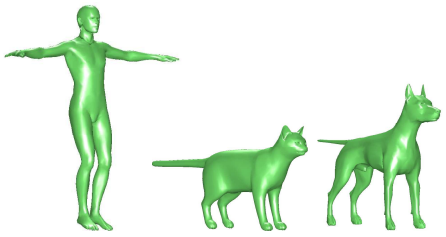
Figure 6.3: The histogram for the six 3D shapes shown in figure 6.2. a) The commute time histogram b) The Euclidean histogram

of the 3D shape, pose invariant segmentation using the k-means clustering on the commute time coordinates.

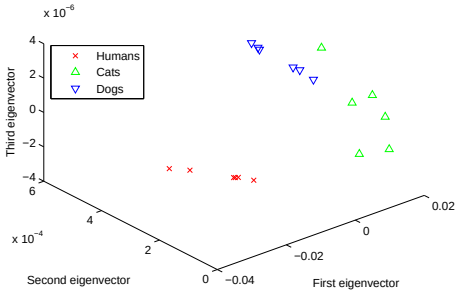
In the second experiment, we construct the shape distribution for six different deformations of each of the 3D shapes shown in figure 6.4(a). Figure 6.3(a) shows the shape descriptors for the six deformations using commute times. The shape descriptors for the same six deformations using Euclidean distances are shown in figure 6.3(b). This shows that the shape descriptor computed using commute time is more robust to shape deformations. We find the distance between each pair of the distributions using Bhattacharyya distance (Bhattacharyya 1943). We project the distance matrix into vector space using classical multi-dimensional scaling (MDS). Figure 6.4 shows that the commute time shape distribution clusters similar shapes better than the Euclidean shape distribution.

6.7 Summary

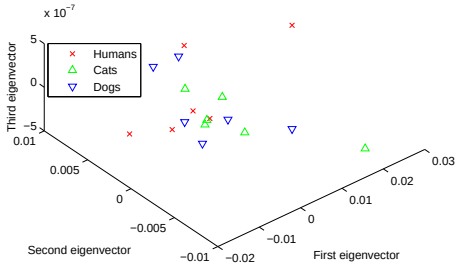
In this chapter we have investigated how the commute time between the vertices on mesh can be used to partition the 3D shape. We also used commute



(a)



(b)



(c)

Figure 6.4: a) Three shapes used in clustering experiment (six deformations of each shape are used). b) The classical MDS projection of the shape similarities as computed using the commute time distributions, with Rand index = 0.77 c) The classical MDS projection of the shape similarities as computed using the D2 distributions, with Rand index = 0.49

time distance to construct the 3D shape distribution for the purpose of 3D shape clustering and 3D shape classification. The empirical results show that commute time is a better choice for shape classification problem.

CHAPTER 7

Conclusions

This chapter summarises the main contributions of the thesis and draws some important conclusions. This includes the novel idea to use a Hermitian property matrix for the purpose of correspondence matching and graph clustering, using the coefficients of symmetric polynomials for the eigenvector direction correction and 3D shape signature using commute time embedding.

7.1 Contributions

The general objective of this thesis is to develop frameworks using graph spectral methods and apply them to a variety of applications from computer vision, for example the corresponding matching and graph clustering problems. First, a spectral graph matching algorithm was developed using the complex spectrum of a Hermitian property matrix. Second, we used the complex coefficients of the elementary symmetric polynomials derived from the eigenvalues and the com-

plex eigenvectors of a Hermitian property matrix for a transformed graph (line graph) of the human skeletal graph captured using Microsoft Kinect device to construct feature vectors. These feature vectors were embedded into pattern space to cluster similar human poses. Third, we used the coefficients of the elementary symmetric polynomials computed from the eigenvectors to make the directions of a pair of eigenvectors consistent with each other for the purpose of correspondence matching. Finally, a robust 3D shape descriptor with respect to changes in pose and topology based on commute time embedding was described. Next, we discussed the contributions and analyzed their strengths and weaknesses, discussing possible improvements of the algorithms and suggesting a potential future extension for more challenging correspondence matching and clustering / classification tasks.

Spectral graph methods for correspondence matching are based on the analysis of the eigenvectors of the proximity matrix constructed from the input feature points. The idea behind the graph spectral methods for computing the feature correspondence is to use the eigenvectors of the graph as signature of the points. These methods are elegant and mathematically well posed. However, they break soon, in the presence of noise and structural corruptions, where the point sets being matched are of different sizes. The novel part of the correspondence matching algorithm we developed in this thesis was to extend the point proximity matrix to the complex domain by augmenting additional angular information to it to construct a Hermitian property matrix. A Hermitian property matrix is complex analog of a real symmetric proximity matrix. The eigenvalues of the Hermitian matrix are real while the eigenvectors are complex. In Chapter 3 we used the complex eigenvectors of a Hermitian property matrix to compute the correspondences between a pair of point sets. The Hermitian property matrix was constructed from the distances between each pair of points and the angular

information. For the experiments on real world data set we used the SIFT orientations computed at each feature point extracted from the input image as angular information. The complex eigenvectors of the Hermitian property matrix established signatures of the feature points that are robust to noise in point position jitter and rotation.

To cope with the problem of noise and structural corruptions, Carcassoni and Hancock (Carcassoni & Hancock 2003) proposed an iterative EM algorithm for alignment of feature point sets. We embedded the complex eigenvectors of the Hermitian property matrix to render the EM algorithm robust to noise and rotation in the input images being matched.

The second contribution of this thesis was the development of a human pose clustering method using four different types of feature vectors constructed from the coefficients of the elementary symmetric polynomials. The polynomials are established from the eigenvalues and the complex eigenvectors of the Hermitian property matrix. The input human skeleton acquired from the Microsoft Kinect device for Xbox 360 was converted to its equivalent line graph. The joints of the human body are represented by the nodes of the graph while edges represent the limbs. The Hermitian property matrix was constructed from the line graph representing a human pose.

The third contribution of this thesis was the development of a method for correction of the sign of eigenvectors for the problem of correspondence matching. Spectral graph methods for correspondence matching are based on the analysis of the eigenvectors of the proximity matrix constructed from the input feature points. Since the sign of eigenvectors are not unique, the eigenvector solver assigns arbitrary signs to the eigenvectors computed for the pair of proximity matrices constructed from the input feature point sets. The correspondence matches can only be computed correctly when the direction (signs) of the corres-

ponding pair of eigenvectors are consistent with each other. We used the coefficients of the elementary symmetric polynomial established from the eigenvectors of the proximity matrices to make the directions of the pair of eigenvectors consistent with each other for the purpose of correspondence matching.

The fourth contribution of this thesis was the development of a 3D shape descriptor which was robust to shape deformations and changes in topology. The proposed descriptor was an extension of the D2 shape descriptor reported by Osada et al. (Osada et al. 2001). We used commute time distance computed from the eigenvalues and the eigenfunctions of the Laplace-Beltrami operator instead of using Euclidean distances between pair of points on the shape.

7.2 Limitations and Future Work

The methods presented in this thesis perform very well. However, several shortcomings can be addressed by further research. Moreover, some of the topics discussed could be extended and investigated further for subsequent improvements.

Although we have experimentally shown that the correspondence matching results obtained by using the complex eigenvectors of the Hermitian property matrix are much better than that of the two state-of-the-art algorithms i.e. Shapiro and Brady point pattern matching algorithm (Shapiro & Brady 1992) and Carcassoni and Hancock EM alignment algorithm (Carcassoni & Hancock 2003). However, it has limitations, which need to be addressed in future research. One of the weaknesses is that the new correspondence matching method that has been developed is computationally expensive for large point sets. The reason for this is that the eigen-decomposition operation is computationally expensive for large Hermitian matrices. To reduce this overhead, we need to find methods to com-

pute and use only the first few eigenvectors.

When we apply the proposed algorithm on real world images, we use SIFT orientations as angular information to construct the Hermitian property matrix for correspondence matching. Therefore, the performance of the proposed methods depends upon the SIFT orientations. The SIFT orientations are computed using a local gradient histogram established in the neighbourhood of the feature points using the Gaussian image at the closest scale to the feature point scale. The orientation histogram is divided into 36 bins of 10 degrees each, totalling 360 degrees. The peaks in the orientation histogram correspond to dominant direction of the local gradients. The highest peak in the histogram and any other peak which is within 80% of the highest peak is used to assign the orientation to the feature point. Therefore, the feature points with multiple peaks are assigned multiple orientations, by creating multiple feature points at the same location but with different orientations. If the a feature point in one image and its corresponding feature point in the second image are assigned a different number of orientations, then the matching results obtained by using the Hermitian matrix is negatively affected. This is due to the fact that increase in the difference between the number of feature points extracted from the two images being matched increases the probability of getting wrong matching results. Therefore we need to remove the extra orientations before we proceed to compute the correspondences. To address this problem we have used cross correlation between the corresponding histograms to remove the extra orientations associated with feature points. However, this method is not very robust and it fails, especially in the case when more than two orientations are assigned to a feature point. Therefore, a more general and robust method needs to be developed.

The proposed method is limited to work on 2D point sets and can handle only the 2D affine transformations. It would be interesting to extend the method

to find correspondences between 3D shapes/meshes.

One possible direction to extend the algorithm presented in Chapter 3 could be the use of RANSAC algorithm. RANSAC is a stochastic algorithm that is based on a heuristic cost function, however, our method is based on the analysis of local consistency and EM algorithm.

In Chapter 4 we used the spectrum of a Hermitian property matrix and the coefficient of the symmetric polynomials to cluster similar human poses. This work can be used in human behaviour analysis. It would be interesting to explore how this work can be used to build a real-time gesture recognition system.

In Chapter 6, we describe a commute-time based 3D shape descriptor that is robust with respect to changes in pose and topology. Commute-time embedding can not detect shape symmetry and hence can not be used to compute correspondences. It would be very interesting to explore the use of curvatures as the angular information to construct a Hermitian property matrix for correspondence matching of 3D shapes.

The algorithms developed in this thesis are not confined to the field of computer vision only. Their applications can be explored in many other research fields including biometrics, molecular chemistry, social networks etc.

List of Symbols

$G(V, E)$	A graph with node set V and edge set E
W_{ij}	The $(i, j)^{th}$ element of edge-weight matrix W
A	The adjacency matrix of a graph
H	The Hermitian matrix of a graph
D	Diagonal matrix with entries, the degrees of nodes of graph
L	Laplacian matrix of a graph
$\mathcal{L}$	The normalized Laplacian matrix of a graph
Λ	The diagonal matrix consisting of eigenvalues
Φ	The column matrix consisting of eigenvectors
λ_i	The i^{th} eigenvalue
$I(x, y)$	A pixel of image I at (x, y) location
ζ	Correspondence probability matrix
$\mathcal{T}$	Transformation matrix
$P_i(\mathbf{x})$	i_{th} coefficient of power symmetric polynomial of **x**
$S_i(\mathbf{x})$	i_{th} coefficient of elementry symmetric polynomial of **x**
F_i	i_{th} feature vector
R_I	Rand index
CT	Commute Time matrix
Z_{ij}	Association matrix

References

Agarwal, A. & Triggs, B. (2006). Recovering 3D Human Pose from Monocular Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(1), 44–58.

Alexander & Bronstein, M. (2009). Nonrigid World 3d Database v 1.0 @ONLINE.

Andriluka, M., Roth, S., & Schiele, B. (2009). Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, (pp. 1014–1021).

Aubry, M., Schlickewei, U., & Cremers, D. (2011). The Wave Kernel Signature: A Quantum Mechanical Approach to Shape Analysis. In *IEEE International Conference on Computer Vision (ICCV) Workshops*, (pp. 1626–1633).

Bach, F. R. & Jordan, M. I. (2004). Learning Spectral Clustering. In *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press.

Belkin, M. & Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Re-

duction and Data Representation. *Neural Computation*, *15*, 1373–1396.

Bhattacharyya, A. (1943). On a Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions. *Bulletin of the Calcutta Mathematics*, *35*, 99–110.

Biggs, N. (1974). *Algebraic Graph Theory*. Cambrridge University Press.

Bunke, H., Foggia, P., Guidobaldi, C., & Vento, M. (2003). Graph Clustering using the Weighted Minimum Common Supergraph. In *International Workshop on Graph Based Representations in Pattern Recognition*, volume 2726.

Caelli, T. & Kosinov, S. (2004). An Eigenspace Projection Clustering Method for Inexact Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 515–519.

Cantzler, H., Fisher, R., & Devy, M. (2002). Improving Architectural 3D Reconstruction by Plane and Edge Constraining. In *Proceedings of British Machine Vision Conference*, (pp. 43–52).

Carcassoni, M. & Hancock, E. R. (2003). Spectral Correspondence for Point Pattern Matching. *Pattern Recognition*, *36*(1), 193–204.

Castellani, U., Mirtuono, P., Murino, V., Bellani, M., Rambaldelli, G., Tansella, M., & Brambilla, P. (2011). A New Shape Diffusion Descriptor for Brain Classification. In *Proceedings of the 14th International Conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part II*, MICCAI'11, (pp. 426–433)., Berlin, Heidelberg. Springer-Verlag.

Chui, H. & Rangarajan, A. (2000). A New Algorithm for Non-Rigid Point Matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 44–51).

Chung, F. R. K. (1997). *Spectral Graph Theory*. American Mathematical Society.

Clarke, J., Carlsson, S., & Zisserman, A. (1996). Detecting and Tracking Linear

Feature Efficiently. In *Proceedings of British Machine Vision Conference*, (pp. 415–424).

Collatz, L. & Sinogowitz, U. (1957). Spektren Endlicher Grafen. *Abhandlungen Aus Dem Mathematischen Seminar Der Universitat Hamburg ABH MATH SEM UNIV HAMBURG*, *21*, 63–77.

Comaniciu, D., Meer, P., & Member, S. (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 603–619.

Costeira, J. & Kanade, T. (1998). A Multibody Factorization Method for Motion Analysis. *International Journal of Computer Vision*, *29*(3), 159–180.

Cross, A. D. J. & Hancock, E. R. (1998). Graph Matching with a Dual-Step EM Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1236–1253.

Cuzzolin, F., Mateus, D., Knossow, D., Boyer, E., & Horaud, R. (2008). Coherent Laplacian 3-D Protrusion Segmentation. In *IEEE Confrence on Computer Vision and Pattern Recognition (CVPR)*.

Cvetković, D., Doob, M., & Sachs, H. (1980). *Spectra of Graphs: Theory and Application*. Pure and applied mathematics. Academic Press.

Delponte, E., Isgrò, F., Odone, F., & Verri, A. (2006). SVD-Matching using SIFT Features. *Graphical Models Special issue on the Vision, Video and Graphics Conference*, *68*(5), 415–431.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39(1)*, 1–38.

Desbrun, M., Meyer, M., Schröder, P., & Barr, A. H. (1999). Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*,

SIGGRAPH '99, (pp. 317–324)., New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Donath, W. & Hoffman, A. (1972). Algorithms for Partitioning Graphs and Computer Logic Based on Eigenvectors of Connection Matrices. *IBM Technical Disclosure Bulletin*, *15*(3), 938–944.

Donoho, D. L. & Grimes, C. (2003). Hessian Eigenmaps: New Locally Linear Embedding Techniques for High-Dimensional Data. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 100, (pp. 5591–5596).

Dornaika, F. & Chung, R. (1999). Stereo Correspondence from Motion Correspondence. (pp. 1070–1075). IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Duda, R. O. & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*, (pp. 98–105). John Wiley and Sons.

Eichner, M. & Ferrari, V. (2010). We Are Family: Joint Pose Estimation of Multiple Persons. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, (pp. 228–242)., Berlin, Heidelberg. Springer-Verlag.

Elgammal, A., Duraiswami, R., & Davis, L. S. (2003). Efficient Kernel Density Estimation using the Fast Gauss Transform with Applications to Color Modeling and Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 1499–1504.

Fiedler, M. (1973). Algebraic Connectivity of Graphs. *Czechoslovak Mathematical Journal*, *23*, 298–305.

Fischler, M. A. & Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, *24*(6), 381–395.

Gangopadhyay, A., Joshi, A., & Wali, R. (2012). A Spectral Clustering Tech-

nique for Studying Post-Transplant Kidney Functions. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, (pp. 201–208)., New York, NY, USA. ACM.

Harris, C. & Stephens, M. (1988). A Combined Corner and Edge Detector. In *Proceedings of Fourth Alvey Vision Conference*, (pp. 147–151).

Hasler, D., Sbaiz, L., Süsstrunk, S., & Vetterli, M. (2003). Outlier Modelling in Image Matching. *IEEE Transactions on Pattern Analysis Machine Intelligence*, *25*, 301–315.

He, X., Yan, S., Hu, Y., Niyogi, P., & jiang Zhang, H. (2005). Face Recognition using Laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 328–340.

Hotelling, H. (1933). Analysis of a Complex of Statistical Variables into Principal Pomponents. *Journal of Educational Psychology*, *24*, 417–441.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data Clustering: A Review. *ACM Computing Surveys*, *31*(3), 264–323.

Jerain, C. & Jain, R. (1991). Structure From Motion - A Critical Analysis of Methods. *IEEE Transactions on Systems, Man and Cybernetics*, *21*(3), 572–588.

Johnson, S. & Everingham, M. (2009). Combining Discriminative Appearance and Segmentation Cues for Articulated Human Pose Estimation. In *Proceeding of MLVMA, 2009. EVERINGHAM: CLUSTERED MODELS FOR HUMAN POSE ESTIMATION 11*.

Jolliffe, I. (2002). *Principal Component Analysis*. American Mathematical Society Second Edition, New York: Springer-Verlag New York, Inc.

Kannan, R. (2000). On Clusterings: Good, Bad and Spectral. In *Journal of the ACM*, (pp. 367–377).

Kawakami, H., Ito, Y., & Kanazawa, Y. (2006). A Robust Method for Detecting

Planar Regions Based on Random Sampling using Distributions of Feature Points. *Systems and Computers in Japan, 37*(4), 11–22.

Kolluri, R., Shewchuk, J. R., & O'Brien, J. F. (2004). Spectral Surface Reconstruction from Noisy Point Clouds. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, (pp. 11–21)., New York, NY, USA. ACM.

Kruskal, J. B. & Wish, M. (1978). *Multidimensional Scaling*. Sage Publications Beverly Hills. CA.

Lafon, S. & Lee, A. B. (2006a). Diffusion Maps. *Applied and Computational Harmonic Analysis, 21*(1), 5–30.

Lafon, S. & Lee, A. B. (2006b). Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning and Data Set Parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*, 1393–1403.

Lee, N. S., Yamasaki, T., & Aizawa, K. (2008). Hierarchical Mesh Decomposition and Motion Tracking for Time-Varying-Meshes. In *ICME*, (pp. 1565–1568). IEEE.

Leuken, R. H., Symonova, O., Veltkamp, R. C., & Amicis, R. (2008). Complex Fiedler Vectors for Shape Retrieval. In *Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, SSPR & SPR '08, (pp. 167–176)., Berlin, Heidelberg. Springer-Verlag.

Ling, H. & Jacobs, D. W. (2007). Shape Classification using the Inner-Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*, 286–299.

Lowe, D. G. (2004). Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision, 60*(2), 91–110.

Lucas, B. D. & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Cconference on Artificial Intelligence - Volume 2*, IJCAI'81, (pp. 674–679)., San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Luo, B. & Hancock, E. (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., & Hancock, E. R. (2002). Spectral Feature Vectors for Graph Clustering. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, (pp. 83–93)., London, UK, UK. Springer-Verlag.

Luo, B., Wilson, R. C., & Hancock, E. R. (2003). Spectral Embedding of Graphs. *Pattern Recognition*, *36*(10), 2213–2223.

Luxburg, U. (2007). A Tutorial on Spectral Clustering. *Statistics and Computing*, *17*(4), 395–416.

MacQueen, J. B. (1967). Some Methods for Classification and Analysis of MultiVariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, (pp. 281–297). University of California Press.

Mateus, D., Horaud, R. P., Knossow, D., Cuzzolin, F., & Boyer, E. (2008). Articulated Shape Matching using Laplacian Eigenfunctions and Unsupervised Point Registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR*.

Meyer, M., Desbrun, M., Schröder, P., & Barr, A. H. (2003). Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In Hege, H.-C. & Polthier, K. (Eds.), *Visualization and Mathematics III*, (pp. 35–57)., Heidelberg. Springer-Verlag.

Microsoft (2010). MS Kinect for XBOX 360. *doi:http://www.xbox.com/kinect*.

Mohar, B. (1992). Laplace Eigenvalues of Graphs - A Survey. *Discrete Mathematics*, *109*(1–3), 171–183.

Mohar, B. (1997). Some Applications of Laplace Eigenvalues of Graphs. In *Graph Symmetry: Algebraic Methods and Applications, Volume 497 of NATO ASI Series C*, (pp. 227–275). Kluwer.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems 14*, (pp. 849–856). MIT Press.

Ohbuchi, R., Minamitani, T., & Takei, T. (2005). Shape-Similarity Search of 3D Models by using Enhanced Shape Functions. *International Journal of Computer Applications in Technology (IJCAT)*, *23*(2/3/4), 70–85.

Osada, R., Funkhouser, T., Chazelle, B., & Dobkin, D. (2001). Matching 3d Models with Shape Distributions. In *Proceedings of the International Conference on Shape Modeling & Applications*, SMI '01, (pp. 154–167)., Washington, DC, USA. IEEE Computer Society.

Ovsjanikov, M., Mérigot, Q., Mémoli, F., & Guibas, L. J. (2010). One Point Isometric Matching with the Heat Kernel. *Computer Graphics Forum*, *29*(5), 1555–1564.

Park, S. H., Lee, K. M., & Lee, S. U. (2000). A Line Feature Matching Technique Based on an Eigenvector Approach. *Computer Vision and Image Understanding*, *77*(3), 263–283.

Pavan, M. & Pelillo, M. (2003). Dominant Sets and Hierarchical Clustering. In *Proceedings 9th IEEE International Conference on Computer Vision (ICCV)*, (pp. 362–369).

Pilu, M. (1997). A Direct Method for Stereo Correspondence based on Singular Value Decomposition. In *IEEE Confrence on Computer Vision and Pattern*

Recognition (CVPR), (pp. 261–266).

Pinkall, U., Juni, S. D., & Polthier, K. (1993). Computing Discrete Minimal Surfaces and Their Conjugates. *Experimental Mathematics*, *2*, 15–36.

Qiu, H. & Hancock, E. R. (2007). Clustering and Embedding using Commute Times. *IEEE Trans. Pattern Anal. Mach. Intell.*, *29*(11), 1873–1890.

Ren, P., Wilson, R. C., & Hancock, E. R. (2011). Graph Characterization via Ihara Coefficients. *IEEE Transactions on Neural Networks*, *22*(2), 233–245.

Robles-Kelly, A. & Hancock, E. R. (2007). A Riemannian Approach to Graph Embedding. *Pattern Recognition*, *40*(3), 1042–1056.

Rogez, G., Rihan, J., Orrite-Uruñuela, C., & Torr, P. H. (2012). Fast Human Pose Detection using Randomized Hierarchical Cascades of Rejectors. *International Journal of Computer Vision*, *99*(1), 25–52.

Rosenberg, S. (1997). *The Laplace on Riemannian Manifold*. Cambridge University Press.

Roweis, S. T. & Saul, L. K. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, *290*, 2323–2326.

Rustamov, R. M. (2007). Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation. In *Symposium on Geometry Processing*, (pp. 225–233).

Sarkar, S. & Boyer, K. L. (1996). Quantitative Measures of Change based on Feature Organization: Eigenvalues and Eigenvectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 478–483).

Schaffalitzky, F. & Zisserman, A. (2001). Viewpoint Invariant Texture Matching and Wide Baseline Stereo. In *IEEE International Conference on Computer Vision*, (pp. 636–643).

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, *10*(5), 1299–1319.

Scott, G. L. & Longuet-Higgins, H. C. (1990). Feature Grouping by Relocalisation of Eigenvectors of Proximity Matrix. In *Proceedings of British Machine Vision Conference*, (pp. 103–108).

Scott, G. L. & Longuet-Higgins, H. C. (1991). An Algorithm for Associating the Features of Two Images. In *Proceedings of the The Royal Society London*, Vol B, No 244, (pp. 21–26).

Shapiro, L. S. (1991). *Towards a Vision-based Motion Framework*. First Year Report, Department of Engineering Science, Oxford University.

Shapiro, L. S. & Brady, J. M. (1992). Feature-based Correspondence: An Eigenvector Approach. In *Image and Vision Computing*, Vol 10, (pp. 283–288).

Shi, J. & Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Shi, J. & Tomasi, C. (1994). Good Features to Track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 593–600).

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., & Blake, A. (2011). Real-time Human Pose Recognition in Parts from Single Depth Images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, (pp. 1297–1304). IEEE Computer Society.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London.

Sun, J., Ovsjanikov, M., & Guibas, L. (2009). A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, (pp. 1383–1392)., Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Tang, J., Liang, D., Wang, N., & zheng Fan, Y. (2007). A Laplacian Spectral Method for Stereo Correspondence. *Pattern Recognition Letters*, *28*, 1391–

1399.

Taubin, G. (1995). A Signal Processing Approach to Fair Surface Design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, (pp. 351–358)., New York, NY, USA. ACM.

Tenenbaum, J. B., Silva, V., & Langford, J. C. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, *290*(5500), 2319–2323.

Tomasi, C. & Kanade, T. (1992). Shape and Motion from Image Streams under Orthography: A Factorization Method. *International Journal of Computer Vision*, *9*(2), 137–154.

Torr, P. H. S. & Murray, D. W. (1993). Outlier Detection and Motion Segmentation. In *Proceedings of the International Society for Optical Engineering (SPIE)*, volume 2059, (pp. 432–443).

Umeyama, S. (1988). An Eigendecomposition Approach to Weighted Graph Matching Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *10*(5), 695–703.

Vedaldi, A. (2006). MATLAB/C Implementation of the SIFT Detector and Descriptor. *doi:http://www.vlfeat.org/∼vedaldi/code/sift.html*.

Weiss, Y. (1999). Segmentation Using Eigenvectors: A Unifying View. In *International Conference on Computer Vision*, (pp. 975–982).

Weng, N., Yang, Y., & Pierson, R. (1997). Three-dimensional Surface Reconstruction using Optical Flow for Medical Imaging. *IEEE Transactions on Medical Imaging*, *16*(5), 630–641.

Wikipedia (2013). Elementry Symmetric Polynomial @ONLINE. *doi:http://en.wikipedia.org/wiki/Elementary_symmetric_polynomial*.

Wilson, R. C., Hancock, E. R., & Luo, B. (2005). Pattern Vectors from Al-

gebraic Graph Theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(7), 1112–1124.

Xiao, B., Hancock, E. R., & Wilson, R. C. (2009). Graph Characteristics from the Heat Kernel Trace. *Pattern Recognition, 42*(11), 2589–2606.

Xu, G. (2006). Discrete Laplace-Beltrami Operator on Sphere and Optimal Spherical Triangulations. *Int. J. Comput. Geometry Appl., 16*(1), 75–93.

Xu, L. & King, I. (2001). A PCA Approach for Fast Retrieval of Structural Patterns in Attributed Graphs. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, 31*(5), 812–817.

Yu, S. X. & Shi, J. (2003). Multiclass Spectral Clustering. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, (pp. 313–319)., Washington, DC, USA. IEEE Computer Society.

Zelnik-manor, L. & Perona, P. (2004). Self-Tuning Spectral Clustering. In *Advances in Neural Information Processing Systems 17*, (pp. 1601–1608). MIT Press.